

Vol. 32, No. 3/4, 2023

Machine
GRAPHICS & VISION

International Journal

Published by
The Institute of Information Technology
Warsaw University of Life Sciences – SGGW
Nowoursynowska 159, 02-776 Warsaw, Poland

in cooperation with
The Association for Image Processing, Poland – TPO

ROBUST LINE-CONVEX POLYGON INTERSECTION COMPUTATION IN E^2 USING PROJECTIVE SPACE REPRESENTATION

Vaclav Skala 

*Dept. of Computer Science and Engineering, Faculty of Applied Sciences
University of West Bohemia, Pilsen, Czech Republic*

www.VaclavSkala.eu

Abstract This paper describes modified robust algorithms for a line clipping by a convex polygon in E^2 and a convex polyhedron in E^3 . The proposed algorithm is based on the Cyrus-Beck algorithm and uses homogeneous coordinates to increase the robustness of computation. The algorithm enables computation fully in the projective space using the homogeneous coordinates and the line can be given in the projective space, in general. If the result can remain in projective space, no division operation is needed. It supports the use of vector-vector operations, SSE/AVX instructions, and GPU.

Keywords: computer graphics, line convex polygon intersection, line convex polygon clipping, Cyrus-Beck algorithm, homogeneous coordinates, projective space, duality principle, vector-vector operations, GPU computing.

1. Introduction

Algorithms for a line and line segment intersection computation with the convex polygon in E^2 and convex polyhedron in E^3 are key parts of many geometrical packages, CAD and GIS systems, etc. An extensive survey of intersection and clipping algorithms can be found in [31]. Fundamental algorithms have been described in many textbooks, see Appendix.

Due to the apparent simplicity of intersection algorithms, they might fail due to the limited precision of computation of the Floating-Point Arithmetic (IEEE 754) used in today's computers. The Cyrus-Beck (CB) algorithm [5] is well known for solving the intersection problem of a line with a convex polygon in the E^2 case or a polyhedron in the E^3 case.

There are two basic principles in the E^2 case:

- the edges of a convex polygon define lines and intersection computation is based on direct intersection computation of the clipped line with an edge of the convex polygon,
- the given line defines a half-plane, which separates convex polygon vertices [20, 24, 28], and positions of the polygon vertices are tested.

The first approach is used by the CB algorithm [5], and the second one was used in [17].

In both cases, the convex polygon can be given as a set of oriented edges with all normal vectors pointing inside or outside the convex polygon and computational complexity is $O(N)$. It means, that the consecutive order of edges is not needed, resp. this property is not used within the CB algorithm.

It should be noted, that in the case of E^2 the convex polygon is given by an ordered sequence of vertices, i.e. clockwise or anti-clockwise, and such property leads to algorithms with $O(\lg N)$ [17]. In the E^3 case an ordering of facets is not possible, however in the triangular mesh case, where neighbors of a triangle are known, the algorithm with $O(\sqrt{N})$ was described [18, 19, 25].

1.1. Cyrus-Beck line clipping algorithm in Euclidean space

The CB algorithm [5] in Euclidean space for a line clipping against a convex polygon in E^2 or against a convex polyhedron in E^3 is well known. It is used in many computer graphics systems and related courses due to its simplicity and applicability for the E^3 case.

However, the CB algorithm has some assumptions:

- it was developed for Euclidean space, i.e. the polygon vertices or the points defining the line p need are given generally in the homogeneous coordinates, they have to be converted to the Euclidean space,
- consistent and known normal vectors orientation of edges, resp. facets, i.e. the normal vectors should all be pointing out or inside,
- generally, an unordered set of edges in the E^2 case, resp. facets in the E^3 case is given. In the E^2 case a polygon is usually given as an ordered set of edges with clockwise or anti-clockwise orientation,
- the given line, which is to be clipped, is given in the parametric form or by two points in the case of line segment clipping.

The CB algorithm is based on direct intersection computation of the given line p in the parametric form and a line on which the polygon edge e_i lies, see Fig. 1, in the

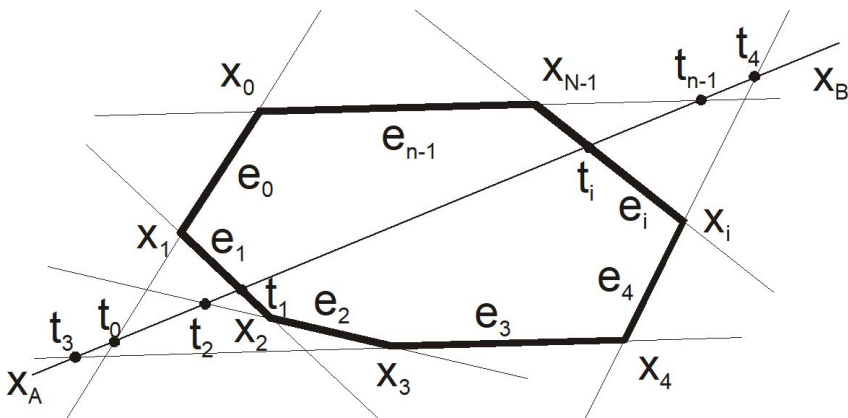


Fig. 1. Cyrus-Beck clipping algorithm against the convex polygon in E^2

implicit form. This leads to a solution of two linear equations (1) (the vector notation is used):

$$\begin{aligned} p: \quad \mathbf{x}(t) &= \mathbf{x}_A + \mathbf{s} t, \quad t \in (-\infty, +\infty), \\ e_i: \quad \mathbf{n}_i^T \mathbf{x} + c_i &= 0, \quad i = 0, \dots, N-1, \\ a_i x + b_i y + c_i &= 0, \end{aligned} \quad (1)$$

where $\mathbf{x}_A = [x_A, y_A]^T$, $\mathbf{s} = [s_X, s_Y]^T$ is the directional vector of the line p , $\mathbf{n}_i = [n_X, n_Y]^T$ is the “normal” vector¹ of the edge e_i , and c_i is related to the e_i distance from the origin.

Solving (1), the parameter t for the intersection point is obtained as:

$$\mathbf{n}_i^T \mathbf{x}_A + \mathbf{n}_i^T \mathbf{s} t + c_i = 0. \quad (2)$$

Then the t_i is the parameter t value for the intersection of the line p and the line on which the edge e_i lies, see Fig. 1.

$$t_i = -\frac{\mathbf{n}_i^T \mathbf{x}_A + c_i}{\mathbf{n}_i^T \mathbf{s}}. \quad (3)$$

The CB algorithm is of $O(N)$ computational complexity with a fixed $O(N)$ pre-computational cost, as coefficients of lines on which the polygon edges lie need to be pre-computed, see Algorithm 1.

It can be seen that there is an instability of the algorithm as if the line p is parallel or nearly parallel to the edge e_i , the expression $\mathbf{n}_i^T \mathbf{s} \rightarrow 0$ and $t_i \rightarrow \pm\infty$. The fraction computation might cause an overflow or high imprecision of the computed parameter t value, see Fig. 1.

It is hard to detect and solve reliably such cases² and programmers usually use a sequence like:

$$\text{if } |\mathbf{n}_i^T \mathbf{s}| < \varepsilon \text{ then a singular case,} \quad (4)$$

which is an incorrect solution as the value ε is a programmer’s choice and the value of $|\mathbf{n}_i^T \mathbf{s}|$ might be also close to the value ε (3).

The CB algorithm for a line clipping is described by the Algorithm 1. It can be easily modified for a line segment clipping just restricting the range of the parameter t to $\langle 0, 1 \rangle$, i.e.

$$\langle t_{\min}, t_{\max} \rangle := \langle t_{\min}, t_{\max} \rangle \cap \langle 0, 1 \rangle. \quad (5)$$

If the final interval of t is empty, i.e. $\langle t_{\min}, t_{\max} \rangle = \emptyset$ (the case $t_{\min} > t_{\max}$), then the line segment does not have an intersection with the convex polygon.

The known modifications of the CB algorithm use a separation function for more reliable detection of “close to singular” cases were described in [16]:

¹The “normal” vector is a bivector having different properties from a vector.

²However, many textbooks do not point out such dangerous construction as far as the robustness and computational stability are concerned and consider it as a singular case.

Algorithm 1 Cyrus-Beck Line Clipping Algorithm

```

1: for  $i := 0$  to  $N-1$  do                                ▷ computation for the given convex polygon
2:   Compute  $(a_i, b_i : c_i)$                                 ▷  $[a_i, b_i : c_i]^T = [\mathbf{n}_i^T : c_i]^T$  for all polygon edges
3: end for
4:
5: procedure CB-CLIP( $\mathbf{x}_A, \mathbf{x}_B$ );                            ▷ line is given by two points  $\mathbf{x}_A, \mathbf{x}_B \in E^2$ 
6:    $t_{\min} := -\infty; t_{\max} := \infty;$                     ▷ set initial conditions for the parameter  $t$ 
7:    $\mathbf{s} := \mathbf{x}_B - \mathbf{x}_A;$                                 ▷ directional vector of the line
8:   for  $i := 0$  to  $N - 1$  do                                ▷ for each edge
9:      $q := \mathbf{n}_i^T \mathbf{s};$ 
10:    if  $\text{abs}(q) < \varepsilon$  then
11:      NOP;                                                    ▷ Singular or close to singular case
12:    else
13:       $t = -(\mathbf{n}_i^T \mathbf{x}_A + c_i) / \mathbf{n}_i^T \mathbf{s};$ 
14:      if  $q < 0$  then  $t_{\min} := \max\{t, t_{\min}\};$ 
15:      else  $t_{\max} := \min\{t, t_{\max}\};$ 
16:      end if
17:    end if
18:  end for                                                    ▷ all convex polygon edges processed
19:  if  $t_{\min} < t_{\max}$  then                                ▷ intersection of a line and the polygon exists
20:     $\{ \mathbf{x}_B := \mathbf{x}_A + \mathbf{s} t_{\max}; \mathbf{x}_A := \mathbf{x}_A + \mathbf{s} t_{\min}; \}$ 
21:    ▷ if  $t_{\min} > t_{\max}$  - NO intersection case
22:  end if
23: end procedure

```

- a separation implicit function $F_i(\mathbf{x})$ defined as
 $F_i(\mathbf{x}) = \mathbf{n}_i^T \mathbf{x} + c_i = a_i x + b_i y + c_i$ for the i^{th} edge [20],
- the parametric form of the given line
 $\mathbf{x}(t) = \mathbf{x}_A + (\mathbf{x}_B - \mathbf{x}_A) t$
for intersection computation with found edges intersected (3).

It can be seen that the CB algorithm is of $O(N)$ complexity and the division operation, which is the most consuming time operation in the floating point representation, is used N times. Also, the division operations cause imprecision and lead to robustness issues.³

³There is a possibility to postpone division operations if the homogeneous coordinates are used, but comparison operations must be modified appropriately [28, 30].

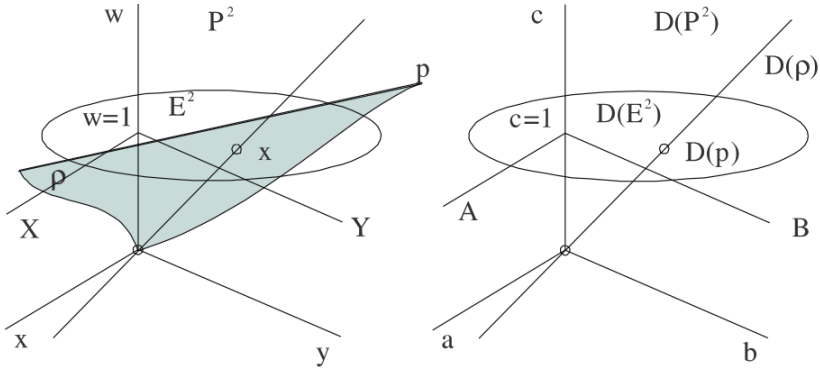


Fig. 2. Projective extension and dual space

2. Projective space

The projective extension of Euclidean space is not a part of standard computer science courses. However, homogeneous coordinates are used in computer graphics and computer vision algorithms, as they enable to represent geometric transformations like translation and rotation by matrix multiplication and also offer to represent a point in infinity.

The mutual conversion between the Euclidean space and projective space in the case of the E^2 space:

$$X = \frac{x}{w}, \quad Y = \frac{y}{w}, \quad w \neq 0, \quad (6)$$

where $\mathbf{X} = (X, Y)$, resp. $\mathbf{x} = [x, y : w]^T$ are coordinates in the Euclidean space E^2 , resp. in the projective space P^2 . The extension to the E^3 is straightforward.

The geometrical interpretation of the Euclidean and the projective spaces is presented in Fig. 2. It should be noted, that a distance of a point $\mathbf{X} = (X, Y)$, i.e. $\mathbf{x} = [x, y : w]^T$, from a line p in the E^2 is defined as:

$$\text{dist} = \frac{aX + bY + c}{\sqrt{a^2 + b^2}} = \frac{ax + by + cw}{w\sqrt{a^2 + b^2}}, \quad \mathbf{p} = [a, b : c]^T, \quad (7)$$

where $\mathbf{n} = (a, b)$ is the normal vector (actually it is a bivector) of the line p and c is related to the orthogonal distance of the line p from the origin. In the E^3 case:

$$\text{dist} = \frac{aX + bY + cZ + d}{\sqrt{a^2 + b^2 + c^2}} = \frac{ax + by + cz + dw}{w\sqrt{a^2 + b^2 + c^2}}, \quad \mathbf{p} = [a, b, c : d]^T, \quad (8)$$

where $\mathbf{n} = (a, b, c)$ is the normal vector (actually it is a bivector) of a plane ρ and d is related to the orthogonal distance of a plane ρ from the origin.

2.1. Principle of duality

A line p given by two points $\mathbf{x}_A = [x_A, y_A : w_A]^T$,
 $\mathbf{x}_B = [x_B, y_B : w_B]^T$ is given using the outer product as:⁴

$$\mathbf{p} = \mathbf{x}_A \wedge \mathbf{x}_B = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_A & y_A & w_A \\ x_B & y_B & w_B \end{vmatrix} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ X_A & Y_A & 1 \\ X_B & Y_B & 1 \end{vmatrix}, \quad (9)$$

$$[y_A w_B - y_B w_A, -(x_A w_B - x_B w_A) : x_A y_B - x_B y_A]^T = [a, b : c]^T,$$

where $w_A > 0$, $w_B > 0$, $\mathbf{p} = [a, b : c]^T$ are coefficients of the line p and \mathbf{i} , \mathbf{j} , \mathbf{k} are the orthonormal basis vectors of the projective space [10].⁵

The projective extension of the Euclidean space enables to use the principle of duality for the intersection of two lines p_1 and p_2 in E^2 using the outer product:

$$\mathbf{x} = \mathbf{p}_1 \wedge \mathbf{p}_2 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix} = \quad (10)$$

$$[b_1 c_2 - b_2 c_1, -(a_1 c_2 - a_2 c_1) : a_1 b_2 - a_2 b_1]^T = [x, y : w]^T.$$

It is because lines and points are dual primitives in the P^2 projective extension [3, 9, 21, 22, 23, 26, 27, 32, 33].

The outer-product $\mathbf{x}_A \wedge \mathbf{x}_B$ is formally equivalent to the cross-product $\mathbf{x}_A \times \mathbf{x}_B$ in the P^2 projective extension case and the non-normalized normal vector of the line p is $\mathbf{n} = [a, b : 0]^T$.

In the E^3 case, the dual primitives are points and planes, i.e.

$$\mathbf{x} = \rho_1 \wedge \rho_2 \wedge \rho_3 = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \end{vmatrix} = [x, y, z : w]^T,$$

$$\rho = \mathbf{x}_A \wedge \mathbf{x}_B \wedge \mathbf{x}_C = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{l} \\ x_A & y_A & z_A & w_A \\ x_B & y_B & z_B & w_B \\ x_C & y_C & z_C & w_C \end{vmatrix} = [a, b, c : d]^T. \quad (11)$$

It should be noted, that the non-normalized directional vector \mathbf{s} of the line p in E^2 is orthogonal to the normal bivector of the line and it is given as:

$$\mathbf{s} = (X_B - X_A, Y_B - Y_A) = (s_X, s_Y) = (-b, a), \quad \mathbf{n} = (a, b). \quad (12)$$

⁴In this case, the outer product is formally equivalent to the cross product

⁵There is a direct connection with the *geometric product* which is defined as $\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \wedge \mathbf{b}$, i.e. $\mathbf{ab} = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \times \mathbf{b}$ [10].

The line p splits the E^2 plane into two half-planes:

$$F_p(\mathbf{x}) = 0, \quad F_p(\mathbf{x}) = \mathbf{p} \cdot \mathbf{x} = \mathbf{p}^T \mathbf{x} = ax + by + cw, \quad (13)$$

where $w > 0$. If $w \rightarrow \pm\infty$ then the point \mathbf{x} is closed or in infinity. It should be noted that the dot-product (scalar product) is a single instruction on GPU.

3. Proposed clipping algorithm in projective space

The CB algorithm computes the parameter t value using division operation N times. However, only two values are needed, if any. It means, that $N - 2$ computations of the parameter t are unnecessary. Also, reliable detection of “singular or close to singular” cases is difficult and time-consuming.

Let us consider the case, when the convex polygon vertices and points defining the line p are given in projective space, i.e. in homogeneous coordinates with $w \neq 0$. In this case, a conversion of $\mathbf{x}_A, \mathbf{x}_B$ and \mathbf{x}_i to Euclidean space using a division operation is needed, if the CB algorithm is to be used. It means, that $2N + 4$ division operations would be needed for the conversion to Euclidean space.

Let us consider the case, when the polygon vertices are generally given in the projective space (6), i.e. using the homogeneous coordinates, as:

$$\mathbf{x}_i = [x_i, y_i : w_i]^T, \quad i = 0, \dots, N - 1 \quad \text{and} \quad w \neq 0, \quad (14)$$

where w_i are is homogeneous coordinate and points \mathbf{x}_A and \mathbf{x}_B define the line p , resp. line segment to be clipped.

$$\mathbf{x}_A = [x_A, y_A : w_A]^T, \quad \mathbf{x}_B = [x_B, y_B : w_B]^T. \quad (15)$$

Then the given line p , given by the points \mathbf{x}_i and $\mathbf{x}_{i \oplus 1}$, intersects the edge e_i **if and only if**:

$$F_p(\mathbf{x}_i) > 0 \quad \mathbf{or} \quad F_p(\mathbf{x}_{i \oplus 1}) > 0, \quad i = 0, \dots, N - 1, \quad (16)$$

i.e. the points \mathbf{x}_i and $\mathbf{x}_{i \oplus 1}$ are on the opposite sides of the line p ⁶.

In the case of the line convex polygon intersection, two intersected edges are detected, if any. It means, that $N - 2$ division operations are saved and no division operation is needed to find out if the convex polygon edge is intersected by the line p .

In the case when the line p intersects the convex polygon edges e_k and e_l the intersection points can be determined as:

- direct intersection computation using the homogeneous coordinates of points (10),
- using the parametric form of the line p (1), but modified for the projective space.

In both cases, all computations support *vector-vector* operations, and therefore they are convenient for GPU or SSE instruction use.

⁶The operator \oplus means addition modulo N , i.e. $a \oplus b = (a + b) \bmod N$.

3.1. Direct intersection coordinates computation

The direct computation of the intersection points is quite simple, as the intersected edges e_k and e_l have been determined in the previous step (16). Using the outer-product (10) (in this case the cross-product) the points of intersections are given as, if any:

$$\begin{aligned} \mathbf{x}_A &= \mathbf{p} \wedge \mathbf{e}_k \equiv \mathbf{p} \times \mathbf{e}_k, & \mathbf{x}_B &= \mathbf{p} \wedge \mathbf{e}_l \equiv \mathbf{p} \times \mathbf{e}_l, \\ \mathbf{e}_k &= \mathbf{x}_k \wedge \mathbf{x}_{k \oplus 1} = [a_k, b_k : c_k]^T, & \mathbf{e}_l &= \mathbf{x}_l \wedge \mathbf{x}_{l \oplus 1} = [a_l, b_l : c_l]^T, \end{aligned} \quad (17)$$

where $\mathbf{x}_A = [x_A, y_A : w_A]^T$, $\mathbf{x}_B = [x_B, y_B : w_B]^T$ and \oplus means mod N operation.⁷

It should be noted that the dot product and cross products are single instructions on GPU [32].

In the case of a line segment clipping, some additional logical conditions are to be used to keep the line segment orientation and respect to situations, when an end-point of the line segment is already inside the given convex polygon (5) [28,29,30].

3.2. Intersection using parametric form

Using the parametric form is similar to the CB algorithm. It is simple, but the projective representation has to be respected.

There are two possibilities:

- linear interpolation conversion from Euclidean space to the projective space, which leads to the linear parameterization of the parameter t ;
- linear parameterization directly in the projective space, which leads to the non-linear monotonic parameterization of the parameter.

The linear parameterization directly in the projective space has significant advantages in this case, as it is simple and robust. The line π given by two points $\mathbf{x}_A = [x_A, y_A : w_A]^T$, $\mathbf{x}_B = [x_B, y_B : w_B]^T$ and $\mathbf{p} = [a, b : c]^T$ are coefficients of the line p , see Fig. 3.

In the projective space P^2 , the line π is given by two points $\mathbf{x}_A = [x_A, y_A : w_A]^T$, $\mathbf{x}_B = [x_B, y_B : w_B]^T$. It forms with the origin 0_P a plane $\rho : ax + by + cw = 0$, where the point $[0, 0 : 0]^T$ represents a point in infinity.

$$\begin{aligned} \mathbf{x}(\tau) &= \mathbf{x}_A + (\mathbf{x}_B - \mathbf{x}_A) \tau, & \tau &\in (-\infty, +\infty), \\ x(\tau) &= x_A + (x_B - x_A) \tau, & y(\tau) &= y_A + (y_B - y_A) \tau, & w(\tau) &= w_A + (w_B - w_A) \tau. \end{aligned} \quad (18)$$

The interpolation (18) is linear but when results are converted to Euclidean space, the parameterization is non-linear but **monotonic**.

The line p in E^2 is given by two points $\mathbf{X}_A = (X_A, Y_A)$ and $\mathbf{X}_B = (X_B, Y_B)$ as:

$$\mathbf{X}(t) = \mathbf{X}_A + (\mathbf{X}_B - \mathbf{X}_A) t, \quad t \in (-\infty, +\infty). \quad (19)$$

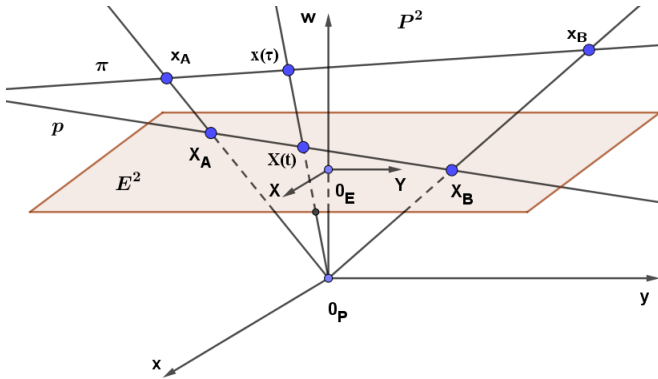


Fig. 3. Parameterization of a line in E^2 and P^2

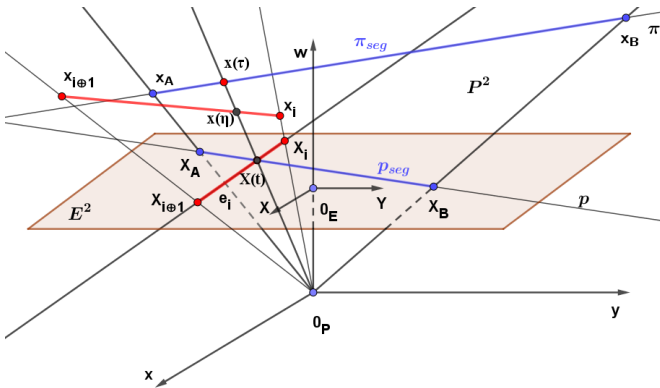


Fig. 4. Intersection of the line π and an edge e_i

It should be noted that $t \neq \tau$, except for $t = \tau = 0$ and $t = \tau = 1$.

In the case of the projective interpolation, see Fig. 4, the system of equations (20) is to be solved:

$$\begin{aligned}
 e_i : \mathbf{p}^T \mathbf{x} &= 0, \quad ax + by + cw = 0, \\
 \pi : \mathbf{x}(\tau) &= \mathbf{x}_A + (\mathbf{x}_B - \mathbf{x}_A) \tau, \quad \tau \in (-\infty, +\infty).
 \end{aligned}
 \tag{20}$$

⁷It should be noted, that instead of using **mod** operation, one "virtual" vertex is added so that $\mathbf{x}_N \equiv \mathbf{x}_0$. This enables us to avoid **mod** operation, which is computationally expensive.

It means, that also homogeneous coordinate w is parametrized (21).

$$\begin{aligned} x(t) &= x_A + (x_B - x_A) \tau = x_A + s_x \tau, \\ y(t) &= y_A + (y_B - y_A) \tau = y_A + s_y \tau, \\ w(t) &= w_a + (w_B - w_A) \tau = w_a + s_w \tau. \end{aligned} \quad (21)$$

This leads to:

$$\begin{aligned} \mathbf{p}^T \mathbf{x}(\tau) &= 0, \quad \mathbf{p}^T \mathbf{x}_A + \mathbf{p}^T (\mathbf{x}_B - \mathbf{x}_A) \tau, \\ \tau &= -\frac{\mathbf{p}^T \mathbf{x}_A}{\mathbf{p}^T \mathbf{s}} \triangleq [\mathbf{p}^T \mathbf{x}_A : \mathbf{p}^T \mathbf{s}]^T = [t : t_w]^T, \end{aligned} \quad (22)$$

where $\mathbf{s} = \mathbf{x}_B - \mathbf{x}_A = [s_x, s_y : s_w]^T$ and τ value can be expressed as a projective scalar value in the projective form as $\tau = [t : t_w]^T$.

Now, the τ values of intersections for the intersected edges are determined. It should be noted, that the τ interpolation is monotonic, but in Euclidean space (X, Y) the interpolation is linear with non-linear parameterization.

It can be seen that division operations are not needed if the result can remain in the projective notation, i.e. no conversion to Euclidean space is required.

The proposed algorithm, with two possible modifications, described above is simple, easy to implement, and convenient for vector-vector implementation. It is based on the projective extension of Euclidean space.

Contrary to the CB algorithm, the algorithm does not require computation of the edges' implicit form, as it uses a separation function.

The first modification is based on the outer product application's implicit representation. The second one is based on the parametric form of the clipped line p , which is more convenient for line segment clipping cases.

It can be seen that in the projective case

- $2N$ division operations are eliminated, if the polygon vertices are given in the projective space, i.e. $w \neq 1$, as the transformations of x_i, y_i to Euclidean space are not needed,
- there is no need to compute edges' coefficients, i.e. a_i, b_i, c_i ,
- $N - 2$ division operations are saved during the run-time and if the intersection points can remain in the projective representation, *no division operation is needed* at all.

It leads to significant improvement in robustness and with additional speed-up as vector-vector operations can be used.

4. Conclusion

This contribution presents a new fully projective algorithm, based on the Cyrus-Beck algorithm, for a line and line segment clipping by a convex polygon using the *vector-vector* operations and supporting GPU implementation, resp. SSE/AVX instructions.

The presented approach eliminates $2N$ division operations in preprocessing of the polygon edges if the polygon vertices are given in the projective space and $N - 2$ division operations in the run-time. It also increases the numerical robustness especially in cases, when the given line is parallel or close to parallel to an edge of the convex polygon.

If the computed results can remain in the projective space, i.e. the conversion to the Euclidean space is not needed, *no division operation is required* by the proposed algorithm. The proposed algorithm can be extended to the E^3 case, i.e. line-convex polyhedron intersection if intersections' points are computed using the parametric form of the given line; however, instead of the separation line two orthogonal planes, which define the clipped line p have to be used, similarly as in [19]. For a deeper study of intersection algorithms, a reader is advised to read "A brief survey of intersection and clipping algorithms" [31].

In future work, the proposed algorithm is to be analyzed from the E^3 case perspective, i.e. line and line segment clipping by a convex polyhedron using the Plücker coordinates [11, 27, 34].

Acknowledgment

The author thanks recent students and colleagues at the University of West Bohemia, Pilsen, Zhejiang University, Hangzhou and Shandong University, Jinan in China for their recent suggestions and constructive comments.

Thanks also belong to the anonymous reviewers, as their comments and hints helped to improve this paper and to several authors of recently published relevant papers [31] for sharing their views and hints provided.

Appendix

The following relevant books are recommended to a reader:

- Salomon, D.: The Computer Graphics Manual [13],
- Salomon, D.: Computer Graphics and Geometric Modeling [12],
- Agoston, M. K.: Computer Graphics and Geometric Modelling: Mathematics [2],
- Agoston, M. K.: Computer Graphics and Geometric Modelling: Implementation & Algorithms [1],
- Lengyel, E.: Mathematics for 3D Game Programming and Computer Graphics [10],
- Foley, J. D., van Dam, A., van Dam, A., van Dam, A., Feiner, S., Hughes, J. F.: Computer graphics – Principles and Practice [7],

- Hughes, J. F., van Dam, A., McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S. K., Akeley, K.: Computer Graphics – Principles and Practice [8],
- Ferguson, R. S.: Practical Algorithms for 3D Computer Graphics [6],
- Shirley, P., Marschner, S.: Fundamentals of Computer Graphics [15],
- Theoharis, T., Platis, N., Papaioannou, G., Patrikalakis, N.: Graphics and Visualization: Principles & Algorithms [35],
- Comninos, P.: Mathematical and Computer Programming Techniques for Computer Graphics [4],
- Schneider, P. J., Eberly, D. H.: Geometric Tools for Computer Graphics [14].

References

- [1] M. K. Agoston. *Computer Graphics and Geometric Modelling: Implementation & Algorithms*. Springer-Verlag, Berlin, Heidelberg, 2004. doi:10.1007/b138805.
- [2] M. K. Agoston. *Computer Graphics and Geometric Modelling: Mathematics*. Springer-Verlag, Berlin, Heidelberg, 2005. doi:10.1007/b138899.
- [3] A. Arokiasamy. Homogeneous coordinates and the principle of duality in two dimensional clipping. *Computers and Graphics*, 13(1):99–100, 1989. doi:10.1016/0097-8493(89)90045-9.
- [4] P. Comninos. *Mathematical and Computer Programming Techniques for Computer Graphics*. Springer-Verlag, Berlin, Heidelberg, 2005. doi:10.1007/978-1-84628-292-8.
- [5] M. Cyrus and J. Beck. Generalized two- and three-dimensional clipping. *Computers and Graphics*, 3(1):23–28, 1978. doi:10.1016/0097-8493(78)90021-3.
- [6] R. S. Ferguson. *Practical Algorithms for 3D Computer Graphics*. A. K. Peters, Ltd., USA, 2nd edn., 2013. doi:10.1201/b16333.
- [7] J. D. Foley, A. van Dam, S. Feiner, and J. F. Hughes. *Computer Graphics – Principles and Practice*. Addison-Wesley, 2nd edn., 1990.
- [8] J. F. Hughes, A. van Dam, M. McGuire, D. F. Sklar, J. D. Foley, et al. *Computer Graphics – Principles and Practice*. Addison-Wesley, 3rd edn., 2014.
- [9] M. Johnson. Proof by duality: or the discovery of “new” theorems. *Mathematics Today*, December:138–153, 1996.
- [10] E. Lengyel. *Mathematics for 3D Game Programming and Computer Graphics*. Course Technology Press, Boston, MA, USA, 3rd edn., 2011.
- [11] N. Platis and T. Theoharis. Fast ray-tetrahedron intersection using Plücker coordinates. *Journal of Graphics Tools*, 8(4):37–48, 2003. doi:10.1080/10867651.2003.10487593.
- [12] D. Salomon. *Computer Graphics and Geometric Modeling*. Springer-Verlag, Berlin, Heidelberg, 1st edn., 1999.
- [13] D. Salomon. *The Computer Graphics Manual*. Springer, 2011. doi:10.1007/978-0-85729-886-7.
- [14] P. J. Schneider and D. H. Eberly. *Geometric Tools for Computer Graphics*. The Morgan Kaufmann Series in Computer Graphics. Morgan Kaufmann, San Francisco, 2003. doi:10.1016/B978-1-55860-594-7.50025-4.
- [15] P. Shirley and S. Marschner. *Fundamentals of Computer Graphics*. A. K. Peters, Ltd., USA, 3rd edn., 2009. doi:10.1201/9781439865521.



- [16] V. Skala. An efficient algorithm for line clipping by convex polygon. *Computers and Graphics*, 17(4):417–421, 1993. doi:10.1016/0097-8493(93)90030-D.
- [17] V. Skala. $O(\lg N)$ line clipping algorithm in E2. *Computers and Graphics*, 18(4):517–524, 1994. doi:10.1016/0097-8493(94)90064-7.
- [18] V. Skala. An efficient algorithm for line clipping by convex and non-convex polyhedra in E3. *Computer Graphics Forum*, 15(1):61–68, 1996. doi:10.1111/1467-8659.1510061.
- [19] V. Skala. A fast algorithm for line clipping by convex polyhedron in E3. *Computers and Graphics (Pergamon)*, 21(2):209–214, 1997. doi:10.1016/s0097-8493(96)00084-2.
- [20] V. Skala. A new approach to line and line segment clipping in homogeneous coordinates. *Visual Computer*, 21(11):905–914, 2005. doi:10.1007/s00371-005-0305-3.
- [21] V. Skala. Duality and intersection computation in projective space with GPU support. In: *Latest Trends on Applied Mathematics, Simulation, Modelling – Proc. 4th International Conference on Applied Mathematics, Simulation, Modelling (ASM'10)*, pp. 66–71. Corfu, Greece, 2010. <http://hdl.handle.net/11025/11797>.
- [22] V. Skala. Duality, barycentric coordinates and intersection computation in projective space with GPU support. *WSEAS Transactions on Mathematics*, 9(6):407–416, 2010. http://afrodita.zcu.cz/~skala/PUBL/PUBL_2010/2010_NAUN-journal.pdf.
- [23] V. Skala. Geometry, duality and robust computation in engineering. *WSEAS Transactions on Computers*, 11(9):275–293, 2012.
- [24] V. Skala. S-clip E2: A new concept of clipping algorithms. *SIGGRAPH Asia Posters, SA*, pp. 1–2, 2012. doi:10.1145/2407156.2407200.
- [25] V. Skala. Algorithms for line and plane intersection with a convex polyhedron with $O(\sqrt{N})$ expected complexity in E3. In: *SIGGRAPH Asia 2014 Posters, SA '14*. Association for Computing Machinery, New York, NY, USA, 2014. doi:10.1145/2668975.2668976.
- [26] V. Skala. Geometric transformations and duality for virtual reality and haptic systems. *Communications in Computer and Information Science*, 434 PART I:642–647, 2014. doi:10.1007/978-3-319-07857-1_113.
- [27] V. Skala. Projective geometry, duality and plücker coordinates for geometric computations with determinants on GPUs. *ICNAAM 2017*, 1863, 2017. doi:10.1063/1.4992684.
- [28] V. Skala. Optimized line and line segment clipping in E2 and geometric algebra. *Ann. Math. Inf.*, 52:199–215, 2020. doi:10.33039/ami.2020.05.001.
- [29] V. Skala. A new coding scheme for line segment clipping in E2. *Lecture Notes in Computer Science*, LNCS-accepted for publication ICCSA 2021:16–29, 2021. doi:10.1007/978-3-030-86976-2_2.
- [30] V. Skala. A novel line convex polygon clipping algorithm in E2 with parallel processing modification. *Lecture Notes in Computer Science*, LNCS 12953 ICCSA 2021:3–15, 2021. doi:10.1007/978-3-030-86976-2_1.
- [31] V. Skala. A brief survey of clipping and intersection algorithms with a list of references (including triangle-triangle intersections). *Informatica (Lithuania)*, 34(1):169–198, 2023. doi:10.15388/23-INFOR508.
- [32] V. Skala, S. A. A. Karim, and E. A. Kadir. Scientific computing and computer graphics with GPU: Application of projective geometry and principle of duality. *International Journal of Mathematics and Computer Science*, 15(3):769–777, 2020. <http://ijmcs.future-in-tech.net/15.3/R-Skala-AbdulKarim.pdf>.

- [33] V. Skala and M. Kuchař. The hash function and the principle of duality. In: *Proc. Computer Graphics International Conference (CGI'01)*, pp. 167–174. Hong Kong, China, 2001. doi:[10.1109/CGI.2001.934671](https://doi.org/10.1109/CGI.2001.934671).
- [34] V. Skala and M. Smolik. A new formulation of Plücker coordinates using projective representation. In: *Proc. 2018 5th International Conference on Mathematics and Computers in Sciences and Industry (MCSI 2018)*, pp. 52–56. Corfu, Greece, 2018. doi:[10.1109/MCSI.2018.00020](https://doi.org/10.1109/MCSI.2018.00020).
- [35] T. Theoharis, N. Platis, G. Papaioannou, and N. Patrikalakis. *Graphics and Visualization: Principles & Algorithms (1st ed.)*. A K Peters/CRC Press, 2008. doi:[10.1201/b10676](https://doi.org/10.1201/b10676).



Vaclav Skala is a professor at the Dept. of Computer Science and Engineering, University of West Bohemia (UWB), Pilsen [Plzen]. He has been with Brunel University in London, U.K., Gavle University, Sweden, Moscow Power Engineering Institute, Russia, etc. His current research is targeted at fundamental algorithms for computer graphics, geometric algebra, meshless (meshfree) methods for scalar and vector fields interpolation and approximation, and applied mathematics. He is the Editor-in-Chief of the Journal of WSCG and Computer Science Research Notes [CSRN]. Vaclav Skala has published over 160+ research-indexed papers with more than 800+(WoS/Publons), 1300+(Scopus), and 2600+(Scholar Google) citations.

A FRAMEWORK FOR FLUID MOTION ESTIMATION USING A CONSTRAINT-BASED REFINEMENT APPROACH

Hirak Doshi * and Uday Kiran Nori 

Department of Mathematics and Computer Science

Sri Sathya Sai Institute of Higher Learning, Andhra Pradesh, India

Email: hd92sssihl.psn@gmail.com, nudaykiran@sssihl.edu.in

**Corresponding author: Hirak Doshi (hd92sssihl.psn@gmail.com)*

Abstract Physics-based optical flow models have been successful in capturing the deformities in fluid motion arising from digital imagery. However, a common theoretical framework analyzing several physics-based models is missing. In this regard, we formulate a general framework for fluid motion estimation using a constraint-based refinement approach. We demonstrate that for a particular choice of constraint, our results closely approximate the classical continuity equation-based method for fluid flow. This closeness is theoretically justified by augmented Lagrangian method in a novel way. The convergence of Uzawa iterates is shown using a modified bounded constraint algorithm. The mathematical well-posedness is studied in a Hilbert space setting. Further, we observe a surprising connection to the Cauchy-Riemann operator that diagonalizes the system leading to a diffusive phenomenon involving the divergence and the curl of the flow. Several numerical experiments are performed and the results are shown on different datasets. Additionally, we demonstrate that a flow-driven refinement process involving the curl of the flow outperforms the classical physics-based optical flow method without any additional assumptions on the image data.

Keywords: fluid motion estimation, evolutionary PDE, image processing, augmented Lagrangian, bounded constraint algorithm.

Mathematics Subject Classification 2020: 35J47, 35J50, 35Q68.

1. Introduction

Variational models for motion estimation have always been one of the central topics in mathematical image processing. Since the seminal work of Horn and Schunck [13] on the variational approach to optical flow motion estimation, many in-depth studies on this topic have been done by developing different variational models of optical flow to obtain useful insights into motion estimation (e.g. [2, 12, 18, 20]). Many of these literature works on motion estimation have been focused on the constancy assumption, e.g., the brightness constancy leading to an algebraic equation, in (u, v) the motion components, called the optical flow constraint (OFC):

$$f_\tau + f_x u + f_y v = 0, \quad (1)$$

where $f(x, y, \tau)$ is the image sequence $f : \Omega \times [0, \infty) \rightarrow \mathbb{R}$ for an open bounded set $\Omega \subset \mathbb{R}^2$, (x, y) are the spatial coordinates and τ is the time variable. However, constancy assumptions can't reflect the reality of actual motion because deformation effects of fluid,

illumination variations, perspective changes, poor contrast etc. would directly affect the important motion parameters. For this reason, physics-dependent motion estimation algorithms have been widely investigated.

1.1. Related Works

A relatively recent work of Corpetti et al. [4,5,6] used the image-integrated version of the continuity equation from fluid dynamics. Since then, there has been a lot of attention on studying physics-based motion estimation. Estimates on the conservation of mass for a fluid in a digital image sequence were discussed by Wildes et al. [21]. The image intensity was obtained as an average of the object density, with the incident light parallel to z -axis such that the 2D projection of the image intensity can be captured as

$$f(x, y, \tau) = \int_{z_1}^{z_2} \rho(x, y, z, \tau) dz.$$

Further using fluid mechanics models, Liu et al. [19] provided a rigorous framework for fluid flow by deriving the projected motion equations. Here the optical flow is proportional to the path-averaged velocity of fluid or particles weighted with a relevant field quantity. In this approach the authors assume that the control surface is planar, there is no particle diffusion by a molecular process, and the rate of accumulation of the particle in laser sheet illuminated volume is neglected, the equations, after neglecting these terms one obtains the continuity equation again. Luttmann et al. [16] computed the potential (resp. stream) function directly by assuming that the flow estimate is the gradient of a potential function (resp. symplectic gradient of a stream function). To overcome the limitations of the global smoothness regularization for fluid motion estimation, Corpetti et al. [5] proposed a second-order div-curl regularization for a better understanding of intrinsic flow structures. A detailed account of various works in fluid flow based on physics is given in [11]. Here the authors observed a physical meaning associated with the terms of the continuity equation (CEC):

$$\underbrace{f_\tau + f_x u + f_y v}_{\text{(a)}} + \underbrace{f(u_x + v_y)}_{\text{(b)}} = 0, \quad (2)$$

where **(a)** corresponds to the brightness constancy given in (1) and **(b)** is the non-conservation term due to loss of particles. Furthermore, a divergence-free approximation of the equation (2) can be obtained by setting the term **(b)** to zero. It is thus natural to study the effect of the term **(b)** in extracting the inherent fluid properties of the flow.

1.2. Our Contribution

In the current work, we develop a generic physics-based framework for fluid motion estimation for capturing intrinsic spatial characteristics and vorticities. It consists of a

two-step technique to compute fluid motion from a sequence of scalar fields or illuminated particles transported by the flow. The technique proposed consists in filtering with an appropriate semi-group the divergence of an initial velocity field estimated through a classical Horn and Schunck (HS) estimator. The vorticity of the initial solution is kept constant while the flow is transformed through its divergence in order to obey a continuity equation for the brightness data which has been used in several optical flow estimators dedicated to fluid flow.

In particular, our method uses a constraint-based refinement approach. As mentioned above, in the two-step technique, the first requirement is an initial flow estimate (u_0, v_0) that obeys the classical optical flow principles like brightness constancy and pixel correspondence. This estimate may not be able to capture the underlying geometric features of the fluid flow. The main idea is to perform a refinement over this crude estimate to capture precise flow structures driven by additional constraints specific to applications. As a concrete example, we choose the initial estimate coming from the Horn and Schunck model [13]. It is well-known that this model is not well-suited for fluid motion estimation. In fact the global smoothness regularization damps both the divergence and vorticity of the motion field. We show in particular how this model can be adapted and refined through our approach. A special feature of our model is the diagonalization by the Cauchy-Riemann operator leading to a diffusion on the curl and a multiplicative perturbation of the laplacian on the divergence of the flow.

There are two main advantages of our method. Firstly, from a theoretical perspective it provides us with an evolutionary PDE setup which allows a rigorous mathematical framework for the well-posedness discussion. Secondly, the contractive semigroup structure on the divergence leads to a simpler numerical analysis. A modified bounded constraint algorithm [17] is employed to theoretically show the convergence of the dual variable introduced by the augmented Lagrangian formulation. The inner iterations of the algorithm use the contractive semigroup of the elliptic term. This approach thus allows us to build a quantitative connection between the optical flow and the fluid flow for various flow visualizations which is often a key problem.

The paper is organized as follows. In Section 2, we give a detailed description of our model. Section 3 is devoted to the mathematical framework. Here we discuss the mathematical well-posedness and the regularity of the solutions. We also show the diagonalization process under the application of the Cauchy-Riemann operator. In Section 4, we show how for a particular choice of additional constraints, our model closely approximates the continuity equation model using a modified augmented Lagrangian formulation. We also employ the bounded constraint algorithm to show the convergence of the Uzawa iterates. Finally, in Section 5, we show our results on different datasets. Section 3.3 concludes the paper.

2. Description of the Model

2.1. General Formulation

Our general formulation is given as

$$J_{\mathbf{R}}(\mathbf{u}) = \beta \int_{\Omega} \phi(f) \psi(\nabla \mathbf{u}) + \alpha \int_{\Omega} \{|\nabla u|^2 + |\nabla v|^2\}, \quad (3)$$

where the constants α, β are weight parameters, the function ψ depends on the components of the flow and its derivatives which essentially captures the underlying geometric structures and the function ϕ corresponds to an image-dependent weight term. A few possible combinations are summarized in Table 1.

As seen from the table, the function ϕ dictates whether the refinement process is image-driven or flow-driven. When $\phi(f) = 1$, there is no influence of the image data on the additional constraint. As a result, the refinement process is completely flow-driven. We assume that both the functions ϕ and ψ are real-valued smooth functions. Moreover, we assume $\phi(f)$ to be a monotone-increasing function. The first term of $J_{\mathbf{R}}(\mathbf{u})$ captures the non-conservation term that violates the constancy assumptions and the second term is the L^2 regularization which governs the diffusion phenomena. In the current work, we particularly focus on the case where $\psi = (\nabla \cdot \mathbf{u})^2$, i.e. where the function ψ penalizes the divergence of the flow. In this case, the refinement functional becomes

$$J_{\mathbf{R}}(\mathbf{u}) = \beta \int_{\Omega} \phi(f) (\nabla \cdot \mathbf{u})^2 + \alpha \int_{\Omega} \{|\nabla u|^2 + |\nabla v|^2\}. \quad (4)$$

2.2. Additional Constraint Involving the Curl of the Flow

In Table 1 we have suggested two such choices for ψ , one penalizing the divergence of the flow and the other penalizing the curl. The operator $\nabla_H := (\partial_y, -\partial_x)$ is called the orthogonal gradient, also referred to as the symplectic gradient in the literature [16].

Tab. 1. Some choices for the functions ϕ and ψ

| $\phi(f)$ | $\psi(\nabla \mathbf{u})$ | Nature of the model |
|-----------|---------------------------------|--|
| f^2 | $(\nabla \cdot \mathbf{u})^2$ | Anisotropic, image-driven, penalizing divergence of the flow |
| 1 | $(\nabla \cdot \mathbf{u})^2$ | Isotropic, flow-driven, penalizing divergence of the flow |
| f^2 | $(\nabla_H \cdot \mathbf{u})^2$ | Anisotropic, image-driven, penalizing curl of the flow |
| 1 | $(\nabla_H \cdot \mathbf{u})^2$ | Isotropic, flow-driven, penalizing curl of the flow |

This geometric constraint captures the rotational aspects of the flow better. In this work, we will particularly demonstrate that a flow-driven refinement process involving the curl of the flow outperforms the classical physics-based optical flow method without any additional assumptions on the image data.

3. Mathematical Framework

3.1. Diagonalization of the System

The associated system of PDEs for the variational formulation is given as:

$$\begin{cases} \frac{\partial u}{\partial t} = \Delta u + a_0 \frac{\partial}{\partial x} [\phi(f)(u_x + v_y)] \text{ in } \Omega \times (0, \infty), \\ \frac{\partial v}{\partial t} = \Delta v + a_0 \frac{\partial}{\partial y} [\phi(f)(u_x + v_y)] \text{ in } \Omega \times (0, \infty), \\ u(x, y, 0) = u_0 \text{ in } \Omega, \\ v(x, y, 0) = v_0 \text{ in } \Omega, \\ u = 0 \text{ on } \partial\Omega \times (0, \infty), \\ v = 0 \text{ on } \partial\Omega \times (0, \infty). \end{cases} \quad (5)$$

Here (u_0, v_0) is the initial flow estimate obtained from the pixel correspondence, $a_0 = \beta/\alpha$ is a positive constant. Since there is no pixel motion at the boundary, it is natural to work with Dirichlet boundary conditions. We will show that the system (5) can be diagonalized by an application of Cauchy-Riemann operator. This special feature is intriguing as well as of great advantage for later analysis. Let us first rewrite the system (5) as

$$\frac{\partial \mathbf{u}}{\partial t} = A\mathbf{u}, \quad (6)$$

where

$$\frac{\partial \mathbf{u}}{\partial t} = \begin{bmatrix} \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial t} \end{bmatrix}, \quad A\mathbf{u} = \begin{bmatrix} \Delta u + a_0 \frac{\partial}{\partial x} [\phi(f)(u_x + v_y)] \\ \Delta v + a_0 \frac{\partial}{\partial y} [\phi(f)(u_x + v_y)] \end{bmatrix}.$$

As we are in the Sobolev setting, the derivatives are taken in a distributional sense. Thus, a key observation is that the order of derivatives can be interchanged. Let us denote by R the Cauchy-Riemann operator matrix

$$R = \begin{bmatrix} \partial_y & -\partial_x \\ \partial_x & \partial_y \end{bmatrix}.$$

Acting R on both sides of (6) leads to

$$R\left(\frac{\partial \mathbf{u}}{\partial t}\right) = R\mathbf{A}\mathbf{u}.$$

This leads to the following transformation of the original coupled system

$$\begin{aligned} \begin{bmatrix} \partial_y & -\partial_x \\ \partial_x & \partial_y \end{bmatrix} \begin{bmatrix} \frac{\partial u}{\partial t} \\ \frac{\partial v}{\partial t} \end{bmatrix} &= \begin{bmatrix} \partial_y & -\partial_x \\ \partial_x & \partial_y \end{bmatrix} \begin{bmatrix} \Delta u + a_0 \frac{\partial}{\partial x} [\phi(f)(u_x + v_y)] \\ \Delta v + a_0 \frac{\partial}{\partial y} [\phi(f)(u_x + v_y)] \end{bmatrix} \\ &= \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \circ k \end{bmatrix} \begin{bmatrix} \partial_y & -\partial_x \\ \partial_x & \partial_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \end{aligned}$$

where, with a slight abuse of notation we denote k for the function $1 + a_0\phi(f)$ and for the multiplicative operator $x \mapsto kx$. Since ϕ is a bounded function and $a_0 > 0$, the multiplicative term k is bounded and strictly positive. We have thus obtained the following decoupling:

$$\frac{\partial}{\partial t}(R\mathbf{u}) = D R\mathbf{u}, \quad (7)$$

where

$$D = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \circ k \end{bmatrix}, \quad R = \begin{bmatrix} \partial_y & -\partial_x \\ \partial_x & \partial_y \end{bmatrix}.$$

The application of the Cauchy-Riemann operator R on the system has resulted in the following relation

$$D = RAR^{-1}.$$

The operator A has been diagonalized by the matrix R . The decoupled system (7) thus takes the following form:

$$\begin{cases} \frac{\partial \xi}{\partial t} = \Delta \xi, \\ \frac{\partial \zeta}{\partial t} = \Delta(k\zeta), \end{cases} \quad (8)$$

where $\xi := u_y - v_x$ is the curl and $\zeta := u_x + v_y$ is the divergence of the flow and $k = 1 + a_0\phi(f)$ is a multiplicative factor.

3.2. Multiplicative Perturbation of the Laplacian

Let us consider the divergence equation from (8)

$$\frac{\partial \zeta}{\partial t} = \Delta(k\zeta), \tag{9}$$

where $k = 1 + a_0\phi(f)$. We make a change of variable $\eta = k\zeta$. This transformation leads to the equation:

$$\frac{\partial \eta}{\partial t} = k\Delta\eta. \tag{10}$$

The operator $k\Delta$ is the multiplicative perturbation of the laplacian. It arises in many physical phenomena e.g. in the theory of wave propagation in non-homogeneous media. The operator has been studied in appropriate weighted function spaces see [1, 7]. In our case the multiplicative perturbation $k\Delta$ leads to an image driven perturbation because of the k factor which depends on the image f . The authors in [1] derive an approximation for the kernel of the associated semigroups by positive linear integral operators

$$G_m(f)(x) = \left(\frac{m}{4\pi k(x)}\right)^{n/2} \int_{\mathbb{R}^n} f(y) \exp\left(-\frac{m}{4k(x)}|x-y|^2\right) dy. \tag{11}$$

Using the above kernel one can design an appropriate stencil for convolution. It is also interesting to note that the stencil size varies with respect to the image intensity. This aspect we will discuss in a forthcoming paper. Let us now consider the Gaussian kernel associated with the operator $k\Delta$

$$G_k(x, t) := \frac{1}{4\pi k(x)t} \exp\left(-\frac{|x|^2}{4k(x)t}\right).$$

The perturbation k plays an important role in controlling the rate of diffusion. If k is large then the Gaussian becomes broader and shorter and if it is small then the Gaussian is thinner and taller. Since Ω is bounded, the perturbation k is bounded. Hence there exists $a_1, a_2 > 0$ such that $a_1 \leq k(x) \leq a_2$. Using this fact, we can obtain the following bound on $G_k(x, t)$.

Lemma 3.1. *Let $n = 2$. Then*

$$\|G_k(x, t)\|_p \leq C_k t^{(1-p)/p},$$

where the constant

$$C_k = \frac{1}{4\pi a_1} \left(\frac{4\pi a_2}{p}\right)^{1/p}.$$

3.3. Wellposedness and Regularity

Let us now consider the abstract IVP associated with the first equation in (8):

$$\begin{cases} \frac{d\xi}{dt} + A_1\xi = 0 \text{ on } [0, \infty), \\ \xi(0) = \xi_0 \in L^2(\Omega). \end{cases} \tag{12}$$

where the initial data

$$\xi_0 = \partial_y u_0 - \partial_x v_0,$$

and (u_0, v_0) is the Horn and Schunck optical flow. Here $A_1 : D(A_1) \rightarrow \mathcal{H}_1$ is an (unbounded) operator

$$\begin{cases} D(A_1) = \{\xi \in H^2(\Omega) \cap H_0^1(\Omega) : A_1\xi \in L^2(\Omega)\}, \\ \mathcal{H}_1 = L^2(\Omega), \\ A_1\xi := -\Delta\xi. \end{cases}$$

The operator A_1 is maximal monotone and symmetric. Hence it is self adjoint. For the well-posedness of the problem (12) we refer to (Theorem 7.7 and 10.1 in [3]). Similarly, the abstract IVP for the second equation becomes

$$\begin{cases} \frac{d\eta}{dt} + A_2\eta = 0 \text{ on } [0, \infty), \\ \eta(0) = \eta_0 \in L^2(\Omega). \end{cases} \tag{13}$$

where the initial data

$$\eta_0 = k(\partial_x u_0 + \partial_y v_0),$$

is the weighted divergence of the Horn and Schunck optical flow. Here $A_2 : D(A_2) \rightarrow \mathcal{H}_2$ is the (unbounded) operator

$$\begin{cases} D(A_2) = \{\eta \in H^2(\Omega) \cap (H_0^1)_k(\Omega) : A_2\eta \in L_k^2(\Omega)\}, \\ \mathcal{H}_2 = L_k^2(\Omega), \\ A_2\eta = -k\Delta\eta. \end{cases}$$

Also $\eta_0 \in L^2(\Omega)$. Here the operator $k\Delta$ is a multiplicative perturbation of the laplacian where k is bounded and strictly positive since ϕ is a monotone increasing function. The

problem will be studied in weighted Sobolev space \mathcal{H}_2 following a similar approach as in [7]. The space \mathcal{H}_2 is a Hilbert space equipped with the inner product

$$\langle w_1, w_2 \rangle_{\mathcal{H}_2} := \int_{\Omega} \frac{1}{k} w_1 w_2 \, dx$$

and the norm

$$\|w\|_{\mathcal{H}_2}^2 = \int_{\Omega} \frac{1}{k} |w|^2 \, dx.$$

Similarly the Hilbert space $\mathcal{H}_3 := (H_0^1)_k(\Omega)$ has the norm [7]

$$\|w\|_{\mathcal{H}_3}^2 := \|w\|_{\mathcal{H}_2}^2 + \|\nabla w\|_{\mathcal{H}_1}^2 = \int_{\Omega} \left(\frac{1}{k} |w|^2 + |\nabla w|^2 \right).$$

The operator A_2 is symmetric and maximal monotone. It is also interesting to note that in our context the weight term k in \mathcal{H}_2 is actually dependent on the image f - bringing in an anisotropy into the discussion. Thus \mathcal{H}_2 is an image dependent Sobolev space. When $\phi(f)$ is a constant function, i.e. the case where the refinement is independent of the image, the norms $\|\cdot\|_{\mathcal{H}_1}$ and $\|\cdot\|_{\mathcal{H}_2}$ coincide upto a constant. In our context, as the values of the image are bounded, k is a bounded function. Also in the pre-processing stage since the images are smoothed with a Gaussian filter we can further assume that k is smooth. We will prove a result on the regularity of the solution for any non-zero time in the diffusion process.

Theorem 3.1. *Let $\eta_0 \in \mathcal{H}_1$. Then the solution of the problem (13) satisfies*

$$\eta \in C^1((0, \infty), \mathcal{H}_2) \cap C([0, \infty), H^2(\Omega) \cap \mathcal{H}_3).$$

For any $\epsilon > 0$ we also have

$$\eta \in C^\infty([\epsilon, \infty) \times \bar{\Omega}). \tag{14}$$

Moreover, $\eta \in L^2((0, \infty), \mathcal{H}_3)$, and

$$\frac{1}{2} \|\eta(T)\|_{\mathcal{H}_1}^2 + \int_0^T \|\nabla \eta(t)\|_{\mathcal{H}_1}^2 \, dt = \frac{1}{2} \|\eta_0\|_{\mathcal{H}_2}^2 \tag{15}$$

holds for $T > 0$.

Proof. We only show the energy estimates as the remaining part of the proof follows very closely to the proof of theorem 10.1 in [3]. It is clear that the operator A_2 is symmetric and maximal monotone. Hence it is self-adjoint. Therefore, by Theorem 7.7 in [3], we have

$$\eta \in C^1((0, \infty), \mathcal{H}_2) \cap C([0, \infty), H^2(\Omega) \cap \mathcal{H}_3).$$

Define $\sigma(t) = \frac{1}{2}\|\eta(t)\|_{\mathcal{H}_2}^2$. Since $\eta \in C^1((0, \infty), \mathcal{H}_2)$ it is clear that σ is C^1 on $(0, \infty)$. Therefore

$$\begin{aligned} \sigma'(t) &= \left\langle \eta(t), \frac{d\eta}{dt}(t) \right\rangle_{\mathcal{H}_2} \\ &= \left\langle \eta(t), k\Delta\eta \right\rangle_{\mathcal{H}_2} \\ &= \left\langle \eta(t), \Delta\eta \right\rangle_{\mathcal{H}_1} \\ &= -\|\nabla\eta(t)\|_{\mathcal{H}_1}^2. \end{aligned}$$

Integrating from ε to T where $0 < \varepsilon < T < \infty$ we get

$$\sigma(T) - \sigma(\varepsilon) = -\int_{\varepsilon}^T \|\nabla\eta(t)\|_{\mathcal{H}_1}^2 dt.$$

Again as $\eta \in C((0, \infty), \mathcal{H}_2(\Omega))$ we have $\sigma(\varepsilon) \rightarrow \sigma(0) = \frac{1}{2}\|\eta_0\|_{\mathcal{H}_2}^2$ as $\varepsilon \rightarrow 0$. Therefore in the limiting case we obtain

$$\sigma(T) + \int_0^T \|\nabla\eta(t)\|_{\mathcal{H}_1}^2 dt = \frac{1}{2}\|\eta_0\|_{\mathcal{H}_2}^2,$$

and (15) holds. Integrating the Hilbert space $(H_0^1)_k(\Omega)$ norm defined above from 0 to T we get

$$\begin{aligned} \int_0^T \|\eta(t)\|_{(H_0^1)_k}^2 dt &= \int_0^T \|\eta(t)\|_{\mathcal{H}_2}^2 dt + \int_0^T \|\nabla\eta(t)\|_{\mathcal{H}_1}^2 dt \\ &= 2 \int_0^T \sigma(t) dt + \int_0^T \|\nabla\eta(t)\|_{\mathcal{H}_1}^2 dt \\ &\leq 2 \int_0^T \sigma(t) dt + \frac{1}{2}\|\eta_0\|_{\mathcal{H}_2}^2. \end{aligned}$$

This shows that

$$\eta \in L^2((0, \infty), \mathcal{H}_3). \quad \square$$

4. A special Case: Approximating the Continuity Equation Model

In this section we show that for a specific choice of the additional constraint $\phi(f) = f^2$, $\psi = (\nabla \cdot \mathbf{u})^2$, our model closely approximates the CEC based model. We justify this theoretically using the modified augmented Lagrangian framework.

4.1. The Augmented Lagrangian Framework

Let $\mathcal{V} = H^1(\Omega) \times H^1(\Omega)$ and $\mathcal{H} = L^2(\Omega)$ denote the Hilbert spaces with the respective norms $\|\cdot\|_{\mathcal{V}}, \|\cdot\|_{\mathcal{H}}$. For simplicity, we fix $\alpha = \beta = 1$. Recall that our refinement model evolves over an initial estimate for which the pixel correspondence problem is already solved upto a certain level. This means that the optical flow constraint (OFC) is satisfied by the flow estimate. Taking this into account we recast the variational problem to a constrained minimization problem:

$$\min_{\mathbf{u}} J_R(\mathbf{u}) = \int_{\Omega} (f \nabla \cdot \mathbf{u})^2 + \int_{\Omega} (|\nabla u|^2 + |\nabla v|^2), \quad (16)$$

subject to the constraint

$$B\mathbf{u} := \nabla f \cdot \mathbf{u} = -f_t =: c. \quad (17)$$

where $\mu_R > 0$ and $\lambda_1 \in \mathcal{H}$ is the Lagrange multiplier. The associated augmented Lagrangian for the problem (16)-(17) is:

$$\mathcal{L}_{\mu_R}(\mathbf{u}, \lambda_1) = J_R(\mathbf{u}) + \frac{\mu_R}{2} \|B\mathbf{u} - c\|^2 + \langle \lambda_1, B\mathbf{u} - c \rangle. \quad (18)$$

Since OFC is a divergence-free approximation of the continuity equation data term, a similar constrained minimization problem can be considered:

$$\min_{\mathbf{u}} J_C(\mathbf{u}) = \int_{\Omega} (f_t + \nabla \cdot (f\mathbf{u}))^2 + \int_{\Omega} (|\nabla u|^2 + |\nabla v|^2), \quad (19)$$

subject to the constraint

$$B\mathbf{u} = c. \quad (20)$$

The associated augmented Lagrangian for the problem (19)-(20) is:

$$\mathcal{L}_{\mu_C}(\mathbf{u}, \lambda_1) = J_C(\mathbf{u}) + \frac{\mu_C}{2} \|B\mathbf{u} - c\|^2 + \langle \lambda_1, B\mathbf{u} - c \rangle. \quad (21)$$

We observe that

$$J_C(\mathbf{u}) = J_{HS}(\mathbf{u}) + J_R(\mathbf{u}) + K(\mathbf{u}),$$

where J_{HS} denotes the Horn and Schunck functional [13] and

$$K(\mathbf{u}) = 2 \int_{\Omega} (B\mathbf{u} - c)(f \nabla \cdot \mathbf{u}).$$

By the Cauchy-Schwarz inequality we have $|K(\mathbf{u})|^2 \leq 2\|B\mathbf{u} - c\|_{L^2}^2 \|f \nabla \cdot \mathbf{u}\|_{L^2}^2$. Thus,

$$J_C(\mathbf{u}) = J_R(\mathbf{u}) + O(\sqrt{\epsilon}) \quad \text{whenever} \quad \|B\mathbf{u} - c\|_{L^2} = O(\epsilon). \quad (22)$$

Heuristically arguing, we can take motivation from (22) and adopt a two-phase strategy where we first determine the minimizer of J_{HS} and use this minimizer as an initial condition in the evolutionary PDE associated with (3). Although our model is not derived by rigorous fluid mechanics, we still demonstrate that our results closely approximate the physics-based models for this particular choice of the functions ϕ and ψ . The first step is to show the equivalence of the variational problem with the associated saddle point problem, see [10, Chapter 3] for further discussions.

Lemma 4.1. *(\mathbf{u}, λ) is a saddle point of (21) iff \mathbf{u} solves the variational problem (19)-(20).*

Observe that the augmented Lagrangian (21) can be reformulated as

$$\mathcal{L}_{\mu_C}(\mathbf{u}, \lambda_1) = J_R(\mathbf{u}) + \frac{\mu_C}{2} \|B\mathbf{u} - c\|^2 + \langle \lambda_1 + 2f\nabla \cdot \mathbf{u}, B\mathbf{u} - c \rangle. \quad (23)$$

The parameters μ_C, μ_R can be chosen as large as necessary. The lagrange multiplier λ_1 which acts as a dual variable is obtained by the Uzawa iteration

$$\lambda_1^{(n+1)} = \lambda_1^{(n)} + 2fd^{(n)} + \rho^{(n)}(B\mathbf{u}^{(n)} - c), \quad (24)$$

where $d^{(n)} = \nabla \cdot \mathbf{u}^{(n)}$ and $\rho^{(n)}$ is a tuning parameter. To show the equivalence of the two unconstrained optimization problems it is necessary to show that the Lagrange multipliers $\{\lambda_1^{(n)}\}$ converge. For this, we rely upon the techniques of bounded constraint algorithm, see [17, Chapter 17] for more details.

4.2. The Bounded Constraint Algorithm

The starting point is the crude pixel correspondence obtained from the Horn and Schunck optical flow $\mathbf{u}^{(0)}$ obtained within a tolerance limit δ_{HS} prescribed by Liu-Shen [15]. Observe that when the optical flow constraint is exactly satisfied then the formulations (16) and (19) coincide. However in reality due to numerical errors and approximations, the constraints are never exactly met. Thus it is natural to look at the equality constraint $B\mathbf{u} = c$ as a bounded constraint $\|B\mathbf{u} - c\|_{\mathcal{H}} \leq \epsilon_1^{(n)}$ where $\epsilon_1^{(n)}$ is a threshold parameter.

Algorithm 1 Bounded Constraint Algorithm

```

1: Set  $\lambda^{(0)}, \rho^{(0)}$ . Choose  $\epsilon_1^{(0)}, \epsilon_2^{(0)}$ .
2: Obtain initial HS optical flow  $\mathbf{u}^{(0)}$ .
3: for  $n = 1, 2, \dots$  until convergence do
4:   update  $\mathbf{u}^{(n)}, d^{(n)}$ 
5:   if  $\|B\mathbf{u}^{(n)} - c\|_{\mathcal{H}} \leq \max\{\epsilon_1^{(n)}, 2\delta_{\text{HS}}\}$  then
6:     if  $\|fd^{(n)}\|_{\mathcal{H}} \leq \epsilon_2^{(n)}$  then
7:       break;
8:     else
9:       update  $\lambda_1^{(n)}$  by (24)
10:       $\rho^{(n+1)} \leftarrow \rho^{(n)}$ 
11:      tighten tolerances  $\epsilon_1^{(n+1)}, \epsilon_2^{(n+1)}$ 
12:    else
13:      update  $B\mathbf{u}^{(n)} - c$ 
14:       $\lambda^{(n+1)} \leftarrow \lambda^{(n)}$ 
15:       $\rho^{(n+1)} \leftarrow 100\rho^{(n)}$ 
16:      tighten tolerances  $\epsilon_1^{(n+1)}, \epsilon_2^{(n+1)}$ 

```

The Algorithm 1 can be viewed in two phases. In the first phase, it is purely a diffusion process while in the second phase, the bounded constraint approximates the continuity equation constraint. This happens because a part of the OFC is already embedded in the CEC. The relaxation δ_{HS} is allowed so that we do not move too far away from the constraint and to ensure that the tolerance $\epsilon_1^{(n)}$ does not become too small.

4.3. Description of the Bounded Constraint Algorithm

The initial Horn and Schunck (HS) estimate $\mathbf{u}^{(0)}$ first obtained. The updates $\mathbf{u}^{(n)}$ in step 4 are obtained by discretizing the Euler-Lagrange equations:

$$\mathbf{P}\mathbf{u}^{(n+1)} = \mathbf{b}\mathbf{u}^{(n)}, \quad (25)$$

where

$$\mathbf{P} := \begin{bmatrix} \alpha + \frac{2\beta f^2}{\Delta x^2} & 0 \\ 0 & \alpha + \frac{2\beta f^2}{\Delta y^2} \end{bmatrix}, \quad \mathbf{b}\mathbf{u}^{(n)} = \begin{bmatrix} \alpha(M * u^{(n)}) + \beta \frac{\partial}{\partial x} [f^2 d^{(n)}] \\ \alpha(M * v^{(n)}) + \beta \frac{\partial}{\partial y} [f^2 d^{(n)}] \end{bmatrix}, \quad (26)$$

$\Delta x, \Delta y$ are grid step sizes and M is the nine-point approximation stencil of the Laplacian. $d^{(n)}$ is updated by the relation

$$d^{(n)} = S(t)d^{(n-1)},$$

where the map $S(t) : u_0 \mapsto u(t) := G_k(\cdot, t) * u_0$ form a linear, continuous semigroup of contractions in \mathcal{H} , G_k is the diffusion kernel associated with the operator $k\Delta$. This semigroup structure allows the process to preserve the spatial characteristics of the divergence and the vorticities.

The convergence criteria stated in lines 5 and 6 are checked subsequently. If both criteria are satisfied then the required approximation is met and the algorithm terminates. If condition in line 6 is not satisfied, then we enter the inner iterations and run through steps 9 to 11, where the Lagrange multiplier λ_1 is updated while the penalty parameter μ_C is not modified.

The outer iteration steps 13 to 16 are executed when the convergence criteria formulated in line 5 fail. In this case, the optical flow constraint is first updated. Then, to preserve the convergence and to ensure no occurrence of spurious updates happens, the Lagrange multiplier is assigned the value from the previous iteration. The tuning parameter is updated with a higher penalty to ensure that the iterates remain within bounds.

4.4. Convergence of the Uzawa Iterates

So far we have discussed how the modified augmented Lagrangian formulation is employed to show the equivalence of the two saddle point problems using the techniques of the Bounded Constraint Algorithm. The discussion is complete only when we prove the convergence of the Uzawa iterates (24). Using the Bounded Constraint Algorithm and the decoupling principle we show the following.

Lemma 4.2. *The Uzawa iterates can be shown to satisfy the bounds*

$$\|\lambda_1^{(n+1)} - \lambda_1^{(0)}\|_{\mathcal{H}} \leq 2M_1 \|f\|_{L^\infty} \|d^{(0)}\|_{\mathcal{H}} + r$$

where

$$r = \max \left\{ \frac{\pi^2}{6} mCM, 2\delta_{HS} \right\}.$$

Proof. Set $n = i$ in Equation (24)

$$\lambda_1^{(i+1)} - \lambda_1^{(i)} = 2f d^{(i)} + \rho^{(i)}(B\mathbf{u}^{(i)} - c), \quad 1 \leq i \leq n.$$

Adding the n equations we obtain

$$\begin{aligned} \lambda_1^{(n+1)} - \lambda_1^{(0)} &= 2f[d^{(n)} + \dots + d^{(0)}] + \rho^{(n)}(B\mathbf{u}^{(n)} - c) + \dots + \rho^{(0)}(B\mathbf{u}^{(0)} - c) \\ &= 2f[S(t)^n d^{(0)} + \dots + d^{(0)}] + \rho^{(n)}(B\mathbf{u}^{(n)} - c) + \dots + \rho^{(0)}(B\mathbf{u}^{(0)} - c). \end{aligned}$$

Therefore,

$$\lambda_1^{(n+1)} - \lambda_1^{(0)} = 2f \left[\sum_{i=0}^n S(t)^i \right] d^{(0)} + \sum_{i=0}^n \rho^{(i)}(B\mathbf{u}^{(i)} - c).$$

Hence,

$$\|\lambda_1^{(n+1)} - \lambda_1^{(0)}\|_{\mathcal{H}} \leq 2\|f\|_{L^\infty} \left| \left[\sum_{i=0}^n S(t)^i \right] d^{(0)} \right| + \sum_{i=0}^n |\rho^{(i)}| \|B\mathbf{u}^{(i)} - c\|_{\mathcal{H}}. \quad (27)$$

Let us first consider the second sum in (27). Following the algorithm we note that $\|B\mathbf{u}^{(n)} - c\|_{\mathcal{H}} \leq C\epsilon_1^{(n)}$. Since the tolerance limit $\epsilon_1^{(n)}$ is tightened after every update, there exists a N such that for $n > N$ we can choose a $M > 0$ such that $\epsilon_1^{(n)} \leq M/(n+1)^2$ as long as $\epsilon_1^{(n)} > 2\delta_{\text{HS}}$. Thus

$$\|B\mathbf{u}^{(n)} - c\|_{\mathcal{H}} \leq C \frac{M}{(n+1)^2}.$$

Also as step 10 of the algorithm suggests the tuning parameter $\rho^{(n+1)}$ is assigned the value of the previous iteration $\rho^{(n)}$. This value is fixed as long as the steps 9-11 run. Let us denote this fixed value by m . Combining these discussions we obtain

$$\sum_{i=0}^n |\rho^{(i)}| \|B\mathbf{u}^{(i)} - c\|_{\mathcal{H}} \leq mCM \sum_{i=0}^n \frac{1}{(i+1)^2},$$

which remains finite as n becomes large. Now suppose it takes n iterations where $\epsilon_1^{(n)} > 2\delta_{\text{HS}}$. From the $(n+1)^{\text{th}}$ iteration when $\epsilon_1^{(n+j)} < 2\delta_{\text{HS}}, j = 1, 2, \dots$ the upper bound becomes $2\delta_{\text{HS}}$. Combining we have

$$\sum_{i=0}^n |\rho^{(i)}| \|B\mathbf{u}^{(i)} - c\|_{\mathcal{H}} \leq r,$$

where

$$r = \max \left\{ \frac{\pi^2}{6} mCM, 2\delta_{\text{HS}} \right\}.$$

which remains finite as $n \rightarrow \infty$. Since $S(t)$ is a contraction we observe that as $n \rightarrow \infty$ the series of operators in the first term of (27) converges to $(I - S(t))^{-1}$ which is a bounded operator. Thus we have $\|(I - S(t))^{-1}d^{(0)}\| \leq M_1\|d^{(0)}\|$. Hence, in the limiting case, we have

$$\|\lambda_1^{(n+1)} - \lambda_1^{(0)}\|_{\mathcal{H}} \leq 2M_1\|f\|_{L^\infty}\|d^{(0)}\|_{\mathcal{H}} + r$$

which is finite. This shows the convergence of the multipliers $\lambda_1^{(n)}$. □

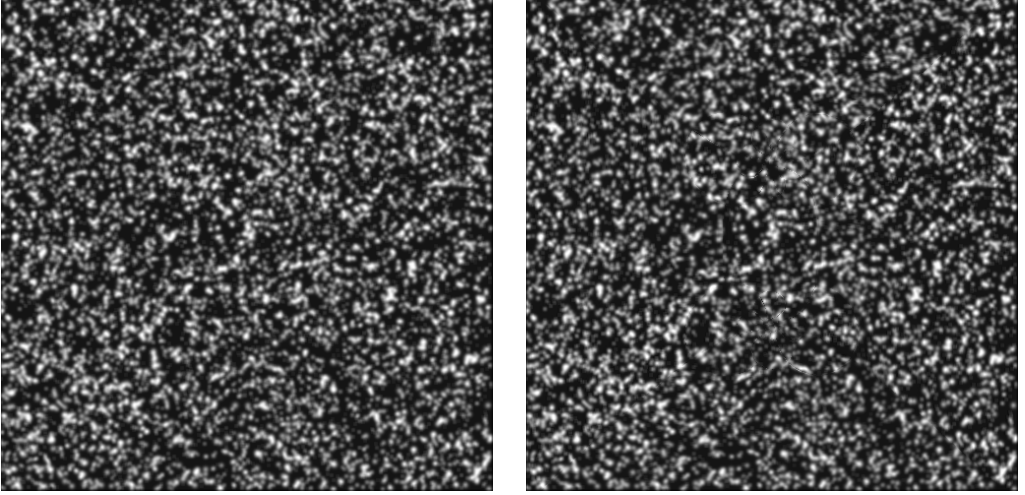


Fig. 1. Oseen vortex pair.

5. Experiments and Results

A direct implementation of the single-phase continuity equation-based refinement requires a very large value of the regularization parameter along with a HS initialization for accelerating convergence and a pyramidal grid for a stable scheme. In this section, we show the results of our two-phase method compared to the continuity equation based method on different datasets.

5.1. Experiments on PIV Dataset

We first tested our algorithm on the oseen vortex pair (see Figure (1)) and compared our results with the continuity equation model. The Oseen vortex pair is a synthetic PIV sequence of dimension 500×500 . The vorticies are placed centered at the positions $(166.7, 250)$ and $(333.3, 250)$. The circumferential velocity is given by $v_\theta = (\Gamma/2\pi r)[1 - \exp(-r^2/r_0^2)]$ with the vortex strength $\Gamma = \pm 7000$ pixels²/s and vortex core radius $r_0 = 15$ pixels. For more details see [14].

Figure (2) indicates that the vorticity plot obtained through our constraint-based refinement algorithm is very close to the continuity equation based model (CEC).

Performing a similar analysis as in [19] we plot the distribution of the x -component of the velocity to obtain Figure (3). This plot compares the distribution of the x -component of the velocity extracted from the grid images for the HS model, CEC model and our constraint-based refinement model. The profiling clearly shows the closeness of

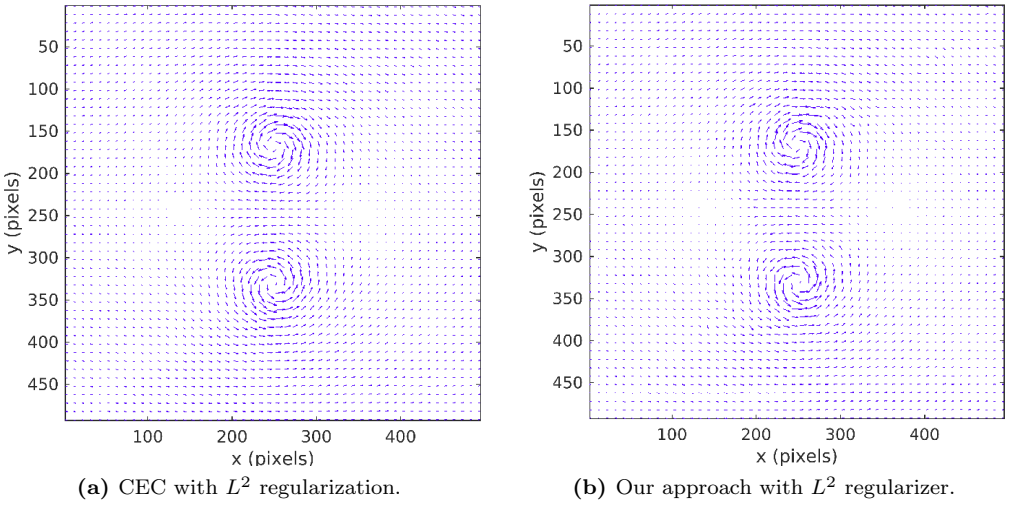


Fig. 2. Vorticity plot for the Oseen vortex pair.

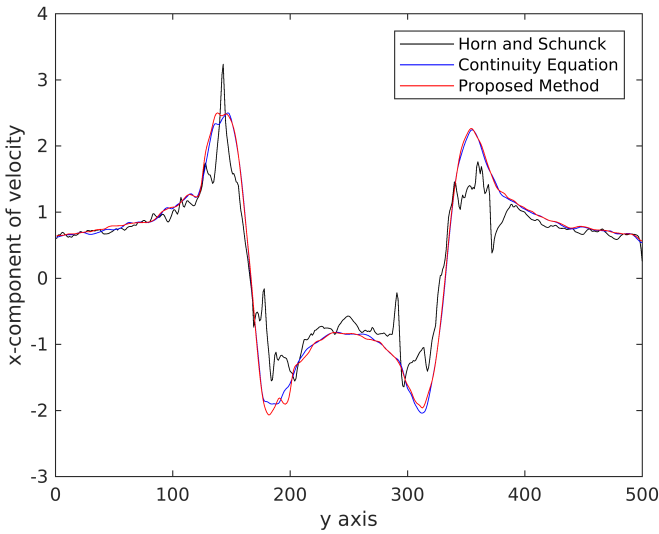


Fig. 3. Distribution of the x -component of the velocity extracted from the grid images for the Oseen vortex pair.

our algorithm to the continuity equation based model. From the figure, it is also seen how the Horn and Schunck model underestimates the flow components, especially near the vortex cores.

For a better quantitative evaluation of our flow two-phase method, we consider additional PIV analytic flows [8]. We show the results for the Poiseuille and Lamb-Oseen sequences, see [9] for the description of these datasets.

Figures 4 and 5 show the vorticity plot for the Poiseuille and the Lamb-Oseen sequences comparing both the methods with the ground-truth flow. Further, we report the End Point Error (EPE) for some of the PIV analytic flows in Table 2. The EPE is computed as

$$EPE = |\mathbf{u}_e - \mathbf{u}_c| = \sqrt{(u_1^e - u_1^c)^2 + (u_2^e - u_2^c)^2},$$

where $\mathbf{u}_e = (u_1^e, u_2^e)$ is the exact optical flow, $\mathbf{u}_c = (u_1^c, u_2^c)$ is the computed optical flow. From the table, it is conclusive that our two-phase method outperforms the CEC method.

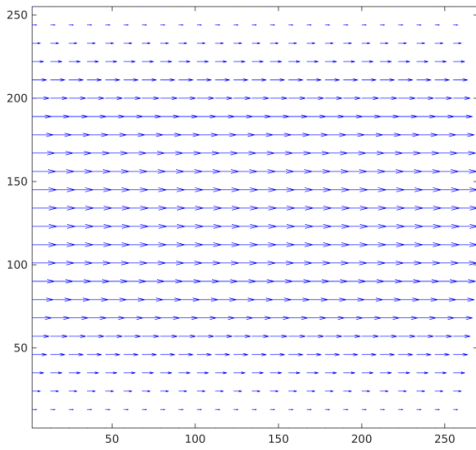
5.2. Experiments on Cloud Sequence

In this sequence, the movement of the fluid exhibits both formation of a vortex as well as a movement of fluid parcels. The distribution of the strength of the vortices in the cloud sequence obeys a Gaussian distribution of mean 0 and standard deviation of $3000 \text{ (pixels)}^2/\text{s}$.

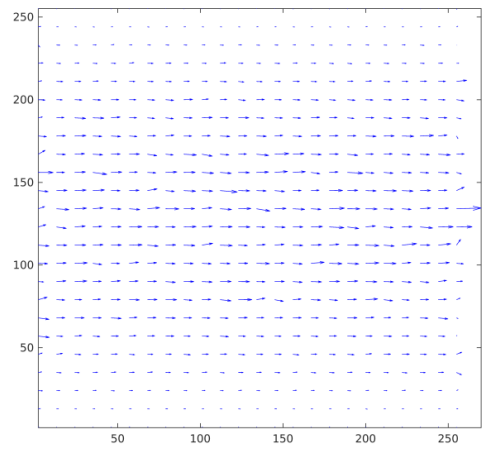
Figure 6 shows the cloud sequence. The comparison of the velocity magnitude plots are shown below: As seen from Figure 7 the isotropic behaviour of the regularization is seen more on the continuity equation based implementation because of the denseness of the flow. Also by increasing the number of iterations we have observed that the effect of diffusion makes the vortices completely circular. The distribution of the x -component of the velocity for the cloud sequence is shown in Figure 8. The Horn and Schunck estimator tends to over estimate at the peaks.

Tab. 2. Comparison of the Average Angular Error (AAE) and End Point Error (EPE) for some PIV analytic flows.

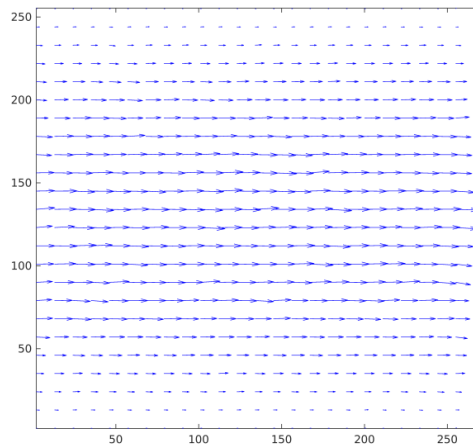
| Algorithm | Poiseuille | Lamb-Oseen | Sink | Vortex | Potential flow |
|-------------------------------|--------------|--------------|--------------|--------------|----------------|
| | EPE | EPE | EPE | EPE | EPE |
| CEC with L^2 regularization | 0.181 | 0.847 | 0.031 | 0.265 | 1.260 |
| Our Two-phase Refinement | 0.106 | 0.841 | 0.029 | 0.261 | 1.226 |



(a) Ground-truth flow.

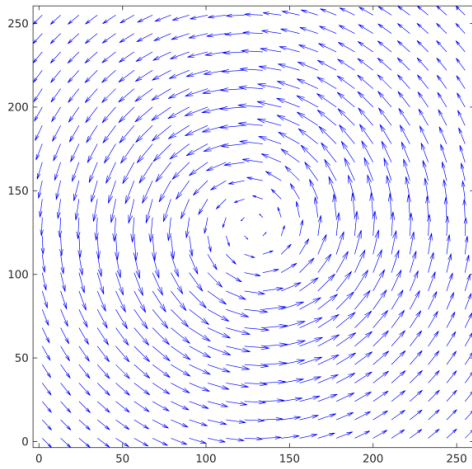


(b) CEC with L^2 .

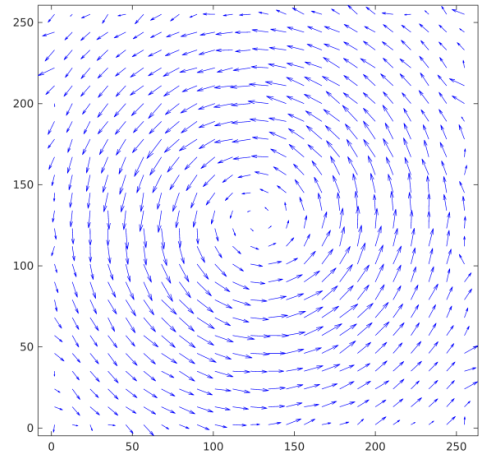


(c) Our two-phase refinement.

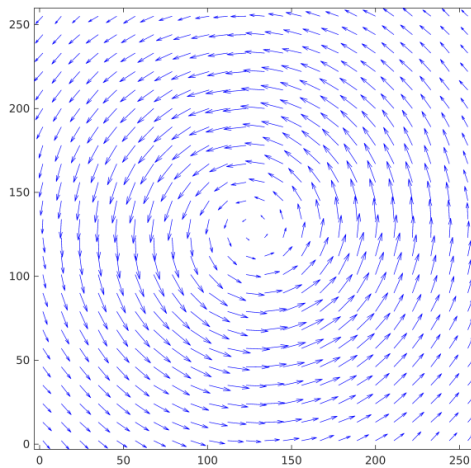
Fig. 4. Vorticity plot for the Poiseuille sequence.



(a) Ground-truth flow.



(b) CEC with L^2 .



(c) Our two-phase refinement.

Fig. 5. Vorticity plot for the Lamb-Oseen sequence.

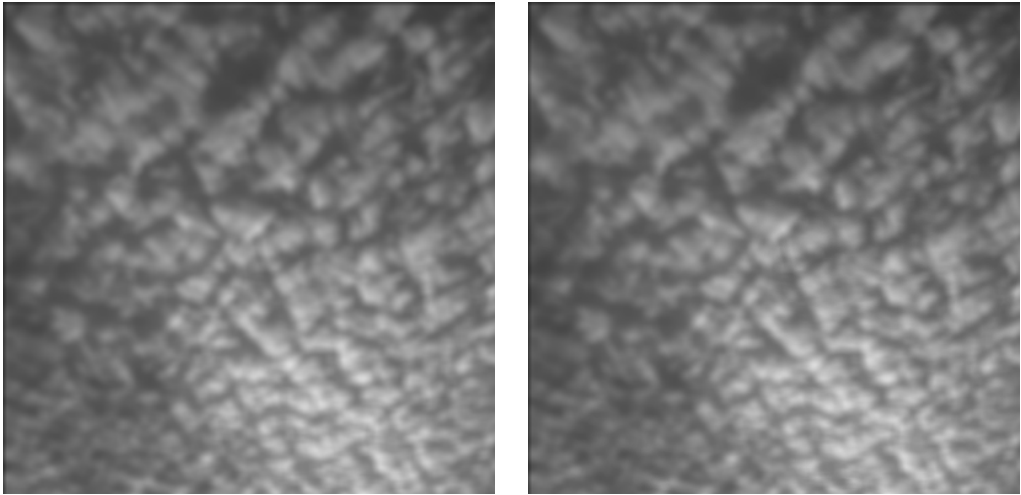


Fig. 6. Cloud sequence

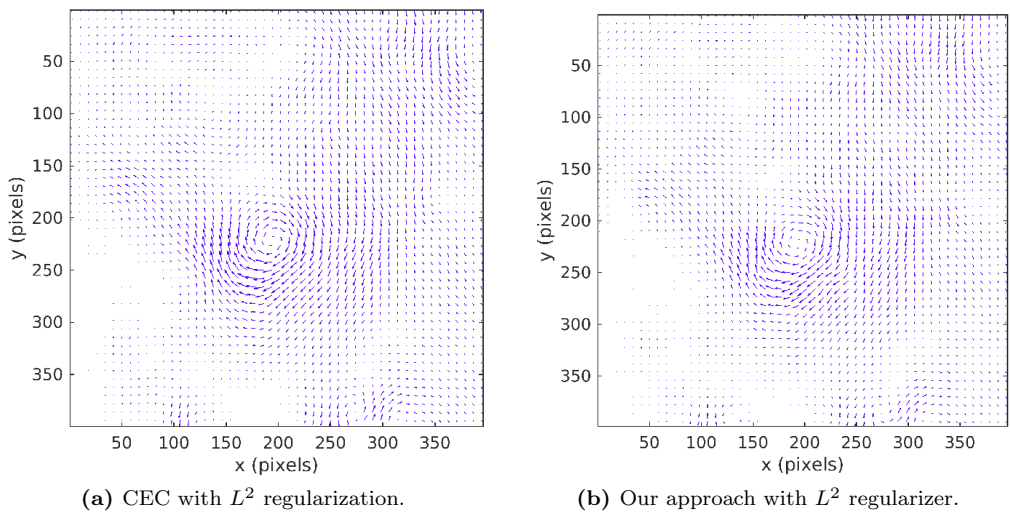


Fig. 7. Vorticity plot for the cloud sequence.

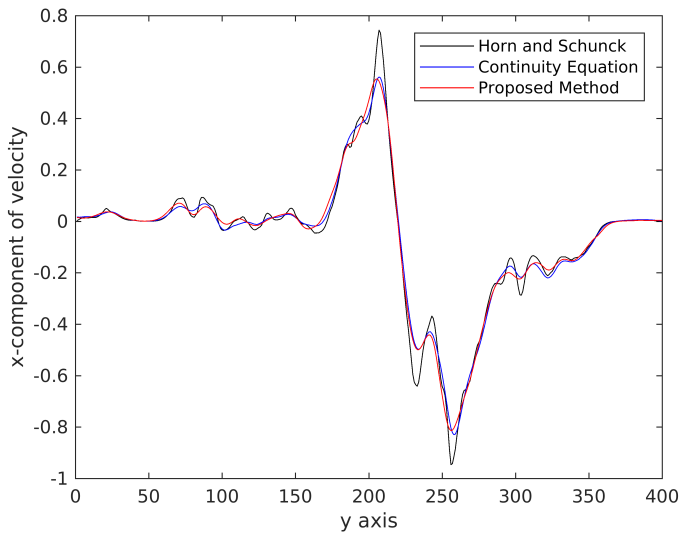


Fig. 8. Comparison between the distributions of the x -component of the velocity extracted from the grid images for the cloud sequence.

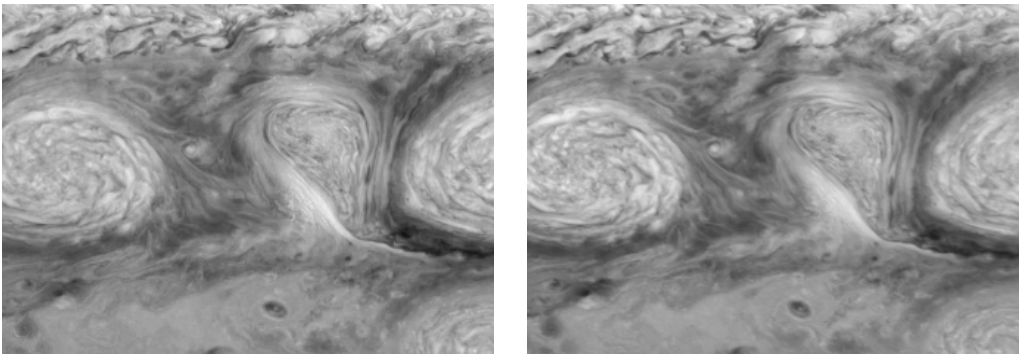
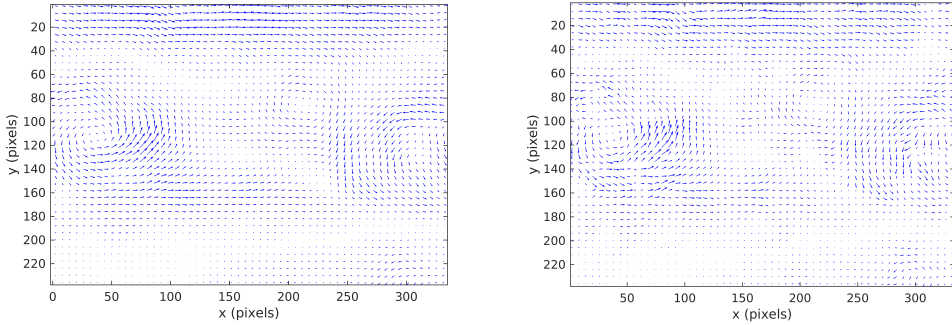


Fig. 9. Jupiter's white oval sequence.

5.3. Experiments on Jupiter's White Oval Sequence

Figure 9 shows Jupiter's white oval sequence. The white ovals seen in the images are distinct storms on Jupiter's atmosphere captured by NASA's Galileo spacecraft at a time-lapse of one hour, see [14].



(a) CEC + L^2 with illumination correction. (b) Our approach with illumination correction.

Fig. 10. Vorticity plot of the Jupiter's white oval sequence.

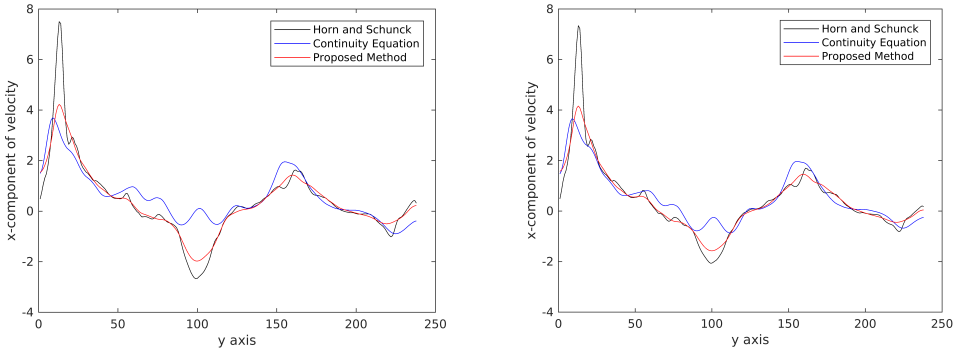
Effect of Illumination Changes on Optical Flow

Due to the time difference between adjacent frames, it was observed that the sun's illumination influenced the subsequent frame considerably in a non-uniform way. To compensate for the illumination effects, it is necessary to account for the illumination variation before applying the optical flow method. In Liu's implementation, an illumination correction is employed by normalizing the intensities and performing a local intensity correction using Gaussian filters. The first plot in Figure 10 shows the results of their implementation of the CEC model.

The following comparison demonstrates the effect of illumination changes on the optical flow computation. As seen from Figure 11 there is a large deviation near the vortex region when illumination correction is not taken into account. The deviation is minimized to a great extent as can be seen from the second image. The reason for our results (even with illumination correction) not being very close to the illumination-corrected CEC-based flow is because of the direct dependency of the process on the image data.

5.4. Demonstration of the Flow-driven Refinement Process

Rather than correcting the illumination changes by modifying the scheme we choose a flow-driven refinement process ($\phi(f) = 1$) and perform a diffusion on the curl component. In order to achieve this, we consider the fourth case from Table (1), $\phi(f) = 1$ and $\psi = (\nabla_H \cdot \mathbf{u})^2$ where $\nabla_H = (-\partial_y, \partial_x)$ is the Hamiltonian gradient. Introducing the symplectic gradient switches the roles of divergence and curl in the Equation (8) and the



(a) Our approach without illumination correction. (b) Our approach with illumination correction.

Fig. 11. Effect of the illumination correction on the distribution of the x -component of the velocity for Jupiter's white oval sequence.

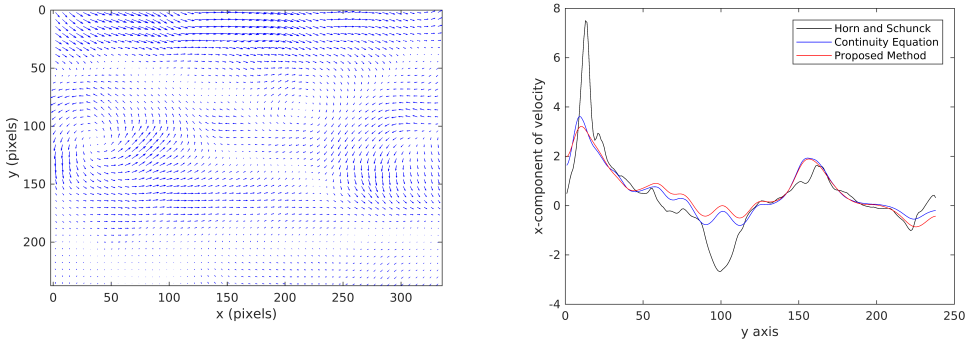


Fig. 12. Plots of the Jupiter's white oval sequence with $\phi(f) = 1, \psi(\mathbf{u}) = (\nabla_H \cdot \mathbf{u})^2$ and without illumination correction.

analysis follows in the same lines. As mentioned earlier this particular choice captures the rotational aspects of the flow much better.

Figure 12 gives the velocity magnitude plot obtained by our constraint-based refinement process $\phi(f) = 1$ and $\psi(\mathbf{u}) = (\nabla_H \cdot \mathbf{u})^2$ of Jupiter's white oval sequence along with the distribution of the x -component of the velocity. The ovals are clearly captured by

our algorithm. From the distribution of the velocity plot, it is also clear that the flow-driven refinement process involving the curl outperforms the CEC-based flows without the additional assumption of illumination correction on the image data.

5.5. Choice of Parameters

In the Liu-Shen implementation of CEC based model, the Lagrange multiplier in the HS-estimator is chosen to be 20 and in the Liu-Shen estimator, it is fixed at 2000. They observed that for a refined velocity field it does not significantly affect the velocity profile in a range of 1000-20,000 except the peak velocity near the vortex cores in this flow. For the image sequences, the best result was obtained for the values $\alpha = 100$ and $\beta = 0.01$. It was also observed experimentally that the numerical scheme converges when the ratio β/α is less than or equal to 10^{-4} .

6. Conclusion

We have proposed a general framework for fluid motion estimation using a constraint-based refinement approach. We observed a surprising connection to the Cauchy-Riemann operator that diagonalizes the system leading to a diffusive phenomenon involving the divergence and the curl of the flow. For a particular choice of the additional constraint, we showed that our model closely approximates the continuity equation based model by a modified augmented Lagrangian approach. Additionally, we demonstrated that a flow-driven refinement process involving the curl of the flow outperforms the classical physics-based optical flow method without any additional assumptions on the image data.

Acknowledgements

The authors dedicate this paper to Bhagawan Sri Sathya Sai Baba, Revered Founder Chancellor, SSSIHL. We would like to thank Dr. Shailesh Srivastava for his insights into obtaining the velocity plots.

Data Availability Statement

The image sequences (data) used for this study were accessed from the repository [22] available publicly as a supplementary material to [14] and also from the web site [8].

Acknowledgements

The authors dedicate this paper to Bhagawan Sri Sathya Sai Baba, Revered Founder Chancellor, SSSIHL. We would like to thank Dr. Shailesh Srivastava for his insights into obtaining the velocity plots.

Data Availability Statement

The image sequences (data) used for this study were accessed from the repository [22] available publicly as a supplementary material to [14] and also from the web site [8].

Statements and Declarations

Competing Interests

The authors declare that they have no competing interests.

Funding



The authors did not receive any financial grant during the preparation of this manuscript.

References

- [1] P. Altomare, S. Milella, G. Musco. Multiplicative Perturbation of the Laplacian and Related Approximation Problems, *Journal of Evolution Equations*, 771–792, 2011. doi:10.1007/s00028-011-0110-6.
- [2] G. Aubert, R. Deriche, P. Kornprobst. Computing Optical Flow via Variational Techniques, *SIAM Journal of Applied Mathematics*, 60:156–182, 1999. doi:10.1137/S0036139998340170.
- [3] H. Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*, Springer, 2011. doi:10.1007/978-0-387-70914-7.
- [4] T. Corpetti, E. Mémin, P. Pérez. Estimating Fluid Optical Flow, *Proceedings of the 15th International Conference on Pattern Recognition (ICPR2000)*, 3:1033–1036, 2000. doi:10.1109/ICPR.2000.903722.
- [5] T. Corpetti, D. Heitz, G. Arroyo, E. Mémin, A. Santa-Cruz. Fluid Experimental Flow Estimation based on an Optical Flow Scheme, *Experiments in Fluids*, 40:80–97, 2006. doi:10.1007/s00348-005-0048-y.
- [6] X. Chen, P. Zillé, L. Shao, T. Corpetti. Optical Flow for Incompressible Turbulence Motion Estimation, *Experiments in Fluids*, 56:8, 2015. doi:10.1007/s00348-014-1874-6.
- [7] P. Eidus, The Perturbed Laplace Operator in a weighted L^2 space, *Journal of Functional Analysis*, 100:400–410, 1991. doi:10.1016/0022-1236(91)90117-N.
- [8] J. Carlier, R. Cemagref. FLUID. Image sequence database. <http://fluid.irisa.fr/data-eng.htm> (Accessed: December 2023).
- [9] J. Carlier. Second set of fluid mechanics image sequences. European project *FLUId Image analysis and Description*, 2005. <https://cordis.europa.eu/project/id/513663>.
- [10] R. Glowinski, P. Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, 1989. doi:10.1137/1.9781611970838.
- [11] D. Heitz, E. Mémin, C. Schnörr. Variational Fluid Flow measurements from Image Sequences: Synopsis and Perspectives, *Experiments in Fluids*, 48:369–393, 2010. doi:10.1007/s00348-009-0778-3.

- [12] W. Hinterberger, O. Scherzer, C. Schnörr, J. Weickert. Analysis of Optical Flow Models in the Framework of Calculus of Variations, *Numerical Functional Analysis and Optimization*, 23(1-2):69–89, 2002. doi:10.1081/NFA-120004011.
- [13] B. K. P Horn, B. G. Schunck. Determining Optical Flow, *Artificial Intelligence*, 17:185–203, 1981. doi:10.1016/0004-3702(81)90024-2.
- [14] T. Liu. OpenOpticalFlow: An Open Source Program for Extraction of Velocity Fields from Flow Visualization Images, *Journal of Open Research Software*, 5:29, 2017. doi:10.5334/jors.168.
- [15] T. Liu, L. Shen. Fluid Flow and Optical Flow, *Journal of Fluid Mechanics*, 614:253–291, 2008. doi:10.1017/S0022112008003273.
- [16] A. Luttmann, E. M. Bollt, R. Basnayake, S. Kramer, N.B. Tuffiaro. A Framework for Estimating Potential Fluid Flow from Digital Imagery, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23:3, 2013. doi:10.1063/1.4821188.
- [17] J. Nocedal, S. J. Wright. *Numerical Optimization*, 2nd Edition, Springer, 2006. doi:10.1007/b98874.
- [18] C. Schnörr. Determining Optical Flow for Irregular Domains by Minimizing Quadratic Functionals of a Certain Class, *International Journal of Computer Vision*, 6:25–38, (1991). doi:10.1007/BF00127124.
- [19] B. Wang, Z. Cai, L. Shen, T. Liu. An Analysis of Physics-based Optical Flow, *Journal of Computational and Applied Mathematics*, 276:62–80, 2015. doi:10.1016/j.cam.2014.08.020.
- [20] J. Weickert, C. Schnörr. A Theoretical Framework for Convex Regularizers in PDE-based Computation of Image Motion, *International Journal of Computer Vision*, 45:245–264, 2001. doi:10.1023/A:1013614317973.
- [21] R. P. Wildes, M. J. Amabile, A. Lanzillotto, T. Leu. Recovering Estimates of Fluid Flow from Image Sequence Data, *Computer Vision and Image Understanding*, 80:246–266, 2000. doi:10.1006/cviu.2000.0874.
- [22] T. Liu. OpenOpticalFlow. GitHub repository, 2021. <https://github.com/Tianshu-Liu/OpenOpticalFlow> (Accessed: December 2023).

RESIDUAL NEURAL NETWORKS IN SINGLE INSTANCE-DRIVEN IDENTIFICATION OF FUNGAL PATHOGENS

Rafał Wyszyński , Karol Struniawski *

Institute of Information Technology

Warsaw University of Life Sciences – SGGW

Warsaw, Poland

*Corresponding author: Karol Struniawski (karol.struniawski@sggw.edu.pl)

Abstract The rise in fungal infections, attributed to various factors including medical interventions and compromised immune systems, necessitates rapid and accurate identification methods. While traditional mycological diagnostics are time-consuming, machine learning offers a promising alternative. Nevertheless, the scarcity of well-curated datasets is a significant obstacle. To address this, a novel approach for identifying fungi in microscopic images using Residual Neural Networks and a subimage retrieval mechanism is proposed, with the final step involving the implementation of majority voting. The new method, applied to the Digital Images of Fungus Species database, surpassed the original patch-based classification using Convolutional Neural Networks, obtaining an overall classification accuracy of 94.7% compared to 82.4% with AlexNet FV SVM. The observed MCC metric exceeds 0.9, while AUC is near to one. This improvement is attributed to the optimization of hyperparameters and top layer architecture, as well as the effectiveness of the Mish activation function in ResNet-based architectures. Noteworthy, the proposed method achieved 100% accurate classification for images from 8 out of 9 classes after majority voting and is high resistant to overfitting, highlighting its potential for rapid and accurate fungal species identification in medical diagnostics and research.

Keywords: Residual Neural Networks, fungal image classification, deep learning, microscopic images, majority voting, machine learning, image processing.

1. Introduction

Fungi kingdom is characterized by species diversity and various life forms. Its organisms can be microscopic to macroscopic [7]. Fungi are eukaryotic organisms, which means that their cells consist of nucleus and an organized inner structure. Unlike plants, fungi do not perform photosynthesis, meaning they do not produce their own food from sunlight. Instead, most of them feed on dead or decaying organic matters, which plays a crucial role in the decomposition of organic material and in whole ecosystems [34]. They have great impact on humans life. Both positive and negative.

There are many different domains where fungi found their utility helping humans. Some of them, such as yeasts, take important part in food production. They are essential to produce bread [2], beer [17], etc. Furthermore, certain fungi possess medicinal properties and are successfully used in the production of pharmaceuticals [11]. On the other hand, over the past several decades, there has been a notable rise in the occurrence

of fungal infections, which has resulted in elevated rates of morbidity and death [23]. The key factors of observed rise are recognized as the use of catheters, wide-spectrum antibiotics, immunosuppression, chemotherapy and radiation [28].

One of the fungi species covered by this research are *Candida*. Although they are part of the normal microbiota of the mouth cavity, digestive system and vaginal canal [32], they can cause infections if they grow out of control and enter deep into the body. *Candida* are responsible for a variety of clinical symptoms, including mucocutaneous overgrowth and bloodstream infections. More than 90% of all invasive infections are due to this species, thanks to their ability to overcome host defense capacity, adhere and create biofilms [30]. *Candida* infection is the most prevalent causative agent of fungal-related biofilm infections and the third most common cause of nosocomial infections in patients seeking emergency medical attention [13]. That ultimately represents the need of accurate and vast identification of species through the diagnostic tests.

The identification of fungal species by mycological diagnostics is a laborious procedure that can take four to ten days. The goal is to replace biochemical tests with machine learning methods, shortening whole diagnosis process by 2-3 days.

Due to a dearth of well-prepared datasets, fungal microscopic pictures are scarce in machine vision and learning applications. Based on the frequency of each fungal infection, the paper by Zieliński et al. [46] introduced the database called Digital Images of Fungus Species database (DIFaS) consisting of nine strains of fungi, responsible for most of the infections. It contains, in total, 176 images of resolution 3600×5760 taken with an Olympus BP74 camera. The strains were cultivated and then stained with Gram method. The original manuscript presented the experimental application of patch-based classification using Convolutional Neural Networks (CNNs), e.g. AlexNet, InceptionV3, DenseNet169, rendering the best overall accuracy of 82.4% for AlexNet FV SVM. The classifier struggled to correctly identify two of the species – *Candida glabrata* (CG) and *Candida neoformans* (CN) resulting in class accuracy of 50%.

Rawat et al. proposed a methodology named MeFunX that leverages a meta-learning-based deep learning architecture, comprising two base learners implemented as CNNs and XGBoost as the meta-learner [31]. Rigorous experimentation demonstrates the outstanding performance of MeFunX, achieving an overall accuracy of 92.49% for the early diagnosis of fungal infections in microscopic images.

Struniawski et al. devised a novel pipeline for the automated identification of soil fungi based on single-instance extraction and deep learning techniques [39]. The approach employs a series of machine vision methods, including thresholding, morphological operations and flood fill algorithms, to isolate individual fungi elements from raw microscopic images. These subimages are subsequently fed into a ResNet50 CNN, achieving an accuracy of 82%. To further enhance the performance, a majority voting scheme is incorporated, resulting in an overall accuracy and F1 score of a remarkable 97%. This pipeline underscores the extraordinary potential of single-instance retrieval, deep

learning and voting mechanisms for accurate and efficient fungi identification. Similarly, the effectiveness of the proposed method of segmentation, majority voting and machine learning methods was demonstrated for identifying mycorrhizal bacteria from raw microscopic images [19, 38]. These findings reinforce the versatility of these techniques for automated microorganism identification, extending their applicability beyond fungi.

The core principle of this research is to testify if the procedure of single object retrieval creating subimaged dataset for training CNNs and then applying majority voting rule for results concatenation can be also applied directly for fungi that are harmful for humans.

2. Methods

A comprehensive analysis is presented of the subimage retrieval procedure and the integration of CNNs and majority voting for image classification. The intricate details of the image preprocessing pipeline are delved into, highlighting the meticulous methods employed to extract the most representative fungal samples from raw microscopic images. Additionally, valuable insights into the training process and optimization strategies are provided, revealing the actions taken to achieve presented performance.

2.1. Dataset information

The DIFaS dataset [46] used in this study consists of 176 microscopic images of fungi, divided into nine distinctive classes (Fig. 1):

- Class 0: *Candida albicans* (CA) – 20 images
- Class 1: *Candida glabrata* (CG) – 20 images
- Class 2: *Candida lusitanae* (CL) – 20 images
- Class 3: *Cryptococcus neoformans* (CN) – 15 images
- Class 4: *Candida parapsilosis* (CP) – 20 images
- Class 5: *Candida tropicalis* (CT) – 20 images
- Class 6: *Maalasezia furfur* (MF) – 21 images
- Class 7: *Saccharomyces boulardii* (SB) – 20 images
- Class 8: *Saccharomyces cerevisiae* (SC) – 20 images

2.2. Subimages retrieval

To enable further image processing, at the very beginning of the algorithm, the images were converted to grayscale. To smooth out small-scale variations in the images, the Gaussian Blur technique was applied [12], improving the quality of the images, which is crucial since samples retrieved in the process are very small. Another procedural step involved the application of the thresholding technique [33]. Various approaches were tested, such as Otsu, Mean, Minimum, and local thresholding with five different block

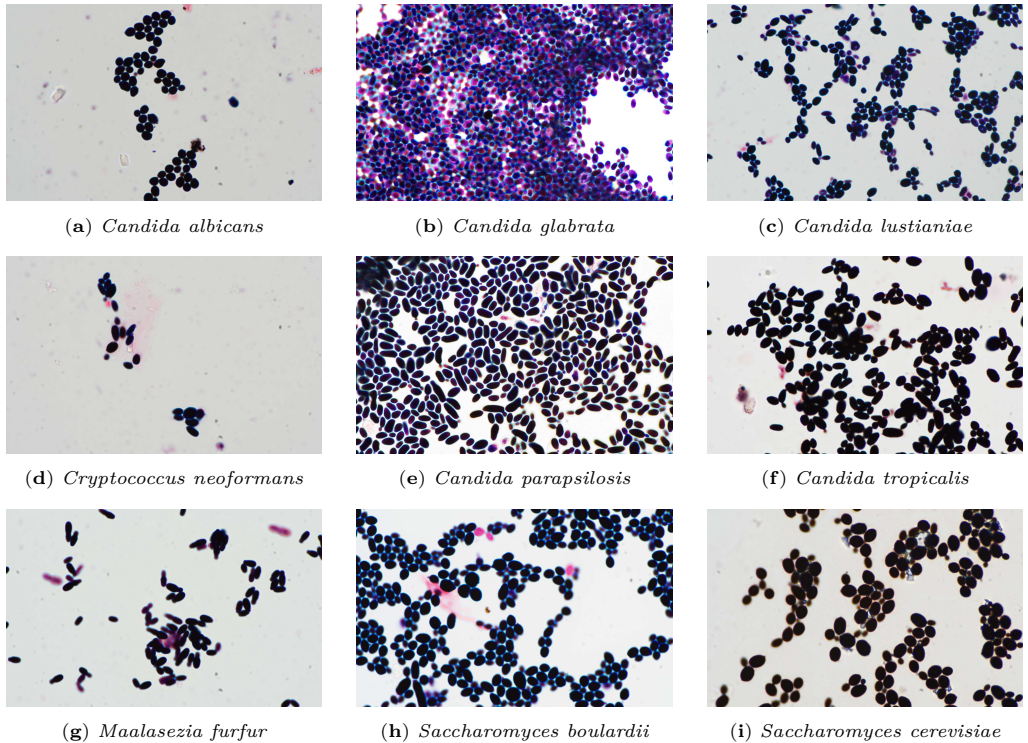


Fig. 1. Microscopic images of each fungus from the dataset.

sizes (35, 45, 55, 65, 75). The best results were obtained for local thresholding [36] with a block size of 55. However, this method introduced noise to the background, which was eliminated by creating another mask that set all pixels with intensity greater than 192 back to black color. This value was chosen experimentally after prior analysis of the background histogram for several images from each class. To separate the samples that were merging on the mask, a morphological operation [18, 37], called binary opening was employed. Thresholding and morphological operations caused gaps in the fungi, which were filled using an algorithm of OpenCV [8] that extracted contours [41] in the mask and then reconstructed them with filling on a new black background. Apart from fungi, the images also contained other objects such as image overexposures or sample contamination. To eliminate them, the image was divided into regions [40] and additional filters were applied. First, regions with solidity lower than 0.75 were removed, as after testing multiple values within the range of 0.7 to 0.9, it was found that this value performed the best in removing unwanted objects while minimizing the removal

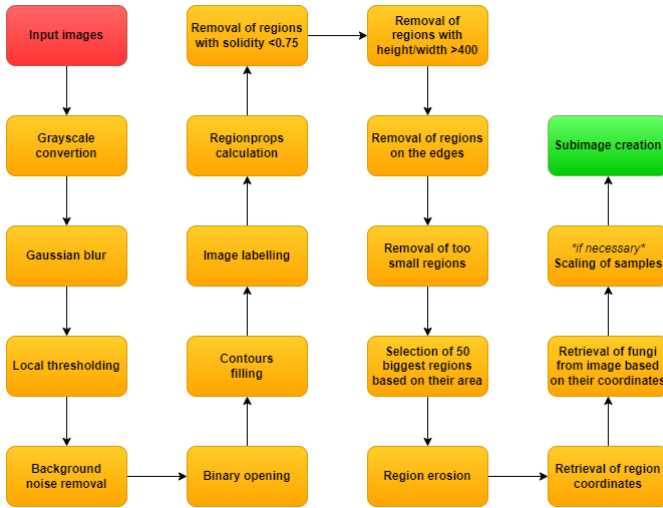


Fig. 2. The data flow diagram illustrating the proposed algorithm for subimages retrieval.

of desired ones. Solidity is the ratio of the region's area to the area of its convex hull, a shape enclosing the object's most extreme points [25]. A greater solidity score denotes a more compact or convex shape, whereas a lower value denotes an extended or uneven shape. Then, random fungi cells from each class were analyzed. It was observed, that the vast majority do not exceed 350 pixels in either height or width. Based on this, regions with a height or width exceeding 400 pixels were discarded, as it was highly possible, that those regions did not contain single cell, but rather a group of several fungi interconnected. Finally, regions with an area smaller than 20% of the difference between the largest and smallest area were removed. This formula excellently dealt with undesired small objects while ensuring universality for each class and image, as minimum threshold was not imposed. It adapted to each case individually. From the remaining ones, 50 objects with the largest surface area were selected. They underwent binary dilation [37] to ensure that the masks contained complete fungal samples with their entire contours. Finally, it was ensured that the samples fit the format of 224×224 pixels as it was required by the utilized neural network. If they were larger, they were scaled to meet the specified parameters. The entire process is presented graphically as a data flow diagram in Fig. 2 and the sample retrieved subimages are shown in Fig. 3.

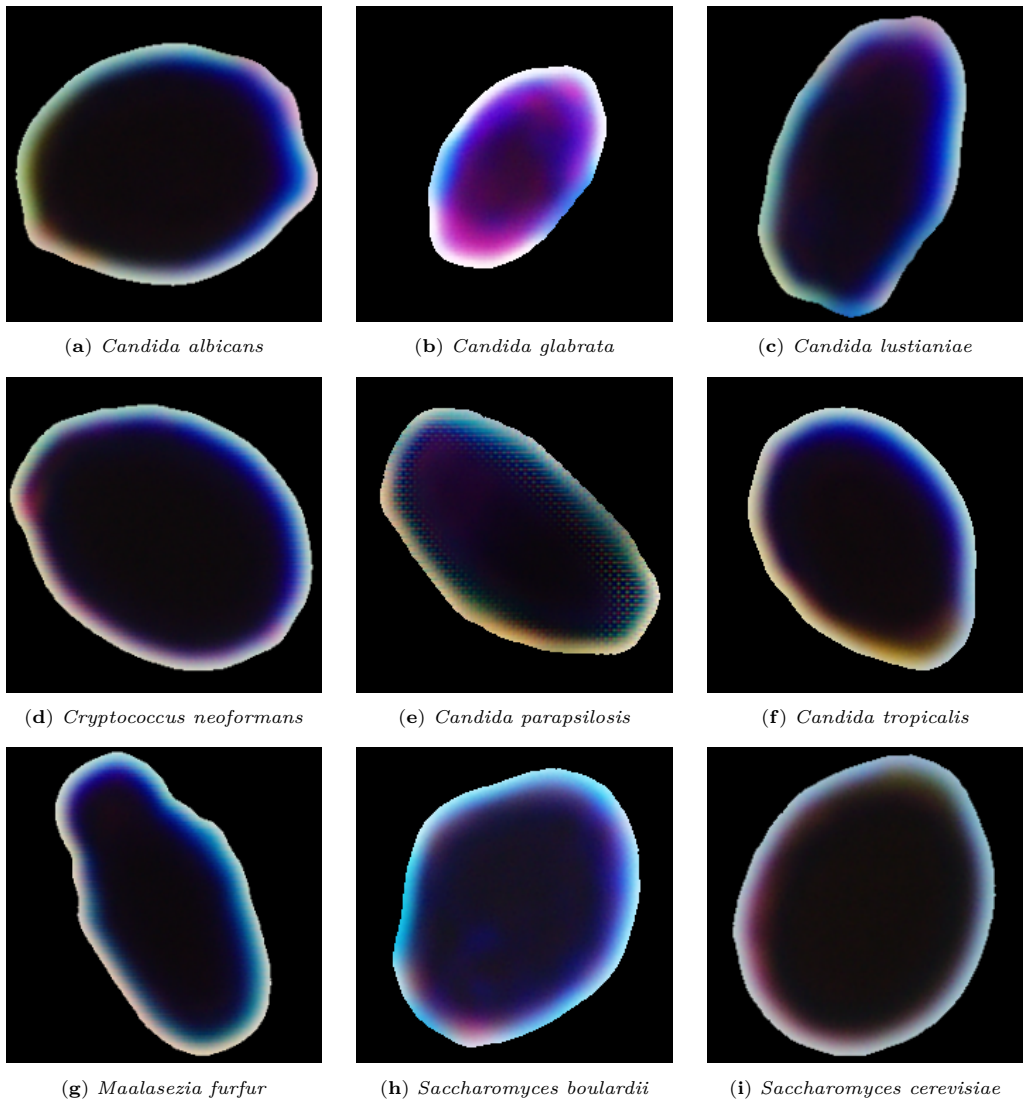


Fig. 3. Sample subimages retrieved from the original dataset.

2.3. Convolutional Neural Network

Convolutional Neural Network [20] is a deep learning architecture inspired by the processes ongoing in human's brain. The deeper the network is, the more advanced shapes it can learn to recognize, much like the brain [35]. Certainly, brain cells are much more complex structures than convolutional layers. Although CNNs are based on simple mathematic operations [21], thanks to increasing computational resources of computers, they found their application in many domains related to Computer Vision (CV), as they can be faster and more precise than humans in specific tasks. They are already successfully utilized in medical image analysis [3,24], facial recognition [42] or autonomus driving [1].

The main difference, that made CNN much more useful in CV than traditional machine learning methods is the fact, that they achieve progressively higher level of abstraction autonomously, unlike traditional methods, which heavily depend on handcrafted features [27]. They also handle better with high-dimensional nature of the image data. There are two main layers responsible for the most of CNN computations. Mentioned earlier convolutional layers [20] and pooling layers [20]. They work quite similarly, yet are responsible for the very different tasks. Convolutional layers use small, learnable fillters (kernels), that move across the image, learning its patterns and spatial hierarchies. Each layer, can contain multiple kernels learning different patterns. When the data travels to deeper layers, previous convolutions connect, recognizing more advanced structures. Pooling layers also work based on a kernels, which traverse across the image. Their purpose is to reduce number of parameters in the image that reduces complexity and improves efficiency of CNNs. They achieve it by aggregating group of pixels around the kernel and computing the value, which represents given small area the best. In the final stage, the network is connected to the fully connected dense layer, with number of neurons corresponding to the number of classes in the dataset. This layer processes outputs from the preceding pooling layer, producing images with associated probabilities. Image with the highest probability is recognized as the identified patch or pattern.

2.4. Residual Neural Network

CNNs were a milestone in the field of CV. Their success led to extremely fast development of this domain. Huge interest in the subject made many scientists and engineers explore new architecture. As understanding of CNNs started to grow, despite their great flexibility and capabilities, some of its limitations started to become transparent. Attempts to train increasingly deeper networks stopped yielding expected outcomes. It happened, because neurons adjust their weights via the backpropagation algorithm [9], which minimizes the loss function. Increase in the networks depth, caused gradient's magnitude to decrease in the deeper layers, which led to slowdown of training process. That is when two undesired issues were defined as vanishing [29] and exploding gradient [29]. The vanishing gradient occurs when the gradient is so small that changes to

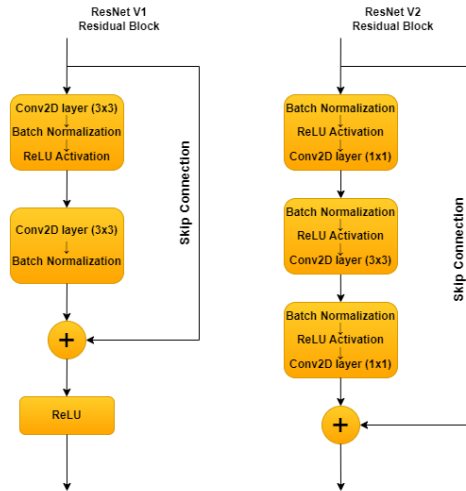


Fig. 4. Residual block architecture of ResNet V1 (left) and ResNet V2 (right).

the tuned parameters throughout the training phase are insignificant. The exploding gradient, on the other hand, arises when the gradient grows so enormous that changes to the tuned parameters throughout the training phase become excessive.

To overcome these challenges, a new type of CNN, the Residual Neural Network (ResNet) was introduced [15]. ResNet deals with the issues of vanishing and exploding gradient using skip connections [15]. These connections enable information to bypass one or more layers increasing network efficiency and its ability to learn more advanced features. They can also allow following layers to learn from information captured in initial ones. There are various versions of the ResNet architecture, such as ResNet34, ResNet101, etc. They differ mainly in terms of depth and width of the network.

This study makes use of the ResNet50v2, an upgraded version of the ResNet50 network, which is one of the most often employed in such tasks [14]. ResNet50 is excellent compromise between the depth and network's performance. The ResNetV2 version introduces changes to the architecture aimed at improving stability and overall network's efficiency. Both versions differ mainly in how the layers are organized within the residual block. In ResNet, convolution is followed by Batch Normalization and ReLU activation. ResNetV2 changed the order, applying Batch Normalization and activation function before convolution. ResNetV2 has also removed the last non-linearity, after the addition, creating identity connection between the input and output [14](see Fig. 4).

2.5. Transfer learning

Along with addressing the gradient descent problem, depth of the networks started to increase, what automatically made models much more complex. Training hundreds of thousands of parameters necessitates powerful computing resources. That is why huge CNN models are typically trained using systems optimized specifically for deep learning workloads. There are a few types of such systems: cloud platforms, clusters, supercomputers [39]. Each of them is expensive and not available for everyone. To make deep learning more affordable for private users, technique known in industry for decades, transfer learning [4], found its perfect place.

The idea of transfer learning is to utilize knowledge gained in different, but related task to solve other problems. Not only does it make computations faster, but also requires less data to achieve high scores, than the models trained from scratch [45]. When so called pre-trained models are pre-trained using sufficiently large data sets, they have basic understanding of shapes, colors etc. from the very beginning. The goal is to fine-tune that knowledge to our own classes and images, by creating a relation between previous and the target task [16]. That is why many of the most common models, such as Inception and ResNets, were pretrained on the ImageNet dataset [10]. ImageNet consists of over 14 millions of images and over 21 thousands of classes. It is a benchmark in object category classification and detection domains.

It is worth pointing out that although ImageNet consists of very wide range of images and task such as microscopic image classification is very specific, transfer learning has been successfully utilized in this area [19].

3. Experiments

Although a pre-trained model was used to classify the dataset, there are various ways to improve its performance. Hyperparameters, such as top-layer topology, batch sizes, activation functions, dropout rate, learning rate and decision whether ResNet is trainable or not on a given layer, may have a tremendous impact on the final results [44], there is no golden mean to choose it. That is why experiments play a crucial part during training. To test many different combinations with limited time and computing resources, the Early Stopping callback was implemented with the patience parameter set to 50, stopping the training process when the value of the validation loss function has not dropped for the last 50 epochs.

Experiments were split into two parts. First, 16 models with multiple configurations of topology, batch sizes and dropout values were trained. Subsequently, subjectively the best model was chosen and tested with many variations of activation functions and learning rate. The attention was directed towards the relatively new and promising Mish activation [26], as well as the implementation of cosine decay for the learning rate [43]:

Model 1: Dense(512, ReLU) \times Dropout(0.3) \times Dense(512, ReLU) \times Dropout(0.3); Learning Rate = 0.001; Batch Size = 32; Trainable = FALSE

Model 2: Dense(1024, ReLU) \times Dropout(0.2) \times Dense(512, ReLU) \times Dropout(0.2); Learning Rate = 0.0001; Batch Size = 32; Trainable = FALSE

Model 3: Dense(512, ReLU) \times Dropout(0.3) \times Dense(512, ReLU) \times Dropout(0.3) \times Dense(256, ReLU) \times Dropout(0.3); Learning Rate = 0.0001; Batch Size = 32; Trainable = FALSE

Model 4: Dense(1024, ReLU); Learning Rate = 0.001; Batch Size = 32; Trainable = FALSE

Model 5: Model 1 with Trainable = TRUE

Model 6: Model 2 with Trainable = TRUE

Model 7: Model 3 with Trainable = TRUE

Model 8: Model 4 with Trainable = TRUE

Model 9: Model 1 with Batch Size = 64

Model 10: Model 2 with Batch Size = 64

Model 11: Model 3 with Batch Size = 64

Model 12: Model 4 with Batch Size = 64

Model 13: Model 1 with Batch Size = 64 and Trainable = TRUE

Model 14: Model 2 with Batch Size = 64 and Trainable = TRUE

Model 15: Model 3 with Batch Size = 64 and Trainable = TRUE

Model 16: Model 4 with Batch Size = 64 and Trainable = TRUE

Model 17: Dense(512, ReLU) \times Dropout(0.3) \times Dense(512, ReLU) \times Dropout(0.3) \times Dense(256, ReLU) \times Dropout(0.3); Learning Rate = 0.001; Batch Size = 64; Trainable = TRUE

Model 18: Model 17 with Learning Rate = 0.0001

Model 19: Model 17 with Cosine Decay Learning Rate with warmup (Warmup Steps = 50; Decay Steps = 950; Initial Learning Rate = 0; Target Learning Rate = 0.01)

Model 20: Model 17 with Cosine Decay Learning Rate with warmup (Warmup Steps = 50; Decay Steps = 950; Initial Learning Rate = 0; Target Learning Rate = 0.1)

Model 21: Model 17 with Cosine Decay Learning Rate with warmup (Warmup Steps = 100; Decay Steps = 900; Initial Learning Rate = 0; Target Learning Rate = 0.001)

Model 22: Model 17 with Cosine Decay Learning Rate with warmup (Warmup Steps = 100; Decay Steps = 900; Initial Learning Rate = 0; Target Learning Rate = 0.01)

Model 23: Model 17 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.1)

Model 24: Model 17 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.01)

Model 25: Model 17 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.001)

Model 26: Model 17 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.0001)

Model 27: Model 17 with Mish activation function at each layer; Learning Rate = 0.01

Model 28: Model 27 with Learning Rate = 0.001

Model 29: Model 27 with Learning Rate = 0.0001

Model 30: Model 27 with Learning Rate = 0.00001

Model 31: Model 27 with Cosine Decay Learning Rate with warmup (Warmup Steps = 50; Decay Steps = 950; Initial Learning Rate = 0; Target Learning Rate = 0.01)

Model 32: Model 27 with Cosine Decay Learning Rate with warmup (Warmup Steps = 50; Decay Steps = 950; Initial Learning Rate = 0; Target Learning Rate = 0.1)

Model 33: Model 27 with Cosine Decay Learning Rate with warmup (Warmup Steps = 100; Decay Steps = 900; Initial Learning Rate = 0; Target Learning Rate = 0.001)

Model 34: Model 27 with Cosine Decay Learning Rate with warmup (Warmup Steps = 100; Decay Steps = 900; Initial Learning Rate = 0; Target Learning Rate = 0.01)

Model 35: Model 27 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.1)

Model 36: Model 27 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.01)

Model 37: Model 27 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.001)

Model 38: Model 27 with Cosine Decay Learning Rate without warmup (Decay Steps = 1000; Initial Learning Rate = 0.0001)

Model 39: Model 27 with Learning Rate = 0.000001

Model 40: Model 27 with Learning Rate = 0.0000001

Tab. 1. Performance comparison of 40 single-instance retrieval and classification models. F1-score denotes the harmonic mean of Precision and Recall, while CN F1 represents the F1-score for class CN against the rest of the classes (OvR), and MCC states as Matthews correlation coefficient.

| First Part | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Model | Precision | Recall | F1 | Accuracy | CN F1 | AUC | MCC |
| 1. | 0.76 | 0.49 | 0.43 | 0.49 | 0.00 | 0.89 | 0.44 |
| 2. | 0.74 | 0.73 | 0.73 | 0.73 | 0.05 | 0.97 | 0.69 |
| 3. | 0.76 | 0.75 | 0.74 | 0.75 | 0.00 | 0.97 | 0.71 |
| 4. | 0.75 | 0.72 | 0.72 | 0.72 | 0.06 | 0.96 | 0.68 |
| 5. | 0.90 | 0.90 | 0.90 | 0.90 | 0.34 | 0.99 | 0.89 |
| 6. | 0.92 | 0.92 | 0.92 | 0.92 | 0.17 | 0.99 | 0.91 |
| 7. | 0.93 | 0.92 | 0.91 | 0.92 | 0.00 | 0.99 | 0.91 |
| 8. | 0.89 | 0.90 | 0.89 | 0.90 | 0.06 | 0.99 | 0.88 |
| 9. | 0.72 | 0.61 | 0.57 | 0.61 | 0.00 | 0.93 | 0.55 |
| 10. | 0.74 | 0.73 | 0.72 | 0.73 | 0.06 | 0.96 | 0.69 |
| 11. | 0.74 | 0.73 | 0.72 | 0.73 | 0.07 | 0.96 | 0.69 |
| 12. | 0.78 | 0.77 | 0.77 | 0.77 | 0.15 | 0.97 | 0.74 |
| 13. | 0.87 | 0.87 | 0.86 | 0.87 | 0.12 | 0.99 | 0.85 |
| 14. | 0.91 | 0.92 | 0.91 | 0.92 | 0.12 | 0.99 | 0.91 |
| 15. | 0.92 | 0.92 | 0.92 | 0.92 | 0.06 | 0.99 | 0.91 |
| 16. | 0.90 | 0.91 | 0.90 | 0.91 | 0.11 | 0.99 | 0.89 |
| Second Part | | | | | | | |
| 17. | 0.91 | 0.91 | 0.90 | 0.91 | 0.12 | 0.99 | 0.90 |
| 18. | 0.91 | 0.93 | 0.92 | 0.93 | 0.00 | 0.99 | 0.92 |
| 19. | 0.90 | 0.91 | 0.91 | 0.91 | 0.18 | 0.99 | 0.90 |
| 20. | 0.87 | 0.15 | 0.04 | 0.15 | 0.00 | 0.63 | 0.00 |
| 21. | 0.76 | 0.75 | 0.73 | 0.75 | 0.00 | 0.97 | 0.72 |
| 22. | 0.90 | 0.89 | 0.89 | 0.89 | 0.00 | 0.99 | 0.88 |
| 23. | 0.69 | 0.15 | 0.05 | 0.15 | 0.00 | 0.63 | 0.04 |
| 24. | 0.82 | 0.81 | 0.80 | 0.81 | 0.00 | 0.98 | 0.79 |
| 25. | 0.84 | 0.85 | 0.84 | 0.85 | 0.06 | 0.98 | 0.83 |
| 26. | 0.86 | 0.86 | 0.85 | 0.86 | 0.00 | 0.98 | 0.83 |
| 27. | 0.65 | 0.54 | 0.48 | 0.54 | 0.00 | 0.90 | 0.49 |
| 28. | 0.88 | 0.88 | 0.88 | 0.88 | 0.14 | 0.99 | 0.86 |
| 29. | 0.86 | 0.88 | 0.87 | 0.88 | 0.00 | 0.99 | 0.86 |
| 30. | 0.91 | 0.91 | 0.91 | 0.91 | 0.32 | 0.99 | 0.90 |
| 31. | 0.88 | 0.87 | 0.86 | 0.87 | 0.00 | 0.99 | 0.85 |
| 32. | 0.90 | 0.11 | 0.02 | 0.11 | 0.00 | 0.62 | 0.00 |
| 33. | 0.59 | 0.53 | 0.48 | 0.53 | 0.00 | 0.90 | 0.46 |
| 34. | 0.70 | 0.68 | 0.66 | 0.68 | 0.00 | 0.96 | 0.63 |
| 35. | 0.87 | 0.15 | 0.04 | 0.15 | 0.00 | 0.63 | 0.00 |
| 36. | 0.83 | 0.82 | 0.81 | 0.82 | 0.00 | 0.98 | 0.79 |
| 37. | 0.91 | 0.92 | 0.91 | 0.92 | 0.17 | 1.00 | 0.91 |
| 38. | 0.92 | 0.92 | 0.91 | 0.92 | 0.00 | 0.99 | 0.90 |
| 39. | 0.89 | 0.89 | 0.88 | 0.89 | 0.00 | 0.99 | 0.87 |
| 40. | 0.89 | 0.88 | 0.87 | 0.88 | 0.00 | 0.99 | 0.86 |

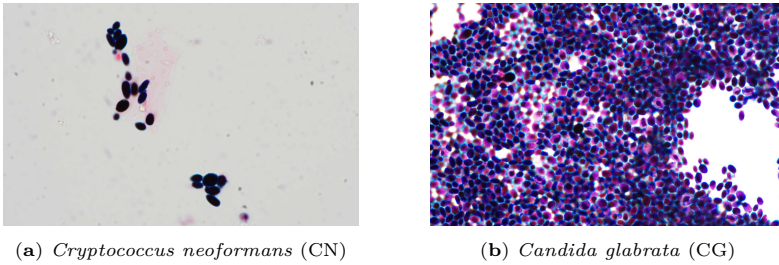


Fig. 5. CN class sample image structure compared to the CA class image.

4. Results

The models were trained on a total of 5832 subimages extracted from the original database. Noteworthy, the dataset is significantly imbalanced, with class 4 (CP) comprising the largest proportion of samples (996) and class 3 (CN) containing the fewest (98). This imbalance stems from the original dataset preparation process, which resulted in fewer basic images for class CN and, more importantly, a fundamentally different structural organization of the images themselves. CN images exhibit a sparsity of visible fungal elements, while other classes present hundreds of fungi instances across the image field (see Fig. 5). Consequently, the subimage imbalance is primarily attributable to the inherent structural disparities between the core images, rather than any shortcomings in the subimage retrieval algorithm. The original dataset authors also observed poor accuracy results for class CN, further corroborating the notion that this limitation lies within the dataset collection process, which could paradoxically contribute to the overall robustness of the proposed approach rendering at the same time low accuracy for the CN class itself [46]. Acknowledged differences can be result of sample preparation variations or natural observed phenomenon for this particular fungi spice that should be further addressed by the microbiologists.

The dataset was divided into training, testing and validation sets in a 7:2:1 ratio, ensuring that 70% of the images were assigned to the training set, 20% to the testing set and the remaining 10% to the validation set, respectively. Table 1 presents the classification results obtained on the testing set for each of the 40 models. At first, six performance metrics were tracked: Precision (the proportion of correctly identified positive instances), Recall (the proportion of positive instances that were correctly identified), F1-Score (the harmonic mean of Precision and Recall), Accuracy (the percentage of correctly classified instances), AUC (Area Under the Curve) and MCC (Matthews's correlation coefficient) that is observed due to the unbalanced input dataset [6]. Initial experiments revealed that while most models achieved satisfactory performance for eight

out of nine classes, they faced difficulties with class CN. To address this issue, an additional metric was monitored: CN class F1-Score. The green cell's color represents the highest values in each statistic for each experiment group.

The final step of the proposed pipeline involves the application of majority voting [22]. Despite its simplicity, this technique has been successfully used in various tasks, consistently enhancing the performance of classifiers [22,39]. The underlying principle behind majority voting is to combine predictions from various sources. In this context, the individual predictions correspond to the classification results obtained for single instances extracted from the input images. These subimages are then concatenated back into the original images, allowing majority voting to combine the predictions for each cell-level classification. Noteworthy, the methodology is primarily suited for monoculture scenarios, where a single fungus species dominates the image. In the case of polycultures, where multiple microorganism species coexist within a single image, majority voting should be replaced with image-level labeling. The performance evaluations for each subimage (see Tab. 1) revealed five promising models emerged as viable solutions. The ultimate selection among these candidates depends on specific business requirements and performance optimization priorities. Models 18 and 15 are clear front-runners in terms of MCC and AUC. Model 15 exhibits superior Precision compared to its counterparts, whereas models 5 and 30 merit attention for their enhanced F1-Score for the underrepresented CN class. The selected five models were subjected to majority voting to demonstrate its performance in this setting. Table 2 compares the results before and after majority voting, the number in bracket indicates increase or decrease for each metric. The inclusion of the majority voting rule consistently resulted in a slight improvement in performance, demonstrating the effectiveness of this method. As shown in the Tab. 2, in 2 out of 5 models, all metrics except AUC recorded a slight increase. Particularly notable was the improvement in F1-Score for the CN class. Following a comprehensive analysis of the obtained results, model 30 emerged as the most promising solution, consistently outperforming its counterparts across various performance metrics, including Recall, Precision, F1-Score, AUC and MCC. Notably, model 30 exhibited exceptional performance in terms of 3rd class F1-Score, surpassing most other models. Figure 7 illustrates the model's learning trajectory, demonstrating rapid convergence to a low loss value. The efficient implementation of transfer learning techniques, coupled with a well-calibrated early stopping mechanism, effectively prevented overfitting. The dropout layers incorporated into the network and the underlying ResNetv2 architecture contributed to the model's robustness. To further evaluate 30th model's versatility, the Micro-Averaged One-versus-Rest ROC curve with AUC values are examined. Fig. 8 shows the model's ability to generalize across classes and adequately handle the underrepresented CN class. Finally, the confusion matrices comparing classification before and after majority voting implementation applied on model 30 are presented in Fig. 6.

Tab. 2. Comparative performance of the leading five classification models post majority voting, with values in brackets indicating changes compared to pre-majority voting results.

| Metrics of selected models with majority voting and without it | | | | | | | |
|--|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Model | Precision | Recall | F1 | Accuracy | CN F1 | AUC | MCC |
| 5. | 0.96(+0.06) | 0.95(+0.05) | 0.94(+0.04) | 0.95(+0.05) | 0.50(+0.16) | 0.98(-0.01) | 0.94(+0.05) |
| 7. | 0.94(+0.01) | 0.92(+0.00) | 0.89(-0.02) | 0.92(+0.00) | 0.00(+0.00) | 0.95(-0.04) | 0.91(+0.00) |
| 15. | 0.96(+0.04) | 0.95(+0.03) | 0.94(+0.02) | 0.95(+0.03) | 0.00(-0.06) | 0.97(-0.02) | 0.94(+0.03) |
| 18. | 0.96(+0.05) | 0.95(+0.02) | 0.94(+0.02) | 0.95(+0.02) | 0.00(+0.00) | 0.95(-0.04) | 0.94(+0.02) |
| 30. | 0.96(+0.05) | 0.95(+0.04) | 0.94(+0.03) | 0.95(+0.04) | 0.50(+0.18) | 0.98(-0.01) | 0.94(+0.04) |

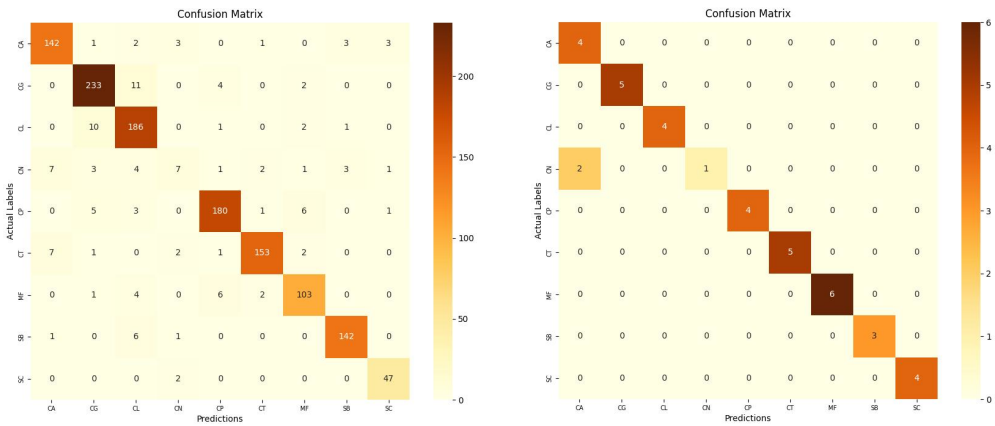


Fig. 6. Comparison of Confusion Matrices for the 30th model for each subimage (left) and after majority voting (right).

5. Conclusion

This paper describes a method for identifying microscopic images of fungi utilizing ResNets and a subimage retrieval mechanism (Fig. 2). The research highlights the significance of hyperparameter and top layer architecture tuning, and its impact on model performance, as demonstrated in Tab. 1. Notably, the best performing model employed the relatively new Mish activation function, despite the widespread use of ReLU in ResNet-based architectures for image classification tasks.

In conclusion, the method presented in this study yields promising results. First, the MCC metric, which is believed to be one of the best metrics when it comes to classifying unbalanced datasets, exceeded 0.9 both before and after majority voting for the top-performing models (Tabs. 1, 2). Furthermore, OvR ROC Curve (Fig. 8a) also presented expected results, as it rapidly approached the value of 1. Despite challenges with the

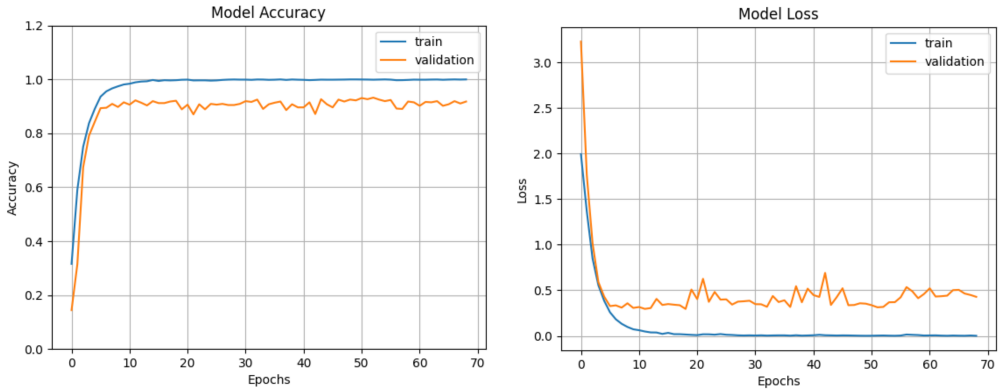


Fig. 7. Accuracy (left) and loss (right) functions during training of the 30th model on training and validation sets.

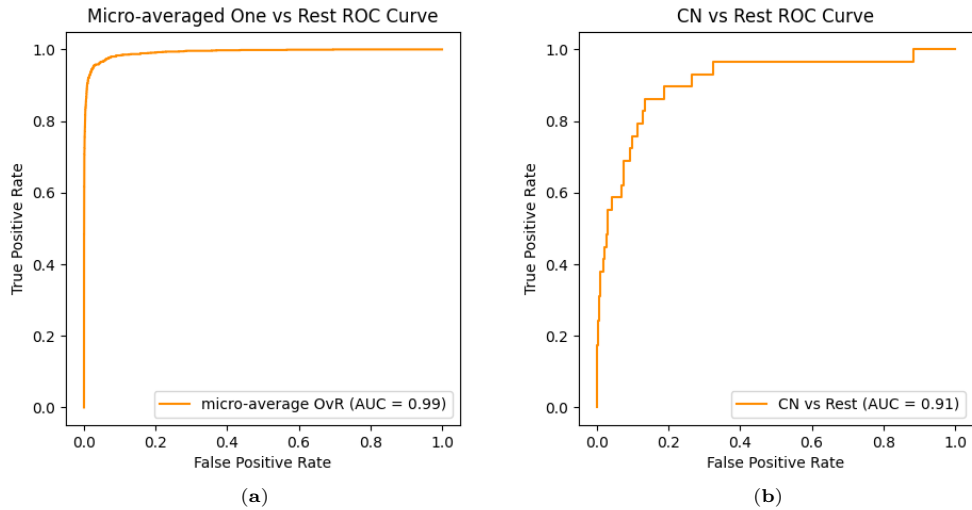


Fig. 8. AUC values and ROC curves of the 30th model on test set, micro-averaged: (a) One vs. Rest and (b) CN vs. Rest.

underrepresented CN class, Fig. 8b illustrating the ROC Curve for the CN versus Rest also showcases a promising results. Worthy of note is the fact, that after application of majority voting (Tab. 2), model 30 classified images belonging to 8 out of 9 classes correctly 100% of the time (Fig. 6). On the other hand, majority voting can also produce negative outcomes due to the fact, that if model struggles with certain class, despite some

of the subimages are classified correctly, majority voting chooses wrong class as the final prediction, decreasing metrics. Model 30 overall accuracy reached 94.74% for the testing set, classifying correctly 36/38 images. In the original paper, B. Zieliński et. al. [46] reached maximum of 82.4% obtained by aggregating patch-based classification.

The primary challenge associated with the dataset is the imbalanced distribution of classes. Despite a comparable number of microscopic images, each image encompasses significantly different sample quantities. Certain images, such as those belonging to the CP class, contained a few hundred fungi, while the CN class consisted of only a handful of samples. Enhancing the efficiency and adaptability of the model would necessitate a larger and more balanced dataset. Furthermore, conducting tests on a more extensive variety of fungal species could yield valuable insights. In a study by Cagatan et al., a VGG16-based method was introduced for identifying *Cryptococcus neoformans*, a fungal pathogen, in patient samples [5]. The initial dataset for this study comprised only 63 images, later augmented to 1000 through the generation of synthetic images using data augmentation techniques. This approach introduces an interesting concept for further system development, specifically in generating artificial images to enhance the input dataset. Convolutional-based single-instance detection methods, such as YOLOv4, represent promising avenues for future advancements, but their practical applicability should be thoroughly evaluated. In contrast to supervised methods like YOLO, the segmentation approach presented in this paper leverages well-established image operations omitting the laborious procedure of image annotation that can be significantly hard due to the fact that the microscopic images contain hundreds of small objects. Images with such intricate structures may be challenging for YOLO. Nevertheless, delving into alternative techniques for single image retrieval, a crucial aspect of the proposed methodology, opens up a promising avenue for further exploration.

References

- [1] A. Agnihotri, P. Saraf, and K. R. Bapnad. A Convolutional Neural Network approach towards self-driving cars. In: *IEEE India Conference (INDICON)*, pp. 1–4. Rajkot, India, 13-15 Dec 2019. doi:10.1109/INDICON47234.2019.9030307.
- [2] A. Ali, A. Shehzad, M. R. Khan, et al. Yeast, its types and role in fermentation during bread making process – a review. *Pak. J. Food Sci.*, 22:170–178, 2012.
- [3] S. Anwar, M. Majid, A. Qayyum, et al. Medical image analysis using convolutional neural networks: A review. *J. Med. Syst.*, 42(11):226, 2018. doi:10.1007/s10916-018-1088-1.
- [4] S. Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica (Slovenia)*, 44(3), 2020. doi:10.31449/INF.V44I3.2828.
- [5] S. Cagatan, T. Mustapha, C. Bagkur, et al. An alternative diagnostic method for *C. Neoformans*: Preliminary results of deep-learning based detection model. *Diagnostics*, 13(1):81, 2022. doi:10.3390/diagnostics13010081.
- [6] D. Chicco, N. Tötsch, and G. Jurman. The Matthews correlation coefficient (MCC) is more reliable

- than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Min.*, 14(1):13, 2021. doi:10.1186/s13040-021-00244-z.
- [7] V. M. Corbu, I. Gheorghe-Barbu, A. S. Dumbrava, et al. Current insights in fungal importance – A comprehensive review. *Microorganisms*, 11(6), 2023. doi:10.3390/microorganisms11061384.
- [8] I. Culjak, D. Abram, T. Pribanic, et al. A brief introduction to OpenCV. In: *Proc. of the International Convention (MIPRO)*, pp. 1725–1730. Opatija, Croatia, 21-25 May 2012. <https://ieeexplore.ieee.org/document/6240859>.
- [9] R. J. W. David E. Rumelhart, Geoffrey E. Hinton. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. doi:10.1038/323533a0.
- [10] J. Deng, W. Dong, R. Socher, et al. Imagenet: A large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. Miami, USA, 20-25 Jun 2009. doi:10.1109/CVPR.2009.5206848.
- [11] B. Dutta, D. Lahiri, M. Nag, et al. Fungi in pharmaceuticals and production of antibiotics. In: R. Mahendra and B. Paul D., eds., *Applied Mycology*, pp. 233–257. Springer, 2022. doi:10.1007/978-3-030-90649-8_11.
- [12] E. Gedraite and M. Hadad. Investigation on the effect of a Gaussian blur in image filtering and segmentation. In: *Proc. International Symposium on Electronics in Marine (ELMAR)*, pp. 393–396. Zadar, Croatia, 14-16 Sep 2011. <https://ieeexplore.ieee.org/xpl/conhome/6034696/proceeding>.
- [13] J. A. Hagerty, J. Ortiz, D. Reich, et al. Fungal infections in solid organ transplant patients. *Surg. Infect.*, 4(3):263–271, 2003. doi:10.1089/109629603322419607.
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in Deep Residual Networks. In: *Proc. of European Conf. of Computer Vision*, vol. 4, pp. 630–645. Amsterdam, The Netherlands, 11-14 Oct 2016. doi:10.1007/978-3-319-46493-0_38.
- [15] K. He, X. Zhang, S. Ren, et al. Deep Residual Learning for Image Recognition. *arXiv*, 2015. ArXiv:1512.03385. doi:10.48550/arXiv.1512.03385.
- [16] A. Hosna, E. Merry, J. Gyalmo, et al. Transfer learning: a friendly introduction. *J. Big Data*, 9:102, 2022. doi:10.1186/s40537-022-00652-w.
- [17] M. Iorizzo, F. Coppola, F. Letizia, et al. Role of yeasts in the brewing process: Tradition and innovation. *Processes*, 9(5):839, 2021. doi:10.3390/pr9050839.
- [18] S. R. Kollem and B. Panlal. Enhancement of images using morphological transformations. *Glob. J. Comput. Sci. Technol.*, 4(1), 2012. doi:10.5121/ijcsit.2012.4103.
- [19] A. Konopka, K. Struniawski, and R. Kozera. Performance analysis of Residual Neural Networks in soil bacteria microscopic image classification. In: *Modelling and Simulation: The European Simulation and Modelling Conference (ESM)*, pp. 144–149. Toulouse, France, 24-26 Oct 2023.
- [20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. 60(6):84–90, 2017. doi:10.1145/3065386.
- [21] C.-C. J. Kuo. Understanding convolutional neural networks with a mathematical model. *J. Vis. Commun. Image Represent.*, 41:406–413, 2016. doi:10.1016/j.jvcir.2016.11.003.
- [22] L. Lam and S. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Trans. Syst. Man Cybern. Syst.*, 27(5):553–568, 1997. doi:10.1109/3468.618255.
- [23] C. Lass-Flörl. The changing face of epidemiology of invasive fungal disease in europe. *Mycoses*, 52(3):197–205, 2009. doi:10.1111/j.1439-0507.2009.01691.x.

- [24] Q. Li, W. Cai, X. Wang, et al. Medical image classification with convolutional neural network. In: *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 844–848. Singapore, 10-12 Dec 2014. doi:10.1109/ICARCV.2014.7064414.
- [25] S. Liu, W. Cai, Y. Song, et al. Localized sparse code gradient in Alzheimer’s disease staging. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5398–5401. Osaka, Japan, 3-7 Jul 2013. doi:10.1109/EMBC.2013.6610769.
- [26] D. Misra. Mish: A self regularized non-monotonic activation function. *ArXiv*, 2020. ArXiv:1908.08681. doi:10.48550/arXiv.1908.08681.
- [27] O. M. Niall, C. Sean, and C. Anderson. In: *Computer Vision Conference (CVC)*, vol. 1, pp. 128–144. Las Vegas, USA, 25-26 Apr 2020. doi:10.1007/978-3-030-17795-9.
- [28] F. C. Odds. *Candida and candidosis*. Bailliere Tindall, 2nd edn., 1988.
- [29] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training Recurrent Neural Networks. *arXiv*, 2013. ArXiv:1211.5063. doi:10.48550/arXiv.1211.5063.
- [30] M. A. Pfaller and D. J. Diekema. Epidemiology of invasive candidiasis: a persistent public health problem. *Clin. Microbiol. Rev.*, 20(1):133–163, 2007. doi:10.1128/CMR.00029-06.
- [31] S. Rawat, B. Bisht, V. Bisht, et al. Mefunx: A novel meta-learning-based deep learning architecture to detect fungal infection directly from microscopic images. *Franklin Open*, 6:100069, 2024. doi:https://doi.org/10.1016/j.fraope.2023.100069.
- [32] C. F. Rodrigues, S. Silva, and M. Henriques. *Candida glabrata*: a review of its features and resistance. *Eur. J. Clin. Microbiol.*, 33(5):673–688, 2014. doi:10.1007/s10096-013-02009-3.
- [33] P. Sahoo, S. Soltani, and A. Wong. A survey of thresholding techniques. *Comput. Graph. Image Process.*, 41:233–260, 1988. doi:10.1016/0734-189X(88)90022-9.
- [34] M. Schaefer, S. Migge-Kleian, and S. Scheu. *The Role of Soil Fauna for Decomposition of Plant Residues*, pp. 207–230. Springer Berlin Heidelberg, 2009. doi:10.1007/b82392_13.
- [35] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. doi:10.1016/j.neunet.2014.09.003.
- [36] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, and K. M. Singh. A new local adaptive thresholding technique in binarization. *ArXiv*, 2012. ArXiv:1201.5227. doi:10.48550/arXiv.1201.5227.
- [37] R. Srisha and A. Khan. Morphological operations for image processing : Understanding and its applications. pp. 17–19. Tamil Nadu, India, 26-27 Apr 2013. <http://www.annauniv.edu/pdf/NCSCV.pdf>.
- [38] K. Struniawski, A. Konopka, and R. Kozera. Identification of soil bacteria with machine learning and image processing techniques applying single cells’ region isolation. In: *Modelling and Simulation: The European Simulation and Modelling Conference (ESM)*, pp. 76–81. Porto, Portugal, 26-28 Oct 2022.
- [39] K. Struniawski, R. Kozera, P. Trzcinski, et al. Automated identification of soil fungi and chromista through convolutional neural networks. *Eng. Appl. Artif. Intell.*, 127, 2024. doi:10.1016/j.engappai.2023.107333.
- [40] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, et al. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. doi:10.7717/peerj.453.
- [41] G. Xie and W. Lu. Image edge detection based on OpenCv. *Int. J. Electr. Electron. Eng. Telecommun.*, 1(1):104–106, 2013. doi:10.12720/IJEEE.1.2.104-106.
- [42] Z. Xie, J. Li, and H. Shi. A face recognition method based on cnn. In: *High Performance Computing and Computational Intelligence Conf.*, vol. 1395, p. 012006. Chengdu, China, 25–27 Oct 2019. doi:10.1088/1742-6596/1395/1/012006.

- [43] K. You, M. Long, J. Wang, and M. I. Jordan. How does learning rate decay help modern neural networks. *arXiv*, 2019. ArXiv:1908.01878. doi:10.48550/arXiv.1908.01878.
- [44] T. Yu and H. Zhu. Hyper-parameter optimization: A review of algorithms and applications. *ArXiv*, 2020. ArXiv:2003.05689. doi:10.48550/arXiv.2003.05689.
- [45] F. Zhuang, Z. Qi, K. Duan, et al. A comprehensive survey on transfer learning. *Proc. IEEE*, 109(1):43–76, 2021. doi:10.1109/JPROC.2020.3004555.
- [46] B. Zieliński, A. Sroka-Oleksiak, D. Rymarczyk, et al. Deep learning approach to describe and classify fungi microscopic images. *PLOS ONE*, 15(6):e0234806, 2020. doi:10.1371/journal.pone.0234806.

RELATIONSHIPS BETWEEN COLORIZATION AND PSEUDO-COLORIZATION OF MONOCHROME IMAGES

Andrzej Śluzek 

*Institute of Information Technology
Warsaw University of Life Sciences – SGGW
Warsaw, Poland*

Abstract This paper investigates the relationship between colorization and pseudo-colorization techniques for converting grayscale images to color. Colorization strives to create visually believable color versions of monochrome images, either replicating the original colors or generating realistic, alternative color schemes. In contrast, pseudo-colorization maps grayscale intensities to pre-defined color palettes to improve visual appeal, enhance content understanding, or aid visual analysis. While colorization is an ill-posed problem with infinitely many RGB solutions, pseudo-colorization relies on mapping functions to deterministically assign colors. This work bridges these techniques by exploring the two following operations: first – deriving pseudo-color from colorized images – this allows for creating stylized or abstract representations from existing colorizations, and second – enriching color diversity in pseudo-colored images – this enhances visual appeal and attractiveness of pseudo-colored images. The paper emphasizes the centrality of decolorization (*rgb-to-gray*) models in both processes. It focuses on the theoretical underpinnings of these problems but complements them with illustrative examples for clarity.

Keywords: colorization, pseudo-colorization, decolorization, rgb-to-gray models, color maps, randomized flood-fill.

1. Introduction and motivation

While both (re)colorization and pseudo-colorization aim to create color versions of monochrome images, their approaches, complexities, and applications differ significantly.

Colorization is the process of converting monochrome images into visually convincing color-rich counterparts. In the case of recolorization, the goal is to replicate the original colors of the image, even though they are unknown to the algorithm, e.g. [3, 6, 16, 18, 22, 23], etc. Alternatively, the objective is to create realistic-looking hypothetical color versions of grayscale images, representing worlds that might not have originally been in color, e.g. [10, 25].

This field holds significant practical and commercial value, particularly in restoring historical photos and movies. Over the past two decades, researchers have proposed numerous and diverse (re)colorization algorithms. However, a fundamental challenge of colorization is its inherent ambiguity. There are infinitely many valid RGB combinations that, when converted back to grayscale, would appear identical.

Thus, colorization techniques typically leverage human knowledge and expectations to guide the algorithms, [22]. Early methods relied on providing reference color images [5, 7] or manual coloring (*scribbling*) of specific image regions [11, 13]. Recently, deep



Fig. 1. Two grayscale images (from a popular SUN dataset [21] and an IR image) and their exemplary colorizations. The results are produced by a method outlined in [24,25] (i.e., without the use of training data, AI tools, semantic analysis, human assistance, etc.).

learning approaches have become dominant, with neural networks designed to learn color patterns suitable for specific domains, semantics, or content, [3,23]. Some techniques additionally incorporate recognition or learning of image domains or objects to further enhance colorization, e.g., [6,18].

While some papers claim fully automatic (re)colorization, e.g. [10,19], they often still rely on implicit human color knowledge through training datasets and semantic identification mechanisms. In contrast, our recent works [24,25] propose a mechanism for truly automatic image colorization, entirely without human intervention, training data, learning processes, or semantic analysis. This approach specifically targets images from domains where original color versions likely never existed.

Fig. 1 summarizes this concept by showcasing grayscale images with various colorized options for each. For the first image (from the real world), one result might resemble the actual colors, although the *ground-truth* is unknown.

Unlike colorization, pseudo-colorization is a simpler process. It assigns a pre-defined color map to image intensities. The key challenge lies in selecting or designing this mapping function to achieve specific goals, such as enhancing visual appeal, improving content perception, or facilitating the visual analysis of objects or processes within the image, e.g. [9,12,15,20].

There are a number of standard color maps, commonly referred to as *hot*, *warm*, *rainbow*, *springtime*, *jet*, *sine*, etc. (see [1,15,17]). Examples are given in Figs. 2 and 3. They are often used in typical applications of pseudo-coloring, including thermography, roentgenography or geographical visualizations.

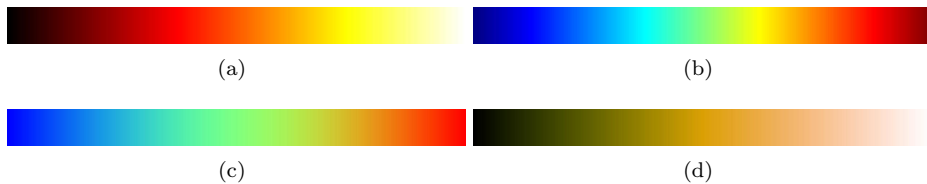


Fig. 2. Some popular color maps used for pseudo-colorization: (a) *hot*, (b) *jet*, (c) *sine* and (d) *warm*.

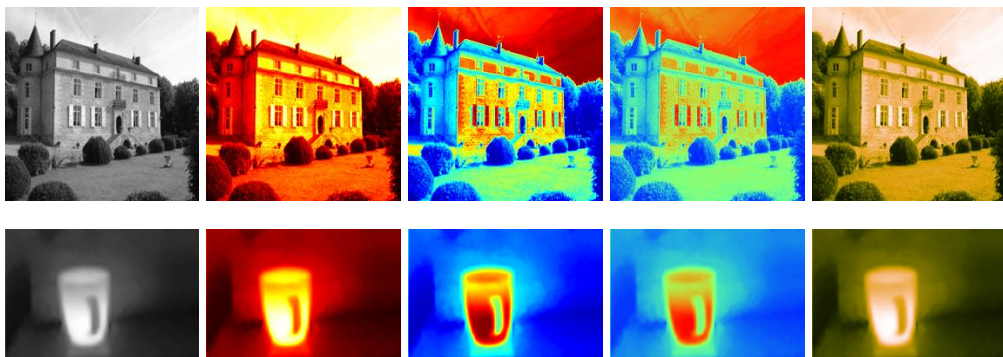


Fig. 3. Two grayscale images and their exemplary pseudo-colorizations obtained by using the corresponding color maps from Fig. 2 (compare to Fig. 1).

Finally, it should be noted that in this paper, we do not consider the so-called *pseudo-coloring problem* (PsCP), which, despite its similar name, is a significantly different task. PsCP involves segmenting a monochrome image into regions and the objective is to assign a set of colors to these regions, with the colors being as visually distinct as possible (e.g., [2, 14]).

Colorization can yield infinitely many valid results for a single grayscale image, while pseudo-colorization produces unambiguous colors based on a pre-defined mapping function. This paper bridges the gap between these two seemingly distinct color-rendering techniques. We aim to explore how to transform colorized images into pseudo-colored ones and vice versa, while preserving the overall color perception of the image. Notably, these methods should be generalizable, functioning across various image content and characteristics.

To our knowledge, no prior research has investigated these connections between colorization and pseudo-colorization. We believe this exploration holds significant potential, driven not only by practical applications but also by sheer scientific curiosity.

To that end, Section 2 presents formal models of image decolorization and colorization

(both regular colorization and pseudo-colorization). In particular, for regular colorization, we revisit models proposed in [24, 25].

In Section 3, the easier problem of converting colorized images into the most similar pseudo-color equivalents is discussed. This operation can be used, for example, to create a more abstract or stylized representation of an image.

The more complex task of transforming pseudo-colored images into visually similar (though, obviously, with much richer colorization effects) fully-colored images is presented in Section 4. Such operations may be required if we intend to make pseudo-colored images more aesthetically pleasing and visually engaging.

In the final Section 5, we summarize the paper and link its outcomes to the related problems of image processing and computer graphics.

Finally, it should be highlighted that this paper presents improved and updated materials originally presented at the 9th Conference on Symbiosis of Technology and IT (SIT) in Kiry, June 2023. It is important to note that these materials have not been previously published.

2. Formal models

2.1. Models of image decolorization

Given a monochrome image, we may want to know what decolorization (*rgb-to-gray*) models describe the conversion of the color original (possibly not even existing) into the grayscale results.

For rendering digital images, the RGB/sRGB color models are universally adopted, and the intensities I of decolorized images are usually defined by linear functions of primary colors, mainly by the Y channel of the YUV (or YIQ) models:

$$I = 0.299 R + 0.587 G + 0.114 B \quad (1)$$

Alternatively, another set of coefficients can be used almost equivalently, as recommended in [8]:

$$I = 0.2126 R + 0.7152 G + 0.0722 B \quad (2)$$

The advantage of Eqs. (1) and (2) is that they take human perception into account, and the perceived brightness of resulting grayscale images looks very similar to the brightness of color originals (see examples in Figs. 4b, c).

However, if we assume that original color images do not exist (i.e., the *ground-truth* colors are unspecified), we can use any set of *rgb-to-gray* coefficients (satisfying $k_R + k_G + k_B = 1$) to ‘decolorize’ the hypothetical color sources:

$$I = k_R R + k_G G + k_B B \quad (3)$$



Fig. 4. A color image and its monochrome counterparts obtained by Eq.1 (b), Eq.2 (c), and by Eq.3 with two random sets of coefficients, $[0.14, 0.11, 0.75]$ in (d) and $[0.44, 0.14, 0.42]$ in (e).

In case of actual color images, their monochrome counterparts obtained by using (3) may sometimes look strange when compared to the original images (see Figs. 4d, e) but, nevertheless, they are visually convincing.

Thus, applying (3) with different coefficients $[k_R, k_G, k_B]$ allows us to generate an infinite spectrum of monochrome versions of a color image. This might seem counter-intuitive, but it aligns with the concept of having infinite RGB combinations that can create a single grayscale image.

Actually, decolorization models are the backbone of other methods discussed in this paper. Their equations indirectly define the colorization results (Subsection 2.3), they are instrumental in converting color images into pseudo-colored ones (Section 3), and in the operation of rich colorization of pseudo-colored images (Section 4).

2.2. Models of pseudo-colorization

Pseudo-colorization models are defined by vector-functions (*color maps*) of the grayscale intensities I :

$$R = R(I), \quad G = G(I), \quad B = B(I) \quad (4)$$

where the $[0 : 1]$ range is generally assumed for the intensities and (subsequently) for the color values.

For the exemplary *sine* model the equations are as follows:

$$R(I) = -0.5 \cos(\pi I) + 0.5, \quad G(I) = \sin(\pi I), \quad B(I) = 0.5 \cos(\pi I) + 0.5, \quad (5)$$

with more examples of color map equations available in [1, 15].

It can be observed that the *color map equations* define parametric curves within the RGB unit cube. The curves start at $[R(0), G(0), B(0)]$ and terminate at $[R(1), G(1), B(1)]$. Normally, the colors assigned to intensities change incrementally, i.e., the curves are continuous. Shapes of such curves for exemplary color maps are given in Fig. 5.

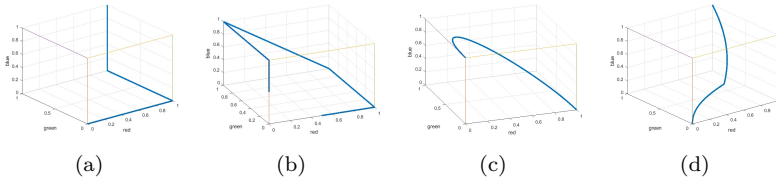


Fig. 5. Curves defined within the *RGB* cube by exemplary color mapping functions: (a) *hot*, (b) *jet*, (c) *sine* and (d) *warm*.

In some tasks, we may want pseudo-colorization results that preserve the perceived brightness of the original monochrome images. To that end, a color $C_u = [R_u, G_u, B_u]$ assigned to the intensity I_u can be replaced by another color $C_{pr} = [R_{pr}, G_{pr}, B_{pr}]$ which satisfies

$$0.299R_{pr} + 0.587G_{pr} + 0.114B_{pr} = I_u \quad (6)$$

and minimizes $\|C_u - C_{pr}\|$ (alternatively, in the above formula, Eq. (2) can be used instead of (1)).

Usually, the solution of (6) is obtained by a simple orthogonal projection of the C_u color onto the $0.299R + 0.587G + 0.114B = I_u$ plane, unless the result lies outside the *RGB* cube. In such cases, additional steps are needed to find the closest color on the boundary of the *RGB* cube.

Such color transformations will be referred to as *YUV* projections. Exemplary results of such projections are given in Fig. 6.

2.3. Models of colorization

Colorization techniques typically assume that the intensity of monochrome images defines the luminance of colored outputs, and only two chrominance channels should be reconstructed. Usually, there are no other formal models used in the popular (re)colorization methods, which are currently dominated by AI.

In this work, however, we propose to use decolorization (*rgb-to-gray*) models, as specified in Subsection 2.1, to define the models of colorization. Thus, given a monochrome image, we assume that it was obtained from a hypothetical color image by applying Eq. 3 with specific values $[k_R, k_G, k_B]$ coefficients.

The number of such models can be very large, depending on the quantization step applied to the coefficients (which can range from 0 to 1, subject to $k_R + k_G + k_B = 1$). For example, there are 5151 decolorization models with the 0.01 quantization increment and 125,751 models with the 0.002 stepsize. Formally, any number of decolorization models can be used. However, in practice, it is best to use only a limited subset, typically between 20 and 40, of the most representative models. This results in a number of

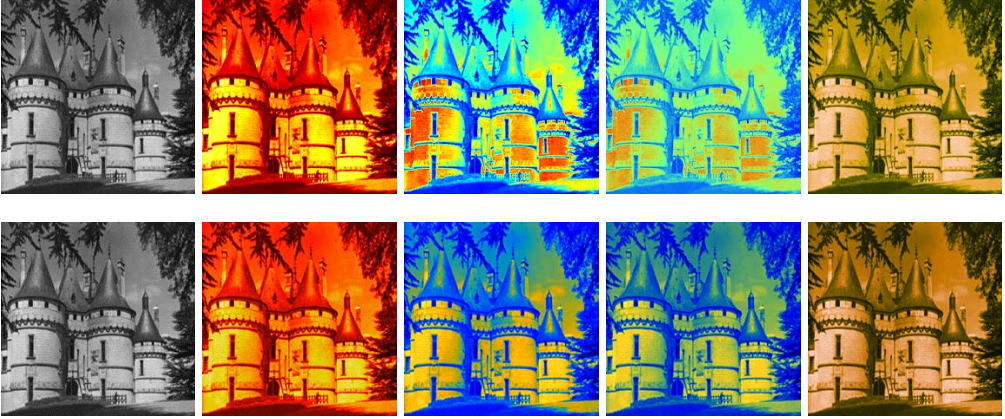


Fig. 6. *Hot*, *jet*, *sine* and *warm* pseudo-colorizations (of an exemplary monochrome image from SUN dataset) before (top row) and after (bottom row) the YUV projections (by using Eq.6).

colorization variants from which the user can choose the most plausible one. If too many *rgb-to-gray* models are used simultaneously, the sheer number of alternative colorizations may be overwhelming for human evaluators. (Details are explained in [26].)

Given, in digital images, the finite number of intensity levels (e.g., from 0 to 255) and, correspondingly, finite numbers of RGB colors, a pixel with intensity I can only be assigned colors that satisfy Eq. (3) with the adopted $[k_R, k_G, k_B]$ values. All such colors should be considered viable choices for coloring an I -valued monochrome pixel. It can be noted that the pool of available colors actually diminishes for darker/lighter intensities (as shown in Fig. 7), and for the extreme values the choice is deterministic (i.e., grayscale *white/black* should remain *white/black* in color).

As an illustration, Fig. 8 shows colors assigned to 208 intensity in two exemplary *rgb-to-gray* models. Note that the first model is actually YUV (and all colors are perceived as having almost the same brightness) while for the second model the perceived brightness of colors varies significantly.

Alternatively, all colors assigned to the selected intensity I_u can be visualized as the polygonal intersection of the $k_R R + k_G G + k_B B = I_u$ plane with the RGB color space cube. An example is given in Fig. 9.

With no prior information provided, all colors available to the I_u intensity can be assigned to a pixel of that value with the same probability. However, if the pixel has an adjacent pixel with an intensity I_1 and its already assigned color C_1 , the probabilities of colors that could be assigned to I_u should be influenced by the intensity and color of the neighbor.

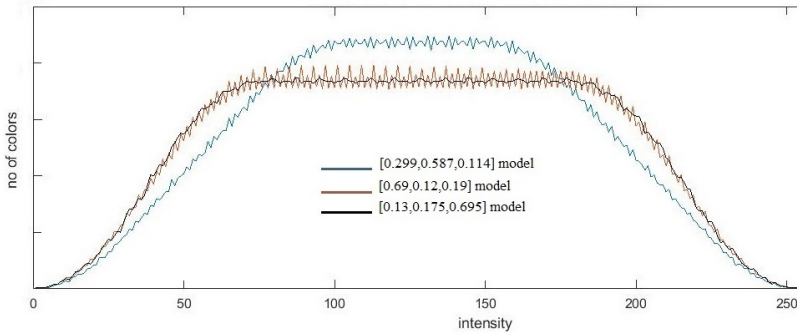


Fig. 7. Changes in the numbers of colors assigned to various intensities for $[0.299, 0.587, 0.114]$, $[0.69, 0.12, 0.19]$ and $[0.13, 0.175, 0.695]$ *rgb-to-gray* models.

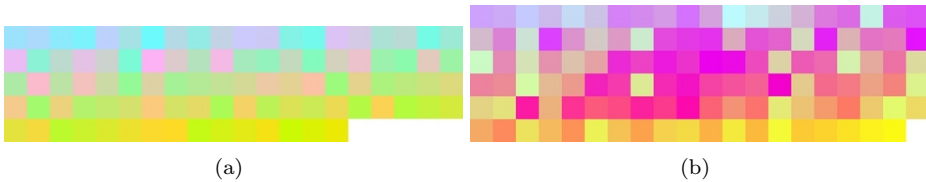


Fig. 8. Exemplary colors assigned to 208 intensity in (a) $[0.299, 0.587, 0.114]$ and (b) $[0.69, 0.12, 0.19]$ models.

Therefore, we use a simple but surprisingly effective (as shown in [24,25,26]) heuristic rule:

The greater the difference in brightness between adjacent pixels, the higher the likelihood that their assigned colors will also differ significantly.

Under this rule, we prioritize colors from the pool available to the level I_u , which are at distances from the color C_1 proportional to the difference in intensity levels $\|I_u - I_1\|$.

Then, pixels can be colored using the following color assignment method:

1. Let's assume a pixel with I_u intensity, which has an already colored neighbor with I_1 intensity and C_1 color. Let $\mathbf{C} = \{C_{u_1}, \dots, C_{u_N}\}$ be the list of colors available to I_u in the adopted *rgb-to-gray* model (see examples in Fig. 8).
2. The neighbor (with I_1 value and C_1 color) contributes a color from the above \mathbf{C} list. First, the list is ordered by the distances of its colors from C_1 , i.e.,

$$\mathbf{C}_{\text{mod}} = \{C_{u_{i_1}}, \dots, C_{u_{i_N}}\}, \text{ where } \|C_{u_{i_n}} - C_1\| \leq \|C_{u_{i_{(n+1)}}} - C_1\| \quad (7)$$

In fact, the displayed lists of colors in Fig. 8 are already ordered (for $I_1 = 40$ and $C_1 = [20, 42, 137]$).

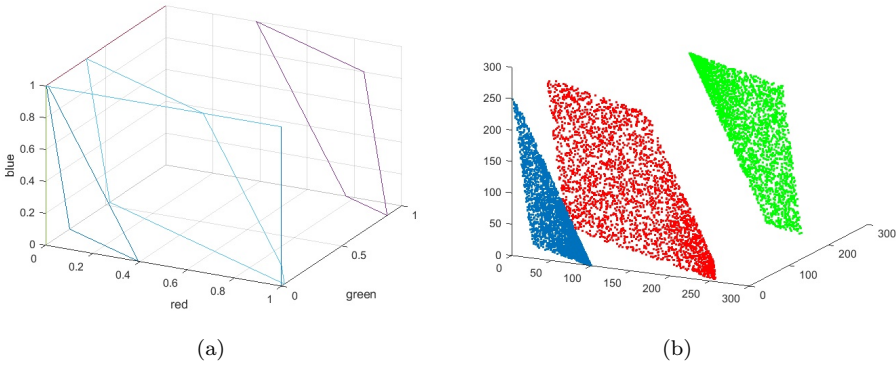


Fig. 9. Distribution of colors assigned to three exemplary intensities in (a) RGB cube and (b) sRGB cube. The assumed *rgb-to-gray* model is $I = 0.299R + 0.587G + 0.114B$. The intensities are 30, 80 and 208 (in sRGB), i.e., 0.118, 0.314 and 0.816 (in RGB).

3. A uniform distribution is used to randomly select a color from a specific sub-range of the **Cmod** list. The location of this sub-range depends on the difference $\|I_u - I_1\|$. In general, smaller differences favor colors from a narrower range near the top of the **Cmod** list, while larger differences favor a wider range of colors towards the end of **Cmod**. Refer to [25] for a detailed explanation of this step.

To achieve a more natural look and avoid unnecessarily regular patterns, the colorization process employs a randomized variant of the queue-based *flood-fill* algorithm, e.g., [4]. This means the active pixel for coloring is randomly chosen from the current queue.

The initial queue consists of the brightest and darkest pixels, for which the color selection is usually deterministic (see Fig. 7). Pixels are then colored using the previously described steps until the queue is empty. However, during colorization, an uncolored pixel might have several already colored neighbors. In these cases, the color selection process (described in Point 3) is repeated multiple times. The final color assigned to the pixel is then the average of the colors obtained from all its colored neighbors.

$$C_u = \frac{1}{K} \sum_{k=1}^K C_{u_k}, \text{ where } K = 1, 2, 3 \text{ or } 4. \quad (8)$$

Despite all these randomizing factors, the results produced by the outlined method are surprisingly repeatable, depending mainly (as expected) on the adopted *rgb-to-gray* model. In fact, the images shown in Fig. 1 are generated by this method, and many more examples can be found in [24, 25, 26].

The visual plausibility of the results can be further improved by applying the YUV

projection outlined in Subsection 2.2. Through visual inspection, it could be identified that several images in Fig. 1 are actually rectified using this approach.

2.4. Remarks on alternative *rgb-to-gray* models

In the decolorization and colorization operations discussed in Subsections 2.1 and 2.3, we assume *rgb-to-gray* models with the fixed values of adopted $[k_R, k_G, k_B]$ coefficients. Nevertheless, the process of colorization can be generalized by assuming that $[k_R, k_G, k_B]$ coefficients are individually assigned to each intensity level. In this case, the following equation should be used instead of Eq. 3 (without affecting the presented methodology of colorization):

$$I = k_R(I)R + k_G(I)G + k_B(I)B \quad (9)$$

However, such a generalization has two weaknesses. First, the number of alternative colorization variants would become unimaginably large. For example, with a step size of 0.002 for the model coefficients, the number of variants would approach $125\,751^{256}$. Second, the *rgb-to-gray* models by Eq. (9) are useless in the decolorization operation because the intensity values are computed from the coefficients, which depend on the unknown intensities themselves. Nevertheless, we will further explore these alternative models in Section 4.

3. Pseudo-colorization of colorized images

The conversion of color (or colorized) images into pseudo-colored ones seems straightforward at first. We simply need to reduce the number of colors to 256, for example by clustering all colors present in an image into 256 classes and then performing the corresponding substitutions.

However, there are certain formal complications. As discussed in Subsection 2.2, pseudo-colorization models are defined by their *color maps*. This means that pseudo-colorization can only be performed on a monochrome image. Therefore, for any color (or colorized) image, we first need to have its monochrome version, which can then be pseudo-colored.

In Subsection 2.1, we assumed that grayscale images are obtained from color images by applying decolorization (*rgb-to-gray*) models defined by Eq. 3, where the $[k_R, k_G, k_B]$ coefficients have specific values. Once these values are assumed or identified, pseudo-colorization can be easily performed as explained in the following description.

First, as highlighted in Subsection 2.3, for the adopted *rgb-to-gray* model, any intensity defines the polygonal intersection with the RGB cube (see Fig. 9). The centers of mass of such polygons, i.e., the average colors assigned to intensities ranging from 0 to 1 (0 to 255), actually form a curve winding (from *black* to *white*) within the cube.

Such curves are equivalent to the curves defining the *color maps* (see Fig. 5) and can be employed as such.

Formally, for the adopted $[k_R, k_G, k_B]$ model, the pseudo-color assigned to I intensity is defined by

$$[R(I), G(I), B(I)] = \text{center of mass}[(\text{RGB cube}) \cap (I = k_R R + k_G G + k_B B \text{ plane})] \quad (10)$$

Fig. 10 shows the curves, i.e. the *color maps*, defined by three exemplary *rgb-to-gray* models. Correspondingly, Figs. 11 and 12 compare outcomes of colorization and pseudo-colorization of selected monochrome images using the same *rgb-to-gray* models.

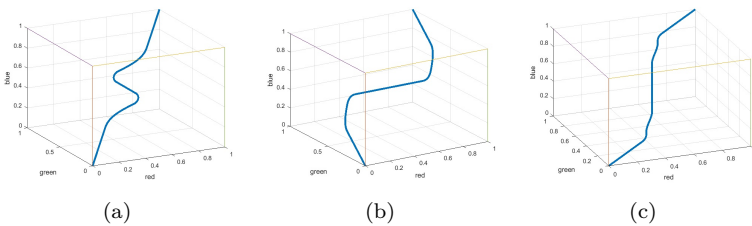


Fig. 10. Curves representing the *color maps* obtained from three *rgb-to-gray* models: (a) $[0.299, 0.587, 0.114]$, (b) $[0.69, 0.12, 0.19]$ and (c) $[0.13, 0.175, 0.695]$.

The presented examples indicate that, despite a reduced number of colors, the pseudo-colored images approximately preserve the coloristic perception of the (fully-)colorized images, although the general impression is simplified and more abstract.

4. Colorization of pseudo-colored images

Colorization of pseudo-colored images is once again based on *rgb-to-gray* models. Given a color map CM (and its equivalent curve, see Figure 5), we attempt to identify an *rgb-to-gray* model $\text{MOD}(\text{CM})$ that would define (as explained in Section 3) a color map as close as possible to CM. Then, a monochrome image can be colorized (using the method outlined in Subsection 2.3) with the identified $\text{MOD}(\text{CM})$ model adopted.

Unfortunately, the color map curves obtained from *rgb-to-gray* models by averaging colors assigned to particular intensities always start at *black* and terminate at *white* (see Fig. 10), while curves of arbitrary color maps can start and terminate at any colors within the RGB cube (e.g., Fig. 5b, c).

Therefore, as the preliminary step, the color map CM is transformed by the YUV

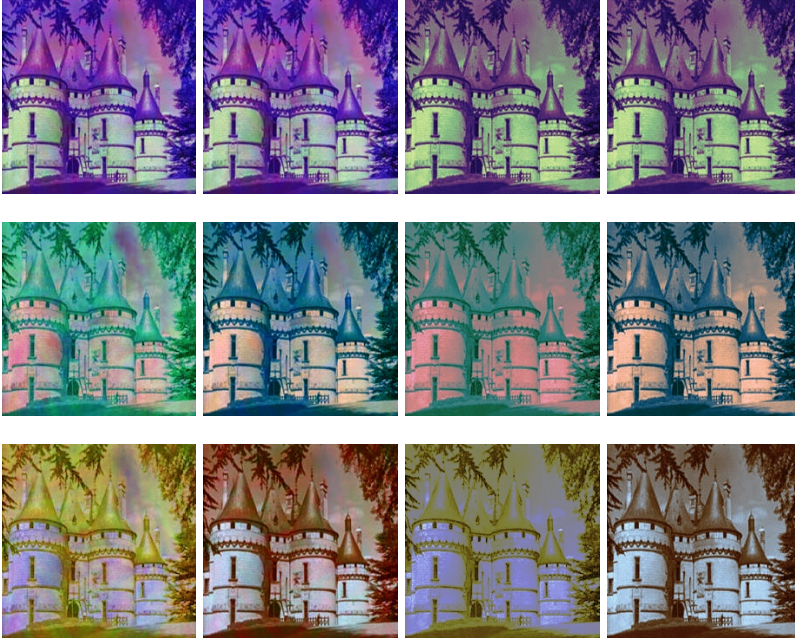


Fig. 11. Results for the grayscale image (from Fig. 6), i.e., its colorized variants (1st column), colorized variants after the YUV projection (2nd column), pseudo-colored variants (3rd column) and pseudo-colored variants after the YUV projection (4th column). The results are obtained using the following *rgb-to-gray* models: [0.299, 0.587, 0.114] (top), [0.69, 0.12, 0.19] (center) and [0.13, 0.175, 0.695] (bottom). Note that the first model is actually YUV, so the results are the same before and after the projection.

projection, and for each intensity I the shift from its original pseudo-color to the YUV-projected pseudo-color is represented by the T_{yuv} vector:

$$T_{yuv}(I) = [R_{YUV}(I), G_{YUV}(I), B_{YUV}(I)] - [R(I), G(I), B(I)], \quad (11)$$

where the original $[R(I), G(I), B(I)]$ pseudo-color is transformed by the YUV projection into $[R_{YUV}(I), G_{YUV}(I), B_{YUV}(I)]$.

Thus, the curves of YUV-projected color maps always start at *black* and terminate at *white* (see examples in Fig. 13).

Next, for each intensity I , we individually identify the decolorization model (e.g., from 125 751 available *rgb-to-gray* models) which minimizes

$$[k_R(I), k_G(I), k_B(I)] = \min \|k_R R_{YUV}(I) + k_G G_{YUV}(I) + k_B B_{YUV}(I) - I\|, \quad (12)$$

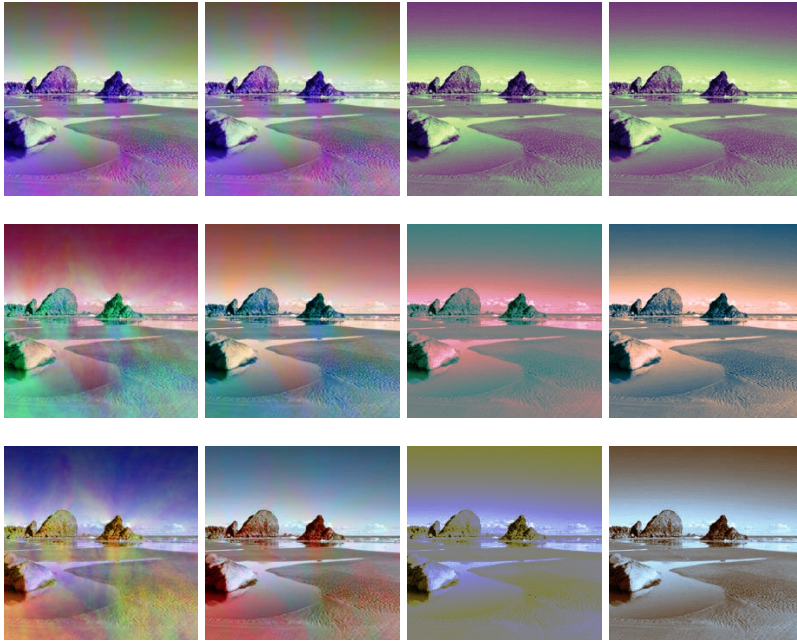


Fig. 12. The same results as in Fig. 11 for another grayscale image (from SUN dataset).

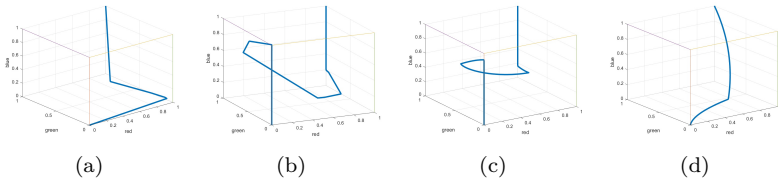


Fig. 13. The YUV-projected curves of exemplary *color maps*: (a) *hot*, (b) *jet*, (c) *sine* and (d) *warm*. Compare the shapes to Fig. 5

i.e., the model which most accurately converts the modified pseudo-color into the corresponding intensity.

The coefficients obtained in (12) define the optimum *rgb-to-gray* model $MOD_{opt}(CM)$, which will be adopted for colorizing monochrome images which are pseudo-colored with the original CM color map.

Crucially, Eq. 12 defines separate $[k_R(I), k_G(I), k_B(I)]$ coefficients for each intensity

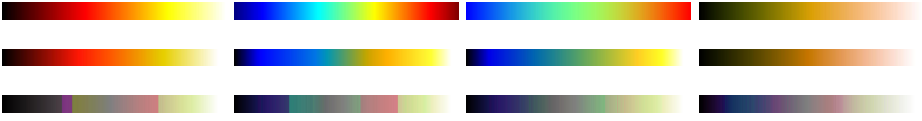


Fig. 14. Exemplary color maps (from left to right: *hot*, *jet*, *sine* and *warm*) with their original pseudo-colors (top row), pseudo-colors after YUV projections (middle row) and colors of the maps converted from the $MOD_{opt}(CM)$ models (bottom row).

level. This implies that Eq. (9) (see Subsection 2.4) truly captures the essence of the $MOD_{opt}(CM)$ model, not Eq. (3). Nevertheless, such a modification does not affect the model of colorization presented in Section 2.3.

The obtained model $MOD_{opt}(CM)$ can be back-converted into the color map using the approach discussed in Section 3. In this approach, for each intensity I its pseudo-color is specified by the center of mass of the polygon defined by the $[k_R(I), k_G(I), k_B(I)]$ coefficients.

Obviously, the color map obtained in this way differs not only from the original CM map but also from the YUV-projected map. This is because the centers of polygons defined by $[k_R(I), k_G(I), k_B(I)]$ coefficients (these centers will be referred to as $[R_{CoM}(I), G_{CoM}(I), B_{CoM}(I)]$) are almost always far from the $[R_{YUV}(I), G_{YUV}(I), B_{YUV}(I)]$ colors (i.e., YUV-projected pseudo-colors of the original CM map) which can be anywhere within the corresponding polygons.

Fig. 14 shows exemplary original color maps, the maps after applying the YUV projections, and eventually compares them to the maps obtained from the corresponding $MOD_{opt}(CM)$ models.

Therefore, the images colorized by the $MOD_{opt}(CM)$ models significantly differ in terms of coloristic perception from the images pseudo-colored by the original CM maps. However, we propose a simple correction that reverses the changes introduced in the color maps so that the corrected images are perceptually consistent with the original pseudo-color images, albeit with much richer diversity of colors.

Given a pixel of (x, y) coordinates and with the original I intensity, its color is corrected as follows:

$$RGB_{new}(x, y) = RGB(x, y) - T_{off} \{I(x, y)\}, \quad (13)$$

where $T_{off}(I) = [R_{CoM}(I), G_{CoM}(I), B_{CoM}(I)] - [R(I), G(I), B(I)]$, i.e. it represents the offset between the original pseudo-color and the corresponding pseudo-color of the map reconstructed from the $MOD_{opt}(CM)$ model.

If the vector resulting from Eq. (13) extends outside the RGB cube, we select the color at the intersection of this vector with the cube boundary.

Figs. 15 and 16 illustratively summarize the results of converting pseudo-colored

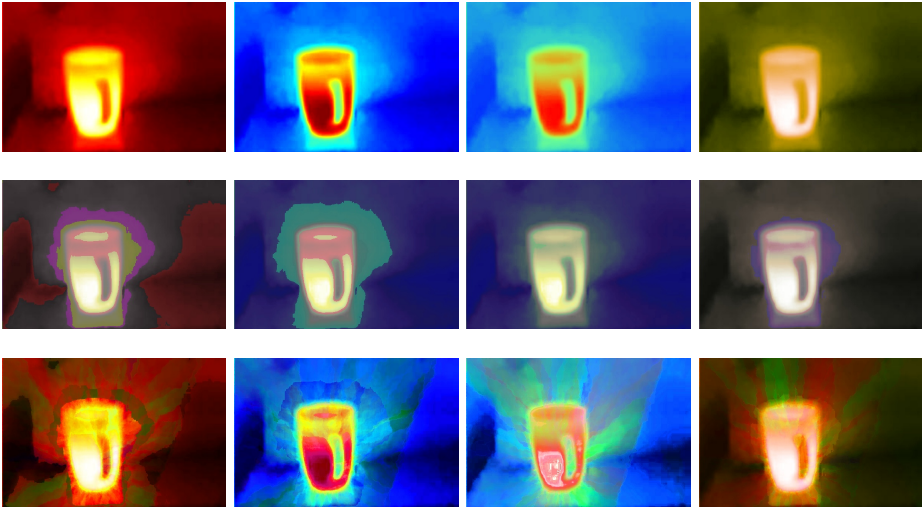


Fig. 15. First row: an image pseudo-colored by (from left to right) *hot*, *jet*, *sine* and *warm* color maps. Second row: the corresponding colored images before the color correction by Eq. 13. Third row: the colorized images after the color correction by Eq. 13.

versions of two previously discussed exemplary images into their fully-colored variants, using four different color maps.

The colorized images correspond well to the pseudo-colored originals in terms of overall coloristic perception, but the added variability of colors enriches the attractiveness and visual appeal of the results.

5. Conclusions

This paper proposes a novel and general approach that bridges the gap between image colorization and pseudo-colorization, which are fundamentally distinct concepts. The key unifying element is the adopted model for image decolorization (*rgb-to-gray*) in which we hypothesize that grayscale image intensities are linear combinations (represented by $[k_R, k_G, k_B]$ coefficients) of primary colors from a potentially non-existent color image.

In image colorization, these coefficients are freely chosen, allowing for the creation of diverse colorized versions of a grayscale image. These variations may not necessarily reflect the “true” colors of the scene, but rather offer aesthetically pleasing visualizations of imagined worlds. For each colorized image, a corresponding pseudo-colored variant can be derived by leveraging the chosen $[k_R, k_G, k_B]$ coefficients.

Colorization of pseudo-colored images follows a different approach. Here, the color

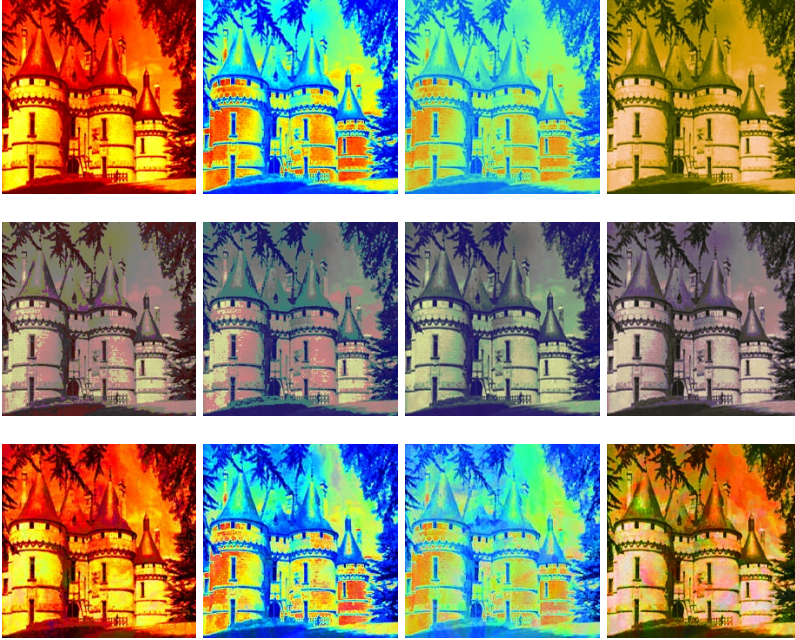


Fig. 16. The same results as in Fig. 15 for another image pseudo-colored by four color maps.

maps themselves define the *rgb-to-gray* models, with unique $[k_R, k_G, k_B]$ coefficients for each intensity level.

Notably, both the conversion from colorized to pseudo-colored images and vice versa result in images with perceptually similar color palettes, containing either a reduced or enriched range of colors.

The proposed methodologies, verified through experimentation, offer a solution to an intriguing challenge, though their immediate practicality might be limited. Nonetheless, this solution is directly applicable for scenarios where preserving the overall color perception of an image is crucial, but the goal is to create either an abstract/stylized representation of a full-color image or a more visually engaging and expressive variant of a pseudo-colored image.

Beyond the scope of this paper, preliminary investigations suggest the potential of this approach for a challenging task: recovering visual artifacts lost during the conversion of color originals to monochrome or pseudo-colored images. This promising avenue merits further exploration in future work.

Furthermore, converting pseudo-colored images into fully colorized ones could serve

as a stepping stone for creating or enriching training datasets for AI-based colorization. This is particularly relevant for domains beyond human color perception, where we only have access to grayscale information, such as infrared, ultraviolet, ultrasound, MRI, or X-ray modalities.

References

- [1] B. Abidi, Y. Zheng, A. Gribok, and M. Abidi. Screener evaluation of pseudo-colored single energy X-ray luggage images. In: *Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) – Workshops*, pp. 35–35, 07 2005. doi:[10.1109/CVPR.2005.521](https://doi.org/10.1109/CVPR.2005.521).
- [2] R. C. Contreras, O. Morandin Junior, and M. S. Viana. A new local search adaptive genetic algorithm for the pseudo-coloring problem. In: Y. Tan, Y. Shi, and M. Tuba, eds., *Advances in Swarm Intelligence*, pp. 349–361. Springer International Publishing, 2020. doi:[10.1007/978-3-030-53956-6_31](https://doi.org/10.1007/978-3-030-53956-6_31).
- [3] E. Farella, S. Malek, and F. Remondino. Colorizing the past: Deep learning for the automatic colorization of historical aerial images. *Journal of Imaging*, 8:269, 10 2022. doi:[10.3390/jimaging8100269](https://doi.org/10.3390/jimaging8100269).
- [4] K. Fishkin and B. Barsky. An analysis and algorithm for filling propagation. In: *Computer-Generated Images. The State of the Art Proceedings of Graphics Interface '85*, pp. 56–76. Springer, 1985. doi:[10.1007/978-4-431-68033-8_6](https://doi.org/10.1007/978-4-431-68033-8_6).
- [5] R. Gupta, A. Chia, D. Rajan, E. Ng, and Z. Huang. Image colorization using similar images. In: *Proc. 20th ACM Int. Conf. on Multimedia (MM'12)*, pp. 369–378, 2012. doi:[10.1145/2393347.2393402](https://doi.org/10.1145/2393347.2393402).
- [6] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics*, 35:1–11, 07 2016. doi:[10.1145/2897824.2925974](https://doi.org/10.1145/2897824.2925974).
- [7] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. In: *Proc. Eurographics Symposium on Rendering (2005)*. The Eurographics Association, 2005. doi:[10.2312/EGWR/EGSR05/201-210](https://doi.org/10.2312/EGWR/EGSR05/201-210).
- [8] ITU-R. Parameter values for the HDTV standards for telecommunication and international programme exchange. Recommendation BT.709-6, International Telecommunication Union, Geneva, 2015. <https://www.itu.int/rec/R-REC-BT.709-6-201506-I/>.
- [9] M. Khan, Y. Gotoh, and N. Nida. Medical image colorization for better visualization and segmentation. In: M. Valdés Hernández and V. González-Castro, eds., *Medical Image Understanding and Analysis*, pp. 571–580. Springer International Publishing, 2017. doi:[10.1007/978-3-319-60964-5_50](https://doi.org/10.1007/978-3-319-60964-5_50).
- [10] C. Lei and Q. Chen. Fully automatic video colorization with self-regularization and diversity. In: *Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3748–3756, 2019. doi:[10.1109/CVPR.2019.00387](https://doi.org/10.1109/CVPR.2019.00387).
- [11] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 23:689–694, 06 2004. doi:[10.1145/1015706.1015780](https://doi.org/10.1145/1015706.1015780).
- [12] K. Moreland. Why we use bad color maps and what you can do about it. In: *Proc. IS&T Int. Symp. Electronic Imaging: Human Vision and Electronic Imaging*, vol. 28, 2016. doi:[10.2352/ISSN.2470-1173.2016.16.HVEI-133](https://doi.org/10.2352/ISSN.2470-1173.2016.16.HVEI-133).
- [13] A. Popowicz and B. Smolka. Fast image colourisation using the isolines concept. *Multimedia Tools and Applications*, 75:15987–16009, 2017. doi:[10.1007/s11042-016-3892-2](https://doi.org/10.1007/s11042-016-3892-2).



- [14] K. Radlak and B. Smolka. Visualization enhancement of segmented images using genetic algorithm. In: *Proc. 2014 Int. Conf. Multimedia Computing and Systems (ICMCS)*, pp. 391–396, 2014. doi:10.1109/ICMCS.2014.6911269.
- [15] A. Rahimian, M. Etehadtavakol, M. Moslehi, and E. Ng. Comparing different algorithms for the pseudo-coloring of myocardial perfusion single-photon emission computed tomography images. *Journal of Imaging*, 8(12):331, 12 2022. doi:10.3390/jimaging8120331.
- [16] A. Salmona, L. Bouza, and J. Delon. Deoldify: A review and implementation of an automatic colorization method. *Image Processing On Line*, 12:347–368, 2022. doi:10.5201/ipol.2022.403.
- [17] X.-Q. Shi, P. Sällström, and U. Welander. A color coding method for radiographic images. *Image and Vision Computing*, 20:761–767, 2002. doi:10.1016/S0262-8856(02)00045-8.
- [18] J. Su, H. Chu, and J. Huang. Instance-aware image colorization. In: *Proc. 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 7965–7974, jun 2020. doi:10.1109/CVPR42600.2020.00799.
- [19] D. Varga and T. Sziranyi. Fully automatic image colorization based on convolutional neural network. In: *Proc. 23rd Int. Conf. Pattern Recognition (ICPR)*, pp. 3691–3696, 2016. doi:10.1109/ICPR.2016.7900208.
- [20] A. Visvanathan, S. Reichenbach, and Q. Tao. Gradient-based value mapping for pseudocolor images. *Journal of Electronic Imaging*, 16(3):033004, 2007. doi:10.1117/1.2778426.
- [21] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbyey to zoo. In: *Proc. 2010 IEEE Conference CVPR*, pp. 3485–3492, 2010. doi:10.1109/CVPR.2010.5539970.
- [22] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul. Grayscale image colorization methods: Overview and evaluation. *IEEE Access*, 9:113326–113346, 2021. doi:10.1109/ACCESS.2021.3104515.
- [23] R. Zhang, P. Isola, and A. Efros. Colorful image colorization. In: *Proc. European Cong. Computer Vision – ECCV 2016*, pp. 649–666. Springer, 2016. doi:10.1007/978-3-319-46487-9_40.
- [24] A. Śluzek. Do we always need ai for image colorization? *Proc. of 4th Polish Conference on Artificial Intelligence*, pp. 31–36, 2023. doi:10.34658/9788366741928.3.
- [25] A. Śluzek. On unguided automatic colorization of monochrome images. In: *WSCG 2023 Proceedings*, vol. 3301 of *Computer Science Research Notes*, pp. 379–384, May 2023. doi:10.24132/CSRN.3301.38.
- [26] A. Śluzek, M. Dudziński, and T. Świsłocki. Automatic colorization of digital movies using de-colorization models and SSIM index. In: *Preproc. 18th Conf. Computer Science and Intelligence Systems (FedCSIS 2023)*, pp. 837–847, September 2023. doi:10.15439/2023F3017.



Andrzej Śluzek received his M.Sc., Ph.D., and D.Sc. (*habilitation*) degrees from Warsaw University of Technology. He is currently a full professor in the Institute of Information Technology (Department of Artificial Intelligence) at Warsaw University of Life Sciences – SGGW.

From 1992 to 2011, he worked at Nanyang Technological University (School of Computer Science and Engineering) in Singapore. From 1994 to 2010, he was the deputy director of the Robotic Research Centre at the same university. From 2011 to 2020, he was an associate professor at Khalifa University (Department of Electrical Engineering and Computer Science) in Abu Dhabi, United Arab Emirates. In 2015, he was granted the title of professor by the President of Poland.

ENHANCING MULTICLASS PNEUMONIA CLASSIFICATION WITH MACHINE LEARNING AND TEXTURAL FEATURES

A. Beena Godbin , S. Graceline Jasmine *

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

**Corresponding author: S. Graceline Jasmine (graceline.jasmine@vit.ac.in)*

Abstract The highly infectious and mutating COVID-19, known as the novel coronavirus, poses a substantial threat to both human health and the global economy. Detecting COVID-19 early presents a challenge due to its resemblance to pneumonia. However, distinguishing between the two is critical for saving lives. Chest X-rays, empowered by machine learning classifiers and ensembles, prove effective in identifying multiclass pneumonia in the lungs, leveraging textural characteristics such as GLCM and GLRLM. These textural features are instilled into the classifiers and ensembles within the domain of machine learning. This article explores the multiclass categorization of X-ray images across four categories: COVID-19-impacted, bacterial pneumonia-affected, viral pneumonia-affected, and normal lungs. The classification employs Random Forest, Support Vector Machine, K-Nearest Neighbor, LGBM, and XGBoost. Random Forest and LGBM achieve an impressive accuracy of 92.4% in identifying GLCM features. The network's performance is evaluated based on accuracy, precision, sensitivity and F1-score.

Keywords: COVID-19, chest X-ray, feature extraction, GLCM, GLRLM, Machine Learning, Random Forest, XGB, SVM.

1. Introduction

Pneumonia, an acute respiratory infection caused by bacteria or viruses, can result from a variety of factors and is a leading cause of mortality, particularly affecting vulnerable populations such as the elderly and children. It can present as a mild ailment in individuals of all ages. In March 2020, the World Health Organization (WHO) declared the emergence of the new Coronavirus 2019, widely known as COVID-19, as a global pandemic [32]. This virus, identified as SARS-CoV-2, is linked to Severe Acute Respiratory Syndrome and is frequently associated with respiratory symptoms. Originating in China, the virus rapidly spread to other nations, leading to a global pandemic that has significantly impacted individuals worldwide, resulting in numerous fatalities. Factors such as the recent discovery of the virus, delayed detection, limited testing capabilities, insufficient medical expertise, and its resemblance to pneumonia-related illnesses have collectively impeded the medical community's ability to effectively combat the virus until relatively recently. Pneumonia, characterized by the accumulation of air or pus in the alveoli of the lungs, can be caused by bacterial, fungal, or viral infections, all of which have the potential to trigger severe allergic reactions [36]. This particular instance manifests symptoms such as coughing, respiratory distress, fatigue, fever, and profuse sweating, which are also commonly observed in patients with COVID-19. The chest radiographs of individuals with COVID-19 exhibit patterns reminiscent of those found

in pneumonia cases, indicating the presence of the virus. The radiological findings of COVID-19 on chest X-rays closely resemble those of pneumonia, as reported by imaging departments. Numerous studies have utilized X-ray imaging for the diagnosis and classification of pneumonia [23]. While comprehensive screening of Coronavirus samples through reverse transcription polymerase chain reaction (RT-PCR) may be insufficient in effectively curbing the global spread of the virus, chest X-rays have proven highly useful in triaging patients infected with COVID-19. The rapid and widespread proliferation of the virus has placed a substantial burden on health and medical organizations worldwide. Consequently, the development of technology capable of distinguishing between patients with pneumonia or normal chest X-rays and those with COVID-19 has become imperative. This urgency arises from the inability of chest X-rays, despite thorough analysis, to differentiate between pneumonia patients and individuals suspected of having COVID-19. Researchers worldwide propose the employment of texture-based cognitive strategies to identify COVID-19. Current research is predominantly focused on extracting texture characteristics from chest X-ray images using methods such as GLCM and GLRLM. The advocated strategy for early detection of potential COVID-19 cases involves categorizing X-ray images into four predefined classes: COVID-19, bacterial pneumonia-affected, viral pneumonia-infected, and normal. Given the severity of the current situation, addressing this matter with utmost urgency is paramount. In this paper, the following contributions are made:

- This paper introduces a method enabling accurate differentiation between COVID-19 infection and pneumonia in chest X-ray images, addressing a crucial diagnostic challenge.
- This research pioneers accuracy in disease identification by combining textural features, specifically GLCM and GLRLM. The amalgamation enhances diagnostic precision, marking a notable advancement in medical image analysis.
- This study contributes by systematically evaluating various machine learning classifiers to enhance the accuracy of COVID-19 detection. The paper evaluates system performance using accuracy, sensitivity, and F1 score, offering a comprehensive snapshot of its diagnostic effectiveness.
- To discern between COVID-19, bacterial pneumonia, viral pneumonia, and healthy individuals, we adopted a tailored approach, training each model independently. This meticulous strategy ensures accurate identification across diverse medical conditions.

Section 2 provides a comprehensive review of the literature. Subsequently, in Section 3 the technical matters related to the methodology are presented. Namely, Section 3.1 expounds on the dataset specifics, Section 3.2 presents the feature engineering approaches, Section 3.3 discusses the machine learning methodologies, and Section 3.4 presents the cross validation techniques. Section 4 presents the outcomes of our empirical investigations. Finally, Section 5 comprises a discourse on the findings and a conclusive statement.

2. Literature Survey

Ardakani et al. [2] suggested a computer-aided diagnostic (CAD) technique for distinguishing COVID-19 pneumonia patients from non-COVID-19 pneumonia patients. To this end, the authors employed a dataset of 612 CT images of pneumonia patients, where 306 patients were diagnosed with COVID-19 and the remaining 306 were diagnosed with non-COVID-19, of which 376 patients were COVID-19 positive. The researchers extracted 20 imaging features from the dataset and subjected them to classification using five different classifiers, namely, Decision Trees (DT), Naive Bayes (NB), K-nearest neighbors (KNN), Support Vector Machines (SVM), and Ensembles. The authors gained the highest level of accuracy, i.e., 91.94%, through ensemble classification. In another study, al-Karawi et al. [5] developed an automated model for COVID-19 analysis in CT scans utilizing CT scan images. In the dataset utilized, a total of 275 COVID-19 cases were identified as positive, while 195 were negative. The CT images were subjected to a Fast Fourier Transform, followed by the application of a Gabor filter for image manipulation. By employing the SVM technique for classification, a commendable accuracy of 95.37% was obtained. Barstugan et al. [8] incorporated 150 CT scans in their study and extracted four different patches from these scans (16×16 , 32×32 , 48×48 , and 64×64) for comparative purposes. SVM was utilized to classify radiomic features obtained using FOS, GLCM, GLRLM, and GLSZM patches. The study further integrated a 10-fold plus DWT (Discrete Wavelet Transform) feature, and the highest accuracy recorded was 99.64%. The examination carried out by Dey and colleagues [14] scrutinized 400 CT scans of persons afflicted with COVID and 400 of non-afflicted with COVID-19, encompassing 200 CT scans for each cohort. A devised strategy enabled the creation of a system capable of segmenting the COVID-19 infected regions into smaller subsections, subsequently extracting data from each area discretely. Machine learning algorithms offer four distinct techniques for classifying entities into groups: Random Forest, Support Vector Machine, K-Nearest Neighbors and Decision Tree. The utilization of the K nearest neighbor algorithm in the conducted investigation resulted in an 88% accuracy rating.

In their scrutiny, Liu and colleagues [22] meticulously scrutinized 61 CT scan images of COVID-19 and 27 CT scan images of pneumonia in general, from which they meticulously extracted 35 statistical textural features. An array of models were meticulously evaluated, including but not limited to Support Vector Machine, Linear Regression, k-Nearest Neighbors and Decision Tree. The authors compared the Ensemble of bagged tree with the aforementioned models. The Ensemble model bagged tree classifiers attained the utmost level of accuracy, which yielded a rate of 94.16%. Ozkaya and colleagues employed an identical dataset and partitioned it into two subsets, namely Subset-1 with dimensions of 16×16 and Subset-2 with dimensions of 32×32 . To identify distinctive features, they employed a design of convolutional neural network architecture

in conjunction with a support vector machine algorithm. The accuracy rate for Subset-2 was 99.28%. With assistance from previously trained deep neural networks, Kassani and colleagues [20] have devised a methodology for extracting features. 117 images of X-ray and 20 images of CT with abnormalities were compared with 117 X-rays and 20 CT scans without abnormalities.

For sorting things into groups, we used XGBoost, Random Forest, Decision Tree, LightGBM, AdaBoost and Bagging algorithm. Bagging tree classifiers are 99% accurate when features are pulled with DenseNet121 and grouped.

CT scans were used in Shi et al.'s study; 1659 COVID-19 and 1028 bacterial pneumonia were classified as negatives. Radiomic and hand-made elements were taken from the infected areas. Linear Regression, Support Vector Machine, Random Forest and Neural Network were compared to a classification method based on Random Forest and LightGBM. In terms of the handmade features, the proposed method gave the best results, with an accuracy level of 89.4% [30].

According to Zheng et al. [39], CT scans could be detected with a 3D deep convolutional neural network (DeCoVNet). 313 participants had COVID-19 while 229 did not. They chose 540 of those to participate in their study. The model has undergone training using a straightforward 2D UNet in an integrated manner. The criteria can be changed to look for COVID-19. There is a best accuracy of 90.8%. The authors developed a three-class system using 618 CT scans with 219 COVID-19, 177 normal persons, and 226 influenza A pneumonia cases. A 3D CNN (Convolutional Neural Network) method was used to divide up the image using a transfer learning model. ResNet-18 classification and location-attention are used to model transfer learning [37]. The precision of their diagnostic outcomes pertaining to COVID-19, IAVP, and uninfected individuals was registered at an impressive 87.7%. Song et al.'s study involved the classification of 88 COVID-19 patients, 100 bacterial pneumonia patients, and 87 healthy individuals through the utilization of deep learning. Specifically the Details Relation neural model was employed, having already undergone training [31]. The researchers achieved a commendable multiclass classification accuracy rate of 94%. Meanwhile, Wang et al. [35] employed a deep learning method, which made use of transfer learning, along with a pre-trained GoogleNet inception model to detect COVID-19 cases. The accuracy of their classifications for 325 COVID-19 positive patients and 740 COVID-19 negative patients was 89.5% for each. A CNN classifier trained on GoogleNet acquired an accuracy rate of 82.14% in an experiment by Alsharman et al. [6]. The study conducted by the researchers included a total of 463 non-COVID-19 images and 349 COVID-19 CT images in their analysis [16].

A team of researchers developed seven Deep CNN models to classify images of pneumonia automatically. A sampling of models included in the table are CNN baselines, VGG16, Xception, DenseNet201, VGG19, InceptionResNetV2, Resnet50, InceptionV3 and MobileNetV2. It was similar to the work by Zhang et al. [38]. Based on deep

learning, they created a diagnosis system for COVID-19 based on 3D ResNet18 and four segmentation models. As per the findings of Rajaraman et al. [29], it was determined that a total of five Convolutional Neural Network (CNN) models were employed in the screening process for detecting the outbreak of the COVID-19 virus. The models used were InceptionV3, VGG16, Xception, NasNetmobile and DenseNet201. Each X-ray picture belongs to one of six groups. In 99.26% of cases, the result was accurate. According to Tsiknakis et al. [34], modified deep CNN models could be used for the screening of COVID on chest X-ray images using the InceptionV3 model. Transfer learning models that have already been trained can find COVID-19 cases automatically, as demonstrated by Ahuja et al. [4]. Oh et al. [25] have conducted analogous research, wherein they have proposed the utilization of a CNN based on patches using ResNet18. To diagnose COVID-19 disease, Elasmaoui and Chawki [17] used 7 deep learning models that had already been trained. Chowdhury et al. [13] developed eight deep CNNs to detect COVID-19. The suggested models were tested using 3487 X-ray images with and without image augmentation. In the present set of images, it can be observed that 423 of them depict the COVID-19 virus, while 1485 portray the typhus bacterium, and the remaining 1579 images display normal conditions.

COVID-19 detection and classification could be improved by using a convolutional neural network model, as suggested by Apostolopoulos et al. [7]. An evaluation of MobileNetv2 was conducted using 3905 x-ray pictures, with the results showing its excellent performance in detecting COVID-19. Rahimzadeh and Attar [28] introduced a modified deep Convolutional Neural Network (CNN) leveraging Xception model and ReNet to identify COVID-19 from chest X-ray images. Their experimental findings reveal that the combined method demonstrates an impressive average accuracy of 91.4%, precision of 72.8%, sensitivity of 87.3%, and specificity of 94.2%. Notably, the model achieved outstanding performance with a remarkable 99.18% accuracy, 97.36% sensitivity, and 99.42% specificity. This highlights the effectiveness of their innovative model in accurately detecting COVID-19 cases. According to another study [1], the Decompose Transfer Compose (DeTraC) Convolutional Neural Network model was modified and revised. An accuracy rate of 97.35%, a sensitivity rate of 98.23%, and a specificity rate of 96.34% were successfully attained [3]. Afshar and colleagues constructed a convolutional neural network architecture founded on capsule networks to identify and diagnose COVID-19.

Our inquiry delved into the effectiveness of COVID CAPS via the utilization of two publicly accessible thoracic X-ray databases. Therefore, we achieved 95.8% specificity, 90% sensitivity, and 95.70% accuracy. The deep CNN technique VGG-16 was employed by Brunese et al. [11] to automatically and quickly identify COVID-19 in chest X-ray images. VGG-16 appears to be 97% accurate at diagnosing COVID-19 based on the results of our study. By using deep learning-based AI, Jin et al. [19] have suggested a method for detecting COVID-19 in CT scan images. Due to the discoveries, the suggested

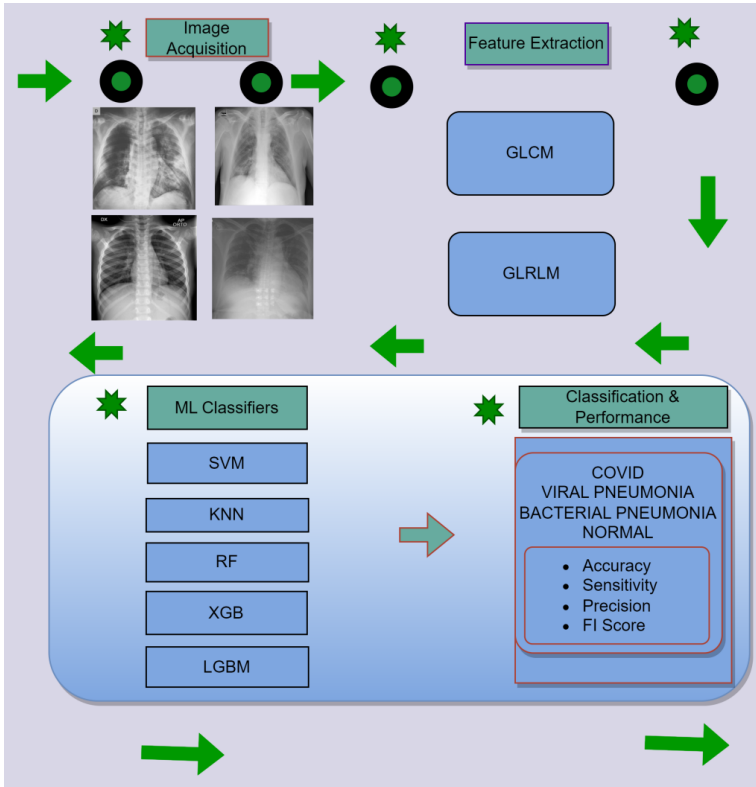


Fig. 1. Workflow of the proposed system.

framework exhibits a 95.8% area under the curve (AUC), a 90.19% level of sensitivity, and a 95.76% level of specificity.

3. Materials and methods

As illustrated in Figure 1, the following section provides an overview of the technologies applied. The first step was to extract numerous features (GLCM, GLRLM) from the CXT pictures using a variety of feature extraction techniques. A technique is proposed to study how different characteristics influence COVID-19 illness categorization. Classification accuracy varies depending on the feature set. An individual or a vector group of the retrieved characteristics is employed to assess their influence on the classification outcomes. Feature extracts and feature vectors were prepared and then the dataset was

Tab. 1. Dataset description.

| Class | Images |
|---------------------|--------|
| Bacterial pneumonia | 2000 |
| Viral pneumonia | 1345 |
| COVID-19 | 2250 |
| Normal | 2370 |
| Total | 7965 |

divided into training set and test set. In order to train the machine learning models to categorize characteristics based on the most popular classifiers, we provided them with training data. The performance of the model was assessed by using 10-fold CV (Cross Validation) as a method of evaluation. Test sets with varying characteristics were evaluated ten times. As each step was completed, the classifier output served as the basis for determining the performance results.

3.1. Data set

The datasets used in this work are X-ray chest [24] and Covid19 radiography [33]. In the experiment, chest X-ray pictures were categorized into four classes: Covid19, Pneumonia bacterial, Pneumonia viral, and Normal. The collection contains 7965 chest X-ray pictures divided into four categories: Covid19 (2250 pictures), Normal (2370 pictures), Pneumonia viral class (1345 pictures), and Pneumonia bacterial class (2000 pictures) shown in Table 1. Sample images from the data set are shown in Fig. 2.

3.2. Feature Extraction Techniques

The study of radiological images numerical features is currently undergoing rapid expansion with the use of artificial intelligence techniques. The initial stage of this work involved analyzing the data for features. With the utilization of texture-based features, the process of identifying tissue sections with varying characteristics becomes simplified, as one can easily identify the connections and distinctive attributes between pixels. Further, statistical features can be derived from the matrices generated through the texture-based techniques after the texture feature has been extracted [15]. In order to extract texture-based attributes, matrices such as the Grey Level Run-Length Matrix (GLRLM) and the Grey Level Co-occurrence Matrix (GLCM) are employed.

3.2.1. GLCM

Image processing and analysis use the Gray-Level Co-occurrence Matrix (GLCM) approach to derive texture from pictures. It is possible to describe the spatial associations

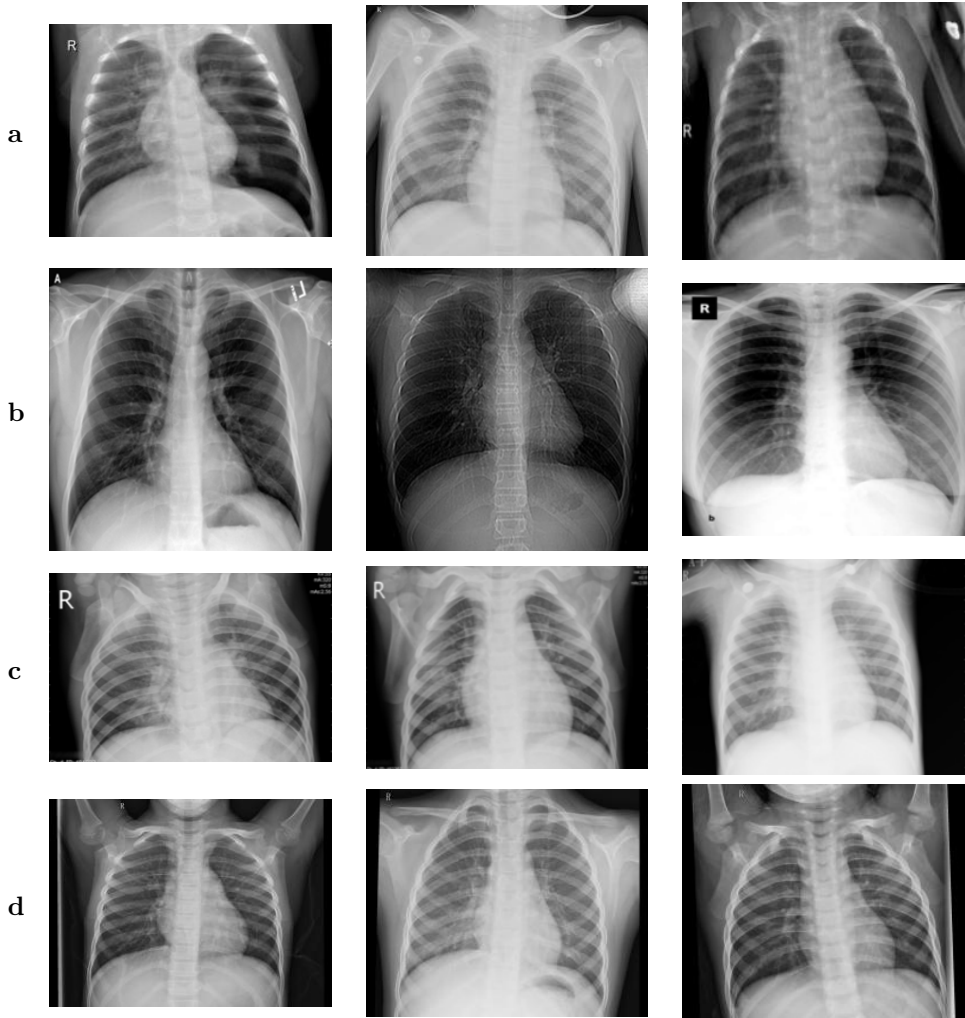


Fig. 2. Samples from the CXT dataset. (a) Images depicting bacterial pneumonia affected patients; (b) images depicting patients with COVID-19; (c) images of viral pneumonia patients; (d) images of normal patients.

between pixels with comparable gray-level values in a picture statistically. In the paper authored by Timo Ojala et al. [26], a comparative analysis is conducted on GLCM texture measures and classification methods utilizing featured distributions. GLCM is based on the idea that the distribution of pixel intensities, and their relation, can provide crucial information about a picture's texture. This is done by determining what pairs of gray-level values will appear at various spatial displacements within the picture.

The following stages are involved in producing a GLCM:

Greyscale: transformation in order to simplify the pixel intensity analysis.

Pixel pairing: finding occurrences of gray level pairs. GLCM matrix entries represent gray-level pairs' frequency of occurrence.

Angular Second Moment (ASM): finding the distribution of pixel pairs over the entire image.

Contrast: finding the differences or variations between pixel intensities.

Dissimilarity: finding the average difference between adjacent gray levels.

Energy: calculating the sum of squared elements in the GLCM.

Homogeneity: finding the proximity of the distribution of elements to the GLCM diagonal.

Maximum Probability: finding the most often occurring gray-level pair in the GLCM.

Sum of Squares: finding the GLCM's variance.

Correlation: measuring the linear dependence between gray-level values of neighboring pixels.

3.2.2. GLRLM

In their paper, Haralick et al. [18] introduced the GLRLM model, an acronym for Gray-Level Run Length Matrix. In the field of image processing feature analysis, the GLRLM is an approach which quantifies the distribution of consecutive gray-level pixels with the same intensity of an image, or the distribution of the gray-level runs. It provides statistical information on the lengths and frequency of these runs, and having this information can help characterize and distinguish textures in a picture much better. GLRLM is a process that is based on the idea that the organization and distribution of runs of comparable gray-level values within a picture may provide key textural information about this picture (see Fig.3). As a result of analyzing these runs, we can extract features that characterize the texture of the picture and identify aspects of its textural quality.

These are some of the most commonly computed GLRLM features:

Short Run Emphasis (SRE): represents how many short runs are there in the picture.

Long Run Emphasis (LRE): shows how many long runs are there in the picture.

Gray-Level Non-Uniformity (GLN): represents comparability, or consistency of the gray level values among runs.

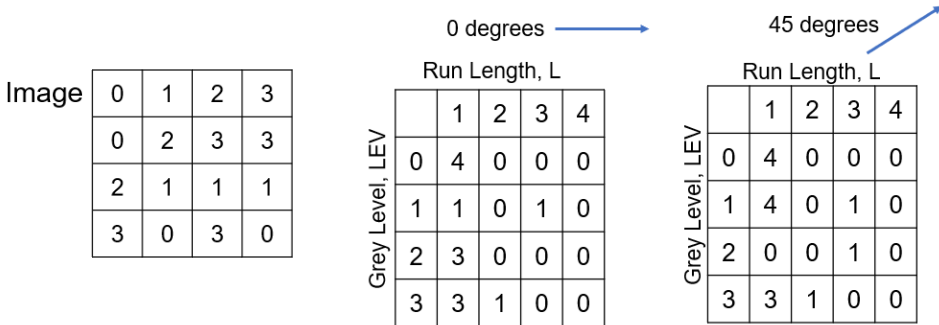


Fig. 3. Calculation of GLRLM (according to [27], license: CC BY 3.0).

Run Length Non-Uniformity (RLN): measures the variation or consistency in run lengths.

Run Percentage (RP): shows the share of the picture area where runs are present.

Run Entropy (RE): represents uncertainty associated with the run length.

3.3. Machine Learning Algorithms

In conjunction with the process of feature extraction, the retrieved features underwent training by machine learning models, and were subsequently evaluated on the test dataset. In light of their unyielding strength, we opted to employ the most formidable and extensively utilized machine learning methodologies. Noteworthy among these techniques for classification were the Support Vector Machine, K-Nearest Neighbor, XG-Boost, LGBM and Random Forest.

3.3.1. Support Vector Machine

The SVM is a machine learning technique used for classification and regression. The method is particularly useful when data cannot be linearly separated or when the decision boundary is complex. Corinna et al. [9] presented a SVM training algorithm for optimal margin classifiers in their paper. Data points are divided into multiple groups using SVM's ideal hyperplane. In order to maximize the margin, the hyperplane is chosen to be as near as possible to the nearest data points from each class. Support vectors consists of the data points nearest to the given class.

Here is the outline of the SVM procedure:

Data representation: Selecting the optimal hyperplane: The SVM algorithm seeks the hyperplane with the highest margin. While minimizing classification error, the hyperplane should divide data points of various classes. This is accomplished by resolving an optimization problem.

Handling non-linear data: In cases where the data aren't linearly separable, SVM uses the "kernel trick". It transcends the data points to a realm of heightened dimensionality. A linear kernel, a polynomial kernel, and an RBF kernel are three common kernel functions.

Training: Through optimization of a cost function, the SVM algorithm learns the hyperplane's parameters and penalizes misclassifications. In order to accomplish this, a convex optimization problem needs to be solved.

Classification: The SVM possesses the remarkable ability to apprehend novel, indiscernible data points through its astute discernment of the side on which they lie on the established hyperplane. A newly-introduced data point is affiliated with a specific class contingent upon its placement on either side of the hyperplane; one side denotes one class and the other side denotes the other one.

3.3.2. K-Nearest Neighbor

In the field of machine learning, the K-nearest neighbor algorithm assumes a prominent role as a supervised decision tree algorithm that possesses the unique ability to tackle classification problems as well as regression problems. Richard et al. presented KNN algorithm in their paper, as reported by Duda et al. in their monograph [15]. The KNN method uses a distance calculation between the test data and all of the training points in order to try and predict which class the test data belongs to based on the data points in the test set. As a result, the number of K points closest to the test data is the number of points that should be chosen. As a result of implementing the KNN algorithm, we shall endeavor to compute the likelihood that the test data is affiliated with the classifications of K training data points. Subsequently, we shall elect the classification that possesses the most elevated probability of belonging to the test data set. As regards the scrutiny of regression analysis, the ascertained value is the mean of the data points from K arbitrarily picked training points. The aforementioned data points are then utilized for the evaluation of the regression.

3.3.3. Random Forest

The Random Forest, a captivating machine learning algorithm, has gained widespread popularity for its exceptional ability to perform classification and regression tasks in machine learning is shown in Figure 4. Breiman proposed the random forest algorithm in his publication, documented as [10]. By combining the predictions from many individual trees, a decision tree system can produce an outcome that is as accurate and reliable as possible.

Random Forest works as follows (see also the Algorithm 1):

As part of the training phase, Random Forest constructs decision trees. The decision trees are constructed using a subset of the original training data as well as the available features. During the process, randomization and bagging take place.

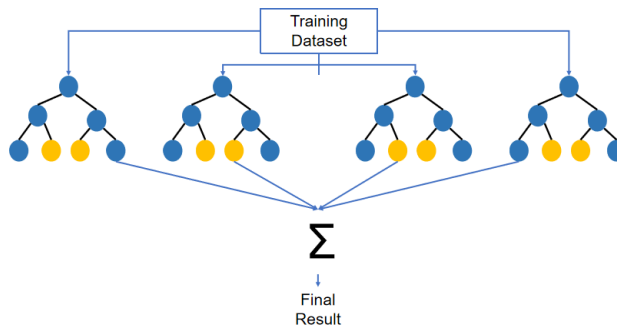


Fig. 4. Random Forest concept scheme.

Rather than considering all features at each node, the Random Forest evaluates the best split based on a random selection of features. In this way, randomness increases the diversity of the ensemble and reduces the correlation between trees.

Training trees: The training data for each decision tree is derived from a different bootstrap sample. As a result, each tree in the ensemble sees a slightly different subset of the original data.

As part of the prediction phase, Random Forest makes predictions based on input features. During classification tasks, the class receiving the most votes is chosen. For regression tasks, it is customary to derive the ultimate prediction through the computation of the mean or median of the individualized predictions.

3.3.4. XGBoost

A gradient boosted trees algorithm is implemented in an efficient and open-source manner using eXtreme Gradient Boosting (XGBoost). In their publication referenced as [12], Tianqi Chen and co-authors put forward a model that employs the XGBoost algorithm. Gradient boosting is a algorithm for supervised learning that combines simple and weaker models to produce accurate predictions of a target variable. With its capacity to manage an extensive range of data categories, connections, spreads, and hyperparameters that are adaptable, the XGBoost algorithm excels in machine learning competitions. You can use it for regression and classification.

3.3.5. LGBM

Guolin Ke et al. [21] proposed the Light Gradient Boosting Model (LGBM). By using lightGBM, gradient boosting models can be trained efficiently and at high performance. LightGBM improves training speed and model accuracy by using a gradient boosting algorithm. Gradient boosting allows LightGBM to combine multiple weak prediction

Algorithm 1 Random Forest

Data:

Training set B with m instances, p features, and target value.

Total number of classes K ; total number of classifiers C in RF.

Procedure:

For $c = 1$ **to** C

Generate a bootstrapped sample B_c from the training set B .

Construct a tree using a random feature subset from bootstrapped sample B_c .

For each node t in the tree

Select randomly $n \approx \sqrt{p}$ or $n \approx \frac{p}{3}$ features.
and cutpoints using the random feature subset.

Identify the best split features
Send down the data using the
best split features and cutpoints.

Develop trained classifier D_c :

Group the C trained classifier models using majority vote:

Predicted label D_c : $D_c(x) = \arg \max_j \sum_{B_c} I(D_c(x) = j)$, for $j = 1, \dots, K$.

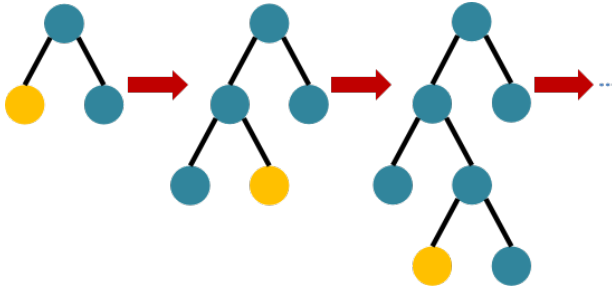


Fig. 5. LGBM-leafwise concept scheme.

models into one strong model (typically decision trees). A sequential correction is made for each successive tree based on the error of the previous tree as symbolically depicted in Figure 5 and outlined in Algorithm 2.

3.4. Cross validation

An important part of machine learning algorithms is the use of cross validation as one of the statistical methods used to evaluate the methods. A significant component of this procedure is the division of the data into two types: training set and validation set. The preeminent approach is the k -fold technique, which is unequivocally the most widely employed methodology. This method involves dividing the dataset into k equally divided parts (folds) for the purpose of training and validating the model of a dataset. During

Algorithm 2 LightGBM Classifier

Input:

x_{train} : training data features
 y_{train} : set of training data labels
 x_{test} : set of test data features
 y_{test} : set of test data labels

Output:

Trained LightGBM classifier model

Initialization:

learning_rate = 0.02
max_depth = 8
random_state = 422

Fit the Model:

evaluation_set = [(x_{test}, y_{test}), (x_{train}, y_{train})]
verbose = 20
evaluation_metric = 'logloss'

Output:

Trained LightGBM classifier model

the validation and training process, the validation of the model is executed through the utilization of distinct folds for each iterations in the training and validation process. Once all the folds have been averaged, the overall performance of the product can be obtained. A k -fold cross validation is shown in Figure 6.

4. Results

This part presents the outcomes of the classification based on CXT images of multiclass pneumonia based on the results of the investigation. The acquisition of all training and test outcomes was achieved through the utilization of a computer equipped with Windows 10 as the operating system and a memory capacity of 8 GB. As part of the analysis, Python 3.7.10 was used in conjunction with Scikit-learn 0.23.19. The types of algorithms used for classification include SVM, RF, KNN, XGBoost, and LGBM. It is possible to tune the hyperparameters of these classifiers in order to control the process of learning. For each method of classification, a number of parameters are controlled in order to achieve the best results. In addition, the hyperparameters for each classification methods were determined through a rigorous grid search coupled with a ten-fold cross-validation over the training set. The attainment of SVM results necessitated the consideration of several pertinent parameters. A conclusion was reached that the XGBoost and RF classifiers can produce similar outcomes for certain parameter variables, contingent upon

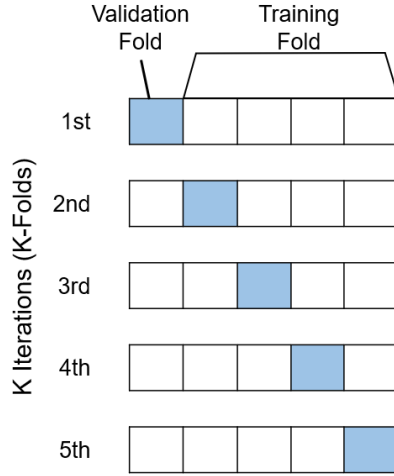


Fig. 6. The concept of k -fold cross validation.

the parameters on which they are trained. The parameters selected for these models are therefore from the range of parameters specified by the authors. A 10-fold Cross Validation technique was used in our study to obtain more reliable and realistic results. The performance of the model was evaluated using accuracy, sensitivity, precision and F1 score. As a proportion, accuracy can be delineated as the quotient of the number of accurate predictions and the aggregate number of forecasts. As a heuristic, precision can be characterized as the ratio of correct number of positive class predictions to the total number of positive class predictions. The assessment of prognostications is appraised through the ratio of accurate affirmative projections and erroneous negatory prognoses. Precision and sensitivity are averaged to calculate F1. Here are the formulas for each metric in terms of TP, TN, FP, and FN for each class i in a multi-class confusion matrix shown in equations (1–8).

$$\text{accuracy} = \text{ACC}_i = \frac{\text{TP}_i + \text{TN}_i}{\text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i}, \quad (1)$$

$$\text{precision} = \text{PRE}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i}, \quad (2)$$

$$\text{sensitivity} = \text{SNS}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, \quad (3)$$

$$\text{F1-score} = \text{F1}_i = 2 * \left(\frac{\text{PRE}_i \times \text{SNS}_i}{\text{PRE}_i + \text{SNS}_i} \right), \quad (4)$$

where the measures for class i are:

TP_i : number of True Positives for class i ,

TN_i : number of True Negatives for class i (sum of all non-class i values in the confusion matrix),

FP_i : number of False Positives for class i (sum of all values in the row for class i , excluding the diagonal element),

FN_i : number of False Negatives for class i (sum of all values in the column for class i , excluding the diagonal element).

The macro-measures for all the classes $i = 1, \dots, N$ are simply the average values:

$$\text{macro-accuracy} = \text{MACC} = \frac{\sum_{i=1}^N \text{ACC}_i}{N} \quad (5)$$

$$\text{macro-precision} = \text{MPRE} = \frac{\sum_{i=1}^N \text{PRE}_i}{N} \quad (6)$$

$$\text{macro-sensitivity} = \text{MSNS} = \frac{\sum_{i=1}^N \text{SNS}_i}{N} \quad (7)$$

$$\text{macro-F1 score} = \text{MF1} = \frac{\sum_{i=1}^N \text{F1}_i}{N} \quad (8)$$

The quantification of accurately predicted positive class instances is denoted as TP. Additionally, TN is also a constituent of the aforementioned metric, or true negative, is the number of examples of a class that are predicted correctly as negatives. False positives can be defined as negative examples that are misinterpreted as positive examples. As a result of false negative, there are examples of positive classes predicted to be negative. Several feature sets were tested to see how well the proposed method performed for different feature sets. The combination of features was tested in many different ways here as well.

Our approach to extracting GLCM or GLRLM features was based on a Python package. The computation is performed to determine the numerical worth of every characteristic attribute with regards to every individual angular measurement, then returns the mean value for each angle degree [1, 28]. In order to calculate our features, we calculate them for the angles of 0, 45, 90 and 135 degrees. Results of using various features are shown in Tables 2 and 3. We incorporate an element of experimentation by varying the distance values used in the analysis. This approach yielded insignificant alterations in the outcome of our trials. The results of Table 2, which rely on GLCM characteristics, reveal that the LGBM classifier demonstrated the highest accuracy at 92.43%, whereas the random forest classifier produced the best F1 score of 96.02%. On the basis of GLRLM features, Table 3 shows that for LGBM classification, the best accuracy

Tab. 2. Results for GLCM features.

| GLCM | | | | |
|------------|--------------|---------------|-----------------|--------------|
| Classifier | Accuracy (%) | Precision (%) | Sensitivity (%) | F1-Score (%) |
| SVM | 91.13 | 92.21 | 93.23 | 93.14 |
| RF | 92.04 | 94.03 | 99.02 | 96.02 |
| KNN | 90.01 | 93.12 | 91.43 | 92.41 |
| XGB | 91.14 | 95.54 | 95.27 | 95.07 |
| LGBM | 92.43 | 96.02 | 95.25 | 96.62 |

Tab. 3. Results for GLRLM features.

| Classifier | Accuracy [%] | Precision [%] | Sensitivity [%] | F1-Score [%] |
|------------|--------------|---------------|-----------------|--------------|
| SVM | 80.21 | 85.21 | 82.18 | 88.00 |
| RF | 89.12 | 100.00 | 99.11 | 100.00 |
| KNN | 82.19 | 99.04 | 98.90 | 99.00 |
| XGB | 86.80 | 98.08 | 97.35 | 100.00 |
| LGBM | 91.20 | 100.00 | 100.00 | 99.21 |

is 91.2%, whereas for RF classifiers and XGBs, the best F1 score is 100%. GLRLM-based machine learning models produce a higher degree of accuracy when compared to GLCM-based models.

Figure 7a shows an example of a confusion matrix for an SVM classifier. A TP of 215 is calculated, a TN of 35, and a FP of 47 is calculated. Figure 7b shows the confusion matrix for a simple LGBM classifier test. In the LGBM classifier for bacterial pneumonia, there are 376 TPs, 0 TNs, and 12 FPs.

The XGB classifier for COVID shown in Figure 8a has 384 TP, 47 TN, and 61 FP. The confusion matrix for Random Forest classifiers is illustrated in Figure 8b. For LGBM classifier for bacterial pneumonia, TP is 447, TN is 60, and FP is 49.

The F1 score was computed using precision and sensitivity as the basis. Figure 9 illustrates the percentage overall performance metrics achieved by all GLCM stride combinations considered.

In Figure 10, the percentage values of performance measures obtained for every GLRLM stride within a multiclass classification methods are presented using the Covid19 identification method. This accuracy value reaches 91.44 percent at its maximum. In a test X-ray images dataset for GLCM stride combinations, the proposed method of multiclass pneumonia identification with machine learning classifiers showed impressive results. Model performance is commonly visualized using ROC curves.

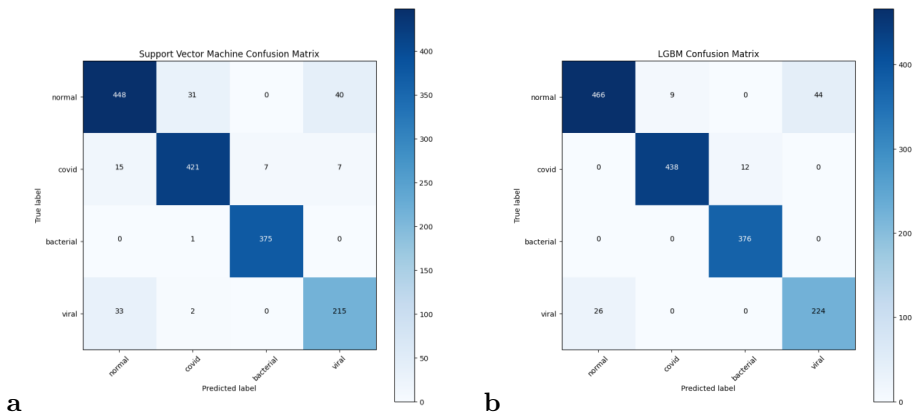


Fig. 7. Confusion matrix for GLCM features. (a) SVM; (b) LGBM.

5. Discussion and conclusion

In the initial step, the present study aims to classify multiclass pneumonia diseases using images from a CXT scanner. Furthermore, the purpose of this study is to examine whether a variety of feature extraction methods can improve classification accuracy. Medical images are characterized by gray levels of intensity, as opposed to sophisticated algorithms and features used in the previous methods. Using intensity-based features, one can analyze important properties of images. COVID-19 requires rapid detection of the diagnosis in order to be successful. CXT used datasets collected from several papers and collected in different ways to develop the proposed method. GLCM classifier has the best results for LGBM and RF with 92.4% and 91.5% accuracy respectively, while XGBoost has 91.5% accuracy for individual feature vectors. Compare GLRLMs with GLCMs, and GLRLMs give lower results. In GLCM and GLRLM, each feature is calculated separately for angles 0, 45, 90, and 135 degrees. Angle-based trials don't matter here. In this paper, we show that gray levels with high gray levels have more value than gray levels with low gray levels. With RF or LGBM classifiers, we get over 92% on GLCM features. GLRLM improved LGBM, XGBoost, and RF by 91.2%, 89.1%, and 86.8%, respectively. Ten-fold cross-validation provided reliable and robust results. According to this study, SVM and KNN classifiers perform less well than RF, XGB and LGBM classifiers. In most cases, RF classifiers will provide the best results. In addition, LGBM and XGB provide significant results as well. Our recommendation is that RF and LGBM classifiers for the purpose of classify COVID-19 classification. In the future, we intend to test our model on a variety of datasets and to improve our model performance to provide an improved diagnosis of COVID-19 from CT and CXT images.

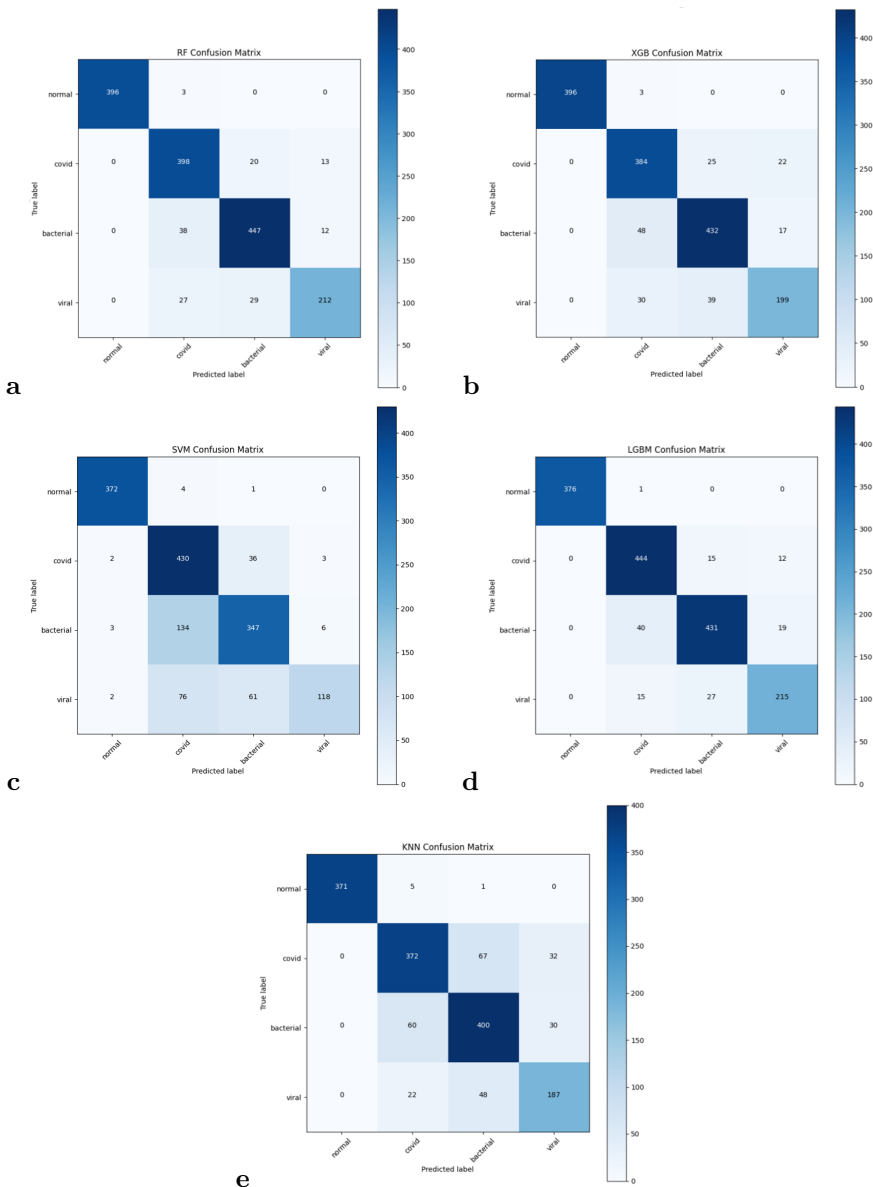


Fig. 8. Confusion matrix for GLRLM features. (a) RF; (b) XGB; (c) SVM; (d) LGBM; (e) KNN.

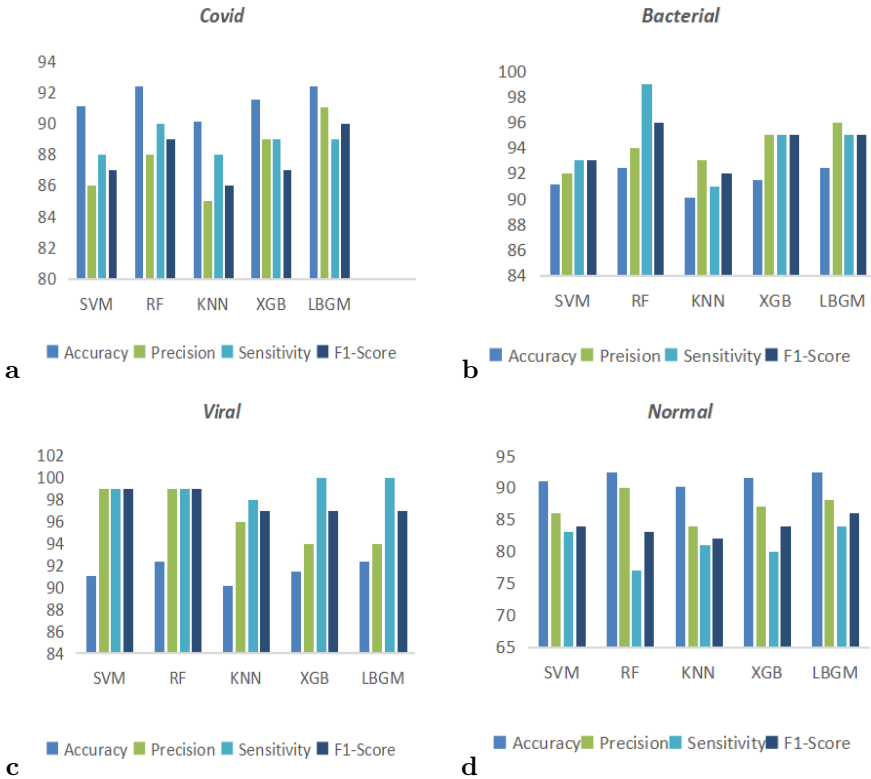


Fig. 9. Performance of classifiers for GLCM features. (a) COVID; (b) bacterial; (c) viral; (d) normal.

References

- [1] A. Abbas, M. M. Abdelsamea, and M. M. Gaber. Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network. *Applied Intelligence*, 51:854–864, 2021. doi:10.1007/s10489-020-01829-7.
- [2] A. Abbasian Ardakani, U. R. Acharya, S. Habibollahi, and A. Mohammadi. COVIDiag: a clinical CAD system to diagnose COVID-19 pneumonia based on CT findings. *European Radiology*, 31:121–130, 2021. doi:10.1007/s00330-020-07087-y.
- [3] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, et al. COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images. *Pattern Recognition Letters*, 138:638–643, 2020. doi:10.1016/j.patrec.2020.09.010.
- [4] S. Ahuja, B. K. Panigrahi, N. Dey, V. Rajinikanth, and T. K. Gandhi. Deep transfer learning-based automated detection of COVID-19 from lung CT scan slices. *Applied Intelligence*, 51:571–585, 2021. doi:10.1007/s10489-020-01826-w.

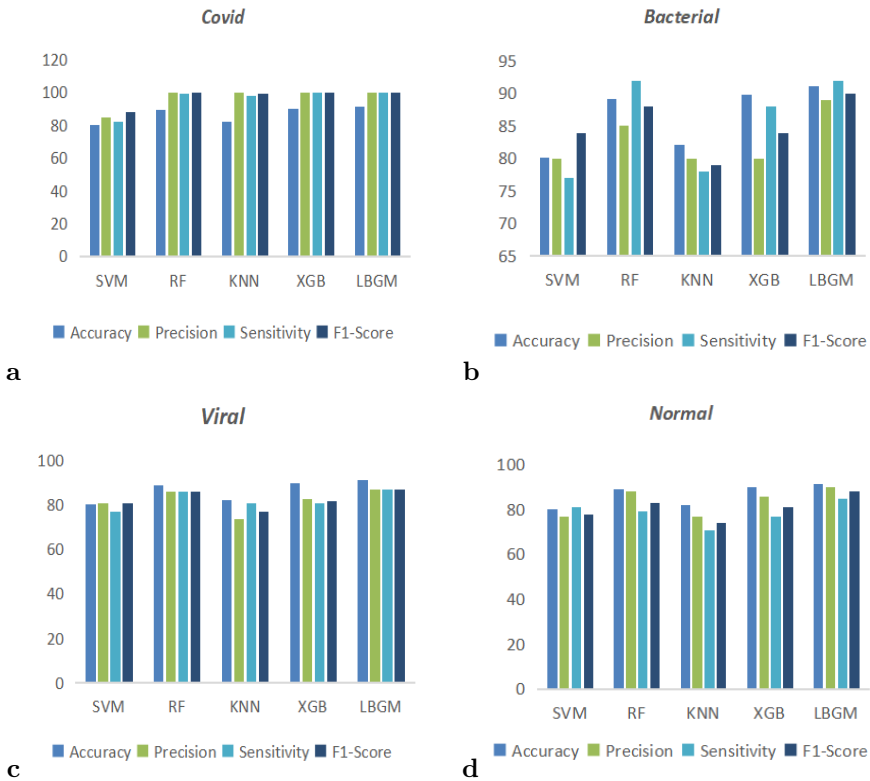


Fig. 10. Performance of classifiers for GLRLM features. (a) COVID; (b) bacterial; (c) viral; (d) normal.

- [5] D. Al-Karawi, S. Al-Zaidi, N. Polus, and S. Jassim. Machine learning analysis of chest CT scan images as a complementary digital test of coronavirus (COVID-19) patients. *MedRxiv*, 2020. MedRxiv.2020.04.13.20063479. doi:10.1101/2020.04.13.20063479.
- [6] N. Alsharman and I. Jawarneh. GoogleNet CNN neural network towards chest CT-coronavirus medical image classification. *Journal of Computer Science*, 16(5):620–625, 2020. doi:10.3844/jcssp.2020.620.625.
- [7] I. D. Apostolopoulos, S. I. Aznaouridis, and M. A. Tzani. Extracting possibly representative COVID-19 biomarkers from X-ray images with deep learning approach and image data related to pulmonary diseases. *Journal of Medical and Biological Engineering*, 40:462–469, 2020. doi:10.1007/s40846-020-00529-4.
- [8] M. Barstugan, U. Ozkaya, and S. Ozturk. Coronavirus (COVID-19) classification using CT images by machine learning methods. *arXiv*:2003.09424. doi:10.48550/arXiv.2003.09424.
- [9] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In: *Proc. 5th Annual Workshop on Computational Learning Theory, COLT '92*, pp. 144–152. Association for Computing Machinery, 1992. doi:10.1145/130385.130401.

- [10] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi:[10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324).
- [11] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone. Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays. *Computer Methods and Programs in Biomedicine*, 196:105608, 2020. doi:[10.1016/j.cmpb.2020.105608](https://doi.org/10.1016/j.cmpb.2020.105608).
- [12] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In: *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794. Association for Computing Machinery, San Francisco, California, USA, 2016. doi:[10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785).
- [13] M. E. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, et al. Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*, 8:132665–132676, 2020. doi:[10.1109/ACCESS.2020.3010287](https://doi.org/10.1109/ACCESS.2020.3010287).
- [14] N. Dey, V. Rajinikanth, S. J. Fong, M. S. Kaiser, and M. Mahmud. Social group optimization–assisted kapur’s entropy and morphological segmentation for automated detection of COVID-19 infection from computed tomography images. *Cognitive Computation*, 12:1011–1023, 2020. doi:[10.1007/s12559-020-09751-3](https://doi.org/10.1007/s12559-020-09751-3).
- [15] R. O. Duda, P. E. Hart, et al. *Pattern Classification and Scene Analysis*, vol. 3. Wiley New York, 1973.
- [16] K. El Asnaoui and Y. Chawki. Using X-ray images and deep learning for automated detection of coronavirus disease. *Journal of Biomolecular Structure and Dynamics*, 39(10):3615–3626, 2021. doi:[10.1080/07391102.2020.1767212](https://doi.org/10.1080/07391102.2020.1767212).
- [17] K. El Asnaoui, Y. Chawki, and A. Idri. Automated methods for detection and classification pneumonia based on X-ray images using deep learning. In: *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*, pp. 257–284. Springer, 2021. doi:[10.1007/978-3-030-74575-2_14](https://doi.org/10.1007/978-3-030-74575-2_14).
- [18] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi:[10.1109/TSMC.1973.4309314](https://doi.org/10.1109/TSMC.1973.4309314).
- [19] C. Jin, W. Chen, Y. Cao, Z. Xu, Z. Tan, et al. Development and evaluation of an artificial intelligence system for COVID-19 diagnosis. *Nature Communications*, 11(1):5088, 2020. doi:[10.1038/s41467-020-18685-1](https://doi.org/10.1038/s41467-020-18685-1).
- [20] S. H. Kassania, P. H. Kassanib, M. J. Wesolowskic, K. A. Schneidera, and R. Detersa. Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: A machine learning based approach. *Biocybernetics and Biomedical Engineering*, 41(3):867–879, 2021. doi:[10.1016/j.bbe.2021.05.013](https://doi.org/10.1016/j.bbe.2021.05.013).
- [21] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, et al. LightGBM: A highly efficient gradient boosting decision tree. In: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, et al., eds., *Advances in Neural Information Processing Systems – Proc. NIPS 2017*, vol. 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.
- [22] C. Liu, X. Wang, C. Liu, Q. Sun, and W. Peng. Differentiating novel coronavirus pneumonia from general pneumonia based on machine learning. *Biomedical Engineering Online*, 19(1):1–14, 2020. doi:[10.1186/s12938-020-00809-9](https://doi.org/10.1186/s12938-020-00809-9).
- [23] H. Mohammad-Rahimi, M. Nadimi, A. Ghalyanchi-Langeroudi, M. Taheri, and S. Ghafouri-Fard. Application of machine learning in diagnosis of COVID-19 through X-ray and CT images: A scoping review. *Frontiers in Cardiovascular Medicine*, 8:638011, 2021. doi:[10.3389/fcvm.2021.638011](https://doi.org/10.3389/fcvm.2021.638011).
- [24] P. Mooney. Chest X-Ray Images (pneumonia) Dataset, 2018. <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>, Kaggle dataset.

- [25] Y. Oh, S. Park, and J. C. Ye. Deep learning COVID-19 features on CXR using limited training data sets. *IEEE Transactions on Medical Imaging*, 39(8):2688–2700, 2020. doi:[10.1109/TMI.2020.2993291](https://doi.org/10.1109/TMI.2020.2993291).
- [26] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996. doi:[10.1016/0031-3203\(28\)95.2900067-4](https://doi.org/10.1016/0031-3203(28)95.2900067-4).
- [27] K. Preetha and Dr. S. K. Jayanthi. GLCM and GLRLM based feature extraction technique in mammogram images. *International Journal of Engineering and Technology*, 7:266, 2018. doi:[10.14419/IJET.V7I2.21.12378](https://doi.org/10.14419/IJET.V7I2.21.12378).
- [28] M. Rahimzadeh and A. Attar. A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. *Informatics in Medicine Unlocked*, 19:100360, 2020. doi:[10.1016/j.imu.2020.100360](https://doi.org/10.1016/j.imu.2020.100360).
- [29] S. Rajaraman and S. Antani. Weakly labeled data augmentation for deep learning: a study on COVID-19 detection in chest X-rays. *Diagnostics*, 10(6):358, 2020. doi:[10.3390/diagnostics10060358](https://doi.org/10.3390/diagnostics10060358).
- [30] F. Shi, L. Xia, F. Shan, B. Song, D. Wu, et al. Large-scale screening to distinguish between COVID-19 and community-acquired pneumonia using infection size-aware classification. *Physics in Medicine & Biology*, 66(6):065031, 2021. doi:[10.1088/1361-6560/abe838](https://doi.org/10.1088/1361-6560/abe838).
- [31] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, et al. Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18(6):2775–2780, 2021. doi:[10.1109/FTCB.2021.3065361](https://doi.org/10.1109/FTCB.2021.3065361).
- [32] A. Tahamtan and A. Ardebili. Real-time rt-pcr in covid-19 detection: issues affecting the results. *Expert Review of Molecular Diagnostics*, 20(5):453–454, 2020. doi:[10.1080/14737159.2020.1757437](https://doi.org/10.1080/14737159.2020.1757437).
- [33] R. Tawsifur, C. D. Muhammad, and K. Amith. COVID-19 radiography database, 2019. doi:<https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database>, Kaggle dataset.
- [34] N. Tsiknakis, E. Trivizakis, E. E. Vassalou, G. Z. Papadakis, D. A. Spandidos, et al. Interpretable artificial intelligence framework for COVID-19 screening on chest X-rays. *Experimental and Therapeutic Medicine*, 20(2):727–735, 2020. doi:[10.3892/etm.2020.8797](https://doi.org/10.3892/etm.2020.8797).
- [35] S. Wang, B. Kang, J. Ma, X. Zeng, M. Xiao, et al. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *European Radiology*, 31:6096–6104, 2021. doi:[10.1007/s00330-021-07715-1](https://doi.org/10.1007/s00330-021-07715-1).
- [36] M. Xu, D. Wang, H. Wang, X. Zhang, T. Liang, et al. COVID-19 diagnostic testing: Technology perspective. *Clinical and Translational Medicine*, 10(4):e158, 2020. doi:[10.1002/ctm2.158](https://doi.org/10.1002/ctm2.158).
- [37] X. Xu, X. Jiang, C. Ma, P. Du, X. Li, et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering*, 6(10):1122–1129, 2020. doi:[10.1016/j.eng.2020.04.010](https://doi.org/10.1016/j.eng.2020.04.010).
- [38] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, et al. Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of COVID-19 pneumonia using computed tomography. *Cell*, 181(6):1423–1433, 2020. doi:[10.1016/j.cell.2020.04.045](https://doi.org/10.1016/j.cell.2020.04.045).
- [39] C. Zheng, X. Deng, Q. Fu, Q. Zhou, J. Feng, et al. Deep learning-based detection for COVID-19 from chest CT using weak label. *MedRxiv*, 2020. MedRxiv.2020.03.12.20027185. doi:[10.1101/2020.03.12.20027185](https://doi.org/10.1101/2020.03.12.20027185).



A. Beena Godbin received the B.Tech. degree in 2005 in information technology and engineering from Anna University, Chennai, Tamil Nadu, and the M.E. degree in computer science and engineering from Anna University, Coimbatore, Tamil Nadu, in 2019. She is currently pursuing the Ph.D. degree with the Vellore Institute of Technology, Chennai. Her research interests include medical image processing, machine learning and deep learning.



Dr. S. Graceline Jasmine, an accomplished professional with a Master's in Computer Application and a Ph.D. in Remotely Sensed Image Processing from VIT, serves as an Associate Professor at the School of Computer Science and Engineering, VIT, Chennai Campus. With 15 years of extensive experience in teaching and research, she has published around 50 papers in international journals and conferences. Currently, she is the research group chair of Imaging and Computer Vision Research Group at VIT Chennai and is engaged in diverse roles, including industrial consultancy and overseeing projects related to Image Processing, Computer Vision, and Remote Sensing. Dr. Jasmine is actively involved in numerous professional bodies and holds certifications such as NASSCOM certified Master Trainer of Associate Analytics and Remote Sensing and Digital Image Analyst certified by ISRO. Her recent projects include AI-based Smart Agriculture, Computer Vision for ST Microelectronics, and 3D weather reconstruction for the Indian Meteorological Department, showcasing her expertise in cutting-edge technologies and applications.

ADDITIONAL LOOK INTO GAN-BASED AUGMENTATION FOR DEEP LEARNING COVID-19 IMAGE CLASSIFICATION

Oleksandr Fedoruk ¹, Konrad Klimaszewski ¹,
Aleksander Ogonowski ¹ and Michał Kruk ²

¹*Department of Complex Systems, National Centre for Nuclear Research, Otwock-Świerk, Poland*

²*Institute of Information Technology, Warsaw University of Life Sciences – SGGW, Warsaw, Poland*

Abstract Data augmentation is a popular approach to overcome the insufficiency of training data for medical imaging. Classical augmentation is based on modification (rotations, shears, brightness changes, etc.) of the images from the original dataset. Another possible approach is the usage of Generative Adversarial Networks (GAN). This work is a continuation of the previous research where we trained StyleGAN2-ADA by Nvidia on the limited COVID-19 chest X-ray image dataset. In this paper, we study the dependence of the GAN-based augmentation performance on dataset size with a focus on small samples. Two datasets are considered, one with 1000 images per class (4000 images in total) and the second with 500 images per class (2000 images in total). We train StyleGAN2-ADA with both sets and then, after validating the quality of generated images, we use trained GANs as one of the augmentations approaches in multi-class classification problems. We compare the quality of the GAN-based augmentation approach to two different approaches (classical augmentation and no augmentation at all) by employing transfer learning-based classification of COVID-19 chest X-ray images. The results are quantified using different classification quality metrics and compared to the results from the previous article and literature. The GAN-based augmentation approach is found to be comparable with classical augmentation in the case of medium and large datasets but underperforms in the case of smaller datasets. The correlation between the size of the original dataset and the quality of classification is visible independently from the augmentation approach.

Keywords: computer vision, deep learning, image classification, generative adversarial networks, medical imaging.

1. Introduction

Computer vision techniques are used in different medical applications for various purposes. They accelerate decision-making while diagnosing patients and support medical personnel on a daily basis. As available algorithms and solutions advance, the problem of medical data accessibility is becoming a bottleneck for new researchers and breakthroughs [25]. Almost all modern algorithms are data-driven and require a lot of data samples to perform well [41]. However, the process of gathering medical data is not easy and often blocked by the high costs of procedures required to obtain the data, patients' personal data access limitations (as GDPR in the European Union or CCPA in the United States of America), rare diseases for which there is just not much data at all.

To overcome that problem, researchers use data augmentation techniques. The main idea is to train models on several modified copies of original data. This reduces overfitting and allows to achieve better training results with less data [36]. In the graphical

data domain, the classical augmentation pipeline includes transformations such as rotating, scaling, changing the brightness of the image, etc. With the rapid development of Generative Adversarial Networks (GAN) [12] it is possible to generate images similar to ones from the given dataset so it is possible to apply GANs as an augmentation pipeline.

In our original experiment, we showed, on a dataset that contains 2000 images per class, that the GAN-based augmentation approach is comparable to but not outperforming classical augmentation [10]. In this paper, we want to move forward and verify if the same is true for even smaller datasets.

The manuscript is organized as follows: in this Section the research problem is described, followed by a literature review in the *Related works* section (Sect. 2). The section on the *Methods* (Sect. 3) describes the *Motivation and methodology* (Sect. 3.1) and the *Dataset* (Sect. 3.2) including its preprocessing. The *GAN-based augmentation* as well as the *Classical augmentation* are both described in Sections 3.3 and 3.4, respectively, followed by Sect. 3.5 on *Image comparison metrics* used to assess their quality. This section ends with a description of the approach used for the *Classification evaluation* for the images, in Section 3.6. In Section 4 on the *Results* the dependence of the classifier output on the sample size is described. The paper ends with a *Conclusion* in Section 5.

Our main contributions include: 1) a study of the multi-class classification performance dependence on the dataset size, from small to moderate samples, of X-ray chest scans; 2) a study of the GAN-based augmentation performance for small datasets using a StyleGAN2-ADA [19] architecture, including Adaptive Discrimination Augmentation, designed to improve GAN training on limited datasets; 3) validation of the StyleGAN2-ADA multi-class training mode to obtain a single generative model for multiple classes; 4) comparison of the GAN-based augmentation to classical augmentation techniques.

This paper presents improved and updated materials originally presented at the 9th Conference on Symbiosis of Technology and IT (SIT) in Kiry, June 2023. These materials have not been previously published.

2. Related works

While chest radiography and computed tomography (CT) scans are not recommended as primary diagnostic tools, they are reported to be highly sensitive in detecting COVID-19 [2]. This led to numerous studies [13] on the applicability of X-ray chest scans in COVID-19 diagnosing. The first to our knowledge ML-based lung imaging classification methods for COVID-19 were works by Xu et al. [42] and by Gozes et al. [14]. Both focus on binary classification of COVID-19 and healthy images with many works that followed [9, 16, 28, 33, 34]. While the binary classification problem for COVID-19 X-ray images is already extensively studied, there were only several attempts to distinguish COVID-19 from other diseases affecting the respiratory system. In particular, the four-class problem is underrepresented in the literature [13]. The lung patterns seen

in COVID-19 are unique but bear a resemblance to those found in pneumonia from other causes [32]. This is concerning, as challenges in distinguishing viral pneumonia from bacterial and fungal pathogen-induced cases have been reported [22]. For instance, A. I. Khan et al. [23] found that a limited dataset leads to problems in the classification of patients with COVID-19 when considering also patients with viral pneumonia and lung opacity. To our knowledge, the CoroNet [23] was the first proposed model that includes four classes with a recent result by E. Khan et al. [24] that achieved 96.13% accuracy utilising a modified EfficientNet-B1 model.

Data augmentation with GANs is not a new idea – multiple papers [36, 37, 43] are showing that this approach is worth deeper investigation, yet the majority of them consider problems with large datasets available [18]. For instance, Bowles et al. [5] study the performance of classical and GAN-based augmentation on data samples ranging from 8 K to 80 K images. Also, GAN-based augmentation for chest X-ray medical imaging (with or without COVID-19 included) has been studied in the literature mostly on moderate-sized datasets. For example, in the proposed IAGAN architecture [26], the models (IAGAN and DCGAN) were trained on two datasets: the “chest X-ray” [21] dataset with two categories – Normal (1575 images) and Pneumonia (4265 images), and the “covid-chestxray” [8] dataset with 3 classes – Normal (8066 images), Pneumonia (5999 images) and COVID-19 (589 images). That results in generative networks trained on more than 4000 images in both cases. In addition, the study focused on synthetic image generation for only two classes while disregarding the possibility of augmentation for the COVID-19 data sample. The very interesting RANDGAN model, proposed by Motamed et al. [27], was also trained on large samples with only two classes — 7493 Normal and 4986 Pneumonia images. In another study [35], a DCGAN architecture tailored for chest X-rays image generation was trained with a dataset of 2000 chest X-rays per 5 classes – Normal, Cardiomegaly, Pleural Effusions, Pulmonary Edema and Pneumothorax. A more recent study by Bali and Mahara used a similar methodology [3] but with DCGAN architecture and a bigger training dataset that contained only two classes – Normal (1314 images) and Pneumonia (3875 images). Finally, Albahli proposed [1] a different GAN architecture – a combination of variational auto-encoder and GAN which was trained on 16 classes with 5000 images per class.

There are only a few studies that consider GAN applications for very small datasets. A very interesting research was performed on a very small dataset of liver lesion images [11]. The dataset contained 3 classes – cysts (53 images), metastases (64 images) and hemangiomas (65 images). Two approaches were tried – DCGAN trained individually per class and Auxiliary Classifier GAN (ACGAN) [29] as a single network that is able to generate images of different classes. The dedicated DCGAN per class approach performed better and resulted in $\approx 7\%$ improvement over the classic augmentation approach while multi-class ACGAN was not able to improve the classification over DCGANs. In the study, both generative networks were trained on classically augmented data samples

and then used as an additional source of synthetic images to further increase of the training dataset. It is worthwhile to investigate whether models with built-in augmentation like StyleGAN2-ADA will behave similarly.

Regretfully, a number of studies on GAN-based augmentation available in the literature [27, 35, 40] omit a comparison with classical augmentation approach. With the training of a GAN being both time and computationally expensive, this limits the proper evaluation of such results. The expected improvement over the simpler method is key information for an informed selection of the most effective augmentation approach. It is also apparent that only several of studies on the topic focus on augmentation for classification problems with more than three classes. Based on our review of the available literature, we find that there is a need for a dedicated study of the multi-class GAN-based augmentation performance in comparison with classical methods, in particular for small and very small datasets. In addition to our knowledge GAN models with built-in augmentation were not evaluated for applicability for small medical data samples.

3. Methods

3.1. Motivation and methodology

In this paper, we test the hypothesis that data augmentation plays a more crucial role in the deep learning process with small datasets than it does with large ones. Also, we try to verify that, in the case of medical imaging, data augmentation based on GAN-generated images could result in bigger data diversity and thus improve deep learning results in comparison to the classical augmentation approach.

We compare 3 data-augmentation approaches with two datasets – the first dataset contains 1000 images per 4 classes and the second contains 500 images per 4 classes respectively. Further in the paper we call the dataset with 1000 images per class a *small* dataset. The dataset with 500 images per class is called *micro*, respectively. To create the small dataset 1000 images of each class were randomly picked from the original dataset after the preprocessing. The micro dataset is a subset of the small dataset and was also created by picking 500 random images of each class. The datasets used are described in detail in the following section.

Data augmentation approaches being compared were: no augmentation at all, classical augmentation, and GAN-based augmentation. To estimate which approach is better we trained a convolutional neural network on data augmented by each approach and evaluate based on different classification quality metrics. The scheme of operations performed is shown in Fig. 1.

3.2. Dataset

The dataset used in the research is the “COVID-19 Radiography Database” developed by a team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka,

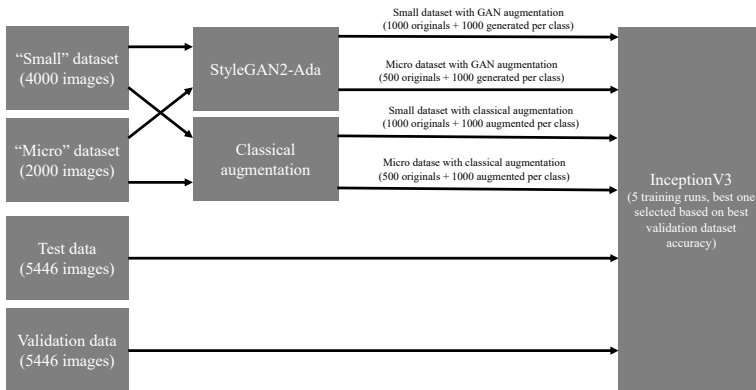


Fig. 1. Visualization of experiment steps and data flow.

Bangladesh, along with their cooperators from Pakistan and Malaysia in collaboration with medical doctors. It is worth noticing that the dataset is being updated and the latest version of it contains way more images than it was when the original experiment was done. As this paper’s goal is an additional investigation of the GAN augmentation technique, we continue to use the same version of the dataset that we used in the original experiment.

The dataset used in this paper contains 3616 images of COVID-19-positive cases, 6012 images of lung opacity (non-COVID lung infection), 1345 images of viral pneumonia, and 10192 images of healthy lungs. Each image is represented in PNG format with dimensions 256×256 . For each image, the dataset authors provided a corresponding lung segmentation mask obtained using a dedicated U-Net model [30]. The examples of the images and segmentations masks are displayed in Fig. 2 and Fig. 3. The dataset was split into 3 subsets: train, validation, and test. The small train subset contains 1000 (500 in the case of the micro dataset) images of each class and is used as a source for classical augmentation, training of the GAN, and for no augmentation approach. The rest of the images were split in half to form validation and test subsets. Also, we use the same test and validation subsets for both experiments with small and micro datasets. The validation subset is used as validation data in target classification CNN training. The test subset was used only as the final trained CNN benchmark which simulates new data from new patients to show real world usage of the trained classification network.

We have preprocessed all images from the original dataset with the following 3-step procedure:

1. All images were cropped to the lung region according to the provided masks.
2. Cropped images were manually reviewed and all images containing any text or graphical annotations/marks were removed.

3. Remaining images were resized into dimensions 128×128 and converted to 1-channel (grayscale) to reduce the amount of data processed while training. The conversion was done by leaving only the first channel of the original images.

After all the described steps, the final version of the dataset used in the experiment contained 3242 images of COVID-19 cases, 2982 images of lung opacity (non-COVID lung infection), 1264 images of viral pneumonia, and 7404 images of healthy lungs. As mentioned earlier 2 experimental cases have been considered in the research. The first experiment was conducted with the small dataset (1000 images per class in the training subset) and the second one with the micro dataset (500 images per class in the training subset). As the original dataset contained 4 classes, there are 4000 images in total in the train subset of the small dataset and 2000 images in the train subset of the micro dataset. Test and validation subsets remained the same for both cases and were prepared during small dataset preparation (see Tab. 1). This made it possible to compare the final results for both experiments as the data those results were calculated on remained the same.

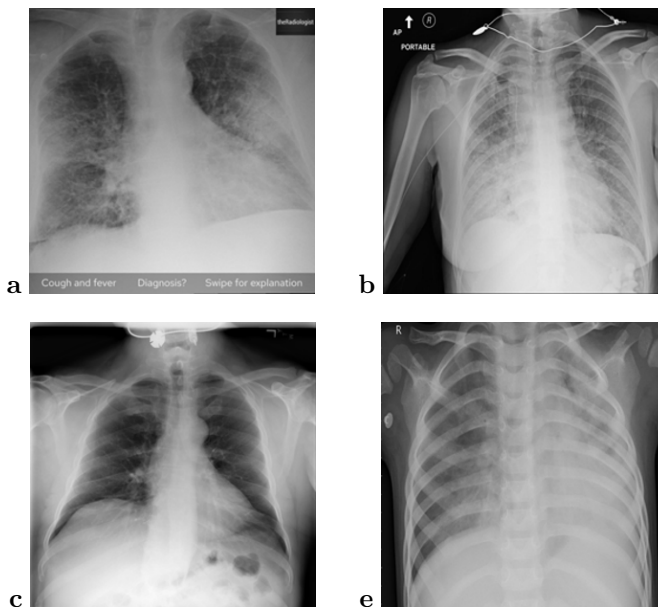


Fig. 2. Examples of unprocessed images from the original dataset. (a) COVID-19; (b) Normal; (c) Lung opacity; (e) Viral pneumonia.

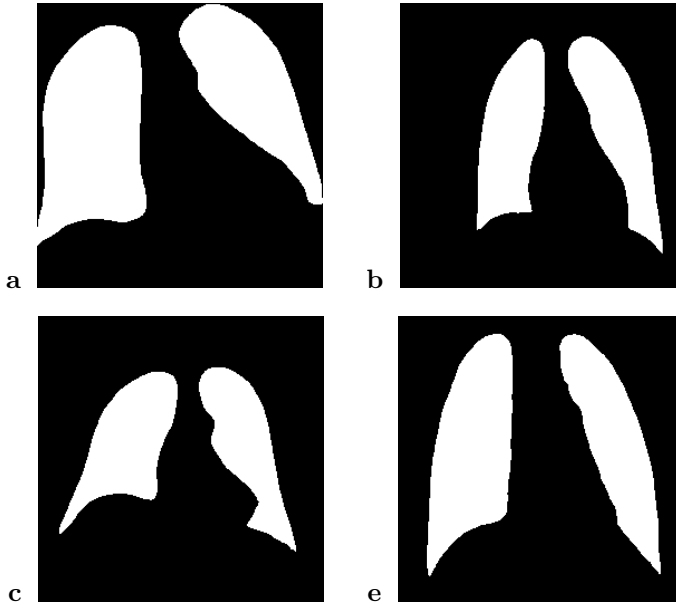


Fig. 3. Example binary masks from the original dataset for the images shown in Fig. 2.

3.3. GAN-based augmentation

In this research we continued to use StyleGAN2 with adaptive discriminator augmentation (ADA) mechanism by NVIDIA [19] as one of its features is the ability to be trained on relatively small datasets and support of class-conditional image generation. For each of the experiments, StyleGAN2-ADA was trained on the corresponding train subset. The training process was monitored, using the validation dataset, to prevent network overfitting. After the training, the epoch with the best Kernel Inception Distance (KID) score was picked as a source of future data generation. The target for r_t ADA heuristic is set to 0.6, both generator and discriminator learning rates were set to 0.0025 while

Tab. 1. Number of images per class in each subset for both small and micro datasets.

| Subset | COVID-19 | Healthy | Lung opacity | Viral pneumonia |
|-------------------|----------|---------|--------------|-----------------|
| Train subset 1000 | 1000 | 1000 | 1000 | 1000 |
| Train subset 500 | 500 | 500 | 500 | 500 |
| Validation subset | 1121 | 3202 | 991 | 132 |
| Test subset | 1121 | 3202 | 991 | 132 |

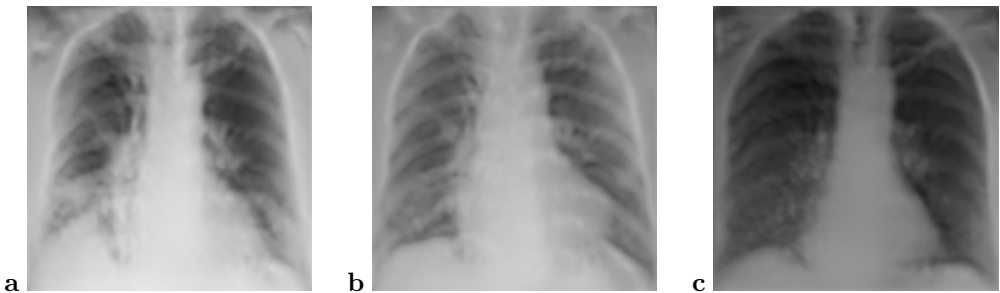


Fig. 4. Kernel Inception Distance values with correlated example image generated by GAN trained on the small dataset. (a) $KID \approx 19.46$; (c) $KID \approx 13.29$; (b) $KID \approx 12.26$.



Fig. 5. Kernel Inception Distance values with correlated example image generated by GAN trained on the micro dataset. (a) $KID \approx 18.82$; (b) $KID \approx 13.30$; (c) $KID \approx 12.89$.

the batch size was set to 32. Following StyleGAN2 authors [20], we use non-saturating logistic loss with R_1 regularization with the regularization term weight γ set to 1.024. Multiclass training was enabled so one network was able to generate images for all target classes after the training was done. All other parameters are set to default values provided by the NVIDIA implementation [19]. The network was trained with a single NVIDIA Tesla K80 GPU and for each case, training took around 7 days.

After the training had been finished we generated 1000 artificial images per class for both experiments, with example images presented in Fig. 4 and 5. The GAN-augmented training dataset for CNN contained 2000 images per class (8000 images in total) for the small dataset and 1500 images per class (6000 in total) for the micro dataset.

3.4. Classical augmentation

Similarly to the GAN-based augmentation described earlier, we have generated 1000 additional images (Fig. 6) by applying classical augmentation with parameters as follows:

- rotation – randomly rotate the image by an angle of up to 5 degrees clockwise or counterclockwise;
- shift – randomly shift the image along cardinal axes within the range of 5% of the specific image size, the empty field is filled with the trace of the last shifted pixels;
- stretch – randomly stretch the image between opposite vertices by up to 5 %;
- zoom – randomly zoom in pictures up to 15% of the specific value of the image size;
- brightness change – randomly brighten or darken the image by up to 40%.

The values of the parameters remain the same as in the previous work and were picked to maximize the accuracy score on the validation subset [10].

3.5. Image comparison metrics

In the previous work, we have used the Fréchet Inception Distance (FID) [15] metric to select the best-performing state of the StyleGAN2-ADA network [10] throughout the training process as it is commonly used to evaluate the quality of images generated by GANs. We calculated a mean FID value, for each training epoch, across all classes between the train subset and generated images. In this research, we added the Kernel Inception Distance (KID) [4] metric as FID is biased for smaller datasets [6]. KID is very similar to FID in that it measures the difference between two sets of samples by calculating the square of the maximum mean discrepancy between vectors of vision-relevant features as extracted by the Inception-v3 [38] classifier network. KID compared to FID has several advantages and performs better with smaller sets and more consistently matches human perception. Similarly to FID, a smaller value of KID means that compared images are more similar to each other, and comparing the same set of images will result in a value equal to 0. Looking at the graphs of both KID and FID values per epoch (Figures 7–10) it is visible that the overall training trend is the same – the value drops with each epoch of training until around epoch 250 and then it starts to grow slowly. But at the same time, KID values change more drastically per each epoch

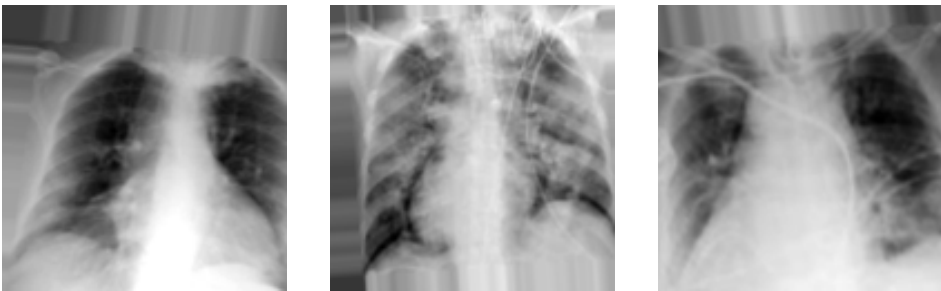


Fig. 6. Examples of images with applied classical augmentation pipeline.

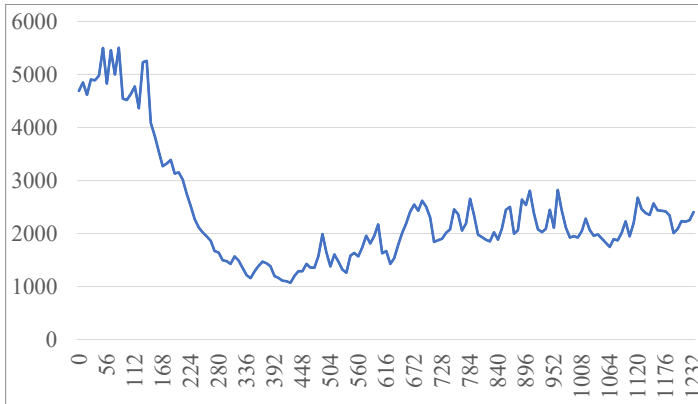


Fig. 7. Fréchet Inception Distance graph for the small dataset (1000 images per class in the train subset) GAN training.

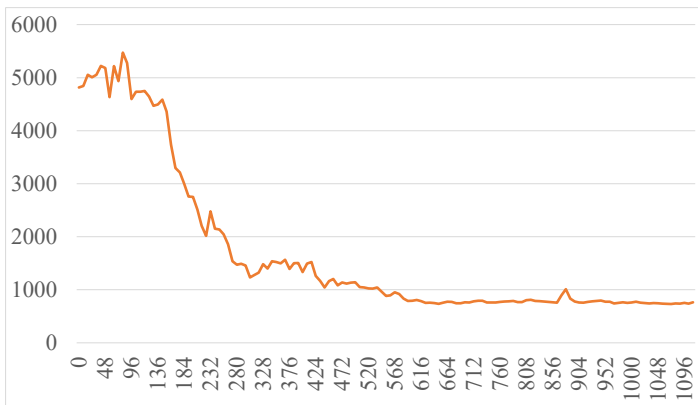


Fig. 8. Fréchet Inception Distance graph for the micro dataset (500 images per class in the train subset) GAN training.

which may indicate, that the KID metric is more sensitive to differences between real and generated images. In addition, as in the original paper, we used RMSE, SRE, and SSIM metrics to verify the quality of generated images [10].

3.6. Classification evaluation

To evaluate and compare the augmentation techniques described above, we trained a convolutional neural network using each of them. The network was trained to classify

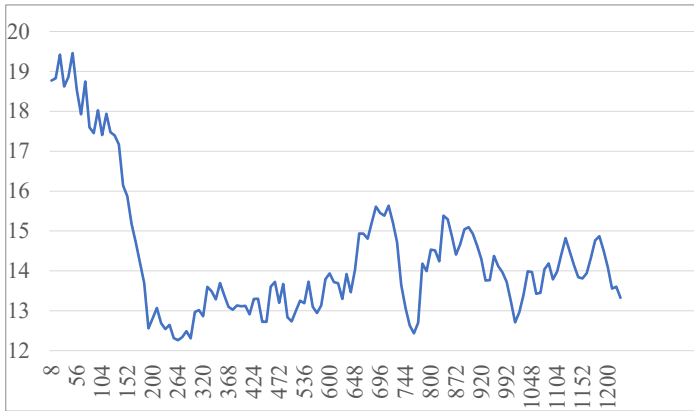


Fig. 9. Kernel Inception Distance graph for case the small dataset (1000 images per class in the train subset) GAN training.



Fig. 10. Kernel Inception Distance graph for the micro dataset (500 images per class in the train subset) GAN training.

4 classes present in the original dataset. We used Inception-v3 with a transfer learning technique as an augmentation benchmark network [38]. We used Keras default implementation with library-provided ImageNet weights [17]. The head of the network was replaced with the following output layers for the transfer learning:

- flatten layer;
- dense layer with 10 neurons and an Exponential Linear Unit (ELU) activation function [7];
- dense layer with 20 neurons and an ELU activation function;
- dense layer with 4 neurons and a softmax activation function.

Categorical cross-entropy was picked as the loss function and the learning rate was set to Keras default of 0.001 [31]. We continue using RMSprop as an optimizer as it was selected on validation accuracy in the original article [10]. The network was trained for 11 epochs with a batch size equal to 32. The total amount of epochs was reduced in comparison with previous work as the best results were achieved in the first 11 epochs in almost all training runs. For the clarity of the experiment, the network was trained 5 times for each experimental case and each augmentation pipeline (no augmentation, classical augmentation, GAN-augmentation). The network frozen weights with the highest validation accuracy score were picked as a final version that was later evaluated on the test subset. To examine and compare the quality of the trained network, several model evaluation metrics were calculated: accuracy, precision, recall, F1, specificity, and Matthew’s correlation coefficient (MCC). Metrics were calculated on the test dataset. Values of the calculated metrics are presented in the **Results** section of the paper.

4. Results

After the classification metrics are calculated on the test dataset, it is visible, that any augmentation approach is better than no augmentation at all. At the same time, GAN-based augmentation and classical augmentation perform comparably for the dataset with at least 1000 images per class in the training dataset, as shown in Table 2 and Fig. 11. Classical augmentation outperforms GAN-based augmentation with datasets containing 500 images per class, as presented in Table 3 and Fig. 12. The overall networks’ results are slightly worse independently of the augmentation approach in comparison with the ones achieved in our previous work [10]. Finally, the classical augmentation shows itself as the best augmentation approach while GAN-based augmentation can achieve comparable results but requires significantly more time and hardware to be performed.

Tab. 2. Classification metrics values for the small dataset (1000 images in the train subset)

| Augmentation pipeline | Accuracy | Precision | Recall | F1 | Specificity | MCC |
|------------------------|----------|-----------|--------|-------|-------------|------|
| No augmentation | 0.85 | 0.845 | 0.769 | 0.805 | 0.931 | 0.74 |
| Classical augmentation | 0.87 | 0.861 | 0.815 | 0.837 | 0.934 | 0.78 |
| GAN-augmentation | 0.862 | 0.822 | 0.815 | 0.819 | 0.936 | 0.76 |

Tab. 3. Classification metrics values for the micro dataset (500 images in the train subset)

| Augmentation pipeline | Accuracy | Precision | Recall | F1 | Specificity | MCC |
|------------------------|----------|-----------|--------|-------|-------------|-------|
| No augmentation | 0.783 | 0.659 | 0.68 | 0.669 | 0.903 | 0.573 |
| Classical augmentation | 0.842 | 0.85 | 0.789 | 0.818 | 0.93 | 0.749 |
| GAN-augmentation | 0.81 | 0.83 | 0.685 | 0.751 | 0.9 | 0.665 |

Tab. 4. Accuracy metric value from the original article (2000 images per dataset)

| Augmentation pipeline | Accuracy |
|-----------------------|----------|
| No augmentation | 0.855 |
| Classic augmentation | 0.891 |
| GAN-augmentation | 0.871 |

We can take a look (Table 4) at the accuracy value obtained in our previous work where 2000 images per class were used [10], there is a visible correlation between classification accuracy and the size of the training dataset independently from data augmentation applied.

5. Conclusion

We have studied the performance of GAN-based augmentation for the classification of lung X-ray medical images as a function of dataset size. The obtained results show that GAN-based augmentation is comparable with classical augmentation for medium and large datasets. Unfortunately, the time and hardware requirements make it unreasonable to use such an approach as the main augmentation technique. In the case of small datasets, the GAN model wasn't able to train well enough to be a source of valuable training data. At the same time, the fact of GAN being able to compete with classical augmentation for larger datasets, potentially allows researchers and medical institutions to solve the problem of medical data availability by sharing synthetically generated images instead of real ones [39]. Therefore, the topic of GAN-based augmentation should be investigated further.

Acknowledgments

This work was completed in part with resources provided by the Świerk Computing Centre at the National Centre for Nuclear Research. This work benefited from the software tools developed in the frame of the EuroHPC PL Project, Smart Growth Operational

Programme 4.2. We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centres: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2022/015617.

References

- [1] S. Albahli. Efficient GAN-based chest radiographs (CXR) augmentation to diagnose coronavirus disease pneumonia. *International Journal of Medical Sciences*, 17(10):1439–1448, 2020. doi:10.7150/ijms.46684.
- [2] H. X. Bai, B. Hsieh, Z. Xiong, K. Halsey, J. W. Choi, et al. Performance of radiologists in differentiating COVID-19 from non-COVID-19 viral pneumonia at chest CT. *Radiology*, 296(2):E46–E54, Aug 2020. doi:10.1148/radiol.2020200823.
- [3] M. Bali and T. Mahara. Comparison of affine and DCGAN-based data augmentation techniques for chest X-ray classification. *Procedia Computer Science*, 218:283–290, 2023. International Conference on Machine Learning and Data Engineering. doi:10.1016/j.procs.2023.01.010.
- [4] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In: *Proc. Int. Conf. Learning Representations (ICRL 2018)*, 2018. <https://openreview.net/forum?id=r11U0zWCW>.
- [5] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, et al. GAN augmentation: Augmenting

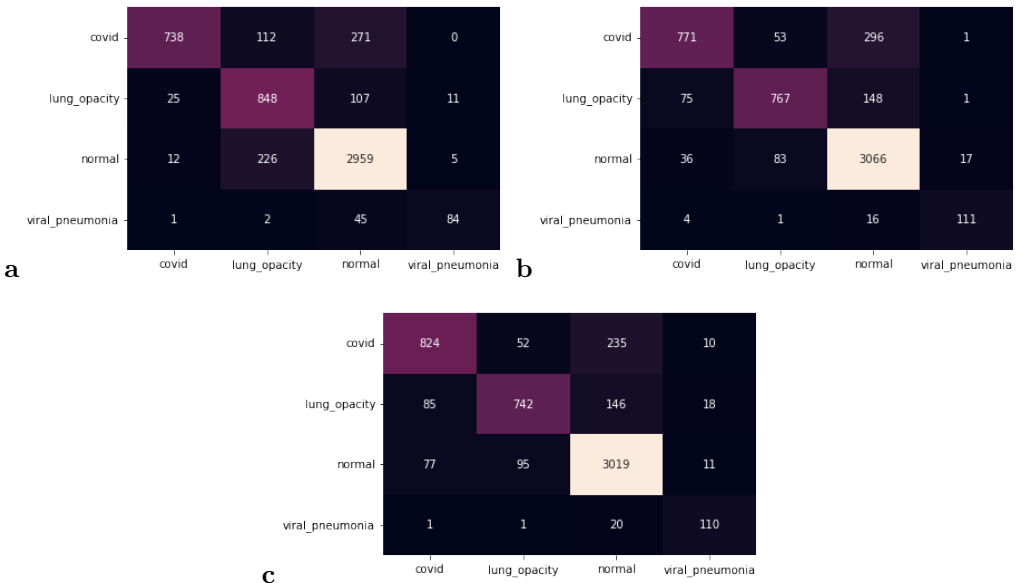


Fig. 11. Confusion matrices for different augmentations of the small dataset. The vertical axis represents predicted values, horizontal axis represents real values. (a) No augmentations; (b) Classical augmentations; (c) GAN-based augmentations.

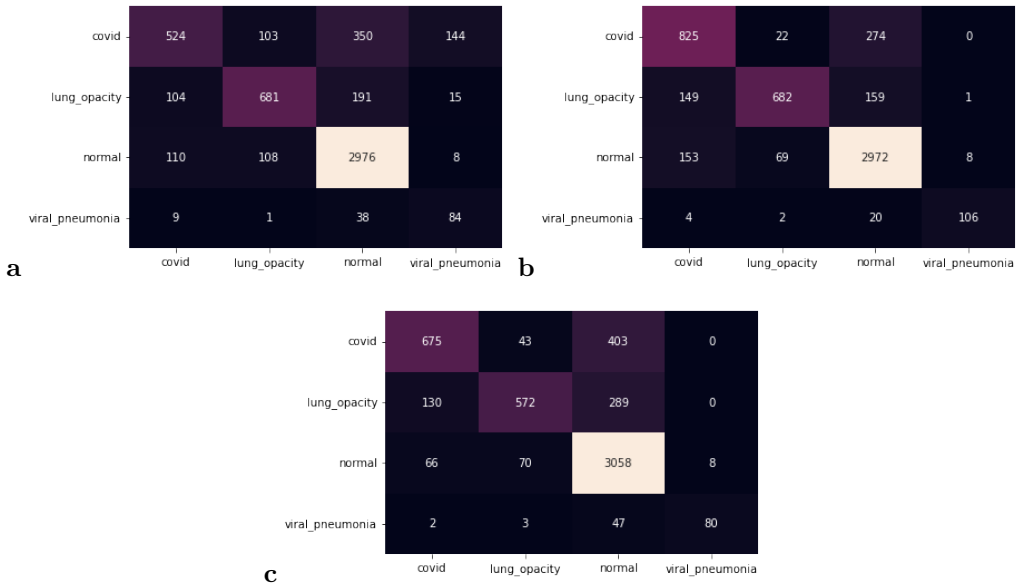


Fig. 12. Confusion matrices for different augmentations of the micro dataset. The vertical axis represents predicted values, horizontal axis represents real values. (a) No augmentations; (b) Classical augmentations; (c) GAN-based augmentations.

training data using Generative Adversarial Networks. *arXiv*, 2018. ArXiv.1810.10863. <https://arxiv.org/abs/1810.10863>.





- [6] M. J. Chong and D. Forsyth. Effectively unbiased FID and inception score and where to find them. In: *Proc. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6069–6078, 2020. doi:10.1109/CVPR42600.2020.00611.
- [7] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In: *Proc. Int. Conf. Learning Representations (ICLR 2016)*, 2016. <https://arxiv.org/abs/1511.07289>.
- [8] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Duong, et al. COVID-19 image data collection: Prospective predictions are the future. *Machine Learning for Biomedical Imaging*, 1:1–38, 2020. doi:10.59275/j.melba.2020-48g7.
- [9] D. Ezzat, A. E. Hassanien, and H. A. Ella. An optimized deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization. *Applied Soft Computing*, 98:106742, Jan 2021. doi:10.1016/j.asoc.2020.106742.
- [10] O. Fedoruk, K. Klimaszewski, A. Ogonowski, and R. Możdżonek. Performance of GAN-based augmentation for deep learning COVID-19 image classification. In: *Proc. Int. Workshop on Machine Learning and Quantum Computing Applications in Medicine and Physics*. Warsaw, Poland, 13–16 Sep 2022. Accepted for publication in AIP Conference Proceedings. <https://events.ncbj.gov.pl/event/141/page/65-home>.
- [11] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, et al. GAN-based synthetic medical

- image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, 321:321–331, Dec 2018. doi:10.1016/j.neucom.2018.09.013.
- [12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, et al. Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, 3, 06 2014. doi:10.1145/3422622.
- [13] W. Gouda, M. Almurafteh, M. Humayun, and N. Z. Jhanjhi. Detection of COVID-19 based on chest X-rays using deep learning. *Healthcare*, 10(2):343, 2022. doi:10.3390/healthcare10020343.
- [14] O. Gozes, M. Frid-Adar, H. Greenspan, P. D. Browning, H. Zhang, et al. Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using Deep Learning CT image analysis. *arXiv*, 2020. ArXiv:2003.05037. doi:10.48550/arXiv.2003.05037.
- [15] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, et al., eds., *Advances in Neural Information Processing Systems: Proc. NIPS 2017*, vol. 30. Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.
- [16] Md. I. Zahirul, Md. I. Milon, and A. Asraf. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in Medicine Unlocked*, 20:100412, 2020. doi:10.1016/j.imu.2020.100412.
- [17] InceptionV3 – Keras Applications API Reference. <https://keras.io/api/applications/inceptionv3/>, [Accessed Dec 2023].
- [18] J. Jeong, A. Tariq, T. Adejumo, H. Trivedi, J. Gichoya, et al. Systematic review of generative adversarial networks (GANs) for medical image classification and segmentation. *Journal of Digital Imaging*, 35, 01 2022. doi:10.1007/s10278-021-00556-w.
- [19] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, et al. Training generative adversarial networks with limited data. In: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., *Advances in Neural Information Processing Systems: Proc. NeurIPS 2020*, vol. 33, pp. 12104–12114. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper_files/paper/2020/hash/8d30aa96e72440759f74bd2306c1fa3d-Abstract.html.
- [20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, et al. Analyzing and improving the image quality of StyleGAN. In: *Proc. CVPR*, 2020. 1912.04958, <https://arxiv.org/abs/1912.04958>.
- [21] D. Kermany. Labeled optical coherence tomography (OCT) and chest X-ray images for classification. *Mendeley Data*, V2, 2018. doi:10.17632/rscbjbr9sj.2.
- [22] D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.e9, Feb 2018. doi:10.1016/j.cell.2018.02.010.
- [23] A. I. Khan, J. L. Shah, and M. M. Bhat. Coronet: A deep neural network for detection and diagnosis of COVID-19 from chest X-ray images. *Computer Methods and Programs in Biomedicine*, 196:105581, Nov 2020. doi:10.1016/j.cmpb.2020.105581.
- [24] E. Khan, M. Z. U. Rehman, F. Ahmed, F. A. Alfouzan, N. M. Alzahrani, et al. Chest X-ray classification for the detection of COVID-19 using deep learning techniques. *Sensors*, 22(3):1211, 2022. doi:10.3390/s22031211.
- [25] J. Li, G. Zhu, C. Hua, M. Feng, B. Bennamoun, et al. A systematic collection of medical image datasets for deep learning. *ACM Computing Surveys*, 56(5), nov 2023. doi:10.1145/3615862.
- [26] S. Motamed, P. Rogalla, and F. Khalvati. Data augmentation using Generative Adversarial Networks (gans) for gan-based detection of pneumonia and COVID-19 in chest X-ray images. *Informatics in Medicine Unlocked*, 27:100779, 2021. doi:10.1016/j.imu.2021.100779.

- [27] S. Motamed, P. Rogalla, and F. Khalvati. RANDGAN: Randomized generative adversarial network for detection of COVID-19 in chest X-ray. *Scientific Reports*, 11(1):8602, Apr 2021. doi:10.1038/s41598-021-87994-2.
- [28] A. Narin, C. Kaya, and Z. Pamuk. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, 24(3):1207–1220, May 2021. doi:10.1007/s10044-021-00984-y.
- [29] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In: D. Precup and Y. W. Teh, eds., *Proc. 34th International Conference on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, pp. 2642–2651. PMLR, 06–11 Aug 2017. <https://proceedings.mlr.press/v70/odena17a.html>.
- [30] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, et al. Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in Biology and Medicine*, 132:104319, 2021. doi:10.1016/j.combiomed.2021.104319.
- [31] RMSprop – Keras Optimizers API Reference. <https://keras.io/api/optimizers/rmsprop/>, [Accessed Dec 2023].
- [32] G. D. Rubin, C. J. Ryerson, L. B. Haramati, N. Sverzellati, J. P. Kanne, et al. The role of chest imaging in patient management during the COVID-19 pandemic: A multinational consensus statement from the fleischner society. *Chest*, 158(1):106–116, Jul 2020. doi:10.1016/j.chest.2020.04.003.
- [33] W. Saad, W. A. Shalaby, M. Shokair, F. A. El-Samie, M. Dessouky, et al. COVID-19 classification using deep feature concatenation technique. *Journal of Ambient Intelligence and Humanized Computing*, 13(4):2025–2043, 2022. doi:10.1007/s12652-021-02967-7.
- [34] K. Sahinbas and F. O. Catak. Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images. In: U. Kose, D. Gupta, V. H. C. de Albuquerque, and A. Khanna, eds., *Data Science for COVID-19 – Computational Perspective*, chap. 24, pp. 451–466. Academic Press, 2021. doi:10.1016/B978-0-12-824536-1.00003-4.
- [35] H. Salehinejad, S. Valaee, T. Dowdell, E. Colak, and J. Barfett. Generalization of deep neural networks for chest pathology classification in X-rays using Generative Adversarial Networks. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 990–994, 2018. doi:10.1109/ICASSP.2018.8461430.
- [36] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019. doi:10.1186/s40537-019-0197-0.
- [37] N. K. Singh and K. Raza. *Medical Image Generation Using Generative Adversarial Networks: A Review*, pp. 77–96. Springer Singapore, 2021. doi:10.1007/978-981-15-9735-0_5.
- [38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception architecture for computer vision. In: *2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi:10.1109/CVPR.2016.308.
- [39] R. Venugopal, N. Shafqat, I. Venugopal, B. M. J. Tillbury, H. D. Stafford, et al. Privacy preserving generative adversarial networks to model electronic health records. *Neural Networks*, 153:339–348, 2022. doi:10.1016/j.neunet.2022.06.022.
- [40] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, et al. CovidGAN: Data augmentation using auxiliary classifier GAN for improved Covid-19 detection. *IEEE Access*, 8:91916–91923, 2020. doi:10.1109/ACCESS.2020.2994762.
- [41] S. E. Whang, Y. Roh, H. Song, and J.-G. Lee. Data collection and quality challenges in deep learning: A data-centric AI perspective. *The VLDB Journal*, 32(4):791–813, 2023. doi:10.1007/s00778-022-00775-9.

- [42] X. Xu, X. Jiang, C. Ma, P. Du, X. Li, et al. A deep learning system to screen novel coronavirus disease 2019 pneumonia. *Engineering*, 6(10):1122–1129, 2020. doi:[10.1016/j.eng.2020.04.010](https://doi.org/10.1016/j.eng.2020.04.010).
- [43] X. Yi, E. Walia, and P. Babyn. Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58:101552, 2019. doi:<https://doi.org/10.1016/j.media.2019.101552>.

RULE-BASED EXPLAINING MODULE: ENHANCING THE INTERPRETABILITY OF RECURRENT RELATIONAL NETWORK IN SUDOKU SOLVING

Pimpa Cheewaprabkakit ^{1,2}, Timothy K. Shih ^{2,*}, Timothy Lau²,
Yu-Cheng Lin ³ and Chih-Yang Lin ⁴

¹*Department of Information Technology, Asia-Pacific International University, Saraburi, Thailand*

²*Department of Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan*

³*Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan*

⁴*Department of Mechanical Engineering, National Central University, Taoyuan, Taiwan*

*Corresponding author: Timothy K. Shih (timothykshih@gmail.com)

Abstract Computer vision has gained significant attention in the field of information technology due to its widespread application that addresses real-world challenges, surpassing human intelligence in tasks such as image recognition, classification, natural language processing, and even game playing. Sudoku, a challenging puzzle that has captivated many people, exhibits a complexity that has attracted researchers to leverage deep learning techniques for its solution. However, the reliance on black-box neural networks has raised concerns about transparency and explainability. In response to this challenge, we present the Rule-based Explaining Module (REM), which is designed to provide explanations of the decision-making processes using Recurrent Relational Networks (RRN). Our proposed methodology is to bridge the gap between complex RRN models and human understanding by unveiling the specific rules applied by the model at each stage of the Sudoku puzzle solving process. Evaluating REM on the Minimum Sudoku dataset, we achieved an accuracy of over 98.00%.

Keywords: rule-based explaining module, recurrent relational network, Sudoku puzzle solving, machine learning.

1. Introduction

Sudoku is one of the most popular intellectual puzzle games [26] that involves logical thinking to fill in numbers. It comprises a 9×9 grid, forming a numerical puzzle with nine rows and nine columns, totalling 81 cells. The grid is further divided into nine 3×3 subgrids, referred to as blocks, each containing nine cells. To initiate the game, a set of given numbers is provided as hints. These hints are placed in some of the cells of the Sudoku puzzle, providing clues to help the player solve the puzzle. In most cases, the more cells that are given, the easier the puzzle tends to be. Currently, to the best of our knowledge, the fewest clues required for a proper Sudoku puzzle is 17. This means that the most challenging Sudoku puzzles now are those with only 17 known cells. The goal is to fill the empty cells with the numbers 1 through 9, ensuring that each number appears only once in each row, column, and block [28]. An example of a Sudoku puzzle and its solution is shown in Fig. 1.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | | 7 | | | | | |
| 6 | | | 1 | 9 | 5 | | | |
| | 9 | 8 | | | | | 6 | |
| 8 | | | 6 | | | | | 3 |
| 4 | | | 8 | 3 | | | | 1 |
| 7 | | | 2 | | | | | 6 |
| | 6 | | | | 2 | 8 | | |
| | | | 4 | 1 | 9 | | | 5 |
| | | | 8 | | | 7 | 9 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| 4 | 2 | 6 | 8 | 5 | 3 | 7 | 9 | 1 |
| 7 | 1 | 3 | 9 | 2 | 4 | 8 | 5 | 6 |
| 9 | 6 | 1 | 5 | 3 | 7 | 2 | 8 | 4 |
| 2 | 8 | 7 | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

Fig. 1. Sudoku puzzle and its solution.

The rapidly evolving realm of computer vision has increasing in various aspects of our daily lives, encompassing domain such as image recognition [19], language translation [25], and critical medical applications like X-ray image analysis for disease diagnosis [10, 27], and game playing [8]. The challenging of Sudoku puzzle has attracted researchers to leverage deep learning techniques for its solution. The fascination lies not only in the puzzles' complexity but also in the diverse strategies required for their solution. Traditional rule-based methods have been prevalent, employing strategies such as elimination, naked singles, and hidden singles. The advent of deep learning has ushered in a revolution of puzzle-solving, introducing adaptive and data-driven approaches to tackle Sudoku's complexities. Despite the remarkable capabilities of deep learning, the reliance on black-box nature of neural network has raised concerns about inner workings and transparency of their decision-making processes, particularly in contexts where machine learning applications make critical decisions. Enhancing the transparency of black-box neural networks becomes particularly crucial in applications requiring abstract reasoning about objects and their interactions, enabling audiences to comprehend the rationale behind the decision process of machine learning. One direct method to achieve this transparency is through the addition of explanations [21]. Existing explanation methods for specific applications such as tracking feature extraction in image recognition to visualize the interpretation of input data [12]. Furthermore, logical methods that integrate logical reasoning into neural networks have been proposed to enhance interpretability throughout the entire process [7, 24]. However, these logical methods may face challenges in generalizing to new data or situations, often relying on hand-crafted rules or assumptions about the data. A similar approach, the expert system, is rule-based [4] but demands a substantial amount of knowledge to be encoded in its rules. This process can be time-consuming and expensive, particularly in complex domains. Macha et al. introduced RuleXAI [13], a tool designed to enhancing explainability in machine learning models. While currently limited to classification, regression, and survival analysis, RuleXAI leverages rule-based explanations and feature relevance to make

models more understandable. However, it may not be perfectly accurate for all model types, especially those with complex, black-box in neural network.

This study introduces the Rule-based Explaining Module (REM), a specialized tool designed to unveil the specific rules applied by the model at each stage of the puzzle-solving process. Therefore, the main contribution of our study is summarized as follows:

1. We present the Rule-based Explaining Module (REM), designed to offer comprehensive, step-by-step explanations of the decision-making processes employed by Recurrent Relational Network (RRN) in Sudoku puzzle solving.
2. We conducted experiments using the Minimum Sudoku and 1 million Sudoku games datasets. The results demonstrated that our model significantly contributes to the transparency and interpretability of the Sudoku solving process.

The remainder of this paper is structured as follows: Section 2 provides a review of related work, Section 3 introduces the proposed Rule-based Explaining module for solving Sudoku, and Sections 4 and 5 present experimental results and conclusions, respectively.

2. Related works

Sudoku is a wildly popular logic-based puzzle game, has captivated individuals of all ages for many years. Its deceptively simple rules and endless variations have sparked a worldwide fascination. The challenge of solving Sudoku puzzles lies in their ability to test both logical reasoning and strategic thinking, especially for more difficult puzzles that captivating players with their intricate patterns and hidden clues.

Over the years, various techniques have been explored for solving Sudoku puzzle. Classical methods like backtracking [20], constraint propagation [14], and genetic algorithms [11] have shed light on Sudoku solving strategies. For instance, the pencil-and-paper method, also known as the human solver approach, efficiently solves easier puzzles but faces difficulties with more challenging ones, especially in the absence of clear clues. In contrast, backtracking, though it guarantees a solution for every valid puzzle, is considerably slower [16]. Subsequently, a hybrid method for solving Sudoku puzzles, integrating traditional backtracking algorithms with pencil-and-paper techniques was introduced [26]. This approach initially utilizes pencil-and-paper strategies, followed by applying backtracking to specific sub-grids, and concludes with pencil-and-paper methods on the remaining cells. This method is designed to improve puzzle-solving efficiency. However, its complexity and computational demands could be a drawback. The integration of algorithmic and intuitive strategies might lead to redundant operations and increasing the time required to solve complex Sudoku puzzles. Musliu and Winter have integrated the structured approach of constraint programming with the iterative nature of local search methods in a hybrid solution [14]. This method leverages the strengths of both: the proficiency of constraint programming in solving constraint

satisfaction problems and the effectiveness of iterated local search in optimization tasks. However, a primary challenge arises in balancing the systematic nature of constraint programming with the adaptive strategy of local search. This balancing act could present difficulties in efficiently finding solutions, especially in the context of complex Sudoku puzzles. Das et al. present an evolutionary algorithm that employs genetic operators, such as crossover and mutation, to generate new candidate solutions [3]. This algorithm may require extensive computational resources and time to converge on a solution. Gaddam et al. propose a method for solving Sudoku puzzles using a combination of deep learning and image processing techniques [6]. This method first utilizes image processing techniques to extract the Sudoku grid from an image and then employs a deep learning model to solve the puzzle. It demonstrates the potential of deep learning for solving Sudoku puzzles. However, the accuracy of this method is dependent on the quality of the input image. If the image is blurry or distorted, the accuracy of the deep learning model may be compromised. These techniques, while laying the foundation for understanding the problem, often lacked the flexibility, solution explanation, and adaptability needed to tackle complex puzzles. Björnsson et al., introduced a search-based approach to generate explainable solutions to Sudoku puzzles [1]. This method involves modelling the perceived human mental effort of using different familiar Sudoku-solving techniques. This model serves as guidance for a search algorithm to identify the correct solutions and present them in a way that is easily understandable to human solvers. However, the method's dependence on a potentially inaccurate model of human mental effort could result in explanations that are not entirely accurate. Another approach, Demystify, introduced by Espasa et al., provides step-by-step explanations for solving various pen-and-paper puzzles, including Sudoku [5]. It utilizes Minimal Unsatisfiable Subsets (MUSes) to solve puzzles through logical deduction, identifying essential puzzle components for progress. While Demystify effectively explains puzzle solutions, it requires human input in the form of high-level logical descriptions. Additionally, its applicability may not be suitable for solving all types of pen-and-paper puzzles. Bogaerts et al. provide step-by-step explanations for constraint satisfaction problems (CSPs), focusing on logic grid puzzles as a specific instance of CSPs [2]. They propose a framework for generating step-wise explanations of the inference steps taken during puzzle-solving. However, this approach is particularly reliant on the availability of a formal rule representation for the CSP domain. Without a well-defined set of rules, the framework may struggle to generate meaningful explanations.

The introduction of neural networks, particularly Recurrent Relational Networks (RRNs), revolutionized the landscape of Sudoku solving. RRNs [17], a type of neural network well-suited for learning long-range dependencies in data, proved adept at capturing the intricate relationships between cells in a Sudoku grid. While RRN's capability to learn from large datasets of Sudoku puzzles enabled them to achieve remarkable accuracy, consistently outperforming classical approaches, their black-box nature makes

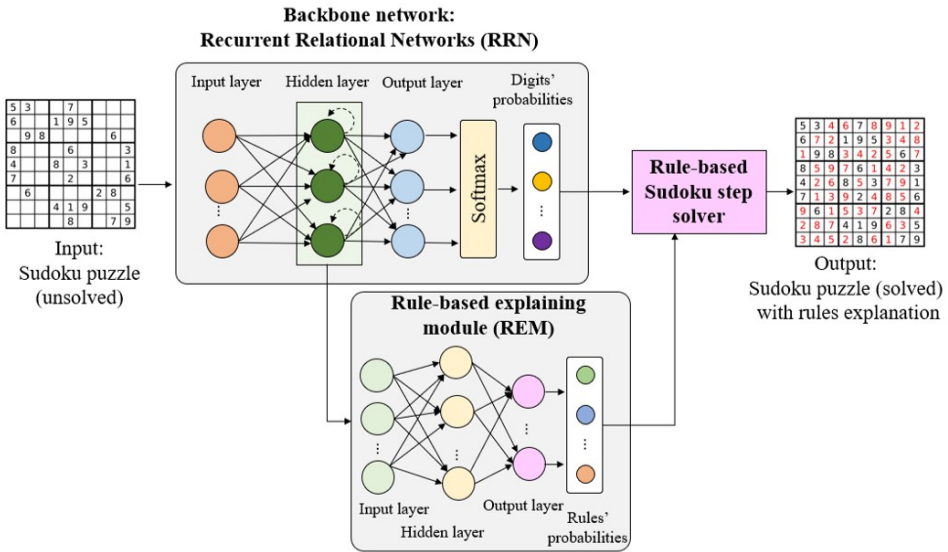


Fig. 2. Overview of the framework.

it challenging to comprehend their decision-making processes. Palm et al. introduced an RRN-based Sudoku solver that utilized a convolutional neural network (CNN) for feature extraction and an RRN for capturing relational dependencies [17]. However, the adoption of neural networks in Sudoku solving has raised concerns about transparency and interpretability, prompting the exploration of explain ability modules.

To address the lack of transparency in RRN-based Sudoku solving, we leveraged rule-based explanation techniques [21], inspired by prior research demonstrating their high accuracy. This integration enhances transparency and interpretability by generating human-readable rules that unveil the model’s decision-making process, these rules offer a more accessible way to understand the process compared to examining the raw model parameters.

3. Proposed method

Our proposed architecture incorporates the Recurrent Relational Network (RRN) [17] and Rule-based Explaining Module (REM), aiming to provide comprehensive, step-by-step explanations of the decision-making procedures in the context of Sudoku puzzle solving as shown in Fig. 2. The process begins by inputting an unsolved Sudoku puzzle

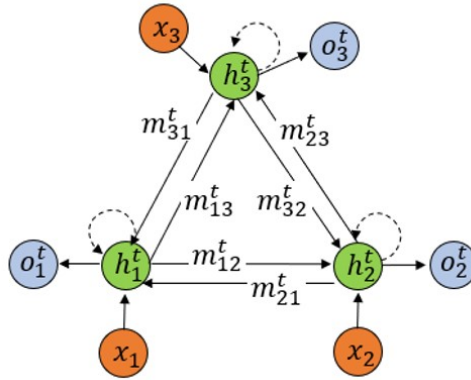


Fig. 3. A Recurrent Relational Network structure with 3 nodes.

zle into the backbone network, the Recurrent Relational Network (RRN). Each input node represents a feature vector (orange circles) corresponding to an individual Sudoku cell. Subsequently, a multi-layer perceptron (MLP) captures patterns and dependencies within the data. Recurrent computation updates all relevant information for each hidden state (green circles). The RRN then outputs probabilities for digits, representing the possible candidate digits for each cell. Following this, all hidden states (green circles) from the RRN are forwarded to the Rule-based Explaining Module (REM) using a multi-layer perceptron. This perceptron maps the hidden states to rules and outputs the probability of selected rules, offering explanations for the decision-making process. Finally, the Rule-based Sudoku step solver module receives output from both the RRN and REM modules. It identifies conditions that trigger specific rules and updates the knowledge base accordingly. This iterative process continues until the Sudoku puzzle is solved, ultimately providing both the solution and explanations for the decision-making steps involved.

3.1. Recurrent relational network (RRN)

RNN is a type of artificial neural network designed to capture long-range dependencies in sequential data. It is able to do this by learning to pass messages between nodes in a graph, which represents the relationships between the elements of the data. RRN is a powerful tool for a variety of tasks, including natural language processing, machine translation, and question answering. The RRN backbone consists of four main components, which are data input, message passing, node hidden state, and the output result as shown in Fig. 3.

3.1.1. Data input

In the context of Sudoku puzzles, there are input nodes, denoted as $i = 1, 2, \dots, 81$, each corresponding to a cell in the Sudoku grid. Each node i possesses an input x_i , representing the feature vector at that specific node.

3.1.2. Message passing

Message passing is responsible for communicating information between nodes. Each node sends a message to each of its neighbours at each iteration of the RRN. The message is a vector of numbers calculated based on the node's current state and its relationship to its neighbour. At each time step t , each node processes a hidden state vector h_i^t . During this process, each node sends a message m_{ij}^t from node i to node j at time step t , where node j represents a neighbouring node, utilizing message function f as illustrated in formula (1)

$$m_{ij}^t = f(h_i^{t-1}, h_j^{t-1}). \quad (1)$$

In Recurrent Relational Network (RRN), the message function f is implemented as a multiple-layer perceptron (MLP), enabling the network to learn the most effective types of messages to send for each situation. To incorporate all relevant information, each node must process all incoming messages, which are then summed together using formula (2). The combination of the MLP and the summation of messages enables RRN to learn complex patterns of communication and information exchange, making them powerful tools for solving tasks that require relational reasoning:

$$m_j^t = \sum_{i \in N(j)} m_{ij}^t, \quad (2)$$

where $N(j)$ represents all neighbouring nodes of node j , comprising nodes in the same row, column, and block as node j . Consequently, messages are currently computed for each node, allowing the model to progress to the next step in updating the network.

3.1.3. Recurrent nodes updates

Recurrent nodes are responsible for storing and updating the network's state, which represents the network's current understanding of the data. The state of a recurrent node is updated based on the messages it receives from its neighbours and the node's own internal state. The updates of recurrent nodes are key to the RRNs' ability to learn long-range dependencies in data. By repeatedly updating the state of each node based on the received messages, RRN can learn to capture the relationships between elements of the data that are separated by long distances in the input sequence. The formula for updating the state of a recurrent node is illustrated in (3):

$$h_j^t = g(h_j^{t-1}, x_j, m_j^t), \quad (3)$$

where g represents the node update function, functioning as a multiple-layer perceptron taking as input the hidden state from the previous iteration h_i^{t-1} , the feature vector of

input information x_j , and the message m_{ij}^t , the g function is trained to execute updates for the hidden state.

3.1.4. The output

After updating the hidden state, we can obtain the output at step t for node i by applying formula (4):

$$O_i^t = k(h_i^t), \quad (4)$$

where k denotes the output function, a multiple-layer perceptron trained to decode the hidden state into the output digit for the Sudoku. It converts the hidden state into output probabilities for a total of 10 different digits using the softmax function. The cross-entropy loss function, defined in formula (5), is used to optimize the model's performance during training. The target digits, represented by $y = y_1, y_2, y_3, \dots, y_81$ denote the correct digit at position i at step t .

$$l^t = - \sum_{i=1}^I \log O_i^t[y_i]. \quad (5)$$

3.2. Rule-based explaining module (REM)

Rule-based explaining is a technique in artificial intelligence employed to explain the reasoning behind decision-making by identifying the conditions that triggered specific rules and the conclusions reached by those rules. In our backbone network utilizing RRN, the message passing in the RRN network encompasses valuable information, including node relationships with its neighbours, which is highly valuable for examination. To extract the explanations from the message, we incorporate a multiple-layer perceptron that learns the rules from the hidden state after message passing and recurrent updating, as defined by formula (6).

$$R_i^t = r(h^t), \quad (6)$$

where R_i^t represents the output of the selected rules used to solve the Sudoku puzzle at step t . The hidden states h^t encompasses all of the RRN graph's hidden states, and the function r is a multiple-layer perceptron that maps the hidden state after message passing to rules at step t . The variable i represents the number of rules, where $i = 1, 2, \dots, n$. The selection of rules is guided by the tasks at hand. For Sudoku puzzles, we employ rules proposed by Hobiger [9] and Riley [22]. From their set of rules, we selected six rules for our experiments as they effectively solve the majority of the Sudoku puzzles in our dataset. Sudoku solving is divided into steps, with each step corresponding to filling in a single digit in the puzzle. Typically, multiple rules can be employed to determine a single digit. Consequently, the rule identification process generates more than one rule at each step. In this scenario, each Sudoku solving step can yield up to six different rule outputs.

| | | | | | | | | |
|-----------|-----------|---------|-------|---------|-----------|-------------|-----------------|-----------|
| 4 5 9 | 3 | 2 | 8 | 4 5 6 9 | 1 5 | 7 | 4 6 9 | 3 6 4 6 9 |
| 4 5 9 | 1 | 6 | 4 5 9 | 8 | 3 | 4 2 9 | 7 | 4 2 9 |
| 4 7 9 | 4 4 7 | 4 3 9 | 4 6 9 | 2 | 4 9 | 8 | 5 | 1 |
| 1 | 3 | 7 | 2 | 9 | 5 8 | 4 5 6 8 | 6 4 5 6 8 | |
| 4 5 6 8 9 | 4 5 6 8 9 | 4 5 9 | 7 | 3 | 1 5 8 | 4 2 5 6 8 9 | 1 2 6 4 5 6 8 9 | |
| 5 8 9 | 5 8 | 2 5 9 | 1 5 | 4 | 6 | 3 | 1 2 8 9 | 7 |
| 2 | 9 | 1 3 4 5 | 4 5 | 7 | 1 4 5 | 5 6 8 | 3 5 6 8 | |
| 7 5 7 | 5 5 | 3 | 8 | 6 | 2 5 9 | 1 | 4 | 2 3 5 9 |
| 4 5 6 8 | 4 5 6 8 | 4 5 | 3 | 1 5 | 1 2 4 5 9 | 7 | 2 6 8 9 | 2 5 6 8 9 |

Fig. 4. An example of Hidden Single.

3.3. Sudoku solving rules

We selected six rules from the rules proposed by Hobiger [9] and Riley [22]. These rules include Hidden Single, Naked Single, Locked Candidates Type 1, Locked Candidates Type 2, Naked Pair, and Hidden Pair. Although there are numerous rules beyond the six we chose, our decision was based on the observation that these specific rules already successfully solved over 98.00% of the most challenging Sudoku puzzles. In our study, our primary focus is on explaining the Sudoku solving process rather than improving accuracy. As such, we believe that utilizing these six rules is sufficient for our purposes.

3.3.1. Rule 1: Hidden single

A Hidden Single occurs when there is only one possible candidate number for a cell within a row, column, or 3×3 block, but that candidate number does not appear in any other cell within that row, column, or block. An example of the hidden single rule is presented in Fig. 4.

Examining row 3 (r3) in Fig. 4, it becomes evident that the cell at row 3, column 4 (r3c4) marked with a green 6, is the sole occurrence of the digit 6 within row 3. Consequently, we can confidently assign the digit 6 to cell r3c4 by eliminating other digit candidates.

3.3.2. Rule 2: Naked single

A Naked Single occurs when there is only one possible candidate number in a row, column, or 3×3 block that can contain a specific digit. An example of the naked single rule is presented in Fig. 5.

| | | | | | | | | |
|----------------------|----------------------|----------------------|------------------------------|----------------------------|------------------------|------------------------|--------------------|------------------|
| 4 | 1 | 2 | 7 | 3 | 6 | 5 | 8 | 9 |
| <small>3</small> | | <small>3</small> | <small>2 5</small> | <small>2 4</small> | <small>2 4 5</small> | <small>1 4</small> | <small>2</small> | <small>6</small> |
| <small>7 9</small> | <small>7 9</small> | <small>9</small> | | <small>4 8</small> | <small>4 5 9</small> | | | |
| 5 | 6 | 8 | | 1 | | 3 | 7 | |
| <small>3 6 9</small> | <small>4 9</small> | <small>4 6 9</small> | <small>8 5</small> | <small>4 7</small> | <small>2 1</small> | <small>4 7</small> | <small>3</small> | |
| 1 | <small>4 5 9</small> | <small>4 5 9</small> | <small>4 6 4 6 4</small> | <small>2 3 2 2 3</small> | <small>7 6 4 9</small> | <small>3</small> | | 8 |
| <small>2 3</small> | | 8 | 7 | <small>1 2 3 4 6</small> | <small>1 2 3 4</small> | <small>6 4 5 4</small> | <small>3 3</small> | |
| <small>2 9</small> | 3 | | | 7 | | 8 | 6 | 5 |
| 8 | <small>2 4</small> | <small>1 4 6</small> | <small>1 2 3 4 5 6 4</small> | <small>2 1 2 3 4 5</small> | <small>7 9 9 7</small> | <small>2 3</small> | <small>2 3</small> | |
| <small>2 6 7</small> | <small>2 5</small> | <small>5 6</small> | 9 | <small>2 6</small> | 8 | 4 | <small>2 3</small> | 1 |

Fig. 5. An example of Naked Single.

Examining the cell at r6c7 in Fig. 5, it has only one possible digit candidate, which is 6. Therefore, we can assign the digit 6 to that cell.

3.3.3. Rule 3: Locked candidates Type 1

The third and fourth rules are more advanced compared to the first two. They employ an indirect method for eliminating potential candidates from a cell. In fact, all rules, except for the first two, are utilized to eliminate possible candidates, eventually leading to the condition where the first two rules can be applied to fill in a digit and complete a step. Locked Candidates Type 1 occurs when all candidates of a specific digit within a block are confined to a row or column, that digit cannot appear outside of that block in that row or column. Fig. 6 illustrates an example of Locked Candidates Type 1.

Observing block 1 (b1) in Fig. 6, digit 5 only appears in row 3 (r3). Consequently, there should not be another instance of digit 5 in row 3 outside of block 1. Therefore, the candidate 5 in cell r3c7 can be eliminated.

3.3.4. Rule 4: Locked candidates Type 2

Locked Candidates Type 2 is the opposite of Locked Candidates Type 1. It occurs when, in a row (or column), all candidates of a specific digit are confined to one block, allowing the elimination of that candidate from all other cells in that block. Fig. 7 provides an example of Locked Candidates Type 2.

Examining row 2 (r2) in Fig. 7, it is observed that all candidate positions for the digit 7 appear only within block 1 (b1). Consequently, the digit 7 must be present in

| | | | | | | | | |
|----------------|----------------|--------------|----------------|----------------|--------------|----------------|----------------|----------------------|
| 9 | 8 | 4 | ^{1 2} | ^{1 2} | ³ | ^{1 3} | ⁶ | ⁵ |
| ³ | ⁶ | 2 | 5 | ¹ | ³ | ^{1 3} | 4 | ^{7 8 9} |
| ⁷ | ⁷ | ⁵ | ⁵ | 1 | 9 | 4 | ⁵ | 2 |
| ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ |
| ⁵ | ^{4 5} | 6 | ¹ | 9 | 7 | 2 | 3 | ^{4 5 8} |
| ⁸ | ⁸ | ⁸ | ⁸ | ⁸ | ⁸ | ⁸ | ⁸ | ⁸ |
| ¹ | ^{4 5} | 3 | 6 | ¹ | 2 | ⁵ | ^{7 8} | ^{4 5 7 8 9} |
| ^{7 8} | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ |
| 2 | ⁴ | 9 | ⁴ | 3 | 5 | 6 | 1 | ^{4 7 8} |
| ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ |
| 1 | 9 | 5 | 7 | 6 | 8 | 4 | 2 | 3 |
| 4 | 2 | 7 | 3 | 5 | 1 | 8 | 9 | 6 |
| 6 | 3 | 8 | ⁴ | ² | ² | 9 | 7 | 5 |
| ⁴ | ⁴ | ⁴ | ⁴ | ⁴ | ⁴ | ⁴ | ⁴ | ⁴ |

Fig. 6. An example of Locked Candidates Type 1.

| | | | | | | | | |
|----------------|----------------|--------------|----------------|----------------|--------------|--------------|--------------|--------------|
| 3 | 1 | 8 | ² | ² | 5 | 4 | ² | 6 |
| ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ | ⁷ |
| ^{4 5} | ^{4 5} | ⁷ | 6 | ⁴ | 3 | 8 | 1 | ² |
| ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ |
| ⁴ | ⁴ | ⁷ | 6 | ^{1 2} | 8 | ¹ | ² | 3 |
| ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ |
| 8 | 6 | 4 | 9 | 5 | 2 | 1 | 3 | 7 |
| 1 | 2 | 3 | 4 | 7 | 6 | 9 | 5 | 8 |
| 7 | 9 | 5 | 3 | 1 | 8 | 2 | 6 | 4 |
| ⁴ | ⁶ | 3 | ^{1 2} | 5 | ⁴ | ⁶ | 7 | ⁸ |
| ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ |
| ⁴ | ⁶ | ⁴ | ^{1 2} | ^{1 2} | ⁴ | ⁶ | 7 | ³ |
| ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ | ⁹ |
| ² | ⁵ | ² | ² | ² | 3 | 9 | 6 | 4 |
| ⁷ | ⁸ | ⁷ | ⁸ | ⁸ | ⁸ | ⁸ | ⁸ | ⁸ |

Fig. 7. An example of Locked Candidates Type 2.

row 2 within block 1. As a result, the digit 7 candidates located outside row 2 in block 1 can be eliminated.

3.3.5. Rule 5: Naked pair

A Naked Pair occurs when there are exactly two candidate numbers for a cell within a row, column, or 3×3 block, and those two candidate numbers also appear together in

| | | | | | | | |
|---|---------|---------|---------|-----|-------|-----|-------|
| 7 | 5 6 | 1 6 | 8 | 4 9 | 1 2 5 | 3 | 2 5 |
| 9 | 2 8 | 1 3 5 | 1 | 3 5 | 4 7 | 4 7 | 6 |
| 4 | 5 3 1 3 | 2 6 7 | 1 5 | 8 9 | | | |
| 6 | 4 2 | 7 8 3 | 9 | 5 1 | | | |
| 3 | 9 7 | 4 5 1 | 6 | 2 8 | | | |
| 8 | 1 5 | 6 9 2 | 3 | 4 7 | 4 7 | | |
| 2 | 7 8 | 4 5 1 6 | 7 8 | 9 3 | | | |
| 1 | 7 | 3 9 7 | 4 5 7 | 8 | 4 5 7 | 6 | 4 5 7 |
| 5 | 3 6 7 8 | 3 6 9 | 3 2 9 7 | 4 | 2 7 8 | 1 | 2 7 |

Fig. 8. An example of Naked Pair.

another cell within the same row, column, or block. This means that those two candidate numbers must be in these two cells, and cannot be appear elsewhere in that row, column, or block. Fig. 8 provides an example of naked pair.

Examining row 8 (r8), candidates 3 and 9 form a pair within a cell. Consequently, the candidate 3 in cell r8c2, (row 8 in this case), can be eliminated

3.3.6. Rule 6: Hidden pair

A Hidden Pair occurs when there are two candidate numbers for a cell within a row, column, or 3x3 block, and these two candidate numbers also appear together in another cell within that same row, column, or block. However, that other cell is already filled with another number. Consequently, all other candidates in those two cells can be eliminated. Fig. 9 provides an example of hidden pair.

Examining column 9 (c9), we observe a pair of candidate digits, 1 and 9, located in cells row 5 column 9 (r5c9) and row 7 column 9 (r7c9). Since 1 and 9 must occupy either of these two cells, any other candidate digits, such as the possible candidate 6 in r5c9, can be eliminated.

3.4. Rule-based sudoku step solver

Many Sudoku solving programs commonly eliminate candidates based on the given puzzle and search through all possibilities to identify the correct candidate digit. In contrast, our Sudoku solver takes a distinct approach. It solves Sudoku in a stepwise, rule-based, using specific rules in each action. The flow chart depicting our approach is presented

| | | | | | | | | | | |
|----------------|---|---|----------------|---|---|----------------|----------------|----------------|----------------|--------------|
| ^{5 6} | 4 | 9 | 1 | 3 | 2 | ^{7 6} | ⁵ | ^{7 8} | ^{7 8} | ⁶ |
| ^{5 6} | 8 | 1 | 4 | 7 | 9 | ^{2 3} | ^{2 3} | ² | ⁶ | ⁶ |
| | 3 | 2 | 7 | 6 | 8 | 5 | 9 | 1 | 4 | |
| ^{4 2} | 9 | 6 | ³ | 5 | 1 | 8 | ⁴ | ^{2 3} | ² | |
| ^{1 4} | 7 | 5 | ³ | 2 | 8 | ^{1 3} | ^{4 6} | ³ | ¹ | ⁶ |
| ^{1 2} | 3 | 8 | ^{7 9} | 4 | 6 | ^{1 2} | ² | | | 5 |
| | 8 | 5 | 3 | 2 | 6 | 7 | ¹ | ⁴ | ¹ | ⁶ |
| | 7 | 1 | 2 | 8 | 9 | 4 | 5 | 6 | 3 | |
| | 9 | 6 | 4 | 5 | 1 | 3 | ² | ² | ² | ² |

Fig. 9. An example of Hidden Pair.

in Fig. 10. The initial step involves assigning possible candidate digits to each cell in the provided Sudoku puzzle. Subsequently, we apply Sudoku rules by examining all potential candidate digits to identify any patterns that conform the established rules. As mentioned in the previous section, the first two rules form the foundation for solving Sudoku puzzles, and we can observe that the Sudoku puzzle can be solved by applying these two rules alone. Following the assessment of the first two rules, the remaining rules are examined one by one to determine if any patterns meet the criteria of each rule. If a pattern satisfying a rule is discovered, we revisit all the rules to ensure no other patterns exist. This process iterates until the step is resolved by either the first or second rule. Conversely, if no pattern satisfying the rule is found, that step cannot be solved, and the solver will cease attempting to solve it. In other words, the Sudoku puzzle cannot be solved within the framework of these six rules.

4. Experimental results

We conducted our experiments using a DGX station with a Nvidia V100 GPU with 32 GB of GPU RAM, employing a batch size of 128 and a learning rate of 2e-5. The training process involved 32 steps because the model stabilized at this step and took 5 days. The total number of trainable parameters was 518006. For testing, we used both the Minimum Sudoku dataset from Gordon Royle and 1 million Sudoku games (1M Sudoku) dataset [18]. Additionally, the Minimum Sudoku dataset [23] was used for training, divided into an 80% training set, a 10% validation set, and a 10% testing set.

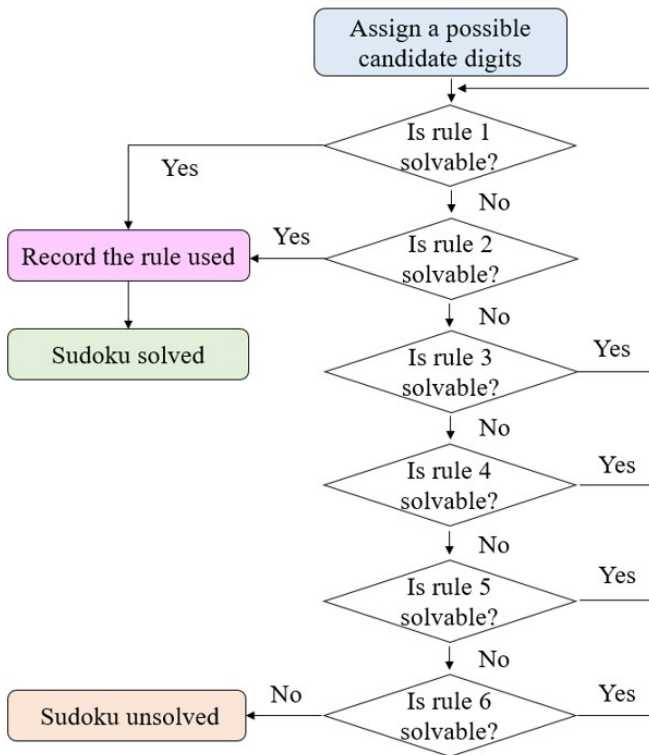


Fig. 10. Flow chart of the Rule-based Sudoku step solver.

Our method demonstrated the ability to solve Sudoku puzzles at a speed of 20 puzzles per second.

4.1. The Minimum Sudoku dataset from Gordon Royle

The Minimum Sudoku dataset [23] comprises 49 151 puzzles, each assigned the difficulty level of 17 given numbers. These puzzles are generated using a backtracking algorithm and is guaranteed to have a unique solution. Leveraging the complexity of these Sudoku puzzles with 17 given numbers during training enables our model to effectively handle a wide range of puzzles, from easy to difficulty levels.

We investigated the model's ability to apply six Sudoku-solving rules independently: Naked Single, Hidden Single, Locked Candidate Type 1, Locked Candidate Type 2, Naked Pair, and Hidden Pair. By testing the model on the Minimum Sudoku dataset,

we aimed to determine how effectively the model can apply each rule to solve Sudoku puzzles by applying 64 steps. The results are correctly verified using the Norvig Sudoku solver [15].

The results of solving the Sudoku puzzle are presented in Tab. 1. This table illustrates that, initially, the model assigns possible number candidates for the puzzle, enabling Sudoku solution with an accuracy of 72.32%. Subsequently, the model employs Rule 1, achieving a Sudoku accuracy of 99.00%. For cells in Sudoku puzzles that remain unsolved after Rule 1, the model applies Rule 2, achieving an accuracy of 98.98%. If any cells persistently resist resolution with Rule 2, the model turns to Rule 3, and so forth, up to Rule 6. The accuracy rates for Rules 3 to 6 are 98.62%, 98.60%, 98.79%, and 98.67%, respectively.

4.2. The 1 million Sudoku games (1M Sudoku) dataset

The 1M Sudoku dataset [18], available on Kaggle was developed by Kyubong Park. Using a computer program, he generated over a million Sudoku puzzles with their corresponding solutions. The dataset encompasses a variety of difficulty levels, ranging from easy to challenging. While several factors can influence a Sudoku puzzle’s difficulty, such as the pattern of given cells, the puzzle’s symmetry, and the existence of hidden singles or doubles, the number of given cells is a crucial factor. Sudoku puzzles with the fewest given cells are generally considered to be more difficult. The majority of puzzles in this dataset are of medium difficulty. We conducted experiments with our model using the 1M Sudoku dataset, performing 64 steps on each puzzle.

The results of solving these Sudoku puzzles are presented in Tab. 2. The contents of this table demonstrates that, in the initial stage, the model assigns possible number candidates for the Sudoku puzzle, enabling a 95.41% success rate in solving Sudoku puzzles. Subsequently, the model employs Rule 1, achieving a perfect accuracy of 100%. For any remaining unsolved cells after applying Rule 1, the model employs Rule 2, also achieving a perfect accuracy of 100%. Since Rule 2 successfully solves all remaining

Tab. 1. The results of solving the Sudoku puzzle on the Minimum Sudoku dataset.

| Accuracy (%) | Rules used | Description |
|--------------|------------|--------------------------------|
| 72.32 | - | Model assigns possible numbers |
| 99.00 | 1 | Naked Single |
| 98.98 | 2 | Hidden Single |
| 98.62 | 3 | Locked Candidate Type 1 |
| 98.60 | 4 | Locked Candidate Type 2 |
| 98.79 | 5 | Naked Pair |
| 98.67 | 6 | Hidden Pair |

Tab. 2. The results of solving the Sudoku puzzle on the 1 million Sudoku games dataset.

| Accuracy (%) | Rules used | Description |
|--------------|------------|--------------------------------|
| 95.41 | - | Model assigns possible numbers |
| 100.00 | 1 | Naked Single |
| 100.00 | 2 | Hidden Single |
| 100.00 | 3 | Locked Candidate Type 1 |
| 100.00 | 4 | Locked Candidate Type 2 |
| 100.00 | 5 | Naked Pair |
| 100.00 | 6 | Hidden Pair |

cells, Rules 3 to 6 become unnecessary. Therefore, Rules 3 to 6 consistently exhibit 100% accuracy when applied. Refer to Tab. 4 for more details. Since our model was trained on the Minimum Sudoku dataset, renowned for its difficulty, it excels in solving Sudoku puzzles, achieving an outstanding 100% success rate.

4.3. Rule-based explanation

Our rule-based explaining module allows us to understand how the RRN model solves Sudoku puzzles by breaking it down step by step based on established rules and inferences. This is demonstrated through examples, such as inputting a Sudoku puzzle from the Minimum Sudoku dataset, where each cell's candidate number represents the probability of it being the correct answer. Fig. 11 depicts a graph showcasing solving accuracy at various steps ranging from 0 to 60 using the Minimum Sudoku dataset. Additionally, Fig. 12 displays a graph representing rule accuracy employed at different steps with the same dataset.

Tab. 3 provides a comprehensive analysis and interpretation of the rule accuracy applied at various steps in the Sudoku-solving process. In the initial stage, the model achieved its highest accuracy near step 32. The model employed rules 1 through 6 to solve the puzzle. This indicates that the Sudoku puzzle is significantly complex, requiring the use of more than two rules to achieve a solution.

In another instance, we utilized input data from the 1M Sudoku dataset. As depicted in Fig. 13, a graph illustrates solving accuracy at various steps, ranging from 0 to 60. Fig. 14 presents a graph illustrating rule accuracy at different steps, with the model achieving its highest accuracy around step 32. Accompanying these figures is Tab. 4, where rules 1 and 2 were employed to solve the puzzle, achieving 100% accuracy. This observation sheds light on why rules 3 to 6 consistently show 100% accuracy. The reason behind this is that the model does not anticipate the utilization of rules 3 to 6, resulting in their consistent correctness as unused rules.

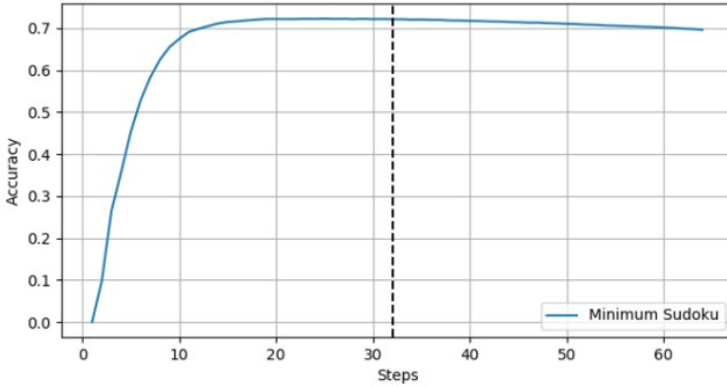


Fig. 11. A graph depicting solving accuracy at different steps using the Minimum Sudoku dataset.

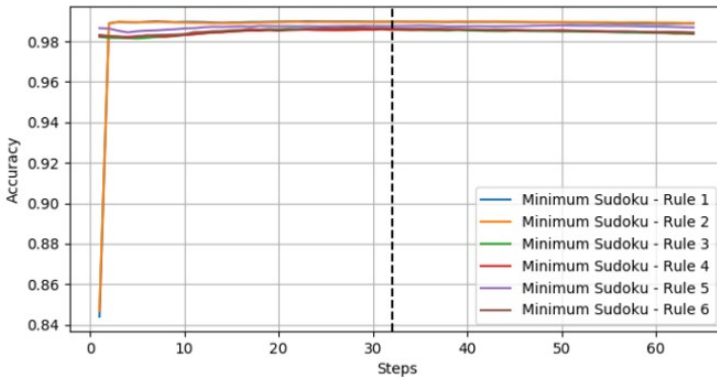


Fig. 12. Illustrating rule accuracy at different steps using the Minimum Sudoku dataset.

5. Conclusion

In this paper, we address concerns regarding the transparency and interpretability of machine learning applications, especially in critical decision-making domains. The opacity of neural networks, often labelled as black-boxes, has raised questions, particularly in Sudoku puzzle-solving scenarios. To tackle this challenge, we introduced the Rule-based Explaining Module (REM) as a tool to understand the complex decision-making processes of RRN during Sudoku puzzle-solving. While our REM has shown promise, there are opportunities for further exploration and improvement. Future research could explore broader applications of REM across diverse datasets. Additionally, extending

Tab. 3. The analysis and interpretation of rule accuracies across different steps in Minimum Sudoku dataset.

| Steps (1-64) | 1 | 16 | 32 | 48 | 64 | Best Step/ accuracy (%) |
|--------------------------------|-------|-------|--------------|--------------|-------|----------------------------|
| Model assigns possible numbers | 9.71 | 71.91 | 72.17 | 71.19 | 69.66 | 24/ 72.32 |
| Rule 1 | 98.89 | 98.96 | 98.98 | 98.94 | 98.88 | 6/ 99.00 |
| Rule 2 | 98.89 | 98.92 | 98.97 | 98.96 | 98.91 | 21/ 98.98 |
| Rule 3 | 98.18 | 98.54 | 98.56 | 98.51 | 98.37 | 20/ 98.62 |
| Rule 4 | 98.25 | 98.54 | 98.56 | 98.55 | 98.41 | 30/ 98.60 |
| Rule 5 | 98.63 | 98.72 | 98.76 | 98.79 | 98.69 | 40/ 98.79 |
| Rule 6 | 98.72 | 98.60 | 98.63 | 98.52 | 98.44 | 26/ 98.67 |

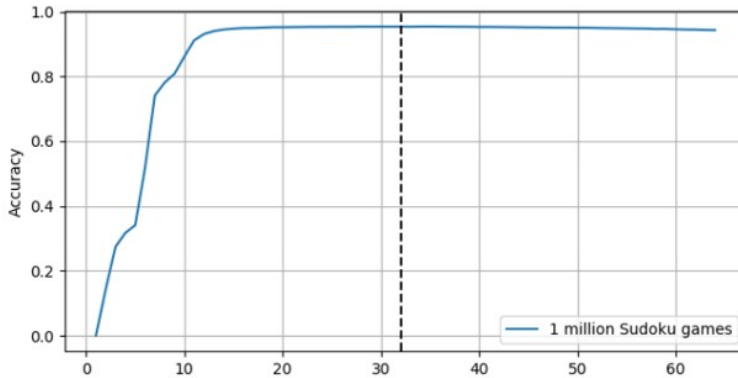


Fig. 13. An example of a graph depicting solving accuracy at different steps.

our approach to other puzzle types or complex decision-making tasks. The development of user-friendly interfaces and visualization techniques could facilitate the practical implementation of REM in real-world scenarios. This work represents a significant step in addressing transparency challenges posed by neural networks, offering a concrete solution in the form of the Rule-based Explaining Module. The success of our proposed method not only contributes to the field of explainable artificial intelligence but also paves the way for broader applications in various domains requiring interpretable decision-making.

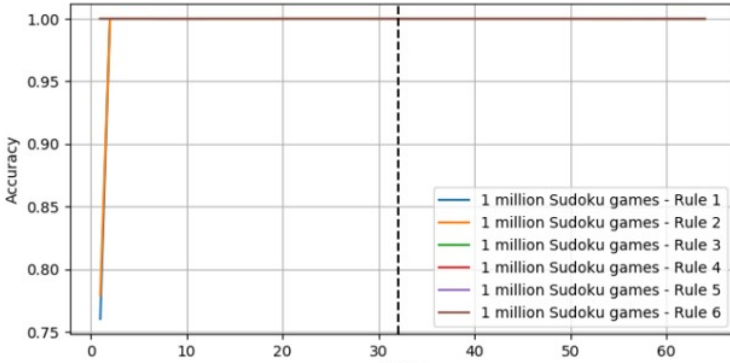


Fig. 14. Illustrating rule accuracy at different steps.

Tab. 4. An example of the analysis and interpretation of rule accuracies across different steps in the 1M Sudoku games dataset.

| Steps (1-64) | 1 | 16 | 32 | 48 | 64 | Best Step/ accuracy [%] |
|--------------------------------|---------------|--------|--------------|--------|--------|----------------------------|
| Model assigns possible numbers | 14.26 | 94.95 | 95.34 | 95.07 | 94.29 | 34/ 95.41 |
| Rule 1 | 100.00 | 100.00 | 99.99 | 99.99 | 99.99 | 1/ 100.00 |
| Rule 2 | 100.00 | 99.99 | 99.99 | 99.99 | 99.99 | 1/ 100.00 |
| Rule 3 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 0/ 100.00 |
| Rule 4 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 0/ 100.00 |
| Rule 5 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 0/ 100.00 |
| Rule 6 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 0/ 100.00 |

References

[1] Y. Björnsson, S. Helgason, and A. Pálsson. Searching for explainable solutions in Sudoku. In: *2021 IEEE Conference on Games (CoG)*, pp. 1–8. Copenhagen, Denmark, 17-20 Aug 2021. doi:10.1109/CoG52621.2021.9618900.

[2] B. Bogaerts, E. Gamba, and T. Guns. A framework for step-wise explaining how to solve constraint satisfaction problems. *Artificial Intelligence*, 300:103550, 2020. doi:10.1016/j.artint.2021.103550.

[3] K. N. Das, S. K. Bhatia, S. Puri, and K. Deep. Solving Sudoku puzzle by evolutionary algorithm. In: *Proc. 21st Asian Technology Conference in Mathematics*. Mathematics and Technology, LLC, Pattaya, Thailand, 14-18 Dec 2016. https://atcm.mathandtech.org/EP2016/contributed/4052016_21261.pdf.

[4] G. D. Engin, B. Aksoyer, M. Avdagic, D. Bozanli, U. Hanay, et al. Rule-based expert

- systems for supporting university students. *Procedia Computer Science*, 31:22–31, 2014. doi:<https://doi.org/10.1016/j.procs.2014.05.241>.
- [5] J. Espasa, I. P. Gent, R. Hoffmann, C. Jefferson, and A. M. Lynch. Using small MUSes to explain how to solve pen and paper puzzles. *ArXiv*, 2021. ArXiv:2104.15040. doi:[10.48550/arXiv.2104.15040](https://doi.org/10.48550/arXiv.2104.15040).
- [6] D. K. R. Gaddam, M. D. Ansari, and S. Vuppala. On Sudoku problem using deep learning and image processing technique. In: *Proc. 3rd Int. Conf. Communications and Cyber Physical Engineering (ICCCE) 2020*, vol. 698 of *Lecture Notes in Electrical Engineering*, pp. 1405–1417, 2020. doi:[10.1007/978-981-15-7961-5_128](https://doi.org/10.1007/978-981-15-7961-5_128).
- [7] O. Gerasimova, N. Severin, and I. Makarov. Comparative analysis of logic reasoning and graph neural networks for ontology-mediated query answering with a covering axiom. *IEEE Access*, 11:88074–88086, 2023. doi:[10.1109/ACCESS.2023.3305272](https://doi.org/10.1109/ACCESS.2023.3305272).
- [8] T. Guns, E. Gamba, M. Mulamba, I. Bleukx, S. Berden, et al. Sudoku assistant – an AI-powered app to help solve pen-and-paper Sudokus. In: *Proc. AAAI Conference on Artificial Intelligence*, p. 16440–16442. AAAI Press, 2023. doi:[10.1609/aaai.v37i13.27072](https://doi.org/10.1609/aaai.v37i13.27072).
- [9] B. Hobiger. Sudoku for Java – HoDoKu, 2013. <https://sourceforge.net/projects/hodoku/>, [Accessed: 15 Oct, 2023].
- [10] A. Hussain, S. U. Amin, H. Lee, A. Khan, N. F. Khan, et al. An automated chest X-ray image analysis for Covid-19 and pneumonia diagnosis using deep ensemble strategy. *IEEE Access*, 11:97207–97220, 2023. doi:[10.1109/ACCESS.2023.3312533](https://doi.org/10.1109/ACCESS.2023.3312533).
- [11] B. Indriyono, N. Pamungkas, Z. Pratama, E. Mintorini, I. Dimentieva, et al. Comparative analysis of the performance testing results of the backtracking and genetics algorithm in solving Sudoku games. *International Journal of Artificial Intelligence and Robotics*, 5(1):29–35, 2023. doi:[10.25139/ijair.v5i1.6501](https://doi.org/10.25139/ijair.v5i1.6501).
- [12] P. Linardatos, V. Papastefanopoulos, and S. B. Kotsiantis. Explainable AI: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2021. doi:[10.3390/e23010018](https://doi.org/10.3390/e23010018).
- [13] D. Macha, M. Kozielski, Ł. Wróbel, and M. Sikora. RuleXAI—A package for rule-based explanations of machine learning model. *SoftwareX*, 20:101209, 2022. doi:[10.1016/j.softx.2022.101209](https://doi.org/10.1016/j.softx.2022.101209).
- [14] N. Musliu and F. Winter. A hybrid approach for the Sudoku problem: Using constraint programming in iterated local search. *IEEE Intelligent Systems*, 32(2):52–62, 2017. doi:[10.1109/MIS.2017.29](https://doi.org/10.1109/MIS.2017.29).
- [15] P. Norvig. Solving every Sudoku puzzle, 15 Jan 2012. <https://norvig.com/sudoku.html>, [Accessed: 15 Oct, 2023].
- [16] E. Onokpasa, D. Bisandu, and D. Bakwa. A hybrid backtracking and pencil and paper Sudoku solver. *International Journal of Artificial Intelligence and Robotics*, 181(47):39–43, 2019. <https://dspace.unijos.edu.ng/jspui/handle/123456789/2769>.
- [17] R. B. Palm, U. Paquet, and O. Winther. Recurrent relational networks. In: *Advances in Neural Information Processing Systems 31 – Proc. 32nd Int. Conf. Neural Information Processing Systems (NeurIPS) 2018*, vol. 31 of *NIPS’18*, p. 3372–3382, 2018. <https://proceedings.neurips.cc/paper/2018/hash/b9f94c77652c9a76fc8a442748cd54bd-Abstract.html>.
- [18] K. Park. 1 million Sudoku games, 2017. <https://www.kaggle.com/datasets/bryanpark/sudoku>, [Accessed: 15 Oct, 2023].
- [19] X. Pengcheng, H. Zhenlin, Z. Liuqi, W. Ning, Z. Hanghang, et al. A realtime image recognition method of Power AI based on quadtree algorithm. In: *Proc. 2023 2nd Int. Conf. Innovation in Technology (INOCON)*, pp. 1–6. Bangalore, India, 3-5 Mar 2023. doi:[10.1109/INOCON57975.2023.10101145](https://doi.org/10.1109/INOCON57975.2023.10101145).

- [20] M. Prabha, S. Radha, P. M. Priya, and B. S. Dhivya. Sudoku solver using minigrid based backtracking algorithm. *International Journal of Research in Engineering, Science and Management*, 5(6):138–140, 2022. <https://journal.ijresm.com/index.php/ijresm/article/view/2180>.
- [21] G. P. Reddy and Y. V. P. Kumar. Explainable AI (XAI): Explained. In: *Proc. 2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pp. 1–6. Vilnius, Lithuania, 27–27 Apr 2023. doi:10.1109/eStream59056.2023.10134984.
- [22] G. Riley. CLIPS rule based programming language code, jan 2016. <https://sourceforge.net/p/clipsrules/code/HEAD/tree/branches/63x/examples/sudoku/>, [Accessed: 15 Oct, 2023].
- [23] G. Royle. Good at Sudoku? Here’s some you’ll never complete. *The Conversation*, 12 Feb 2012. [Accessed: 15 Oct, 2023]. <https://theconversation.com/good-at-sudoku-heres-some-youll-never-complete-5234>.
- [24] S. Shi, H. Chen, W. Ma, J. Mao, M. Zhang, et al. Neural logic reasoning. In: *Proc. 29th ACM Int. Conf. Information & Knowledge Management (CIKM '20)*, p. 1365–1374. Boise, ID, USA, 21–25 Oct 2020. doi:10.1145/3340531.3411949.
- [25] Y. Singh, P. Kumar, S. Goel, P. Garg, T. Srivastava, et al. Anuvadak: Language system using machine learning techniques. In: *Proc. 2023 Int. Conf. Artificial Intelligence and Smart Communication (AISC)*, pp. 742–745. Greater Noida, India, 27–29 Jan 2023. doi:10.1109/AISC56616.2023.10085373.
- [26] A. A. Suha Binta Wadud and M. Abdullah-Al-Wadud. An improved hybrid method combining backtracking with pencil and paper for solving sudoku puzzles. In: *Proc. Int. Symp. Electrical, Electronics and Information Engineering (ISEEIE) 2021*, p. 438–441. Association for Computing Machinery, Seoul, Republic of Korea, 19–21 Feb 2021. doi:10.1145/3459104.3459176.
- [27] Y. Tian. Artificial intelligence image recognition method based on convolutional neural network algorithm. *IEEE Access*, 8:125731–125744, 2020. doi:10.1109/ACCESS.2020.3006097.
- [28] P.-S. T. P.-S. Tsai, T.-F. W. P.-S. Tsai, J.-Y. C. T.-F. Wu, and J.-F. H. J.-Y. Chen. Integrating of image processing and number recognition in Sudoku puzzle cards digitation. *Journal of Internet Technology*, 23(7):1573–1584, 2022. doi:10.53106/160792642022122307012.

GUESSING QUANTUM STATES FROM IMAGES OF THEIR ZEROS IN THE COMPLEX PLANE

Maciej Janowicz , Andrzej Zembrzusi 

*Department of Applied Mathematics, Institute of Information Technology,
Warsaw University of Life Sciences – SGGW, Warsaw, Poland*

Abstract The problem of determining the wave function of a physical system based on the graphical representation of its zeros is considered. It can be dealt with by invoking the Bargmann representation in which the wave functions are represented by analytic functions with an appropriate definition of the scalar product. The Weierstrass factorization theorem can then be applied. Examples of states that can be guessed from the pictorial representation of zeros by both the human eye and, possibly, by machine learning systems are given. The quality of recognition by the latter has been tested using Convolutional Neural Networks.

Keywords: scientific visualization, zeros of wave functions, Bargmann representation, Weierstrass factorization theorem, Convolutional Neural Networks

1. Introduction

It is, indeed, difficult to imagine physical sciences and their development without such or that pictorial representations of reality including representations of *states* of a physical system. Even solving problems in elementary or high-school physics almost always demands creating some figures to grasp the intuitive essence of the exercise. The situation in quantum mechanics is not different. As a matter of fact, graphical representations of physical states are even more needed because of several counter-intuitive features of that theory.

A basic way to represent states of quantum systems is by providing their wave functions. In many cases it is inconvenient, or even meaningless, to use the most standard representation of the wave functions as square-integrable functions defined on the suitable Cartesian product of the sets of real numbers. One of many possibilities is to consider the Bargmann representation in which the wave function belongs to the set of entire functions in the complex plane (or Cartesian product of many such planes).

Taking into account a special – “stiff” – behavior of entire functions in the complex planes, and the presence of several powerful theorems which, in principle, allows to reconstruct of the whole function from its values at specific points, one can reasonably ask whether it is possible to guess the “nature” of the quantum states just from a picture showing their zeros in the plane. The answer is, in principle, negative for the reasons specified later, but the number of specific interesting cases in which at least a qualitative description of the state can be done is sufficient to warrant further consideration.

The main body of the work is organized as follows. In Section 2 major facts about the

quantum states in the Bargmann representation are provided. Section 3 offers several examples of graphical representations of zeros of analytic wave functions that allow a trained human eye to easily guess what the quantum state is with a minimum of additional knowledge. Section 4 contains the description of the artificial neural network used in the paper and the results of its application. Some concluding remarks are contained in Section 5.

2. Quantum states in the Bargmann representation

After one of the major scientific revolutions which took place in years 1925-1935, quantum mechanics has been established as the most fundamental conceptual framework of the whole physics. Among its basic concepts is that of the wave function. A wave function is a complex function that represents the quantum state of a physical system. With the help of the wave function, we can calculate the probabilistic properties of the system, especially expectation values of observable physical quantities.

In the simplest case the wave function (also called state function or state vector) is a square-integrable complex function of four real variables including the time that serves as a parameter. The same wave function can, however, be represented in many different ways. A remarkable representation that is invoked in this work in which wave functions are holomorphic functions of complex variables is called *Bargmann* or *Bargmann-Segal* representation [2, 3, 7]. The state vectors are entire (i.e. holomorphic in the whole complex plane) functions of the complex variable z that are of the order one, that is, they cannot grow faster than

$$\exp(A|z|)$$

as the modulus $|z|$ goes to infinity with A being a constant.

One can define an important quantity called the scalar product of two functions $\psi(z)$ and $\phi(z)$ as follows:

$$\langle \phi(z) | \psi(z) \rangle = \frac{1}{\pi} \int \bar{\phi}(z) \psi(z) \exp(-|z|^2),$$

where we have restricted ourselves to functions of one complex variable z , and $\bar{\phi}(z)$ denotes the function that is complex-conjugated to $\phi(z)$. The integration is to be performed over the entire complex plane.

The quantity

$$Q(z) = \frac{1}{\pi} |\psi(z)|^2 \exp(-|z|^2)$$

is called (Husimi) Q -function [8]. It is sometimes used for visual representations of the quantum states as well as computation of expectation values of the so-called anti-normal products of operators.

The Bargmann representation is particularly well suited to describe the quantum aspects of electromagnetic fields in resonators or cavities. In such systems, it is very convenient to represent the quantized fields as system modes of oscillations (called just modes). If for some reason, we can restrict ourselves to precisely one such mode, the wave function in the Bargmann representation can be reduced to that of just one complex argument. If that single distinguished mode of electromagnetic oscillations is populated with just n quanta (photons in this case), the wave function is of the form:

$$\psi(z) = \psi_n(z) = \frac{1}{N} z^n,$$

where N is a normalization factor to ensure that the scalar product of $\psi_n(z)$ with itself is equal to 1. In the above simple case, we have $N = \sqrt{\pi n!}$.

The above state functions are called *Fock states*. These states are the most elementary ones. Remarkably, it is rather difficult (though possible) to obtain them in real-life experiments.

However, in many cases, the state functions are sums of the above elementary states. For instance,

$$\phi(z) = \frac{1}{N} (1 + z^2 + (1/2)z^3)$$

is a valid quantum state, being a *coherent superposition* of three Fock states. The first term, 1, corresponds to the state with zero photons (vacuum), the second – to the state with two photons, and the third – to the state with three photons. The coefficients (complex in the generic case) are weights with which elementary states contribute to the whole wave function.

Another important class of states consists of exponential functions of the form:

$$\psi_\alpha(z) = \exp(\alpha z),$$

where α is a complex number. They are called *coherent states*. The electromagnetic field produced by a laser is close to being in a coherent state. The light of a laser has excellent coherence properties; this justifies the name.

Let me now invoke the Weierstrass factorization theorem that allows one to build a holomorphic function from the location of its zeros.

To begin with, we follow Weierstrass as quoted by Rudin, and introduce the following *elementary factors* [6]:

$$E_p(z) = (1 - z) \exp\left(z + \frac{z^2}{2} + \dots + \frac{z^p}{p}\right).$$

Their only zero is at $z = 1$. They are all close to 1 if $|z| < 1$ and p is large.

Then the following holds.

Theorem (Weierstrass). *Let f be an entire function, suppose that $f(0) \neq 0$, and let z_1, z_2, z_3, \dots be the zeros of f , listed according to their multiplicities. Then there exists an entire function g and a sequence $\{p_n\}$ of nonnegative integers, such that*

$$f(z) = e^{g(z)} \prod E_{p_n} \left(\frac{z}{z_n} \right).$$

Now, if f has a zero of the order k at $z = 0$, the above theorem applies to $f(z)/z^k$ so that the *non-zero* value at $z = 0$ is not an essential assumption. Unfortunately, the above factorization is *not* unique.

In the case of entire functions of the order q , the Weierstrass results has been strengthened by Hadamard:

Theorem (Hadamard). *If f is an entire function of finite order ρ and m is the order of the zero of f at $z = 0$, f admits a factorization:*

$$f(z) = z^m e^{g(z)} \prod_{n=1}^{\infty} E_{p_n} \left(\frac{z}{z_n} \right).$$

3. Examples of quantum states and pictorial representations of their zeros

Let me describe here what one can expect from the pictorial representations of the zeros of the (Bargmann-space) state functions. We can look for those zeros not in the whole plane but only inside a contour C . In what follows below we consider C to be a circle with the radius R .

In the case of a pure Fock state with an arbitrary photon number n , one can see only a single mark at $z = 0$ for any n . Unfortunately, using just one picture we cannot distinguish between various ns .

In the case of a coherent superposition of Fock states, i.e., a polynomial of the n th order in z , we can have up to n different zeros and n different marked points in the picture.

In the case of coherent states, the picture is *empty* because the simple exponential function has no zeros in the complex plane.

However, a superposition of the Fock and coherent states has, in the generic case, infinitely many zeros. Thus, with growing R we will see more and more points (and marks) to appear within the contour. What is more, the number of photons populating the most populated Fock state in the superposition can be easily guessed from the picture.

In this Section several examples of pictorial representations of the zeros of wave functions in the Bargmann representation are provided in Figs. 1-8.

Let us first observe that it would be quite difficult and sometimes impossible to find out the zeros of the wave function from the corresponding density given by the Q -function. Thus, the left-hand side of each of the figures has an independent value.

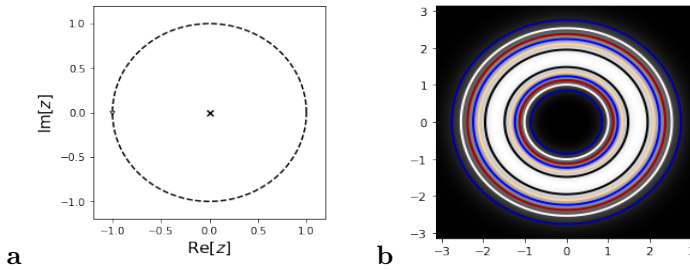


Fig. 1. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = z^3$. (a) Zeros of the wave function; (b) Q -function.

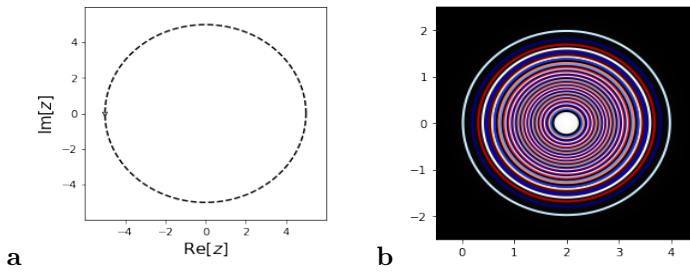


Fig. 2. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = \exp(2z)$. (a) Zeros of the wave function; (b) Q -function.

And let us now consider the principal problem stated in the previous section. Can we get at least some quantitative knowledge about the quantum state by just having a look at the left-hand sides of the figures contained in this section? Let us say that we know in advance that they show zeros of some state functions. In Fig. 1a we can

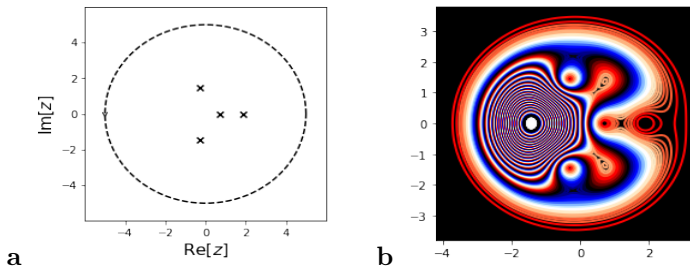


Fig. 3. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = z^4 - 2z^3 + 2z^2 - 5z + 3$. (a) Zeros of the wave function; (b) Q -function.

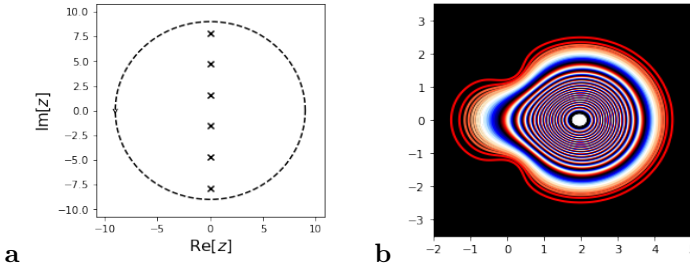


Fig. 4. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = 1 + \exp(2z)$. (a) Zeros of the wave function; (b) Q -function.

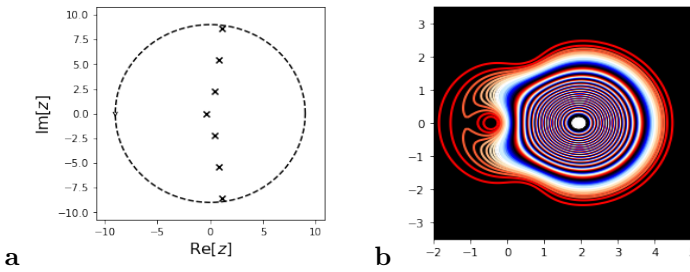


Fig. 5. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = z + \exp(2z)$. (a) Zeros of the wave function; (b) Q -function.

immediately infer that very likely, the state has just a single zero at $z = 0$. This means that it must be a Fock state with n photons though we cannot know what the number n is equal to. In Fig. 2a we see that a state has no zeros at all. Though there are, in principle, uncountably many such functions, an exponential one is clearly distinguished

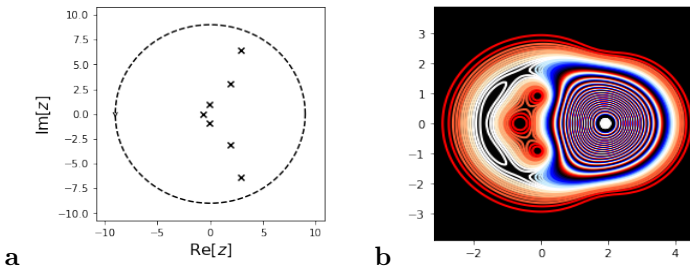


Fig. 6. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = z^3 + \exp(2z)$. (a) Zeros of the wave function; (b) Q -function.

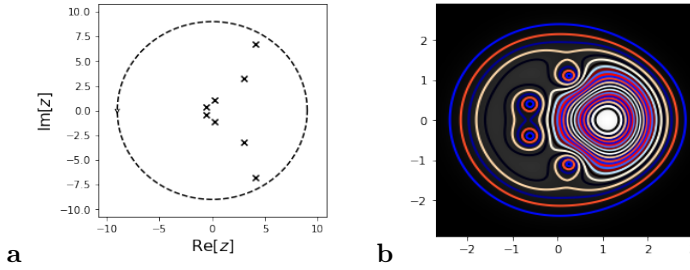


Fig. 7. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = z^4 + \exp(2z)$. (a) Zeros of the wave function; (b) Q -function.

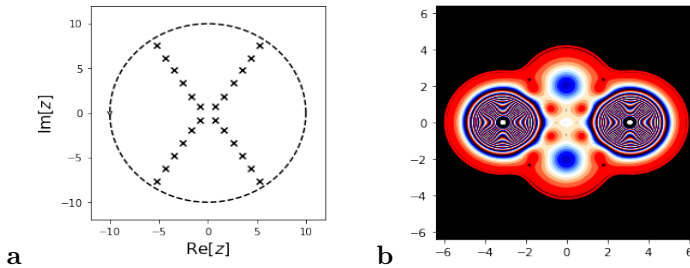


Fig. 8. Zeros and density in the complex plane associated with the (unnormalized) wave function $\psi(z) = 1 + z^2 + \cos(2z) + (1/4) \cos(3.15iz)$. (a) Zeros of the wave function; (b) Q -function.

by the physical context. Thus, we may reasonably suspect that we have to do with a coherent state (though we cannot say too much about the factor multiplying z in the exponent). Fig. 3a shows four zeros. If we can change the radius of the contour and there is no change in the number of zeros contained within it, we can claim with a reasonable margin of error that the state is a superposition of Fock states and the state which contributes the largest amount of quanta has precisely four of them. Looking at Figs. 4a-7 we can suspect that, on this occasion, the number of zeros is infinite (again, we would have to change the radius of the contour to confirm this). The difference in the number of photons carried by the Fock-state component is very nicely mirrored in the figures, though, in this case, the right-hand sides of the figures would be very helpful to establish the probable nature of the superposition. Finally, the X-like nature of the distribution of zeros in Fig. 8a suggests that a special combination of coherent states containing sines and/or cosines are involved.

4. Classification of states using convolutional neural networks

To check whether automatization of the recognition of quantum states based on the pictures of their zeros, we have decided to employ a convolutions neural network to solve the following classification problem.

We have generated pictures showing zeros of two families of functions.

The first family (denoted by (A)) has been of the form:

$$W_i(z) = \sum_{j=0}^K a_{i,j} z^j,$$

with $i = 0, 2, \dots, N - 1$. Several values of N has been tried from $N = 100$ to $N = 10000$. The degree of the polynomials K has been set equal to 5. All the coefficients $a_{i,j}$ have been real random numbers uniformly sampled from the interval $(0, 1)$. Physically, all $W_i(x)$ s correspond to a superposition of Fock states.

The second (B) family of functions and corresponding pictures of their zeros has been given by:

$$P_i(z) = \sum_{j=0}^K b_{i,j} z^j + c_i \exp(2z),$$

where again i runs from 0 to $N - 1$. The coefficients $b_{i,j}$ and c_i have again been real random numbers uniformly sampled from the interval $(0, 1)$. Physically, this function corresponds to a superposition of Fock states and a coherent state.

For both families of coefficients, $3N$ plots have been created with circular marks of zeros for three radii of circles within which the zeros have been determined. Those radii have been chosen equal to 3, 6, and 9. All together we generated $2 \cdot (3N)$ pictures.

Thus, our neural network has been trained to solve binary, balanced classification problems to distinguish between two families of triples of pictures corresponding to (A) and (B).

The training set has consisted of 80% of pictures, that is, $0.8 \cdot 2 \cdot 3N = 4.8 \cdot N$, the test set has had $1.2 \cdot N$ pictures.

The number of epochs in training the network has been 20. The architecture of the network has been illustrated in Fig. (9).

Despite this, the results are quite encouraging as can be seen from Figures (10, 11) as well as from the report contained in Figure (12).

The standard metrics took satisfactory values even for N as small as 300. The network has had no difficulties in classifying the plots.

In addition, we have also generated a family (denoted by (C)) similar to $P_i(z)$, but this time containing the sine function:

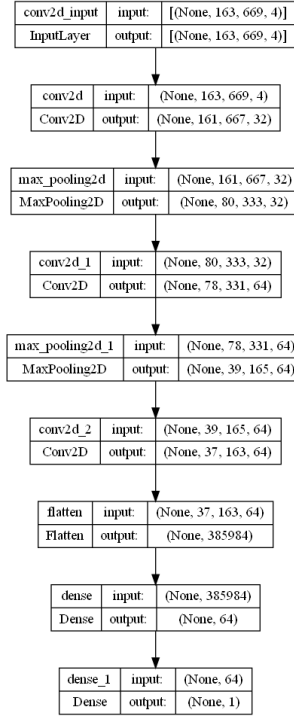


Fig. 9. Architecture of the convolutional neural network used in this work.

$$R_i(z) = \sum_{j=0}^K b_{i,j} z^j + c_i \sin(2z).$$

From the quantum-optical point of view such a (wave) function pertains to a superposition of several Fock states and two coherent states.

We have trained the convolutional neural network to solve the classification problem for triples of pictures belonging to (A) or (C).

The results for the $R_i(z)$ have been less impressive but they seem still to be quite satisfactory as can be seen from Figs. (13, 14, 15)

as well as from the report contained in Figure (12)

There is no need to stress that the above results have a rather preliminary character serving as a kind of “proof of concept”. Much more complex classification problems dealing with multiple classes and using more advanced classifiers will have to be considered.

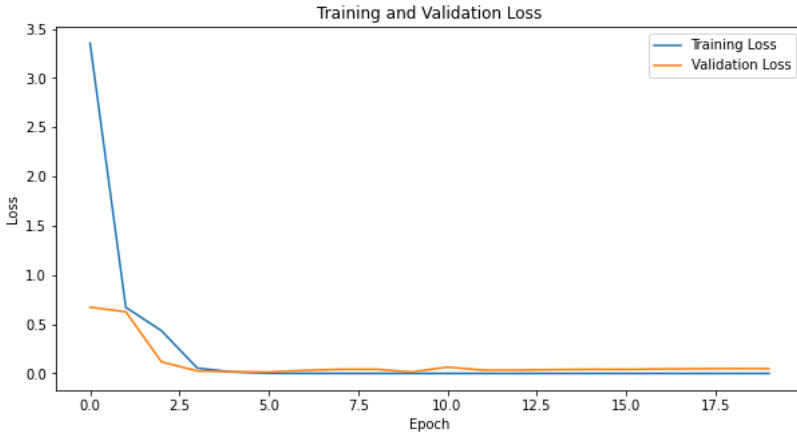


Fig. 10. Dependence of the loss function on the training epoch for the classification of images generated by zeros of the functions $W_i(z)$ and $P_i(z)$.

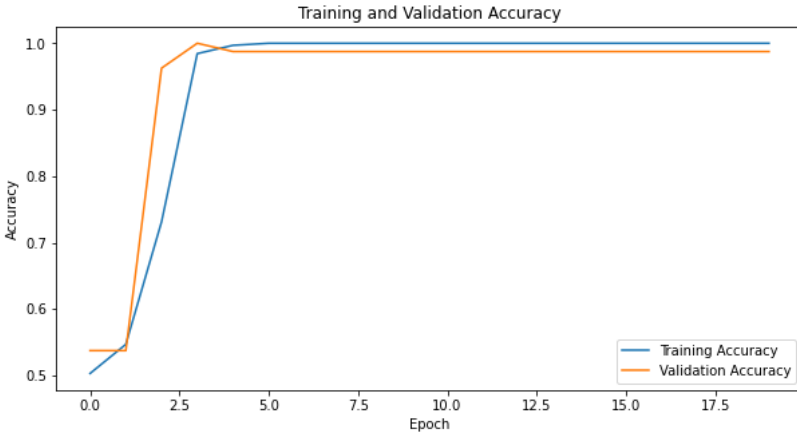


Fig. 11. Dependence of the classification accuracy on the training epoch for the images generated by zeros of the functions $W_i(z)$ and $P_i(z)$.

What is more, semi-supervised and/or self-supervised learning methods will likely be of advantage in the context of graphical recognition of quantum states in the context of quantum tomography [1, 9].

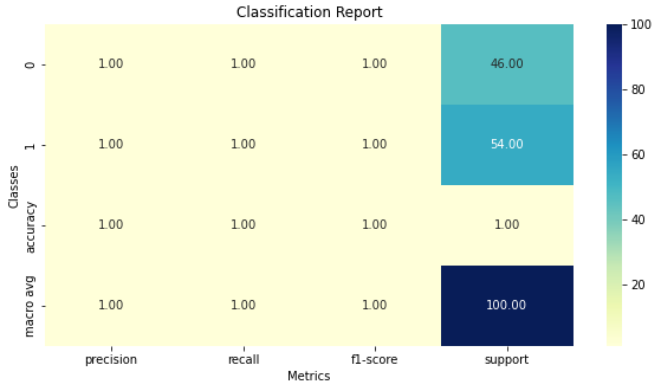


Fig. 12. Classification report for the images generated by zeros of the functions $W_i(z)$ and $P_i(z)$.

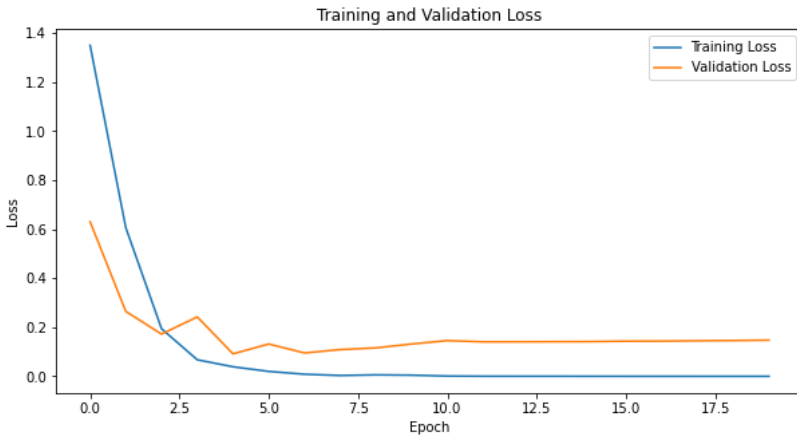


Fig. 13. Dependence of the loss function on the training epoch for the classification of images generated by zeros of the functions $W_i(z)$ and $R_i(z)$.

5. Concluding remarks

An obvious question is whether we can find a similar representation of multi-mode states. Let us first consider, another quantity (more general than the wave function) that characterizes a quantum system. This is the density matrix. For a single mode, the density matrix depends on two complex variables: $\rho = \rho(w^*, z)$. Of particular interest is often the *diagonal part* of the density matrix, $\rho(z^*, z)$. This means that, of course, ρ is not a holomorphic function and we have no reconstruction theorems like that of Weierstrass

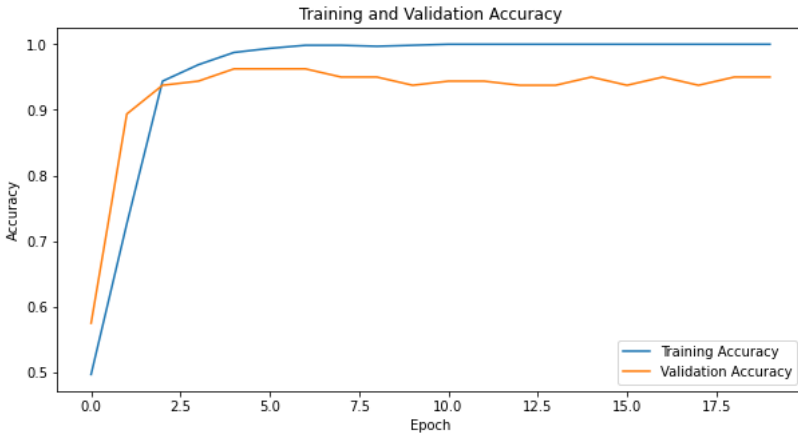


Fig. 14. Dependence of the classification accuracy on the training epoch for the images generated by zeros of the functions $W_i(z)$ and $R_i(z)$.

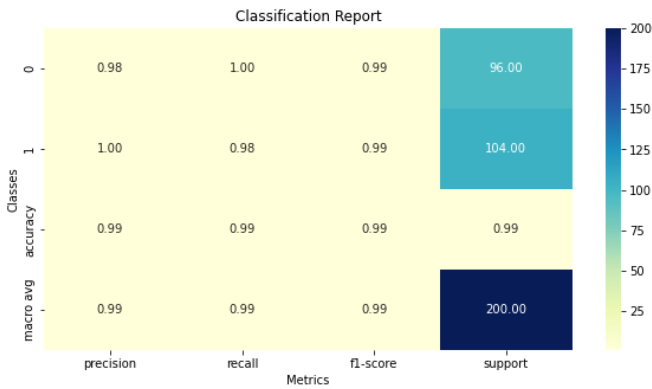


Fig. 15. Classification report for the images generated by zeros of the functions $W_i(z)$ and $R_i(z)$.

or Hadamard in our disposal. Still, zeros of such diagonal elements (being the zero of Q -functions) can be obtained and plotted to serve as a pictorial representation of the state.

Now, if we have a multi-mode wave function, then at least two routes are possible depending on the number of modes. If this number is equal to 2, we can obtain an interesting parallel with the knot theory [5]. Indeed, suppose (unnormalized) $\psi(z_1, z_2)$ is equal to, say, $z_1^2 + z_2^3$. In that case, the zeros satisfy the equations $z_1^2 = -z_2^3$ so that they say, $z_1^2 + z_2^3$, then In that case, the zeros satisfy the equations $z_1^2 = -z_2^3$ so that they

are located on a complex curve (there are no isolated zeros of holomorphic functions of many variables). The common part of this curve with a sphere S^3 forms a simple knot. Thus, in a certain sense, the knots can also serve as a two-dimensional generalization of the pictures in Figs. 1a-8a. Finally, let us mention here that there is a pretty obvious analogy of the zeros of multi-mode quantum states and the Calabi-Yau surfaces. They are known for their very complicated behavior and any visualization methods applied to them would be helpful or even desirable. One line of tackling this problem has been outlined in [4].

References

- [1] S. Ahmed, C. S. Muñoz, F. Nori, and A. F. Kockum. Classification and reconstruction of optical quantum states with deep neural networks. *Physical Review Research*, 3(3):033278, 2021. doi:10.1103/PhysRevResearch.3.033278.
- [2] J. C. Baez, I. E. Segal, and Z. Zhou. *Introduction to Algebraic and Constructive Quantum Field Theory*. Princeton University Press, 1992.
- [3] V. Bargmann. On a Hilbert space of analytic functions and an associated integral transform part I. *Communications on Pure and Applied Mathematics*, 14(3):187–214, 1961. doi:10.1002/cpa.3160140303.
- [4] A. J. Hanson. A construction for computer visualization of certain complex curves. *Notices of the American Mathematical Society*, 41(9):1156–1163, 1994.
- [5] J. Milnor. *Singular points of complex hypersurfaces*. Princeton University Press, 1969.
- [6] W. Rudin. *Real and Complex Analysis*. McGraw Hill, Boston, USA, 3rd edn., 1987.
- [7] I. E. Segal. Mathematical problems of relativistic physics. In: M. Kac, ed., *Proc. Summer Seminar*, vol. II of *Lectures in Applied Mathematics*. American Mathematical Society, Boulder, Colorado, USA, 1960. Chapter VI.
- [8] D. Walls and G. J. Milburn. *Quantum Optics*. Springer, Berlin, 2nd edn., 2008.
- [9] T. Xin, S. Lu, N. Cao, G. Anikeeva, D. Lu, et al. Local-measurement-based quantum state tomography via neural networks. *NPJ Quantum Information*, 5(1):109, 2019. doi:10.1038/s41534-019-0222-3.

THE DEVELOPMENT OF A GENERATIVE APPROACH FOR JOINT SUPER-RESOLUTION IMAGE RECONSTRUCTION FROM HIGHLY SPARSE RAW DATA IN THE CONTEXT OF MR-PET IMAGING

Krzysztof Malczewski 

*Institute of Information Technology, Warsaw University of Life Sciences – SGGW, Warsaw, Poland
(krzysztof_malczewski@sggw.edu.pl)*

Abstract The present study introduces a rapid and efficient approach for reconstructing high-resolution images in hybrid MRI-PET scanners. The application of sparsity, compressed sensing (CS), and super-resolution reconstruction (SRR) methodologies can significantly decrease the demands of data acquisition while concurrently attaining high-resolution output. G-guided generative multilevel networks for sparsely sampled MR-PET input are shown here. Compressed Sensing using conjugate symmetry and Partial Fourier methodology speeds up data collection over k-space sampling methods. GANs and k-space adjustments are used in this image domain technique. The employed methodology utilizes discrete preprocessing stages to effectively tackle the challenges associated with the deblurring, reducing motion artifacts, and denoising of layers. Initial trials offer contextual details and accelerate evaluations. Preliminary experiments provide contextual information and expedite assessments.

Keywords: GAN, WGAN, super-resolution, compressive sensing, medical modalities

1. Introduction

Commercial PET-MRI imaging equipment with synergistic capabilities debuted in 2010. According to source [34], they have a competitive advantage. Clinical imaging is improved by hybrid imaging technology. MR/PET combines MRI soft tissue morphology with PET functional imaging. These advances are driven by how well different imaging modalities provide correlated, not duplicative, findings. The effective outcome was achieved by integrating functional imaging from Positron Emission Tomography with CAT's soft tissue analysis, two oncological technologies. The use of ^{18}F -FDG with CAT scanners has been recognized in relevant research. Medical imaging using FDG-PET can identify and quantify malignant cells' metabolic rate. In therapy management, CT scans can detect even the smallest wounds that PET scans may miss due to their limited range or technological restrictions. Respiration, locomotion, and circulation are typical causes. The soft tissue contrast of MRI is well-known.

The main argument favors MRI over CT. This approach helped treating neurological problems, brain tumors, craniofacial defects, abdominal wall masses, mass-like lesions, and other conditions. Despite following MRI protocols to the letter, emission tomography is effective. PET and MRI are compatible, as indicated in reference [1]. PET and CT use different radiation wavelengths and can be combined to improve their efficacy. PET

and MRI image capture methods differ. MR-images may affect PET signal acquisition because they require a visually appealing and steady field. The above method meets medical image processing requirements. Photomultipliers cannot detect PET signals in strong magnetic fields. To overcome this limitation, a mobile table connects magnetic resonance (MR) and positron emission tomography (PET) scanners in different places. Supine patients undergo PET and MR imaging without movement. The current architectural design prevents simultaneous collecting of unprocessed data, prolonging diagnostic procedures and increasing patient problems. The integration of MRI-PET may solve this obstacle. According to references [35, 36], super-resolution techniques have improved medical image processing.

Deep learning algorithms can accurately replicate complicated relationships between low-resolution and high-resolution pictures, even under demanding situations, advancing Single Image Super-Resolution (SISR). Image quality improved after enhancing. Structured Convolutional Neural Networks (CNNs) help Super-Resolution Convolutional Neural Networks (SRCNNs) and their accelerated variations provide better Single Image Super-Resolution results for two-dimensional natural images. This phenomena is observed in sources [4, 28].

Patch, edge, sparse coding, prediction, and statistics have been conventional algorithm groupings for decades. These methods cost less than deep learning to compute. Deep learning has improved convolutional neural network use, advancing super-resolution. Despite deep-learning advances, medical picture super-resolution remains unsolved. Medical imaging uses 3D volumes. CNNs used to ignore the input's three-dimensional structure. Because 3D models require more memory and compute computational power than 2D models, their usefulness is limited. Convolutional neural networks (CNNs) optimize pixel or voxel-level error, measured by mean squared error (MSE) between the predicted model's output and a high-resolution reference. Research in [46] suggests that using MSE and PSNR as metrics for assessing picture accuracy may be unreliable. Mean Squared Error enhancement reduces only picture sharpness and perceptual accuracy.

Generative Adversarial Networks (GANs) have gained significant popularity and are extensively employed in many applications such as image super-resolution, modality switching, and synthesis. The aforementioned domains have been extensively examined in the literature [19, 24, 27].

The utilization of 3D Multi-Level Densely Connected Super-Resolution Networks (mDCSRN) has the potential to address the aforementioned issues. A highly linked network reduces the weight of a mDCSRN [5].

Enhancing intensity difference optimization increases model size and speed while preserving performance. GAN training improves system efficacy, according to research. A common deep learning neural network architecture includes a generator (\mathbb{G}) and a discriminator (\mathbb{D}). The generator and discriminator compete to minimize the difference

between generated and actual data during training. The Generative Adversarial Network was introduced by Goodfellow and colleagues in 2014 [11].

Super-Resolution computer vision applications use Generative Adversarial Networks and GANs with adversarial and perceptual loss functions are designed to perform picture Super-Resolution (SR). Superior textures are restored to lower-resolution images. The network can retrieve exact textures and high-frequency components. However, its scope is limited. Generational Adversarial Networks can modify data and introduce noise. Super-resolution and other methods were evaluated for the task of improving image quality [21, 41]. MRI distortions are generated by imaging plane motion. Motion is needed for super-resolution. A recent study suggests that convolutional neural networks can enhance medical image quality [6, 26, 37]. Researchers developed SRCNN [7], a deep convolutional network, for super-resolution reconstruction. CNNs were first used in Super-Resolution. Shi et al. introduced a sub-pixel convolutional layer as an alternative to the deconvolutional layer [42].

The training method becomes simpler. Simple linear network designs underpin the methods. The link between neural network depth and over-parameterization is growing. Previous research indicates that recursive networks can effectively handle difficulties by applying weights repeatedly [25, 44]. Increased network depth improves performance, but deeper networks are more prone to gradient outbursts. Hyun et al. utilized Convolutional Neural Networks and k-space rectification methods to replace missing k-space data regions with original data [18]. Thus, it is crucial to improve the effectiveness of mitigating the aliasing artifacts.

A primary constraint associated with Magnetic Resonance Imaging concerns the phase of the assessment. The expeditious acquisition of MRI data has garnered significant attention from a multitude of researchers. Improvement is necessary in the phase encoding intervals utilized during the sampling of k-space. This phenomenon typically leads to a decline in the visual accuracy of the image. The implementation of the proposed k-space sampling pattern would yield advantages in resolving the matter. As per the author's description, the procedure of populating k-space entails obtaining subsets along a designated phase encoding direction. The methodology utilized in this approach involves the utilization of blades that are similar to those found in a propeller. The implementation of Hermitian symmetry results in the halving of the complex space. This feature enables the retrieval of the missing k-space component. This methodology improves the understanding of components that occur frequently.

This paper elucidates a methodology based on Generative Adversarial Network that has been employed for the purpose of reconstructing Compressed Sensing Magnetic Resonance Imaging (CS-MRI), taking cues from previous studies. The methodology that has been put forth involves the amalgamation of Generative Adversarial Networks that are reliant on images, along with k-space corrections. The aforementioned methodology demonstrates enhanced efficacy in contrast to singular and non-sequential techniques for

correcting k-space. The current approach incorporates the fusion of deformable image registration and Generative Adversarial Networks, and has been extended to incorporate the fusion of multiple frames of visual data. The Wasserstein generative adversarial network (WGAN) was employed to optimize the algorithm's performance and promote model convergence during the training phase.

The results have been the subject of rigorous scrutiny in focused research investigations. The main aim of the methodology described in this manuscript is to improve the accuracy and quality of hybrid scanner images with regards to boundary demarcation, while also decreasing the time required for acquisition.

This work presents a novel algorithm that is suitable for use with MR/PET and integrates super-resolution, accurate estimation of movement, and streamlines the examination. The results have faced significant scrutiny in focused empirical inquiry. The primary objective of the methodology described in this manuscript is to enhance the precision and excellence of images concerning the identification of boundaries, while concurrently reducing the time required for acquisition (see Figure 1).

The primary findings of this work include the following:

1. The framework algorithm demonstrates a comprehensive approach towards the joint reconstruction process of MR-PET images. Various aspects such as sparse sampling trajectories, synchronizing of k subspaces, deblurring, noise reduction, motion compensation, and subsequently, increasing the resolution of an image are key areas of focus in this study.
2. The present study introduces a novel model for reconstructing MR-PET images using a generative super-resolution approach.
3. The methodology provided employs the joint sparsity of both the MR and PET modalities.
4. The sparsity of the MR and PET raw data has resulted in an acceleration of the input data collecting process.
5. The technique has been specifically intended for gathering visual information across different scales. This issue is often simplified by other authors.
6. The algorithm is capable of extracting visual features at different scales. This subject matter is frequently oversimplified by other writers.
7. The employed methodology entails distinct preprocessing stages to address the challenges of blur and noise removing layers.
8. The proposed technique employs a reconstruction strategy for magnetic resonance imaging that leverages convolutional neural networks. This approach aims to restore low-quality images derived from highly sparse raw data.
9. The aforementioned methodology employs the compressed sensing framework to prioritize the minimization of data acquisition durations.
10. The reconstruction layer of the procedure is nested with the author's deformable motion estimation procedure.

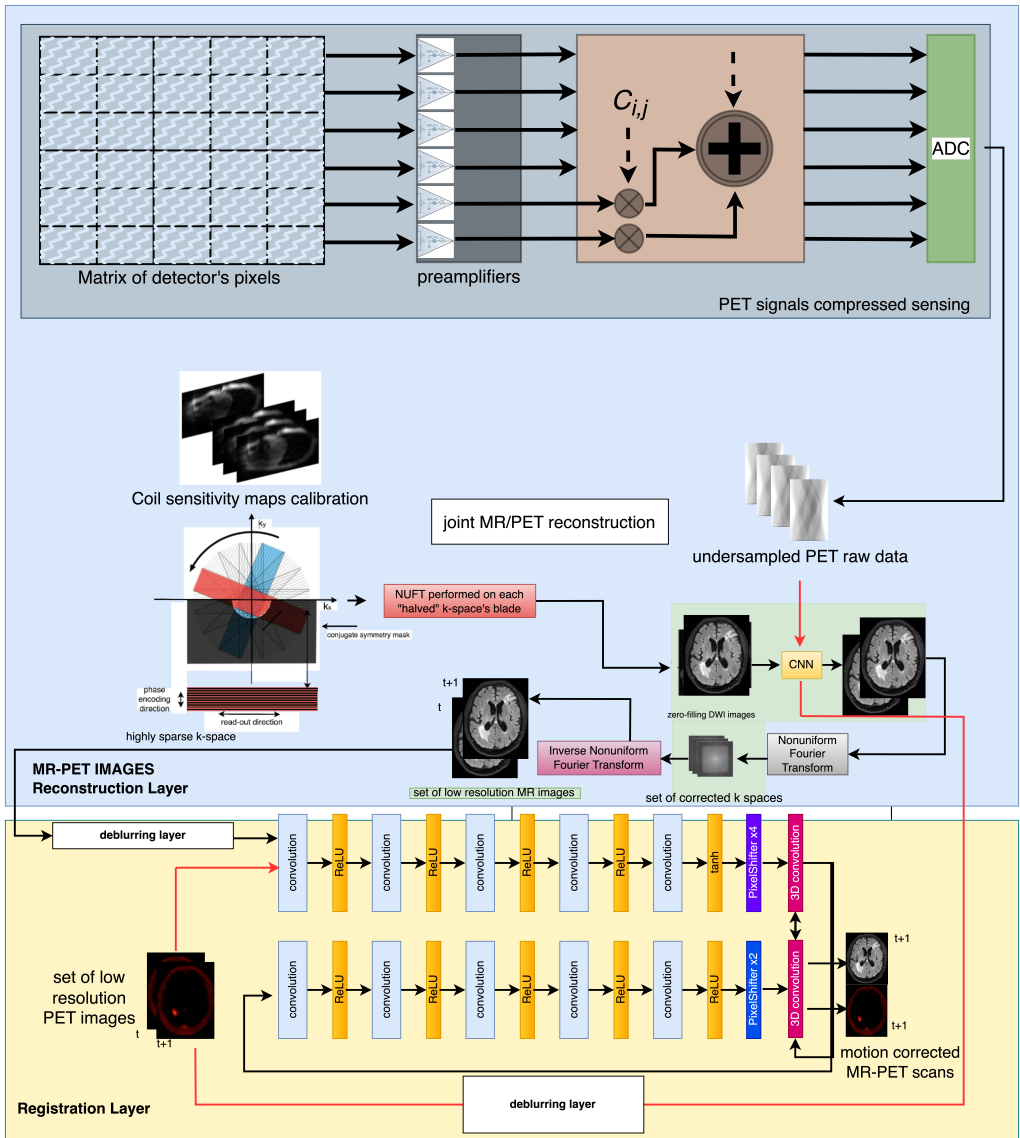


Fig. 1. The process for reconstructing low resolution MR-PET images. The method takes into account the lack of overlap between the MR and PET modalities. At its core, the deblurring net is nested. The raw PET signals are compressively detected in the upper part. The lower part

2. Joint sparseness on MR/PET

Although the modalities used have various physical bases, reconstruction often uses side-by-side projections. Despite the potential that both images use the same geometry, the assertion implies discrete image reconstruction processes. The reconstruction process can be simplified by sharing critical inter-technique data and recognizing similarities between objects [34]. Integrating this characteristic with other extracted structures may reduce motion abnormalities. Instead of reconstructing phases, the combined sparsity algorithm is used. It uses structural similarities to improve spatial resolution and eliminate involuntary patient movement during image capture by combining two sparse datasets. This algorithm solves the optimization problem within concurrent constraints.

In MRI and PET the Compressed Sensing in conjunction with Partial Fourier transform and the exploitation of conjugate symmetry have been used to induce sparsity in the data sets. The expression of joint sparsity can be formulated as follows:

$$\left\| \begin{matrix} \mathbb{S}(x_{\text{MRI}}^i) \\ \mathbb{S}(x_{\text{PET}}^i) \end{matrix} \right\|_2 = \sqrt{(\mathbb{S}(x_{\text{MRI}}^i))^2 + (\mathbb{S}(x_{\text{PET}}^i))^2}.$$

where x_{MRI}^i and x_{PET}^i are three-dimensional image volumes coming from MRI and PET, respectively. This equation involves the sparsifying transform \mathbb{S} applied to these image volumes. The regularization carried out at every voxel can be expressed in the following manner:

$$\Upsilon^i = \|\mathbb{S}(\text{in}M_{\text{MRI}}^i)\| - \|\mathbb{S}(\text{in}P_{\text{PET}}^i)\|,$$

where $\text{in}M_{\text{MRI}}^i$ and $\text{in}P_{\text{PET}}^i$ refer to MR and PET input data streams. The symbol Υ^i denotes joint sparsity regularization. The regularization parameters present in the process serve to mitigate the overlap of non-coherent features in MR and PET images.

The methodology presented in this scholarly article utilizes the concept of joint sparsity, specifically in the domains of MRI and PET, as illustrated in Figure 1.

3. Sparse sampling versus MR/PET raw data

The core method integrates and synthesizes data from multiple modalities. The current study compresses PET data volume, as cited in reference [34]. Positron-emitting radioactive elements are mixed whenever possible to reduce readout channels. Consolidating their output signals increased PET scan resolution. MR/PET hybrid scanners integrate super-resolution and compressive sensing through structural components. Sparse depiction of the detector's structure is obvious. The sparsity attribute can create new multiplexing setups. Random matrices with constrained isometry can be generated using several stochastic methods. The notion of greatest likelihood guides sensing matrix frameworks. Research indicates that creating detected matrices results in the lowest

reconstruction error in MS [30]. This scientific article describes a method that uses a limited number of channels to create spatial and temporal domains. PET input data is sparsely sampled. Each output can be interpreted mathematically as a linear combination of photodetector pixels with weights $c_{k,n}$ (refer to Figure 1). The number of sensors is lowered via 4:1 subsampling. MR-PET joint sparsity and shared product characteristics allow motion model parameters from MR data sets to improve PET image quality. Super-resolution is achieved using the same method as shown in Figure 1. This process used to enhance low-resolution MR and PET scans. This study uses PROPELLER, Poisson Disc, and Partial Fourier sampling. In this strategy, compactification inhibits signal recovery.

The encoding procedure uses 2D 3x3 convolutional layers iteratively, like CNNs. After layer processing, leaky rectified linear units, batch normalization, and 2x2 maximum pooling are used for downsampling.

This paper presents an approach to reconstruct low-resolution magnetic resonance pictures. K-space blades with high sparsity achieve this. To reduce data collection time, the sampling method reduces data density and uses a conjugate symmetric mask. To correct motion and blur, deblurring and registration layers improve low-resolution images.

U-net design was trained using the mean squared error loss function, which is mathematically represented by the statement: every zero-filled image is linked to a completely sampled image, denoted by S_{true} . Adam's optimizer, previously discussed in reference [48], is used to reduce the loss function. The study used a training rate of 0.0001 and ran the process for 100 epochs. Only 32 images were used for training. The hyperparameters were determined using empirical observations.

$$\beta^i = \begin{cases} \underset{\beta^i}{\operatorname{argmin}} \|S_{true} - f_{\beta^i}(|F^{-1}(y_0)|)\|, & i = 0; \\ \underset{\beta^i}{\operatorname{argmin}} \|S_{true} - f_{\beta^i} - f_{\beta^i}(S^i)\|, & \text{otherwise.} \end{cases}$$

4. The application of Generative Adversarial Networks within the framework of Super Resolution image reconstruction

The model framework that has been built is illustrated in Figure 1. The system consists of two components, specifically the deformable motion estimation and the reconstructing network. The second component consists of two blocks: a generating block and a discriminating block.

The effectiveness of the generative adversarial network framework in the domain of motion correction relies on its ability to improve picture restoration and support the reconstruction of missing raw data. The function indicated above have the power to

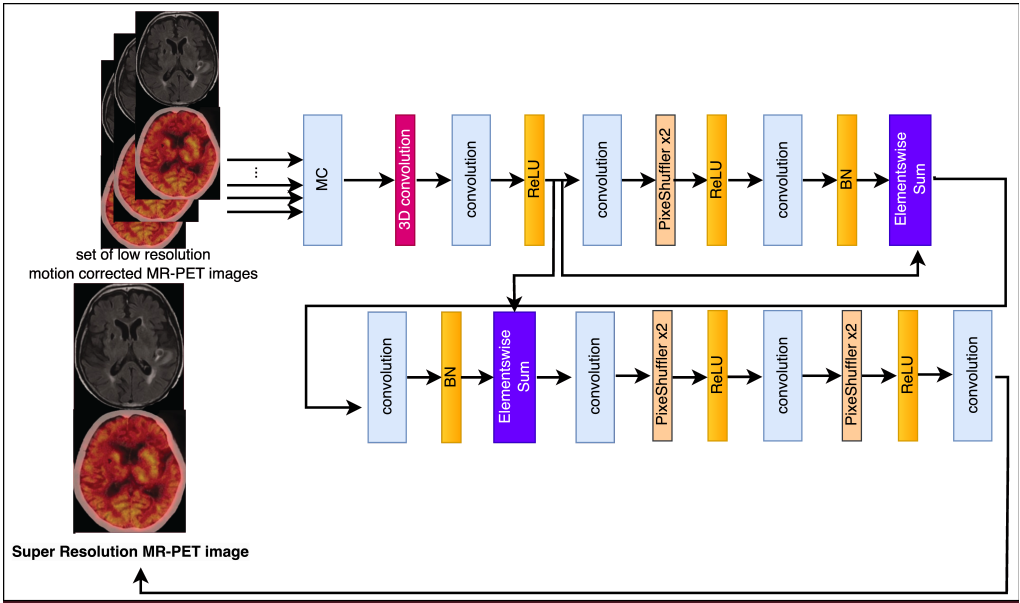


Fig. 2. The diagram illustrating the structure and flow of the generator net.

produce shots of exceptional quality. The main goal of the generator is to generate samples that have a significant level of resemblance to authentic data, while the discriminator aims to effectively categorize samples as either real or artificial.

$$\min_G \max_D V(D, G) = \min_G \max_D \mathbb{E}_x [\log (\mathbb{D}(x))] - \mathbb{E}_y [\log (1 - \mathbb{G}(y))].$$

The variables y and x represent motion deformed and corrected illustrations, respectively. With the exception of the core layer, encoder blocks are comprised of five convolutional layers and $\frac{n}{2}$ feature maps, each containing n mappings. The encoder blocks and decoder blocks share an architectural design, but transposed convolutions replace all convolutional layers. A method for estimating spatial transformation parameters is used in image registration technology, as detailed in [14]. Following this, the displacement discrepancy between frames is corrected. Displacement parameters can change the spatial location of frames in sequences that depict the same subject but were shot at different times and places.

Multiple pairings from the registration module are blended with I_{LR} frames using a 3D convolutional layer. The user-generated output is sent into the generator network. The study used a generator network design (denoted as \mathbb{G}) based on the SR-GAN architecture (see Figure 2). A single residual block is used in the \mathbb{G} -network to reduce the

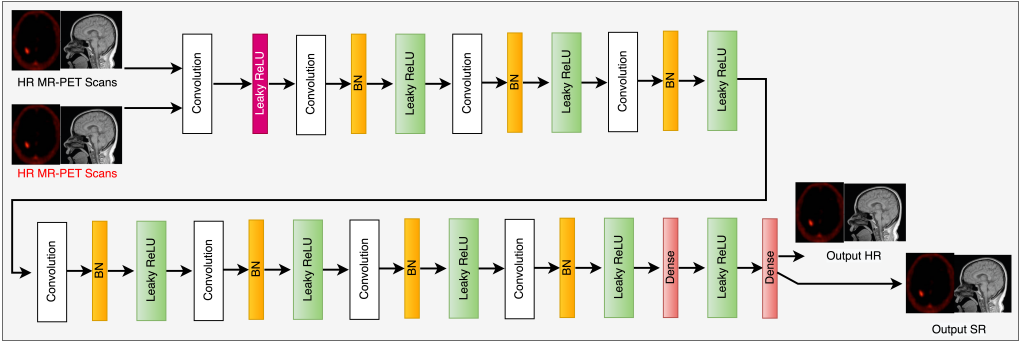


Fig. 3. The flowchart illustrating the architecture and functionality of the discriminator net.

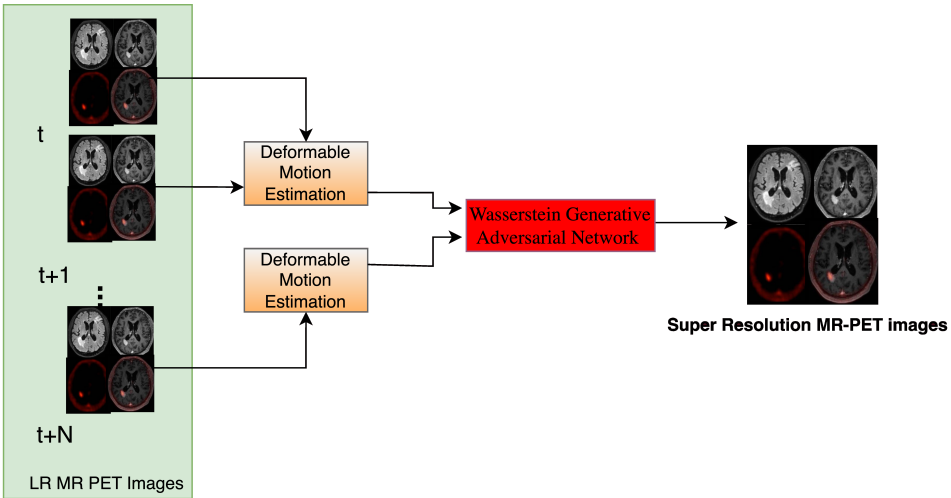


Fig. 4. The proposed magnetic resonance super resolution image reconstruction algorithm.

number of parameters and maintain generalization. To attain the required detail, the residual network uses two sub-pixel convolutional layers. The architecture of the discriminator, denoted as \mathbb{D} in Figure 3, consists of eight convolutional layers. As network levels rise, attributes correlate positively. Convolutional kernel reduction reduces feature dimensionality. Two modifications were made to address SR-GAN reconstruction and network training/convergence difficulties. In the initial phase, the discriminator \mathbb{D} ignored the *sigmoid* activation function in the output layer. In addition, parameter modifications were limited to a constant value of c (0.01) relative to their absolute

magnitude. The investigation focuses on insufficient security protocols during training and complex model convergence, as supported by references [16, 17, 29]. The anomaly is due to the low overlap between genuine and counterfeit distributions. Disregarding the statistical measure JS divergence, which compares distributions, may prevent network convergence. Arjovsky and colleagues found that the Wasserstein distance accurately measures the distribution separation even when overlap is low.

5. Methods used for the reconstruction of high-resolution MR-PET images

5.1. Reconstruction of the images and the loss function

The methodology commences by reconstructing the MR and PET images with low resolution, utilizing subspaces that have been inadequately sampled. Refer to [33] for this procedure.

The method uses blur, noise removing and motion estimation layers, with its main reconstruction process arranged as shown in Figures 2, 3, and 4.

The WGAN [16] shows that the Wasserstein distance improves confrontation network formation. The definition of Wasserstein distance is as follows:

$$\mathbb{W}(\mathbb{P}_{\text{ref}}, \mathbb{P}_{\text{gen}}) = \frac{1}{K} \substack{\text{sub} \\ \|\|f\|\|_{L \leq K} \\ \mathbb{W} \in \prod (\mathbb{P}_{\text{ref}}, \mathbb{P}_{\text{gen}})} \mathbb{E}_{(x,y) \sim \mathbb{P}_{\text{ref}}} [f(x)] - \mathbb{E}_{x \sim \mathbb{P}_{\text{gen}}} [f(x)].$$

The equation above uses the symbol $\prod (\mathbb{P}_{\text{ref}}, \mathbb{P}_{\text{gen}})$ to represent all possible joint probability distributions between \mathbb{P}_{ref} and \mathbb{P}_{gen} . The discrimination function of the adversarial network is $f_{\mathbb{W}}$, as shown in the equation. This limits the discriminator's input sample derivative to a predetermined range. The variable \mathbb{W} in the domain \mathbb{D} undergoes a modification procedure limited to the range of $-c$ to c . This technique emphasizes the gradient update generator and reduces the disappearing gradient problem. The function denoted by $f_{\mathbb{W}}$ satisfies the following equation:

$$L = \mathbb{E}_{x \sim \mathbb{P}_{\text{ref}}} [f_{\mathbb{W}}(x)] - \mathbb{E}_{x \sim \mathbb{P}_{\text{gen}}} [f_{\mathbb{W}}(x)].$$

As the variable L increases, it becomes feasible to estimate the Wasserstein distance between the probability distributions \mathbb{P}_{ref} and \mathbb{P}_{gen} through approximation. The former word refers to legitimate information diffusion, whereas the latter applies to synthesized information. The discriminator and generator loss functions are precisely specified as follows:

$$\begin{aligned} \mathbb{D}_{\text{loss}} &= \mathbb{E}_{x \sim \mathbb{P}_{\text{gen}}} [f_{\mathbb{W}}(x)] - \mathbb{E}_{x \sim \mathbb{P}_{\text{ref}}} [f_{\mathbb{W}}(x)], \\ \mathbb{G}_{\text{loss}} &= \mathbb{E}_{x \sim \mathbb{P}_{\text{gen}}} [f_{\mathbb{W}}(x)]. \end{aligned}$$

The training method is defined by the discriminator loss function, \mathbb{D}_{loss} . To evaluate GAN training, the Wasserstein distance between real and generated data distributions should decrease. Distance magnitude is negatively correlated with this measure.

The objective of this strategy is to enhance the learning procedure of the generator, represented as \mathbb{G} . The objective of this task is to evaluate the relationship between the input sequence I_t^{LR} (with values ranging from 1 to N) and its corresponding counterpart I_t .

The task was successfully accomplished by employing a feedforward Convolutional Neural Network. The neural network underwent training using the parameter $\Psi_{\mathbb{G}}$. The parameters of the neural network, denoted as $\Psi_{\mathbb{G}} = \{U_{1:L}; b_{1:L}\}$, with L layers, are obtained by minimizing the loss function $l_{\mathbb{G}}$ for the Super-Resolution generation network as described in reference [31]:

$$\Psi_{\mathbb{G}}^* = \underset{\Psi_{\mathbb{G}}}{\operatorname{argmin}} \frac{1}{N} \sum_{t=1}^N l_{\mathbb{G}}(\mathbb{G}_{\Psi_{\mathbb{G}}} I_t^{\text{LR}}, I_t^{\text{HR}}).$$

The current study utilizes a loss function, referred to as $l_{\mathbb{G}}$, that is based on previous scholarly research and has been appropriately acknowledged and cited in [38].

$$l_{\mathbb{G}} = l_{\text{MSE}} + 10^{-6} l_{\text{gen}}.$$

where l_{MSE} is defined by one of the equations below. The comprehensive net loss function of the SR-GAN model encompasses the loss functions of both the generating and discriminating blocks, denoted as $l_{\mathbb{G}}$ and $l_{\mathbb{D}}$, respectively.

$$l_{\mathbb{D}} = \frac{1}{N} \sum_{n=1}^N \left(\log(1 - \mathbb{D}_{\Psi_{\mathbb{D}}}(\mathbb{G}_{\Psi_{\mathbb{G}}} I_n^{\text{SR}})) \right) - \log(\mathbb{D}_{\Psi_{\mathbb{D}}}(I_n^{\text{HR}})).$$

The discriminator generator reconstruction equation is given. This generator, $\mathbb{G}_{\Psi_{\mathbb{G}}} I^{\text{SR}}$, rebuilds the original picture I^{HR} . The reconstructed images are denoted by $\mathbb{D}_{\Psi_{\mathbb{D}}}(\mathbb{G}_{\Psi_{\mathbb{G}}} I^{\text{SR}})$ and $\mathbb{D}_{\Psi_{\mathbb{D}}}(I^{\text{HR}})$. This variable represents the number of target pictures. The variables l_{MSE} and $l_{\mathbb{G}}$ are defined as follows:

$$l_{\text{MSE}} = \frac{1}{r^2 H W} \sum_{x=1}^W \sum_{y=1}^H (I_{x,y}^{\text{HR}} - \mathbb{G}_{\Psi_{\mathbb{G}}}(I^{\text{LR}})_{x,y})^2.$$

$$l_{\mathbb{G}} = \sum_{n=1}^N -\log \mathbb{D}_{\Psi_{\mathbb{D}}}(\mathbb{G}_{\Psi_{\mathbb{G}}}(I_n^{\text{LR}})).$$

The researchers added a registration loss component to the model's loss function to improve high-frequency texture information recovery. The expected difference between

spatial transformation calculations and observations is denoted as RLT. The major goal is to minimize complex information loss during geometric translation of consecutive frames. This method helps restore the HR scan. The RLT loss function is as follows.

$$\text{RLT} = \sum_{i=\pm 1} \|I'_{t+i}{}^{\text{LR}} - I_t{}^{\text{LR}}\|^2.$$

The equation described above represents the result obtained by applying the registration net to the picture $I_{t+i}{}^{\text{LR}}$. This process yields the image represented as $I'_{t+i}{}^{\text{LR}}$. The equation that expresses the length of the center of gravity (called also diameter in the Feret sense) of the pathological structure, symbolized as l_{loss} , is expressed in the following way:

$$l_{\text{loss}} = l_{\text{MSE}} + 10^{-6}l_{\mathbb{G}} + \varrho\text{RLT}.$$

The RLT weight coefficient, denoted by ϱ , has been assigned a value of 0.001 in accordance with the findings of the experiments. In relation to the notion of Wasserstein Generative Adversarial Networks (WGANs), it is feasible to omit the terms $l_{\mathbb{G}}$ and $l_{\mathbb{D}}$, leading to an adjustment of the loss function:

$$l_{\mathbb{D}} = \frac{1}{N} \sum_{n=1}^N \mathbb{D}_{\Psi_{\mathbb{D}}}(\mathbb{G}_{\Psi_{\mathbb{G}}}(I_n^{\text{SR}})) - \mathbb{D}_{\Psi_{\mathbb{D}}}(I_n^{\text{HR}}).$$

5.2. Registration of MR scans

The registration net is demonstrated using a multi-scale approach, which has been successful in traditional methods [34]. The procedure requires the target frame (I_t^{LR}) and the surrounding frame ($I_{t-R:t+R}^{\text{LR}}$) as input. Pyramidal registration is used to train spatial transformation parameters for image motion correction. Two sets of pictures are registered separately through the registration layer for a three-frame input. The parameters of the net are optimized by the minimization of the mean-squared error between the converted frames and the target frames. This parameter is denoted by $\omega_{\delta,t+1}^*$. This learning technique enhances the neural network's motion correction on the image dataset.

$$\omega_{\delta,t+1}^* = \underset{\omega_{\delta,t+1}}{\text{argmin}} \|I_t^{\text{LR}} - I'_t{}^{\text{LR}}\|^2.$$

The symbol $I'_t{}^{\text{LR}}$ represents the registration layer's result after the registration procedure.

The order is acknowledged. Figure 1 depicts the network layer setup for registration. Traditional methods for modeling deformable registration have been shown effective using a multi-scale framework, as shown in [8], [42], and [49].

This study uses a strategy to obtain a spanning tree with minimal aggregate edge costs. Nodes $i \in P$ represent distinct items, such as pixels or groups of pixels. The

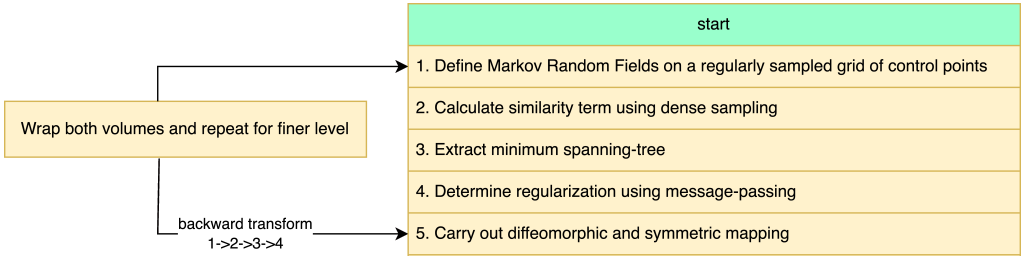


Fig. 5. The estimation of motion fields.

system links each node to a set of hidden labels representing motion fields, represented as $w_i^l = \{f_i^l, g_i^l, h_i^l\}$. The optimization-based energy function has two components: the data cost (S) and the pair-wise regularization cost $R(w_i^l, w_i^m)$, which applies to all nodes l connected to m .

$$E(w_i) = \sum_{j \in P} S(w_j^l) + \chi \sum_{l, m \in N} R(w_i^l, w_i^m). \tag{1}$$

The cost function estimates pixel similarity in two images. The parameter χ determines the influence of the regularization term and is used for weighting. The first element of the equation 1 is the data term, whereas the latter element is the regularization parameter.

The observed behavior is unaffected by adjacent entity displacements. The variable χ is used for weighting and determines the influence of the regularization term. In (1), the first component represents the data term and the second represents the regularization parameter.

5.3. The MR-PET images blur removal net

The work aims to restore a clear and accurate image, I_S , from a blurred image, I_B , without knowing the blur kernel. The deblurring process uses a convolutional neural network (\mathbb{G}_{ρ_G}), also known as the Generator. An estimation determines the best I_S image for each I_B value. In addition, the critic network (\mathbb{D}_{ρ_D}) is included in the training phase, and both networks engage in adversarial training. Integration of content and adversarial losses creates the composite loss function:

$$\mathcal{L} = \mathcal{L}_{\text{GAN}} + \lambda \cdot \mathcal{L}_X.$$

In all experiments, λ was set to 100. This study does not condition the discriminator like in Isola et al. [19], because input-output discrepancies are not penalized. The loss function in the case of this GAN is defined as:

$$\mathcal{L}_{\text{GAN}} = \sum_{n=1}^N -\mathbb{D}_{\rho_D} (\mathbb{G}_{\rho_G} (I^B)).$$

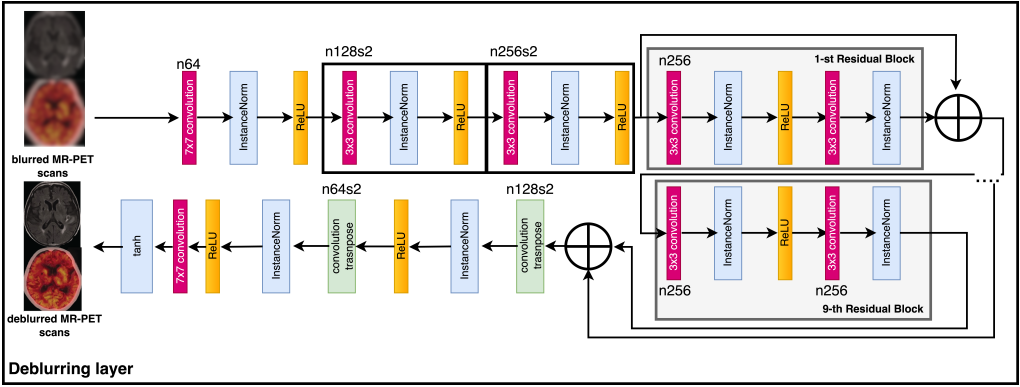


Fig. 6. Deblurring net.

Common data loss functions include the L1 or Mean Absolute Error (MAE) loss and the L2 or Mean Squared Error (MSE) loss. Using these functions as the sole optimization aim produces uncertain abnormalities in images. According to [40], the observed irregularities are due to the mean value of plausible solutions at the pixel level within the pixel space. Using the L2-loss technique in the perceptual loss function allows for mathematical expressions to calculate dissimilarity between the synthesized image and the reference image’s CNN feature maps. The terminology is expressed as follows:

$$\mathcal{L}_X = \frac{1}{U_{k,n} B_{k,n}} \sum_{x=1}^{U_{k,n}} \sum_{y=1}^{B_{k,n}} \left(\emptyset_{k,n} (I^S)_{x,y} - \emptyset_{k,n} \left(\mathbb{G}_{\rho_G} (I^B)_{x,y} \right) \right)^2 .$$

The symbol $\emptyset_{k,n}$ represents the feature map derived from the n -th convolution operation within a pre-trained network designed for MRI analysis [25]. The feature map is acquired subsequent to activation and prior to the k -th maxpooling layer. The variables $U_{k,n}$ and $B_{k,n}$ denote the dimensions of the feature maps.

5.4. MR-PET images denoising procedure

Magnitude pictures are the main representation in MRI-PET, making denoising difficult. The magnitude pictures are derived from real and imaginary components [10]. The presence of noise in magnitude images can be attributed to the Rician distribution, which exhibits a higher level of complexity compared to conventional additive noise. Denoising results depend on the model’s precision. This is due to its ability to ignore the core physical process and change it via sample-based learning. The main goal of MRI-PET noise mitigation is to improve diagnostic image quality by reducing noise. Noise-corrupted MR-PET images are represented by x , while noise-free images are represented

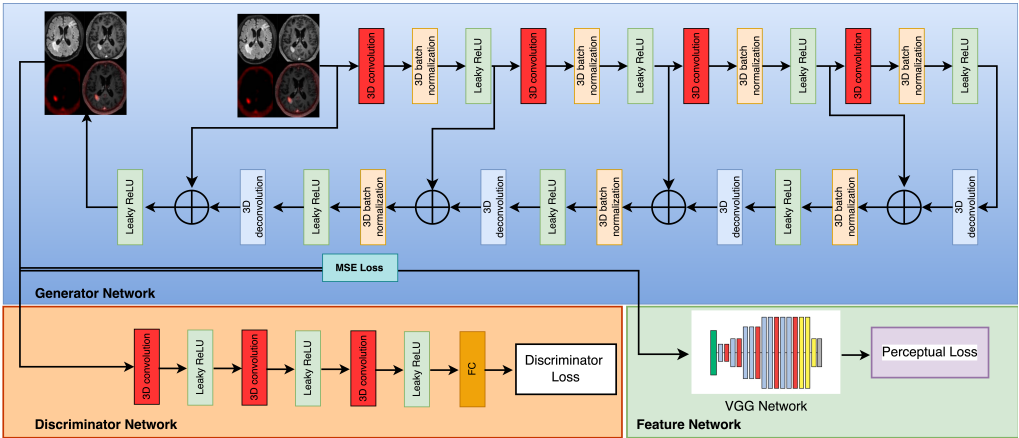


Fig. 7. Denoising net.

by y . Two matrices, x and y , with real-valued elements and the same dimensions, $m \times n$. Entities are connected as:

$$x = \varrho(y) .$$

The noise generation function is represented by the mapping function ϱ . Deep Learning is known for operating like a black box regardless of noise statistics. To optimize the denoising process of MR-PET, it is crucial to streamline the search for the most suitable approximation of the function ϱ^{-1} . The denoising technique entails the elimination of undesired noise from a provided signal or dataset.

$$\operatorname{argmin}_f \left\| \hat{y} - y \right\| .$$

The variable \hat{y} reflects the anticipated value of y , based on the function $f(x)$, which provides the most accurate approximation of the inverse of ϱ .

According to statistical analysis, it can be inferred that samples x and y originate from distinct data distributions. Specifically, the variable x denotes the distribution of a noisy picture (P_n), while the variable y denotes the distribution of a noise-free image (P_{gen}). The denoising technique employs a mapping algorithm to alter the distribution. The function f establishes a mapping between samples drawn from the distribution P_n and a distribution denoted as \mathbb{P}_{gen} , which is identical to the actual data distribution P_r .

The discriminative model is specifically designed to differentiate between samples generated by a generative model and real data samples. The generative model utilizes the provided input sample to generate a novel sample that has a high degree of similarity

to the underlying data distribution.

$$\begin{aligned} \mathcal{L}_{\text{WGAN}}(\mathbb{D}) = & -\mathbb{E}_{y \sim P_r} [\mathbb{D}(y)] + \mathbb{E}_{x \sim P_n} [\log \mathbb{D}(y)] + \mathbb{E}_{x \sim P_n} [D(G(x))] + \\ & + \psi \mathbb{E}_{\hat{x} \sim P_{\hat{x}}} \left[\left(\left\| \nabla_{\hat{x}} D(\hat{x}) \right\|_2 - 1 \right)^2 \right]. \end{aligned} \quad (2)$$

The final part of (2) is a gradient penalty factor, with ψ as a penalty coefficient. To construct the probability distribution, $P_{\hat{x}}$, points are uniformly sampled along straight lines from the actual data distribution P_r and generator distribution P_{gen} . Below is the loss function formulation for the generator \mathbb{G} :

$$\mathcal{L}_{\text{WGAN}}(G) = \mathbb{E}_{x \sim P_r} [\log \mathbb{D}(y)] + \mathbb{E}_{x \sim P_n} [\log (1 - D(G(x)))].$$

In activities that need pixel-level adjustments, the Mean Squared Error (MSE) loss function is utilized rather frequently. The main goal is to reduce the differences between the original image and the generated image at a pixel level. The computation described above can be derived using the following methodology:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{abc} \|\mathbb{G}(x) - y\|^2.$$

The variables a , b , and c represent the dimensions of the image. A recent study has demonstrated that the utilization of the Mean Squared Error (MSE) loss function has the potential to yield a substantial peak signal-to-noise ratio. Nevertheless, a decline in specificity, particularly with commonplace particulars, could have a substantial impact on clinical diagnosis [27].

The problem at hand is effectively tackled by the proposed loss function, which incorporates perceptual loss as documented in references [3], [10], and [22]. The utilization of a pre-existing neural network facilitates the extraction of pertinent data from both authentic and counterfeit photographic representations. Perceptual similarity quantifies the extent of dissimilarity in the attributes of reference and synthesized images. The next section provides an explanation of the perceptual loss function:

$$\mathcal{L}_{\text{perceptual}} = \frac{1}{abc} \|\omega(\mathbb{G}(x)) - \omega(y)\|_F^2.$$

The variable ω denotes the feature extractor, whereas a , b , and c denote the dimensions of the feature map. In this study, the VGG-19 network is employed for the purpose of extracting visual features [43]. The VGG-19 convolutional neural network consists of a total of nineteen layers, comprising sixteen convolutional layers and three fully connected layers. The scope of feature extraction is constrained to the initial sixteen layers. To implement the VGG network-based perceptual loss, the following procedures should be followed:

$$\mathcal{L}_{\text{VGG}} = \frac{1}{abc} \|\text{VGG}(\mathbb{G}(x)) - \text{VGG}(y)\|_F^2.$$

The generator \mathbb{G} is coupled to a joint loss function that includes MSE, VGG, and discriminator losses.

The architectural design of the discriminator network, denoted as \mathbb{D} , is illustrated in Figure 7. Each of the model's three convolutional layers uses 32, 64, or 128 filters. A homogeneous kernel size of $3 \times 3 \times 3$ was used to configure the convolution layers. The top layer is totally merged and gives a distinctive result. The pre-trained VGG-19 network extracts features. For further information, refer to the main source document [43]. Pan and Yang [39] found that transfer learning eliminates the need for network retraining for MR-PET scans, so

$$\mathcal{L}_{\text{RED-WGAN}} = \delta_1 \mathcal{L}_{\text{MSE}} + \delta_2 \mathcal{L}_{\text{VGG}} + \delta_3 \mathcal{L}_{\text{WGAN}}(G) .$$

The suggested RED-WGAN network configuration is shown in Figure 7. Three components make up the system: a generator network (denoted as \mathbb{G}), a discriminator network (denoted as \mathbb{D}), and a feature extractor (VGG network). Similar short connections connect the convolutional and deconvolutional layers. Each layer performs three-dimensional convolution, Leaky-ReLU activation, and batch normalization except for the final layer. The final layer only conducts 3D-convolution and Leaky-ReLU. This study uses a $3 \times 3 \times 3$ kernel configuration with a filter sequence of 32, 64, 128, 256, 128, 64, 32, and 1.

6. The implementation of sparse sampling in MR/PET raw data

The methodology aims to consolidate all data from several modalities. According to [34], this study suggests that PET data volume can be compressed. To reduce readout channels, positron-emitting radioactive elements are mixed whenever possible. They produced a higher-resolution PET image by consolidating their output signals. MR/PET hybrid scanners integrate super-resolution and compressive sensing through structural components. Sparse depiction of the detector's structure is obvious. The above characteristic allows sparse-sense to generate novel multiplexing setups. Making meaningful sensing matrices is essential in computer science. Several stochastic approaches can generate constrained isometry random matrices. Maximal likelihood is used to build sensing matrices in the framework. Research indicates that using identified matrices results in the lowest mean-square reconstruction error [30]. This publication describes a method that uses a minimal number of channels to create discrete space and time domains. This interprets PET input data as compressed PET signals. Each readout is interpreted using a linear combination of photodetector pixels, with weights represented as $c_{k,n}$ and shown in Figure 5. The number of sensors was lowered via 4:1 subsampling. MR-PET joint sparsity and shared product characteristics enable motion model parameter generation from MR data sets to enrich PET pictures. Super-resolution is achieved using the same method as shown in Figure 1.

7. Results

The algorithm was proven effective by this investigation. The technique was tested using Biograph mMR-Simultaneous MR-PET scanner data. Therefore, two separate sets were created. For reference, the baseline dataset includes all active channels. In the subsequent dataset, a reduction of 15% in the number of channels is achieved by deactivating eight detectors that are uniformly distributed.

To prepare the dataset, data from MR-PET was organized into static frames that were free from motion artifacts. The algorithm proposed in this study was utilized to reconstruct an image for each frame. The normal clinical approach was employed, which entails conducting 3 iterations, dividing the data into 24 subgroups, and applying a 5×5 Gaussian post-smoothing technique. The reference/target picture for the reconstruction process was obtained from the initial static frame, while the successive static frames provided the source images for reconstruction.

During a simulated three-minute scan, participants assumed various positions and orientations of their heads at random time intervals. Each study demonstrated a distinct variety of movements, resulting in varying quantities of static frames.

To augment the insufficient quantity of data for constructing a sufficiently large dataset to train the neural network, additional picture volumes were synthesized from the five patient trials. A total of one hundred picture volumes of MR-PET were simulated by implementing random changes to the line-of-response (LOR) data. Thereafter, the raw dataset was subjected to transformation and histogramming, and then forwarded to the reconstruction method, as previously discussed, in order to generate image volumes. To mitigate the computational expenses involved with training the neural network, the images underwent a resizing process. Specifically, the dimensions of the images were modified from $400 \times 400 \times 109$ to $128 \times 128 \times 96$. This was achieved by cutting the backdrop, which consisted of voxels with zero values, and subsequently rescaling the resulting image. Cross-validation was employed, utilizing a 4:1 ratio for the allocation of training and test data. The training process consisted of 100 epochs, each consisting of 20 steps per epoch, and used a batch size of 4. The learning rate was set to 10^{-4} .

The super-resolution image reconstruction model was trained using an NVIDIA DGX device with a graphics processing unit dubbed A-100, utilizing resources from Google Colab Pro. The learning strategy employed by the generator involves the utilization of merged low-resolution (LR) images with dimensions of $60 \times 60 \times 2$. One high-resolution (HR) image of 240 by 240 units is the goal. Content, perceptual, and adversarial losses comprise the generator's loss function. MRI images in HR and SR are shown to the Discriminator. The discriminator, or binary classifier, optimizes using binary cross-entropy. The Adam optimizer optimizes the generator and discriminator. Tensorflow and Keras are utilized to implement the proposed network in Python. The network for super resolution image reconstruction has undergone training for a total of 50 epochs.

The Adam optimizer is utilized by both the Generator and the Discriminator to optimize the network parameters. The initial learning rate for Adam is 0.001, which is then reduced to 0.000001 after half the number of epochs.

The generation process incorporates three distinct forms of losses, including content loss, perceptual loss, and adversarial loss. The Discriminator utilizes both the SR MRI image and the HR image. The classifier in question is a binary classifier that employs binary cross-entropy as a means of optimization. The utilized optimizer is an Adam optimizer, which is applied to both the generator and discriminator components. Moreover, the MR-PET dataset comprised 1000 pairs of pictures, each containing both blurred and sharp images, with a resolution of 640×640 . The deblurring network that has been shown has demonstrated exceptional performance in terms of structured self-similarity. It is comparable to the current state-of-the-art in terms of peak signal-to-noise ratio and offers visually appealing outcomes. The utilization of L2 distance in pixel space is not employed by the network, hence missing direct optimization for the PSNR measure.

The denoising model was trained with 1% and 4% noise added due to the limited understanding of the noise level in the actual data. In this study, the VGG network used for low resolution MR-PET image reconstruction was pre-trained using the Medical ImageNet dataset. The convolutional output of the VGG16 model was utilized as the encoded embedding of the de-aliased output and the ground truth. Subsequently, the mean squared error (MSE) was calculated between these two outputs. Create distinct networks for various undersampling ratios using the constant mutual hyperparameters: $\alpha = 10$, $\beta = 0.2$, $\theta = 0.003$, starting learning rate of 0.0001, batch size of 30. It is worth mentioning that the hyperparameters, namely α , β and θ represent the weights assigned to various loss components throughout the training process. The Adam optimization algorithm was employed, utilizing a momentum value of 0.4. The learning of each model was conducted via early stopping, with the learning rate being reduced by half every 4 epochs. The MR-PET reconstruction model proposed in this study demonstrates robustness and requires minimal parameter adjustment. Consequently, we employed the same hyperparameters for subsequent tests, employing different undersampling ratios, varied undersampling masks, and both with and without noise.

The models were implemented using a high-level Python wrapper called TensorLayer [9].

Subsampled sinograms are divided into two pieces with sparse orthogonal domains. A hybrid conjugate gradient method was used to iteratively recreate the PS sinogram. A system of equations was solved using blocked relaxations. Reducing component total variation (TV) improves the piece-wise smooth model of the initial component. After integrating the two pieces, the sinogram was created and used to enhance the PS sinogram. This method produces quantified PET images with fewer readout channels. The evaluation distinguishes two information groups. The first thing that needed to be accomplished was to evaluate the super-resolution (SR) image reconstruction technique by

directly comparing it to native and naive approaches. Investigating the precision and dependability of the magnetic resonance (MR) sample was one of the secondary objectives of this investigation. In the present investigation, both in vivo and phantom data were evaluated. The outcomes of the simulation are presented in Figure 8. Compressed Sensing, conjugate symmetry, and the Partial Fourier method speed up data collection while preserving the unique k-space trajectories. The present framework module aims to integrate compressed sensing and super-resolution into MRI scanners. This work used phantom input files to demonstrate compressed sensing (CS) challenges for magnetic resonance imaging (MRI).

8. Discussion

Over the past decade, MR-PET and other integrated scanning technologies have grown in importance. Understanding the purpose of these tools is the first step to becoming a notable figure in the field. Even with motion blur, the study's strategy reduced artifacts from insufficiently sampled data. This article describes a new super-resolution technique for high-sensitivity compressed MR/PET signals. As expected, the algorithm improves image resolution without changing the technology.

Table 1 shows that CS quality ratios affect the Peak Signal-to-Noise Ratio (PSNR) values. The PSNR is calculated using multiple methodologies using ground truth images. One hundred simulations were run. To establish statistical significance of quality measures for each simulation scenario, the PSNR was iteratively calculated and averaged. The best results were achieved using a 50% compression ratio. Reducing the number of input samples improves the PSNR, as seen in Tables 2 and 3. The symbols N , M , SD , $t(99)$, p refer to: number of tests performed, mean value, standard deviation, t -value with the confidence level of 99 percent, p -value, respectively. The examination duration decreases directly with this value. Motion distortions can be reduced while sacrificing resolution with this strategy. Rapid convergence, picture prior, and blur kernel detection are prioritized in this method.

Preliminary trial data can provide context for efficient test completion. Motion estimation techniques may reduce diagnostic imaging image artifacts, improving diagnosis accuracy. Figures 8 and 9 show improved result resolution and quality. The above results are preliminary and may change. A qualitative study of twenty patients' neuroimages showed the algorithm's benefits. Using a combined MR/PET scanner, 30 oncological patients provided PET and MRI data. The research used 30 simulated brain PET data volumes and patient model MR scans in phantom studies. The Peak Signal-to-Noise Ratio for each reconstruction procedure was calculated quantitatively.

To test the null hypothesis, a t-test was used to compare image quality ratings of images reconstructed from highly sparse sampling spaces using the proposed method and completely sampled ground truth images. A radiologist found that the offered technique

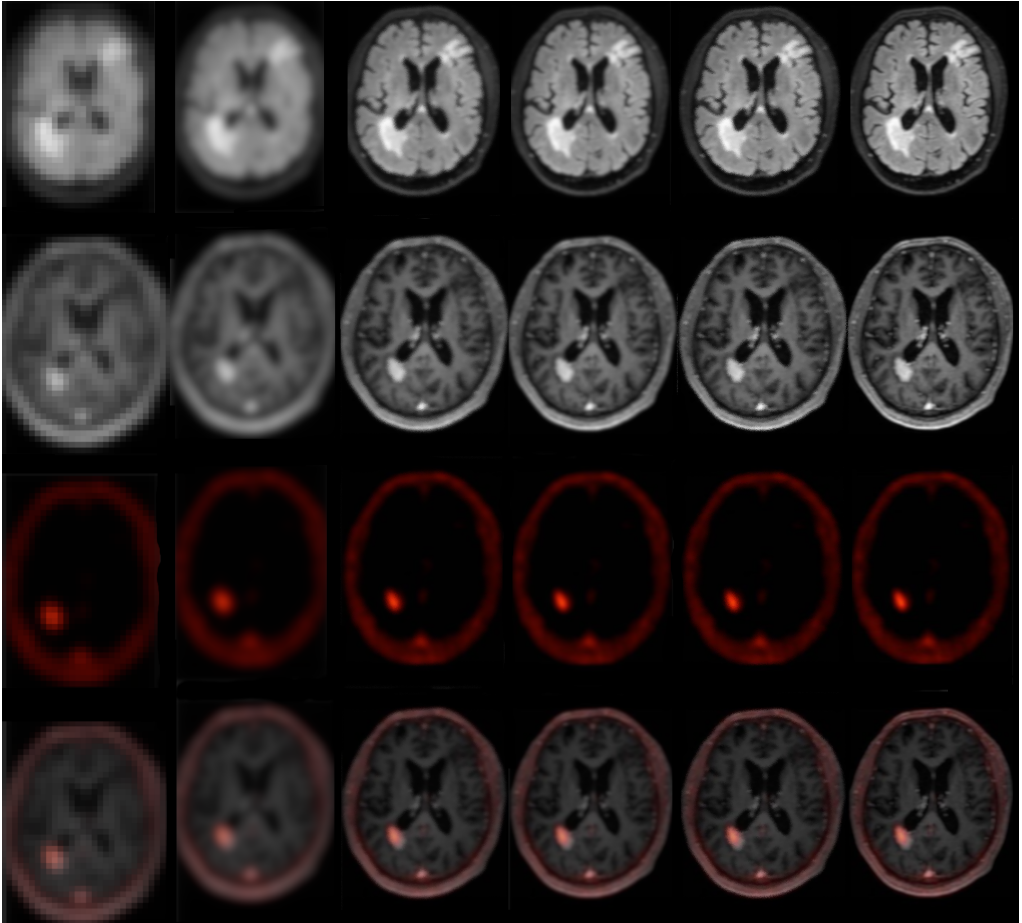


Fig. 8. The following is an illustration of a clinical trial. Images are numbered in a horizontal direction, from left to right. The present study involves the reconstruction of an image using the regular sampling scheme, without motion correction and SRR applied (1), B-spline curve (2) and Yang's method (3) were employed for the reconstruction process, along with Lim's method (4), Zhang's procedure (5), Kim's algorithm (6). Additionally, a proposed sampling scheme and motion compensation were utilized for super-resolution purposes. The aforementioned techniques were applied without introducing any additional information. The compression ratio is 50%. See Figure 9 for more results.

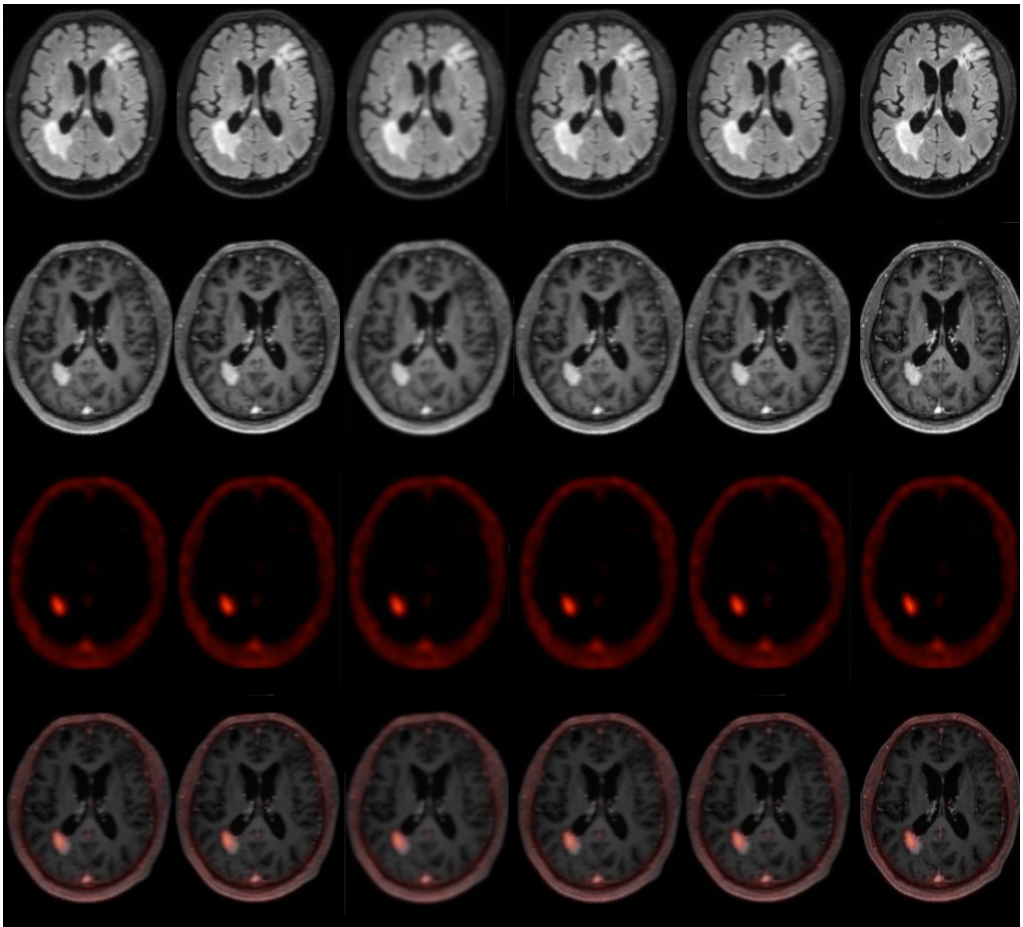


Fig. 9. The following is an illustration of a clinical trial (continued). Images are numbered in a horizontal direction, from left to right. The present study involves the reconstruction of an image using Mahapatra's method (7), Liu et al. procedure (8). Moreover, Dong's method (9) and Pham's method (10) were employed for the reconstruction process, along with Shi method (11), and the author's method (12) (see Tab. 2 for references). Additionally, a proposed sampling scheme and motion compensation were utilized for super-resolution purposes. The aforementioned techniques were applied without introducing any additional information. The compression ratio is 50%.

Tab. 1. Results of measuring the performance of the algorithm using various raw data sampling schemes on the data presented in Figure 9.

| Raw data sampling* [%] | PSNR [dB] | N | M | SD | $t(99)$ | p -value |
|------------------------|-----------|-----|-------|------|---------|------------|
| 20 | 26.76 | 100 | 26.76 | 0.04 | 0.322 | 0.143 |
| 40 | 32.33 | 100 | 32.33 | 0.05 | -0.274 | 0.147 |
| 60 | 33.98 | 100 | 33.98 | 0.03 | -1.299 | 0.191 |
| 80 | 34.16 | 100 | 34.16 | 0.02 | -0.643 | 0.056 |
| 100 | 34.88 | 100 | 34.88 | 0.06 | 1.001 | 0.064 |

*After making a comparison with scans that have been fully sampled, the phrase refers to the percentage of the input samples that are still present. For example, a ratio of sixty reveals that forty percent of the samples gleaned through a comprehensive examination were discarded.

Tab. 2. The statistical parameters associated with the efficiency metrics of the model depicted in Figs. 8 and 9.

| High resolution reconstruction method | PSNR [dB] | N | M | SD | $t(99)$ | p -value |
|---------------------------------------|-----------|-----|-------|------|---------|------------|
| no SRR, no MC | 24.88 | 100 | 24.88 | 0.04 | 0.265 | 0.533 |
| B spline curve | 26.01 | 100 | 26.01 | 0.03 | -0.321 | 0.399 |
| Yang et al. [47] | 29.80 | 100 | 29.80 | 0.02 | -0.928 | 0.294 |
| Lim et al. [29] | 29.67 | 100 | 29.67 | 0.02 | -0.912 | 0.232 |
| Zhang et al. [49] | 31.01 | 100 | 31.01 | 0.03 | -0.023 | 0.349 |
| Kim et al. [25] | 31.03 | 100 | 31.03 | 0.04 | -0.903 | 0.211 |
| Mahapatra et al. [32] | 30.01 | 100 | 30.01 | 0.04 | -0.429 | 0.473 |
| Liu et al. [30] | 29.93 | 100 | 29.93 | 0.05 | -1.003 | 0.321 |
| Dong et al. [7] | 28.55 | 100 | 28.55 | 0.04 | -1.003 | 0.218 |
| Pham et al. [23] | 31.66 | 100 | 31.66 | 0.02 | -1.003 | 0.362 |
| Shi et al. [42] | 32.03 | 100 | 32.03 | 0.03 | -1.003 | 0.412 |
| The suggested SRR algorithm | 33.98 | 100 | 33.98 | 0.03 | -0.992 | 0.102 |

had better image metric values and better anatomical structure representation than the other algorithms. Additionally, a stringent Bowker symmetry test was used to examine image quality disparities. Markov random field (MRF) optimization allows structural representation-based registration approaches to be developed. The target registration

Tab. 3. The present work aims to evaluate the efficacy of various reconstruction techniques for in-vivo brain imaging. Specifically, it investigates the use of motion correction (MC) and upscaling (HR) using the method given by the author. See Figs. 8 and 9 for the pertinent brain pictures. The Peak Signal-to-Noise Ratio (PSNR) values corresponding to the four scenarios stated before are displayed in the second column of Table 1.

| input | Sparse-sampling ratio [%] | MC | SRR | PSNR [dB] | N | M | SD | $t(99)$ | p -value |
|-------|---------------------------|-----|-----|-----------|-----|-------|------|---------|------------|
| LR | 50 | No | No | 25.77 | 100 | 25.77 | 0.03 | -1.211 | 0.198 |
| LR | 50 | Yes | No | 26.65 | 100 | 26.65 | 0.03 | -1.045 | 0.245 |
| HR | 50 | Yes | No | 28.19 | 100 | 28.19 | 0.03 | -1.217 | 0.193 |
| SR | 50 | Yes | Yes | 33.98 | 100 | 33.98 | 0.03 | -1.299 | 0.191 |

error (TRE) measure was used to evaluate motion estimation algorithms:

$$\text{TRE} = \frac{1}{N} \sum_{i=1}^N \sqrt{(T_{L_x}^i - T_{\mathbb{D}_x}^i)^2 + (T_{L_y}^i - T_{\mathbb{D}_y}^i)^2 + (T_{L_z}^i - T_{\mathbb{D}_z}^i)^2}.$$

Using a linear combination of radial basis functions as a ground truth, the compelled deformation is T_L . In contrast, the motion estimation algorithms shown in Table 4 determine the deformation parameters of $T_{\mathbb{D}}$. The variable N represents the number of landmarks manually identified following medical expert guidance. The author's MC algorithm was tested using multiple image registration methods to find the most efficient one. All research registration procedures were statistically analyzed using mean and standard deviation. Table 4 shows the t -test TRE and p -values. Statistical study shows that the author's strategy differs significantly from alternative strategies with p -values below 0.002. Target Registration Error (TRE) mean and standard deviation values of 1.4 and 0.2 voxel were obtained by registering input and output images. The aforementioned values outperformed alternative strategies, as seen in Table 4.

To evaluate motion correction results for each participant, the disparity between images with motion and reference images without motion parameters was calculated. The statistical significance of data changes with and without artificial motion was examined using paired t -tests. Statistical analyses show the improvement is significant. Please refer to tables 2-3 for a comparison of the proposed approach to existing image resolution enhancement algorithms. This study used numerous motion registration algorithms, i.e., the methods of Wachinger et al. [45], Groppe et al. [13], Jenkinson et al. [20], Yang et al. [47], Greve et al. [12], Kadipasaoglu et al. [23], MIND [15], Branco et al. [2], as well as WGAN deformable registration procedure, i.e., the author's procedure. This approach has been expanded to incorporate directions on handling erroneous target images from a clinical scanner with better resolution to improve its viability. Super-resolution (SR) has been used to identify the relationship between low- and high-resolution scanner image domains.

Tab. 4. Statistical parameters of several registration methods in relation to the implemented technique.

| Motion compensation procedure | TRE [voxels] | | |
|--|--------------|-----------|----------------|
| | <i>M</i> | <i>SD</i> | <i>p-value</i> |
| not applied | 4,90 | 2,60 | <0,002 |
| Wachinger et al. [45] | 2,72 | 0,78 | <0,005 |
| Groppe et al. [13] | 2,41 | 0,27 | <0,005 |
| Jenkinson et al. [20] | 3,55 | 0,37 | <0,004 |
| Yang's et al. [47] | 2,01 | 0,37 | <0,004 |
| Greve et al. [12] | 3,01 | 0,29 | <0,006 |
| Kadipasaoglu et al. [23] | 1,66 | 0,31 | <0,003 |
| MIND [15] | 1,82 | 0,19 | <0,004 |
| Branco et al. [2] | 1,73 | 0,16 | <0,009 |
| WGAN deformable MC – the author's method | 1,40 | 0,17 | <0,002 |

This study involved an evaluation of the author's methods in comparison to several advanced super resolution image reconstruction algorithms. The current study focuses on the reconstruction of an image using a regular sampling scheme, without the application of motion correction and super-resolution reconstruction (SRR) techniques. The reconstruction process involves the utilization of B-spline curve and Yang's method [47], as well as Lim's method [29], Zhang's procedure [49], Kim's algorithm [25], and Liu et al.'s procedure [30]. Furthermore, the reconstruction procedure utilized Dong's approach [7], Pham's method [23], Shi's method [42], and the author's method.

This study improved neural network training to accurately map low-resolution magnetic resonance and positron emission tomography images to ground-truth subimages. Its potential benefits, particularly its ability to create high contrast and resolution, are the main reason for its expected success. MR/PET technology will be integrated with sparsely sampled input data super resolution image reconstruction techniques in this study. Budgetary issues hinder this methodology's performance and these issues should be considered. The long-term balancing of MR/PET's increased expenses is unknown. Two data sets were used in the experiment. Experiments compared the compressively sensed super-resolution picture reconstruction approach to simpler and less advanced methods. The magnetic resonance (MR) sample design's efficacy was the study's secondary goal. The proposed reconstruction approach and alternative algorithms were

used at different compression rates in MRI, with subjective and objective picture evaluations. This study examined in vivo and phantom inputs. These images show computer-generated model results. Compared to unmodified k-space trajectories, compressed sensing, conjugate symmetry, and Partial-Fourier (PF) technologies accelerate data collecting. Compressed sensing (CS) can reconstitute sparse signals by projecting them into a low-dimensional linear subspace. The theoretical certainty of the methodology given gives it great promise. The study uses k-space modifications and Generative Adversarial Networks (GANs) in the image domain. Generative Adversarial Networks (GANs) can include image-specific prior knowledge. In the picture and k-space domains, iterative calculations using Wasserstein Generative Adversarial Networks (WGANs) and k-space correction approaches are used. The method shows potential in tackling k-space rectification error. Compared to other methods, the analyzed strategy reconstructs images with better quality and fewer aliasing artifacts. The suggested method reduces aliasing artifacts better than existing and non-iterative methods. No matter the sample frequency for Cartesian and radial sampling masks, the suggested approach has a higher peak signal-to-noise ratio than the others.

In addition, the study used empirical data in the form of Magnetic Resonance images, which contain actual values rather than true k-space data from MRI scans. Pictures with complex numerical values are the theme. In Generative Adversarial Networks, a fake connection between input and output layers is essential. Preprocessing is essential for complex data sets. T1-weighted images and other magnetic resonance imaging techniques will be used to measure clinical value and examine radiologists. This study proposes changes to increase image quality and reduce data collection time. Even with misregistration distortions, the proposed technique can eliminate sparse data artifacts. The strategy uses compressed sensing, raw data sparsity, and super-resolution reconstruction to improve k-space filling efficiency or fidelity. Image complexity decreases as MR/PET picture fidelity improves. Edge representation improves with higher high-frequency component sampling rates. The technique reviewed shows promise for hybrid scanner integration without hardware adjustments.

The reconstruction approaches for MR/PET use either whole raw data or pre-existing data as reference standards. Refer to Tables 2 through 3 for a comprehensive review of clinical trials where acceleration factors of up to 2 led to diagnostically viable scans and radiologists acknowledged the higher resolution. The main outcomes of this study encompass the subsequent findings:

- The algorithm described in the framework showcases a comprehensive methodology for the collaborative reconstruction of MR-PET data. This work places significant emphasis on several critical areas, including sparse sampling trajectories, synchronization of k subspaces, deblurring, noise reduction, motion correction, and ultimately, enhancing the resolution of a picture.

- The current investigation presents an innovative framework for the reconstruction of MR-PET images through the utilization of a generative super-resolution methodology.
- The provided methodology utilizes the combined sparsity of both the MR and PET modalities.
- The limited availability of MR and PET raw data has led to an increase in the rate at which the input data is processed.
- The methodology has been specifically designed for the purpose of collecting visual data at various scales. Other authors frequently oversimplify this matter.
- The system demonstrates the ability to extract visual cues across various scales. Other writers often oversimplify this topic issue.
- The technology used involves certain preprocessing phases to tackle the difficulties associated with blur and noise removal layers.
- The suggested method utilizes a neural network-based reconstruction algorithm for magnetic resonance imaging. The objective of this approach is to recover images of poor quality that are obtained from extremely limited raw data.
- The methodology described above utilizes the compressed sensing framework in order to prioritize the effort to minimize the duration of data collecting.
- The author's deformable motion estimation approach is buried within the reconstruction layer of the procedure.

The presented system uses compressed raw data, an advanced SR-GAN architecture, and a denoising module to pre-process low-resolution MR-PET images. The network can super-resolve low-resolution and noisy MR-PET images and recreate high-resolution MR images. The methods offered helps solve a problem where artifacts and noise diminish the peak signal-to-noise ratio in MR-PET images, reducing the generative adversarial network's effectiveness. The proposed solution yields better picture reconstruction quality than previous methods, as shown by empirical data. Therefore, this can improve diagnostic procedure suggestions for healthcare providers.

References

- [1] G. Antoch and A. Bockisch. Combined PET/MRI: a new dimension in whole-body oncology imaging? *European Journal of Nuclear Medicine and Molecular Imaging*, 36(S1):113–120, 2008. doi:10.1007/s00259-008-0951-6.
- [2] M. P. Branco, A. Gaglianese, D. R. Glen, D. Hermes, Z. S. Saad, et al. ALICE: A tool for automatic localization of intra-cranial electrodes for clinical and high-density grids. *Journal of Neuroscience Methods*, 301:43–51, 2018. doi:10.1016/j.jneumeth.2017.10.022.
- [3] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. In: *Proc. Int. Conf. Learning Representation (ICLR)*, 2015. Proceedings published in arXiv, see <https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.html>. doi:10.48550/arXiv.1412.7022.



- [4] H. Chang, D. Y. Yeung, and Y. Xiong. Super-resolution through neighbor embedding. In: *Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. Washington, USA, 2004. doi:10.1109/CVPR.2004.1315043.
- [5] Y. Chen, F. Shi, A. G. Christodoulou, Y. Xie, Z. Zhou, et al. Efficient and accurate MRI super-resolution using a generative adversarial network and 3D multi-level densely connected network. In: A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, eds., *Proc. Conf. Medical Image Computing and Computer Assisted Intervention (MICCAI) 2018*, vol. 11070 of *Lecture Notes in Computer Sciences*, pp. 91–99. Springer International Publishing, Granada, Spain, 16–20 Sep 2018. doi:10.1007/978-3-030-00928-1_11.
- [6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. doi:10.1109/TIP.2007.901238.
- [7] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In: *Proc. European Conference on Computer Vision (ECCV)*, vol. 8692 of *Lecture Notes in Computer Science*, p. 184–199. Springer, Zurich, Switzerland, 6–12 Sep 2014. doi:10.1007/978-3-319-10593-2_13.
- [8] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In: *Proc. European Conference on Computer Vision (ECCV)*, vol. 9906 of *Lecture Notes in Computer Science*, p. 391–407. Springer, Amsterdam, The Netherlands, 2016. doi:10.1007/978-3-319-46475-6_25.
- [9] H. Dong, A. Supratak, L. Mai, F. Liu, A. Oehmichen, et al. TensorLayer: A versatile library for efficient deep learning development. In: *Proc. 25th ACM International Conference on Multimedia, MM '17*, p. 1201–1204. Association for Computing Machinery, 2017. doi:10.1145/3123266.3129391.
- [10] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In: *Advances in Neural Information Processing Systems 28 – Proc. NIPS 2015*, vol. 28 of *NeurIPS Proceedings*, pp. 262–270, 2015. <https://proceedings.neurips.cc/paper/2015/hash/a5e00132373a7031000fd987a3c9f87b-Abstract.html>.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, et al. Generative adversarial nets. In: *Advances in Neural Information Processing Systems 27 – Proc. NIPS 2014*, vol. 27 of *NeurIPS Proceedings*, pp. 2672–2680, 2014. <https://proceedings.neurips.cc/paper/2015/hash/a5e00132373a7031000fd987a3c9f87b-Abstract.html>.
- [12] D. N. Greve and B. Fischl. Accurate and robust brain image alignment using boundary-based registration. *NeuroImage*, 48(1):63–72, 2009. doi:10.1016/j.neuroimage.2009.06.060.
- [13] D. M. Gropp, S. Bickel, A. R. Dykstra, X. Wang, P. Mégevand, et al. iELVis: An open source MATLAB toolbox for localizing and visualizing human intracranial electrode data. *Journal of Neuroscience Methods*, 281:40–48, 2017. doi:10.1016/j.jneumeth.2017.01.022.
- [14] J. Gu, H. Lu, W. Zuo, and C. Dong. Blind super-resolution with iterative kernel correction. In: *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, p. 1604–1613. Long Beach, CA, USA, 2019. doi:10.1109/CVPR.2019.00170.
- [15] M. P. Heinrich, M. Jenkinson, M. Bhushan, T. Matin, F. V. Gleeson, et al. MIND: Modality independent neighbourhood descriptor for multi-modal deformable registration. *Medical Image Analysis*, 16(7):1423–1435, 2012. Special Issue on the 2011 Conference on Medical Image Computing and Computer Assisted Intervention. doi:10.1016/j.media.2012.05.008.
- [16] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, et al. Meta-SR: A magnification-arbitrary network for super-resolution. In: *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, p. 1575–1584. Long Beach, CA, USA, 15–20 Jun 2019. doi:10.1109/CVPR.2019.00167.

- [17] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In: *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, p. 4700–4708. Honolulu, HI, USA, 21–26 Jul 2017. doi:[10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
- [18] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo. Deep learning for undersampled MRI reconstruction. *Physics in Medicine & Biology*, 63(13):135007, 2018. doi:[10.1088/1361-6560/aac71a](https://doi.org/10.1088/1361-6560/aac71a).
- [19] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 5967–5976. Honolulu, HI, USA, 2017. doi:[10.1109/CVPR.2017.632](https://doi.org/10.1109/CVPR.2017.632).
- [20] M. Jenkinson, P. Bannister, M. Brady, and S. Smith. Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841, 2002. doi:[10.1006/nimg.2002.1132](https://doi.org/10.1006/nimg.2002.1132).
- [21] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, et al. Edge-enhanced GAN for remote sensing image superresolution. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5799–5812, 2019. doi:[10.1109/TGRS.2019.2902431](https://doi.org/10.1109/TGRS.2019.2902431).
- [22] J. Johnson, A. Alahi, and F. F. Li. Perceptual losses for real-time style transfer and super-resolution. In: *Proc. Eur. Conf. Computer Vision (ECCV)*, vol. 9906 of *Lecture Notes in Computer Science*, pp. 694–711. Springer, 2016. doi:[10.1007/978-3-319-46475-6_43](https://doi.org/10.1007/978-3-319-46475-6_43).
- [23] C. M. Kadipasaoglu, C. Morse, K. Pham, C. Donos, and N. Tandon. SAMCOR: A robust and precise co-registration algorithm for brain CT and MR imaging. *Interdisciplinary Neurosurgery*, 30:101637, 2022. doi:[10.1016/j.inat.2022.101637](https://doi.org/10.1016/j.inat.2022.101637).
- [24] K. Kataoka, Y. Shiraiishi, Y. Takeda, S. Sakata, M. Matsumoto, et al. Aberrant PD-L1 expression through 3'-UTR disruption in multiple cancers. *Nature*, 534(7607):402–406, 2016. doi:[10.1038/nature18294](https://doi.org/10.1038/nature18294).
- [25] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In: *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1637–1645. Las Vegas, NV, USA, 27 Jun 2016. doi:[10.1109/CVPR.2016.181](https://doi.org/10.1109/CVPR.2016.181).
- [26] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. ReconNet: Non-iterative reconstruction of images from compressively sensed measurements. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 449–458. Las Vegas, NV, USA, 27 Jun 2016. doi:[10.1109/CVPR.2016.55](https://doi.org/10.1109/CVPR.2016.55).
- [27] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 105–114. Honolulu, HI, USA, 2017. doi:[10.1109/CVPR.2017.19](https://doi.org/10.1109/CVPR.2017.19).
- [28] K. A. Lidke, B. Rieger, T. M. Jovin, and R. Heintzmann. Superresolution by localization of quantum dots using blinking statistics. *Optics Express*, 13(18):7052–7062, 2005. doi:[10.1364/OPEX.13.007052](https://doi.org/10.1364/OPEX.13.007052).
- [29] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In: *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, p. 1132–1140. Honolulu, HI, USA, 21–26 Jul 2017. doi:[10.1109/CVPRW.2017.151](https://doi.org/10.1109/CVPRW.2017.151).
- [30] C. Liu and D. Sun. On Bayesian adaptive video super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):346–360, 2014. doi:[10.1109/TPAMI.2013.127](https://doi.org/10.1109/TPAMI.2013.127).
- [31] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, et al. Robust video super-resolution with learned temporal dynamics. In: *Proc. 2017 IEEE Int. Conf. Computer Vision (ICCV)*, pp. 2526–2534. Venice, Italy, 22–29 Oct 2017. doi:[10.1109/ICCV.2017.274](https://doi.org/10.1109/ICCV.2017.274).

- [32] D. Mahapatra, B. Bozorgtabar, and R. Garnavi. Image super-resolution using progressive generative adversarial networks for medical image analysis. *Computerized Medical Imaging and Graphics*, 71:30–39, 2019. doi:[j.compmedimag.2018.10.005](https://doi.org/10.1016/j.compmedimag.2018.10.005).
- [33] K. Malczewski. Image resolution enhancement of highly compressively sensed CT/PET signals. *Algorithms*, 13(5), 2020. doi:[10.3390/a13050129](https://doi.org/10.3390/a13050129).
- [34] K. Malczewski. Super-resolution with compressively sensed MR/PET signals at its input. *Informatics in Medicine Unlocked*, 18, 2020. doi:[10.1016/j.imu.2020.100302](https://doi.org/10.1016/j.imu.2020.100302).
- [35] K. Malczewski. Diffusion weighted imaging super-resolution algorithm for highly sparse raw data sequences. *Sensors*, 23(12):5698, 2023. doi:[10.3390/s23125698](https://doi.org/10.3390/s23125698).
- [36] K. Malczewski and R. Stasinski. High resolution MRI image reconstruction from a PROPELLER data set of samples. *International Journal of Functional Informatics and Personalised Medicine*, 1(3), 2008. doi:[10.1504/IJFIPM.2008.021394](https://doi.org/10.1504/IJFIPM.2008.021394).
- [37] A. Mousavi, A. B. Patel, and R. G. Baraniuk. A deep learning approach to structured signal recovery. In: *Proc. 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Monticello, IL, USA, 29 Sep – 02 Oct 2015.
- [38] K. Ning, Z. Zhang, K. Han, S. Han, and X. Zhang. Multi-frame super-resolution algorithm based on a WGAN. *IEEE Access*, 9:85839–85851, 2021. doi:[10.1109/ACCESS.2021.3088128](https://doi.org/10.1109/ACCESS.2021.3088128).
- [39] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi:[10.1109/TKDE.2009.191](https://doi.org/10.1109/TKDE.2009.191).
- [40] H. Pedersen, S. Kozerke, S. Ringgaard, et al. k-t PCA: temporally constrained k-t BLAST reconstruction using principal component analysis. *Magnetic Resonance in Medicine*, 62(3):706–716, 2009. doi:[10.1002/mrm.22052](https://doi.org/10.1002/mrm.22052).
- [41] G. Qian, Y. Wang, J. Gu, C. Dong, W. Heidrich, et al. Rethinking learning-based demosaicing, denoising, and super-resolution pipeline. In: *Proc. 2022 IEEE Int. Conf. Computational Photography (ICCP)*, pp. 1–12. Pasadena, CA, USA, 1-5 Aug 2022. doi:[10.1109/ICCP54855.2022.9887682](https://doi.org/10.1109/ICCP54855.2022.9887682).
- [42] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, p. 1874–1883. Las Vegas, NV, USA, 27-30 Jun 2016. doi:[10.1109/CVPR.2016.207](https://doi.org/10.1109/CVPR.2016.207).
- [43] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In: *Proc. Int. Conf. Learning Representation (ICLR)*, 2015. Proceedings published in arXiv, see <https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:accepted-main.html>. doi:[10.48550/arXiv.1409.1556](https://doi.org/10.48550/arXiv.1409.1556).
- [44] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2790–2798, 21-26 Jul 2017. doi:[10.1109/CVPR.2017.298](https://doi.org/10.1109/CVPR.2017.298).
- [45] C. Wachinger and N. Navab. Entropy and Laplacian images: Structural representations for multi-modal registration. *Medical Image Analysis*, 16(1):1–17, 2012. doi:[10.1016/j.media.2011.03.001](https://doi.org/10.1016/j.media.2011.03.001).
- [46] A. W. A. Wahab, M. A. Bagiwa, M. Y. I. Idris, S. Khan, Z. Razak, et al. Passive video forgery detection techniques: A survey. In: *Proc. 2014 10th Int. Conf. Information Assurance and Security (IAS)*, pp. 29–34. Okinawa, Japan, 28-30 Nov 2014. doi:[10.1109/ISIAS.2014.7064616](https://doi.org/10.1109/ISIAS.2014.7064616).
- [47] F. Yang, M. Ding, X. Zhang, W. Hou, and C. Zhong. Non-rigid multi-modal medical image registration by combining L-BFGS-B with cat swarm optimization. *Information Sciences*, 316:440–456, 2015. doi:[10.1016/j.ins.2014.10.051](https://doi.org/10.1016/j.ins.2014.10.051).

- [48] Z. Zhang, D. Gao, X. Xie, and G. Shi. Dual-channel reconstruction network for image compressive sensing. *Sensors*, 19:2549, 2019. doi:[10.3390/s19112549](https://doi.org/10.3390/s19112549).
- [49] Z. Zhang, Z. Wang, Z. Lin, and H. Qi. Image super-resolution by neural texture transfer. In: *Proc. 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, p. 7974–7983. Long Beach, CA, USA, 15-20 Jun 2019. doi:[10.1109/CVPR.2019.00817](https://doi.org/10.1109/CVPR.2019.00817).

LIQUID DETECTION AND INSTANCE SEGMENTATION BASED ON MASK R-CNN IN INDUSTRIAL ENVIRONMENT

Grzegorz Gawdzik ^{1,2} and Arkadiusz Orłowski ²

¹*Security and Defense Systems Division, Lukaszewicz Research Network –*

Industrial Research Institute for Automation and Measurements PIAP, Warsaw, Poland

²*Institute of Information Technology, Warsaw University of Life Sciences – SGGW, Warsaw, Poland*

Abstract The goal of the paper is to present an efficient approach to detect and instantiate liquid spilled in the industrial and industrial-like environments. Motivation behind it is to enable mobile robots to automatically detect and collect samples of spilled liquids. Due to the lack of useful training data of spilled substances, a new dataset with RGB images and masks was gathered. A new application of the Mask-RCNN-based algorithm is proposed which has the functionalities of detecting the spilled liquid and segmenting the image.

Keywords: AI, Mask-RCNN, liquid detection, dataset.

1. Introduction

Robotics is a domain in which machine vision is frequently used. It has multiple applications, including e.g. automotive domain for lines and signs detection, or manufacturing industry for pick & place, assembly and quality control operations. This paper is focused on a specific usage of machine vision in mobile robotics for liquid spill detection. Liquid leakages or liquid spills are something typical in industrial environment, especially in chemical production including oil, detergents, adhesives, cosmetic chemicals, and other specialty chemicals. It is common to see chemical substances on the floor, close to the open tanks covers or leaking tank flanges. Some of those chemicals are hazardous or even toxic for humans and shall be treated and removed in a specific way. To understand the type of a substance, it shall be sampled and analysed. This operation can be performed manually by human or with the use of a dedicated tool or machine. The goal for this paper is to propose a solution that would enable automatic sampling of spilled liquids. The solution is solely based on RGB data analysis and enables the detection of different types of substances that were spilled on the factory-like floor. The solution provides both bounding box of detected substances as well as masks that can be used for further processing and utilized for substance localization in 3D environment. The proposed algorithm is based on convolutional neural networks and is trained in a supervised manner. Dataset used for training is a custom one and was prepared exclusively for this purpose. Multiple databases were searched to gather required data, however due to niche application no available dataset was found.

2. Dataset

Dataset consisting spilled liquids in a factory-like environment was not available in any open access database. The searched terms were related to stains (incl. stains on the clothes), oil leakage, oil slick, spill of oil, liquid leak, fluid leak, or their variations. Among verified databases there were for instance: COCO dataset [9,10] (over 80 classes), Kaggle datasets [20] (over 280 000 datasets), OpenML database [16] (over 5300 datasets), Penn Machine Learning Benchmarks [15,17,18,19] (over 400 datasets). Therefore, a new dataset was required to partially fill the gap and to provide basic images related to spilled liquids. The dataset consists of objects of one class only, which is a liquid. Liquid class encloses two different types of liquids which are tea and coffee. In the future the dataset will be expanded by other types of liquids, especially chemical or chemical-like substances typical for chemical industry. Database was gathered using a camera with Full HD resolution and 30 fps in different lighting conditions, both natural and artificial. The above-mentioned substances were spilled on two different surface types, which are industrial monocolour epoxy floor (epoxy) and light terracotta tiles (terracotta). Gathering of data for a database was divided into five steps:

1. recording of short videos (approx. 10s) for each example of spilled liquid showing a spill from different angles;
2. reduction of the recorded frames by keeping only every tenth frame;
3. division of the video into separate images;
4. annotation of the images by drawing polygons and adding contextual information like type of substance and type of surface;
5. transfer of the frames with an annotation file into one dataset.

For dataset preparation 29 short videos were recorded. Dataset consists of images and a file with annotations. Examples of selected frames together with extracted from annotations masks can be seen in Fig. 1. The dataset in its current form (without extracted masks) is available in the repository [3].

Videos were annotated with the use of CVAT programme [12] both in local and server-based versions. Extracted video frames with annotations were saved in a COCO-dataset-like format including following information: bounding box of the object, polygon describing the edges of the object (segmentation), area of the polygon (in image pixels), category (always one – liquid) and attributes showing a type of substance and type of surface.

3. Solution

The proposed solution is based on transfer learning, with a use of Mask R-CNN algorithm as a foundation. The algorithm was selected due to its well known performance in detection and segmentation tasks [5]. The inputs for the algorithm are images and

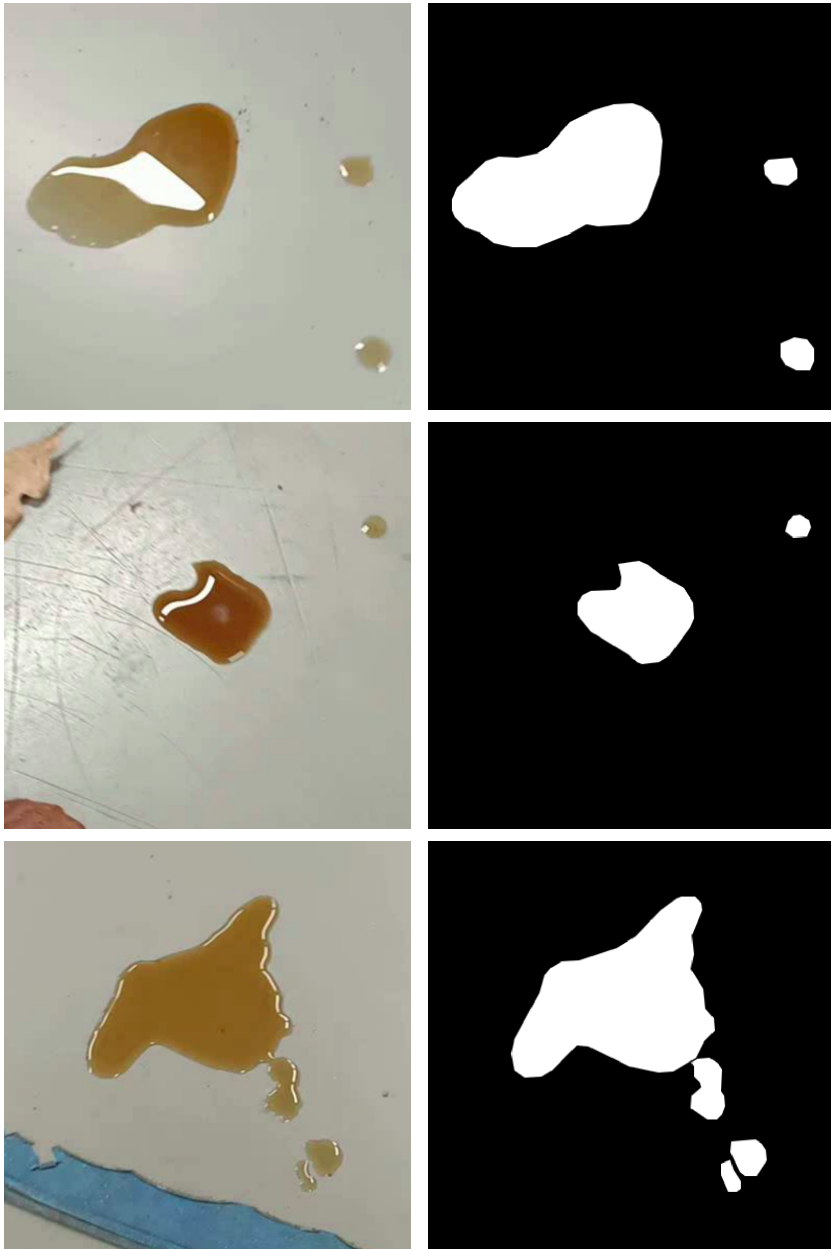


Fig. 1. Samples of the dataset. Images presenting RGB images and created masks.

masks. To comply with requirements the masks had to be extracted from the dataset (based on annotations) and saved as a set of images associated with original RGB ones. The created masks are presented in Fig. 1.

3.1. Environment and Tools

The algorithm was trained, tested and verified on two machines.

The first one is Lenovo ThinkPad T16p with 12th Gen Intel® Core™ i7-12700H×20, 32 GiB RAM, NVIDIA GeForce RTX 3050 Laptop GPU (4GiB VRAM) and installed Ubuntu 20.04.6 LTS.

The second one is Clevo x370 SNW-G with 13th Gen Intel® Core™ i9-13900HX×32, 64 GiB RAM, NVIDIA GeForce RTX 4090 Laptop GPU (12GiB VRAM) and installed Ubuntu 20.04.6 LTS.

To guarantee same software version on both machines the conda environment [11] was prepared and installed. As the main framework PyTorch [13] in version 1.13 with CUDA support was used. The PyTorch-based algorithm was trained and tested both with a use of CUDA and CPU. CUDA was used with NVIDIA GeForce RTX 4090 Laptop GPU and CPU with Intel® Core™ i7-12700, due to limitations of NVIDIA GeForce RTX 3050 Laptop GPU.

The usage of two calculation means emphasised a dramatic difference in training speed on both machines, even though the values of parameters and number of epochs were the same. Training with GPU for 10 epochs lasted around 4 minutes, while training on CPU lasted around 5 hours. Due to this difference, further calculations were performed on the GPU and therefore all outcomes presented in the paper are based on the outputs from the algorithm run on the GPU.

3.2. Algorithm

Mask R-CNN-based algorithms are widely used in multiple domains especially were detectable object presence and its exact shape is required. The algorithm has application in the fields of geospatial surveying [2, 22], object detection in technical processes [21], atmospheric analysis [23], fruits detection [6], biomedical studies [7] and many others.

The algorithm was created with the use of PyTorch framework. The implementation is based on a finetuned Mask R-CNN model, pre-trained on COCO train2017 dataset [1, 10] with default weights. Used Mask R-CNN model is founded on top of ResNet-50-FPN [5] architecture. Model's head for bounding box prediction and image segmentation was changed and trained on the dataset to output both bounding box and segmentation mask of a detected object. The algorithm classifies two classes, which are *liquid spill* – class 1, and *background* – class 0. In the algorithm a few hyperparameters was used including number of batches, learning rate, number of epochs and split ratio. As the optimiser a stochastic gradient descent with momentum was selected.

3.3. Training and Testing Datasets

To create a training and testing sets, the dataset had to be split, however not to bias the results it was decided that there will be no permanent division of the sets. On the contrary, at each run of the algorithm, the dataset was randomly divided into two sets of 70% to 30%, respectively. To reduce randomness of results each run (with unchanged parameters' values) was repeated 10 times. The approach was selected to keep the results objective and not to skew them at the dataset division level. The inputs are twofold as the algorithm accepts two tuples, which are: 1) RGB images, 2) annotations, including bounding boxes, labels, masks and image ID to reference with RGB images.

3.4. Hyperparameters

Based on the initial algorithm performance, it was observed that the hyperparameters with the biggest impact were: 1) batch size and 2) learning rate. The number of epochs did not change the results significantly and after looping for 6-7 times the loss value was stabilised. According to initial results, 1000 additional runs of the algorithm was performed, with hyperparameters that were set as follows: batch size from 1 to 10, learning rate from 0.001 to 0.01 and number of epochs to 10. The others were set as follows: momentum factor to 0.9 and weight decay to 0.0005.

During training the images were shuffled in each run to generalise the training set and limit impact of the consecutive images from one video. While testing the images were not shuffled.

3.5. Predictions

Predictions are composed of a set of values. Each prediction consists of information about detected objects, including:

1. coordinates of bounding boxes in reference to analysed RGB image,
2. class number (in this case always liquid, as another class is a background),
3. prediction score,
4. coordinates of vertices creating a polygon of the segmentation mask.

Based on the predictions, a mask is generated to visualise a successful detection. Examples of the predicted masks are presented in Fig. 2.

4. Evaluation

The algorithm was run multiple times with consecutively changed parameters. During each run the algorithm calculated the loss value after each epoch and adjusted the weights. The sample of calculated train loss curves can be seen in Fig. 3. The differences are based on the different hyperparameters chosen at the beginning of each run.



Fig. 2. Examples of visualised predictions that led to liquid detection and masks creation.

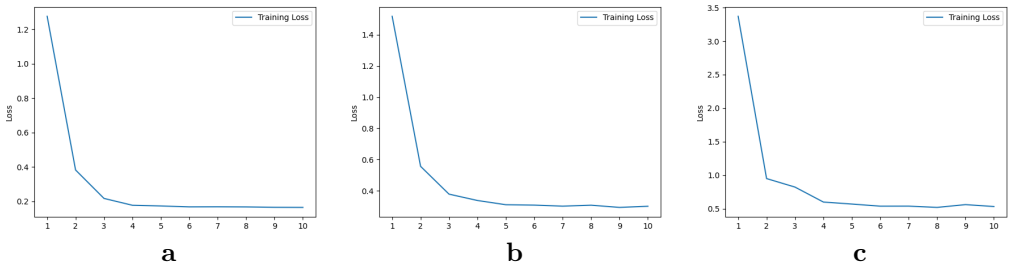


Fig. 3. The training loss curves for runs with different parameters: (a) Batch Size = 2, RL = 0.001, loss value: 0.165; (b) Batch Size = 5, RL = 0.001, loss value: 0.301; (c) Batch Size = 10, RL = 0.005, loss value: 0.532.

To measure the prediction algorithm results, an evaluation method based on MS COCO [8] and available in PyTorch Computer Vision library [14] was used. The method provides 10 classification quality measures of mean average precision and recall for both bounding box and segmentation masks detections.

To decide whether the prediction should be considered a True Positive case usually a threshold is set for the Intersection over Union (IoU) of the predicted and ground truth regions. In our study we have used a set of thresholds, starting from 0.5 up to 0.95. So, a prediction with the value of IoU higher than the selected threshold is considered as a True Positive one. Based on the True Positive, False Positive and False Negative

results, the average precision (AP) and average recall (AR) with different thresholds is calculated.

For evaluation purposes, ten IoU thresholds between 0.5 and 0.95 with step of 0.05 were used to calculate the mean values for average precision (mAP) and average recall (mAR):

$$\text{mAP}_{\text{COCO}} = \frac{\text{AP}_{0.50} + \text{AP}_{0.55} + \dots + \text{AP}_{0.95}}{10} \quad (1)$$

The evaluation method takes under consideration the size of the objects and divide them into 3 groups: small with size lower than 32×32 pixels; medium with size between 32×32 pixels and 96×96 pixels; large with size higher than 96×96 pixels.

The measures mAP and mAR are presented below and were calculated both for bounding boxes and masks.

Mean average precision

- mean average precision (mAP) over 10 consecutive IoU thresholds, from 0.5 to 0.95, step 0.05 (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95) for all scales;
- mAP at IoU = .50 for all scales;
- mAP at IoU = .75 for all scales;
- mAP over 10 consecutive IoU thresholds for small scale;
- mAP over 10 consecutive IoU thresholds for medium scale;
- mAP over 10 consecutive IoU thresholds for large scale.

Mean average recall

- mean average recall (mAR) over 10 consecutive IoU thresholds, from 0.5 to 0.95, step 0.05 (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95) for all scales;
- mAR over 10 consecutive IoU thresholds for small scale;
- mAR over 10 consecutive IoU thresholds for medium scale;
- mAR over 10 consecutive IoU thresholds for large scale.

As it was mentioned before the algorithm was run 100 times with different sets of parameters. Each run was repeated 10 times, which ends up with 1000 runs. To calculate and select optimal parameters for the algorithm, the results of 5 measures of classification quality were taken under consideration, which are: Bounding Box Average Precision, Bounding Box Average Recall, Segmentation Average Precision, Segmentation Average Recall and training loss. First four measures were calculated over 10 IoU consecutive thresholds and provides general precision and recall of the model. Based on the results there is no set of parameters that provides outstanding results in comparison to other sets, however there is a clear trend. Tables 1 and 2 present sets of five best and five worst results, respectively, based on loss value.

Tab. 1. Five best results based on the loss value. No.: batch no.; BS: batch size; LR: learning rate; bb mAP: bounding box mean average precision, bb mAR: bounding box mean average recall, s mAP: segmentation mean average precision, s mAR: segmentation mean average recall; TL: training loss.

| No. | BS | LR | bb mAP | bb mAR | s mAP | s mAR | TL |
|-----|----|-------|--------|--------|-------|-------|-------|
| 4 | 1 | 0.004 | 0.839 | 0.878 | 0.873 | 0.901 | 0.117 |
| 5 | 1 | 0.005 | 0.850 | 0.886 | 0.888 | 0.914 | 0.117 |
| 6 | 1 | 0.006 | 0.840 | 0.871 | 0.886 | 0.906 | 0.120 |
| 7 | 1 | 0.007 | 0.844 | 0.875 | 0.887 | 0.908 | 0.117 |
| 8 | 1 | 0.008 | 0.859 | 0.886 | 0.890 | 0.910 | 0.119 |

Tab. 2. Five worst results based on the loss value. For denotations see Tab. 1.

| No. | BS | LR | bb mAP | bb mAR | s mAP | s mAR | TL |
|-----|----|-------|--------|--------|-------|-------|-------|
| 71 | 8 | 0.001 | 0.413 | 0.622 | 0.571 | 0.813 | 0.500 |
| 81 | 9 | 0.001 | 0.433 | 0.646 | 0.588 | 0.827 | 0.478 |
| 82 | 9 | 0.002 | 0.436 | 0.630 | 0.596 | 0.834 | 0.463 |
| 91 | 10 | 0.001 | 0.432 | 0.611 | 0.580 | 0.790 | 0.489 |
| 92 | 10 | 0.002 | 0.443 | 0.641 | 0.586 | 0.817 | 0.465 |

It seems that batch size has a highest impact on the results achieved during algorithm training and testing and the lower the value is, the better results are achieved.

Learning rate has lower impact but it seems that the best results can be achieved with learning rate different from minimal or maximal values. The full table with all runs can be found in the repository [4].

The results present averaged outputs of the model from 10 runs of testing dataset with selected set of parameters. The detailed measures showing results for different IoU and area values was presented in Table 3 and 4. The tables show the outputs from a sample run with following parameters: batch size = 1, learning rate = 0.005, epochs number = 10, dataset split = 70%.

5. Conclusion

The achieved results seem to be promising. They present high values of the classification quality measures. Moreover, the results show a clear relation between batch size, learning rate and achieved outcomes. Used dataset is recorded with a high resolution camera which is expected to be used in real case scenario. The liquids in that case will be located in the close range to the camera, therefore the evaluation shall be focused on

Tab. 3. Mean Average Precision: mAP, and Mean Average Recall: mAR, for bounding boxes.

| Measure | IoU | Area | Value |
|---------|-----------|--------|-------|
| mAP | 0.50:0.95 | all | 0.855 |
| mAP | 0.50:0.95 | small | 0.790 |
| mAP | 0.50:0.95 | medium | 0.756 |
| mAP | 0.50:0.95 | large | 0.947 |
| mAR | 0.50:0.95 | all | 0.877 |
| mAR | 0.50:0.95 | small | 0.820 |
| mAR | 0.50:0.95 | medium | 0.782 |
| mAR | 0.50:0.95 | large | 0.964 |

Tab. 4. Mean Average Precision: mAP, and Mean Average Recall: mAR, for segmentation masks.

| Measure | IoU | Area | Value |
|---------|-----------|--------|-------|
| mAP | 0.50:0.95 | all | 0.899 |
| mAP | 0.50:0.95 | small | 0.819 |
| mAP | 0.50:0.95 | medium | 0.847 |
| mAP | 0.50:0.95 | large | 0.978 |
| mAR | 0.50:0.95 | all | 0.908 |
| mAR | 0.50:0.95 | small | 0.833 |
| mAR | 0.50:0.95 | medium | 0.853 |
| mAR | 0.50:0.95 | large | 0.982 |

metrics considering detections over large scale objects (96×96 pixels and more). The mean average recall and precision values for segmentation received in this study were very high and close to 90%. The mean average precision and recall values for bounding boxes were a bit lower and reached 86% and 88%, respectively.

In the next stages of the research it is planned to use the current results as a benchmark and to test other algorithms with modified architecture to identify those with best performance, and to increase the dataset with images comprising new liquids and surfaces on which the substances are spilled.

References

- [1] awsaf49. COCO 2017 Dataset, 2017. <https://www.kaggle.com/datasets/awsaf49/coco-2017-dataset>, [Accessed: May 2023].
- [2] A.-A. Dalal, Y. Shao, A. Alalimi, and A. Abdu. Mask R-CNN for geospatial object detection. *International Journal of Information Technology and Computer Science (IJITCS)*, 12(5):63–72, 2020. doi:10.5815/ijitcs.2020.05.05.
- [3] G. Gawdzik. Liquid dataset. PIAP Cloud Resources, 2023. <https://cloud.piap.pl/index.php/s/ApiXNzt4ZUUSRks>, [Accessed: 10 Dec 2023].

- [4] G. Gawdzik. Table of results for all runs of the liquid detection. PIAP Cloud Resources, 2023. <https://cloud.piap.pl/index.php/s/cE3mJ8CrCYWUkJ9>, [Accessed: 10 Dec 2023].
- [5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In: *Proc. 2017 IEEE international Conference on Computer Vision (ICCV)*, pp. 2980–2988. Venice, Italy, 22–29 Oct 2017. doi:10.1109/ICCV.2017.322.
- [6] W. Jia, Y. Tian, R. Luo, Z. Zhang, J. Lian, et al. Detection and segmentation of overlapped fruits based on optimized Mask R-CNN application in apple harvesting robot. *Computers and Electronics in Agriculture*, 172:105380, 2020. doi:10.1016/j.compag.2020.105380.
- [7] H. Jung, B. Lodhi, and J. Kang. An automatic nuclei segmentation method based on deep convolutional neural networks for histopathology images. *BMC Biomedical Engineering*, 1:24, 2019. doi:10.1186/s42490-019-0026-8.
- [8] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, et al. Detecion Evaluation. In: *COCO. Common Objects in Context* [10]. [Accessed: Dec 2023]. <https://cocodataset.org/#detection-eval>.
- [9] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, et al. Microsoft COCO: Common Objects in Context. *arXiv*, 2015. ArXiv:1405.0312. doi:10.48550/arXiv.1405.0312.
- [10] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, et al., eds. *COCO. Common Objects in Context*, 2020. [Accessed May 2023], <https://cocodataset.org>.
- [11] Anaconda, Inc. Conda Documentation, 2023. <https://docs.conda.io>, [Accessed: 10 Dec 2023].
- [12] CVAT.ai Corporation. CVAT Open Data Annotation Platform, 2023. <https://www.cvat.ai>, [Accessed: 10 Dec 2023].
- [13] PyTorch Foundation, a project of The Linux Foundation. PyTorch Get Started, 2023. <https://pytorch.org/>, [Accessed: 10 Dec 2023].
- [14] TorchVision maintainers and contributors. TorchVision: PyTorch’s Computer Vision library. GitHub repository, 2016. <https://github.com/pytorch/vision>, [Accessed: Jun 2023].
- [15] R. S. Olson, W. La Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore. PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(36):1–13, 2017. doi:10.1186/s13040-017-0154-4.
- [16] J. van Rijn, J. Vanschoren, B. Bischl, M. Feurer, G. Casalicchio, et al. OpenML Datasets. <https://www.openml.org/search?type=data>, [Accessed: Dec 2023].
- [17] J. D. Romano, T. T. Le, W. La Cava, J. T. Gregg, D. J. Goldberg, et al. Penn Machine Learning Benchmarks. <https://epistasislab.github.io/pmlb/>.
- [18] J. D. Romano, T. T. Le, W. La Cava, J. T. Gregg, D. J. Goldberg, et al. PMLB v1.0: an open-source dataset collection for benchmarking machine learning methods. *Bioinformatics*, 38(3):878–880, 2021. doi:10.1093/bioinformatics/btab727.
- [19] J. D. Romano, T. T. Le, W. La Cava, J. T. Gregg, D. J. Goldberg, et al. PMLB v1.0: an open source dataset collection for benchmarking machine learning methods. *arXiv*, 2021. ArXiv:2012.00058v2. doi:10.48550/arXiv.2012.00058.
- [20] D. Sculley, J. Moser, W. Cukierski, J. Rose, M. O’Connell, et al. Kaggle Datasets, 2023. <https://www.kaggle.com/datasets>, [Accessed: Jan 2023].
- [21] S. Sibirtsev, S. Zhai, M. Neufang, J. Seiler, and A. Jupke. Mask R-CNN based droplet detection in liquid–liquid systems, Part 2: Methodology for determining training and image processing parameter values improving droplet detection accuracy. *Chemical Engineering Journal*, 473:144826, 2023. doi:10.1016/j.cej.2023.144826.

- [22] H. Su, S. Wei, M. Yan, C. Wang, J. Shi, et al. Object detection and instance segmentation in remote sensing imagery based on precise Mask R-CNN. In: *Proc. 2019 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1454–1457. Yokohama, Japan, 28 Jul – 2 Aug 2019. doi:10.1109/IGARSS.2019.8898573.
- [23] P. Su, J. Joutsensaari, L. Dada, M. A. Zaidan, T. Nieminen, et al. New particle formation event detection with Mask R-CNN. *Atmospheric Chemistry and Physics*, 22(2):1293–1309, 2022. doi:10.5194/acp-22-1293-2022.

Grzegorz Gawdzik, M.Sc., Eng., graduated from Warsaw University of Life Sciences – SGGW, Faculty of Forestry, with specialization in GIS and 3D scanning. Currently works as a researcher in the Security and Defence Systems Division of Łukasiewicz – PIAP and a Ph.D. student at Warsaw University of Life Sciences – SGGW. His major research interests are: 3D scanning and environment reconstruction; perception for situational awareness; deep learning; pattern recognition; image detection and segmentation; machine learning; reinforcement learning; robotic manipulation.

IRIS RECOGNITION BASED ON LOCAL GREY EXTREMUM VALUES WITH CNN-BASED APPROACHES

Kamil Malinowski * and Khalid Saeed 

Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland

**Corresponding author: Kamil Malinowski (kamil.malinowski@sd.pb.edu.pl)*

Abstract One of the most important steps in the operation of biometric systems based on iris recognition of the human eye is pattern comparison. However, the comparison of the recorded pattern with the pattern stored in the database of the biometric system cannot function properly without effective extraction of key features from the iris image. In the presented work, we propose an iris recognition system based on image feature extraction and extreme grey shade analysis. Harris-Laplace, RANSAC and SIFT descriptor algorithms were used to find and describe key points. In the experimental part, two methods were used to compare descriptors: the Brute Force method and the Siamese Network method. IIT Delhi Iris Database (version 1.0), MMU v2 database, UBIRIS v1, UBIRIS v2 image databases were used for the study. The proposed method utilizes a different approach when using the generalized corner extraction algorithm (Harris-Laplace algorithms) for comparing iris patterns. In addition, we prove that the use of the descriptor and the Siamese neural networks significantly improves the results obtained in the original method based on paths alone in the case of well contrasted infrared images with very low resolutions.

Keywords: biometrics, iris, grey extremum values, encoding.

1. Introduction

The iris is the opaque structure of the human eye. It is an element of the uveal membrane, located between the lens and the cornea. Its central element of variable diameter is called the pupil. Each iris has a pattern of discoloration, lining, and folds [1].

When looking for the optimal choice of a given biometric feature as an identification tool, many different factors should be considered. Each of the solutions used today has its strengths and weaknesses. One of the most important advantages of iris-based imaging systems is the statistically low rate of recognition errors. When making a choice, one should consider the adequacy of the solution to the satisfying needs related to identification. Nowadays, biometrics is the most important element in capturing systems. In addition, two-component systems, using biometrics and traditional solutions, are widely available and easy to use. An example may be the increasingly common biometric passports. It has become common to use biometrics in mobile devices, smartphones, tablets, laptops, where the standard is to install fingerprint readers and the iris of the eye. Biometrics is also becoming present in banking. More and more banks are working on the implementation of biometric systems as an effective identity control for customers, thus increasing the same level of security for their services. The iris pattern is very distinctive

and may not be sufficient to *uniquely* distinguish people. There are varieties of works that have already shown evidence of aging of the iris pattern [2].

Identity verification solutions based on biometric methods are used to control access to resources, also fulfilling the role of blocking unauthorized access attempts.

There are various neural network architectures that have been used for iris pattern recognition. Each of them has drawbacks that are worth discussing.

One of the main challenges of UniNet [3] is its accuracy, which largely depends on the quality of the data entered into it. A study by Zao et al. [3] shows that the accuracy of the UniNet algorithm in iris recognition is about 98.4%, so there is still room for improvement. Another disadvantage of this technology is the need for proper lighting and positioning of the eye, which can be difficult in some situations, such as performing identification at long distances or in low light, as discussed in the work of Hajari et al. [4].

The disadvantage of DRFNet [5] and GraphNet [6] is that they require a large amount of training data, which can be difficult to obtain. In addition, their implementation can require significant computing power.

Iris recognition using Siamese neural networks and point descriptors is one of the modern approaches in the iris recognition problem. The idea behind the creation of Siamese neural networks was to develop a method for comparing similarities for very complex data samples. The advantage of this method is the ability to compare data samples that have different characteristics and types. A well-constructed Siamese network is able to indicate subtle differences between two seemingly identical data samples [7].

When combined with point descriptors such as SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Features), ORB (Oriented Fast and Rotated BRIEF) [8,9,10,11], Siamese networks can create a highly accurate iris recognition system. The point descriptors are responsible for detecting characteristic points on the image and generating vectors describing these points. One advantage of this approach is that point descriptors can be used to recognize the iris, even if it is not fully visible in the image. Additionally, Siamese networks with point descriptors have the ability to generalize and can operate with high efficiency on training and test data from different sources [8,9,10,11].

A novel method for image matching using extreme grey shade values and the SIFT descriptor was described in a publication by Zhao et al. [10]. The method is based on the use of extreme grey shade values of the image, which have unique characteristics. The authors proposed using the SIFT descriptor to extract characteristic points on the image and create vectors describing these points. Then, the selected extreme grey shade values are also added to the feature vector. The next step is to use a classification algorithm that can learn to recognize image based on the created feature vectors. The results of the authors' experiments show that the proposed method achieves an efficiency of 99.33% in image recognition. It is worth noting that despite the promising results, the method requires further research and testing on larger data sets to confirm its effectiveness and applicability in real applications.

Iris recognition systems can be vulnerable to fraud attempts, such as trying to present artificially generated iris images. Research focused on increasing resilience to such types of fraud is crucial for enhancing system security. The diversity of methods for comparing iris patterns makes such fraud attempts challenging. The aim of our study was to develop a new method for verifying individuals based on the iris by using properly extracted key points with associated descriptors, which could serve as another alternative to existing methods. Can a method using descriptors of points extracted from paths of extreme value for greyscale be effective in solving the problem of comparing iris patterns? We made a significant improvement to the original algorithm in [12]. Our modification involved extracting key points and analysing the SIFT descriptors of these points using Siamese neural networks. As a result, the algorithm has become resistant to various lighting conditions and changes in the position of the registered object. Information about the iris structure and its characteristics can be extracted from the paths of extreme values for shades of grey, which can be an extension of the methods previously described and can significantly improve their efficiency. The merit of descriptors combined with appropriate extraction of key points is to enhance the features and increase the diversity of iris patterns. The authors noted the great potential and effectiveness of comparing patterns using a technique based on comparing outlier paths based on shades of grey. Unlike the original method, ours proved more efficient for more than one set of irises. The original extracting extreme value paths approach used in proposed method is discussed in detail in Section 3.

2. State of the art

Nowadays, iris extraction is becoming more efficient and accurate thanks to developing technologies. With the increasing number of available iris databases and developing extraction algorithms, it is possible to achieve very high accuracy in iris recognition. One of the most important developments in the field of iris extraction is the introduction of methods using artificial neural networks. Neural networks make it possible to recognize irises more accurately and quickly, which contributes to the efficiency of this method.

However, it is worth remembering that iris extraction is still a process that requires high precision and accuracy. Many factors must be taken into account, such as image quality, distance from the camera and the health of the eye, in order to obtain accurate results. Therefore, research is still being conducted to develop more efficient iris extraction methods and to improve the quality of data sets.

The current trend in research on comparing iris patterns is the use of CNN (Convolutional Neural Network). A disadvantage of using CNN is its sensitivity to the quality of training data. Researchers use artificial neural networks in various ways to solve the problem of iris recognition. Lee et al. [13] used three CNN (Convolutional Neural Network) models for extracting features from images of the iris. The developed model uses

a non-square filter, and each CNN model is composed of eight convolutional layers and three fully connected layers. In this method, two additional regions are extracted – iris and periorbital region containing information about the shape of eyelids, eyebrows, and skin colour. From these regions and blurred and normalized iris, feature vectors are extracted and compared using SVM (Support Vector Machine). The proposed solution is sensitive to eyelid shape change, light reflection noise, and eyelashes. Yang et al. [14] pre-trained ResNet-18 model was used as an encoder of the created system – as an encoder skeleton for extraction of multi-level features. High-level functions have made it possible to capture more contextual information. The low-and high-level functions are combined by the Spatial Awareness Function Combine (SAFFM) module. Minimum Shifted and Masked Distance (MMSD) is used to compare the encoded irises. The authors achieved the Equal Error Rate (EER) factor of 0.27% for the developed method. Chen et al [15] used proposed method called NSNet (convolutional neural network based on the attention mechanism). Raw image without iris extraction was taken as input for feature extraction and recognition. The average EER (Equal Error Rate) factor is 0.343%. Winston et al. [16] tried to solve the problem of limited availability of data sets, which has a direct impact on the accuracy of classifiers. They have empirically proved that Adam based optimization is good at learning iris features using deep learning. According to the conducted research, the hybrid network of deep learning with SVM is the most appropriate method of recognizing the patterns of the iris of the eye, reaching the accuracy of 97.8%. Liu et al. [17] using image blur with three filters increased the accuracy of the methods of recognizing iris patterns using deep learning techniques. Chen et al. [18] is another work that uses CNN to compare the irises of the eye. The proposed method used a novel loss function called T-Center loss to enhance the discriminant ability of deep models. To avoid the gradient explosions and identify the appropriate hyperparameter, their approach simultaneously normalizes the feature vectors and feature center vectors. Despite the sensitivity of fuzzy, mirror reflections and reflections confirmed by the authors, the method gave satisfactory results. Liu et al. [19] 2-channel CNNs were used to recognize the iris. In the 2-channel CNN, the authors introduced four key innovations, including a large-scale hybrid iris identification and verification framework, a radial attention layer for weighing different regions of the iris, online expansion schemes to increase resilience, and structural reduction to lower computational load to improve performance. Ahmadi et al. [20] proposed a method based on two-dimensional Gabor kernel (2-DGK), polynomial filtering, and step filtering to solve the problem of iris recognition. The accuracy of the method is 95.36%. The same authors tried to improve their work and hence in [21], they proposed an algorithm using “hybrid radial basis function neural network (RBFNN) with genetic algorithm (GA) for matching task” and obtained an accuracy of 99.99%, but this time the procedure took much time (860.70 s).

Classic methods of iris pattern recognition have several advantages compared to those

using CNN. One of them is the requirement of incomparably fewer computational resources and memory. They are also simpler to understand and implement independently. Wang et al. [22] used an improved algorithm, based on wavelet packet transformation, to improve the iris recognition. The article uses the db4 wavelet base and Shannon entropy to decompose a normalized iris image. The iris recognition system uses Hamming distances. The authors declare the recognition effectiveness of the developed method at 96.3%. Bala et al. [23] the authors managed to improve the method based on the Xor-Sum Code (IXSC), allowing it to be used to recognize the iris both in the visible and infrared light. The EER ratio is at the level of 8.27%. Galdi et al. [24] proposed a multi-classifier based on three descriptors: colour, texture, and clusters. The method achieved an EER of 0.29. The authors have released the source code of the method they developed. This made it possible to compare the results obtained by us. Lv et al. [25] in their method used an odd symmetric 2D Log-Gabor filter to analyse the phase and amplitude of the iris texture in relation to different frequencies and orientations, and use feature fusion to eliminate noise. Abbasi et al. [26] using a binary genetic algorithm, they choose the best combination of various wavelet transforms, Fourier transforms, and Gabor filter. The proposed method has achieved a FAR (False Acceptance Rate), of 0 and a FRR (False Rejection Rate) of 0.092. Barpanda et al. [27] to extract iris features in their method they use wavelets from the Cohen-Daubechies-Feauveau 9/7 filter bank. This method has been improved [28] by using the Mel-frequency cepstral coefficients (MFCCs) to differentiate iris tissues. Gad et al. [29] segment the iris using the Delta-Mean (DM) method proposed by them. At the stage of extracting the features of the iris, an algorithm is used that combines the frequency and location of the features – multi-algorithm mean. The average accuracy of the algorithm is 99.48% while EER is 0.28. Yao et al. [30] using Harr and log-Gabor transforms, they achieved a recognition accuracy of 95%.

3. Proposed method

The goal of our research was to create a new method for comparing human iris patterns using known and publicly available algorithms. The algorithm should detect subtle differences in data samples, which would allow it to more precisely and effectively recognize iris patterns recorded in various environments. We proposed to use the Harris-Laplace, RANSAC (Random Sample Consensus) and SIFT descriptor algorithms to find and describe key points in the iris. The extracted key points of the studied iris are compared to those in the database using two Brute Force methods and Siamese neural networks. The proposed method is unique and easy to implement. For the sake of our goal, we used three iris bases in our research, namely IIT Delhi Iris Database (Version 1.0) [31], MMU.v2 database [32], UBIRIS v1 [33]. Since the iris images in the aforementioned databases were recorded in a restricted environment, we chose the UBIRIS v2 database [34] to

compare the performance of proposed method in an unrestricted environment. In the above-mentioned collection of the irises of the eye, the outer and inner boundaries are not always a perfect circle, which is directly related to the angle at which the image was recorded. Therefore, the assumption that the center of the pupil of the eye is located in the center of the captured image may be a mistake. For the initial determination of the pupil area, we used the diagrams developed in our previous articles [35,36]. Images from UBIRIS v1 [33] have been converted to greyscale. Proposed method does not take into account iris rotation. Unfortunately, in the case of the UBIRIS v2 database [34], classical iris segmentation algorithms such as the one developed by us are not effective enough for the proposed iris pattern recognition method to work properly. For this database, we used the method developed by Omar et al. [37]. For the other bases, we describe below the iris extraction method we developed in previous articles.

The boundaries of the iris in images recorded at an angle other than a right angle are shaped like an ellipse. To eliminate image distortion and convert the elliptical boundaries of the iris into a circle in the area of the pupil, a rectangle is circumscribed. Having information about three points lying at the vertices of the rectangle (i'_1, j'_1) , (i'_2, j'_2) , (i'_3, j'_3) , we are able to find the affine transformation of this object into a square:

$$\begin{aligned} (i_1, j_1) &\rightarrow (i'_1, j'_1), \\ (i_2, j_2) &\rightarrow (i'_2, j'_2), \\ (i_3, j_3) &\rightarrow (i'_3, j'_3). \end{aligned} \tag{1}$$

For the points belonging to the vertices of the square (i_1, j_1) , (i_2, j_2) , (i_3, j_3) , one should find the transformation coefficients $(a_{00}, a_{01}, a_{02}, a_{10}, a_{11}, a_{12})$ by solving the system of equations (2):

$$\begin{bmatrix} i_1 \\ j_1 \\ i_2 \\ j_2 \\ i_3 \\ j_3 \end{bmatrix} = \begin{bmatrix} i'_1 & j'_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & i'_1 & j'_1 & 1 \\ i'_2 & j'_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & i'_2 & j'_2 & 1 \\ i'_3 & j'_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & i'_3 & j'_3 & 1 \end{bmatrix} \begin{bmatrix} a_{00} \\ a_{01} \\ a_{02} \\ a_{10} \\ a_{11} \\ a_{12} \end{bmatrix}. \tag{2}$$

In such a transformed image, determining the pupil center (x_0, y_0) inscribed in the square becomes a trivial task. The next stage of isolating the iris of the eye is to identify two points $(x_1, y_1)(x_2, y_2)$ lying on the outer border of the iris [34]. From the indicated points and the pupil center, the radius of the circle is determined, to which these points belong:

$$R = \frac{\sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2} * \sin\left(\frac{\pi}{2} - \tan^{-1}\left|\frac{y_1 - y_2}{x_1 - x_2}\right|\right)}{\sin\left(\pi - 2\left(\frac{\pi}{2} - \tan^{-1}\left|\frac{y_1 - y_2}{x_1 - x_2}\right|\right)\right)}. \tag{3}$$

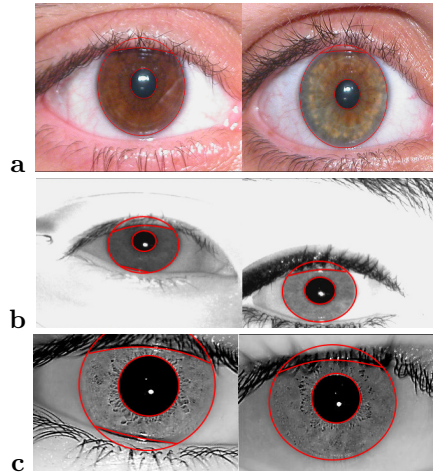


Fig. 1. Iris segmentation – (a) UBIRIS v1 [33], (b) MMU.v2 database [32], (c) IIT Delhi Iris Database (Version 1.0) [31].

The result of determining the boundaries of the outer and inner iris of the eye is shown in Fig. 1.

Iris normalization is aimed at transforming the area of the iris separated at an earlier stage into an area of constant size, regardless of the previously separated area of the iris. Obtaining consistent sizes is essential for the iris comparison procedure. Normalization ensures resistance to discrepancies in the size of the irises caused by the dilating pupil – resulting from different environmental conditions in which the image was recorded. All of images have been reduced to the size of 240×340 pixels.

In this work, the conversion of the Cartesian coordinates (x, y) to coordinates in the non-concentric polar system (p, θ) was used:

$$\begin{aligned} p &= \log \sqrt{(x - x_c)^2 + (y - y_c)^2}, \\ \theta &= \text{atan2}(y - y_c, x - x_c), \end{aligned} \quad (4)$$

where (x_c, y_c) – pupil center coordinates.

The result of applying the mathematical transformation to the iris image is an image with a constant size of 240×60 pixels (Fig. 2). To enhance texture details of the iris, we used adaptive histogram equalization (CLAHE).

Proposed algorithm is based on changes in the intensity of points in stripes of constant size. As in the work of Rathgeb et al. [12], point intensity paths are extracted.

The pre-processed iris image I is divided into 15 stripes with a width of 4 points

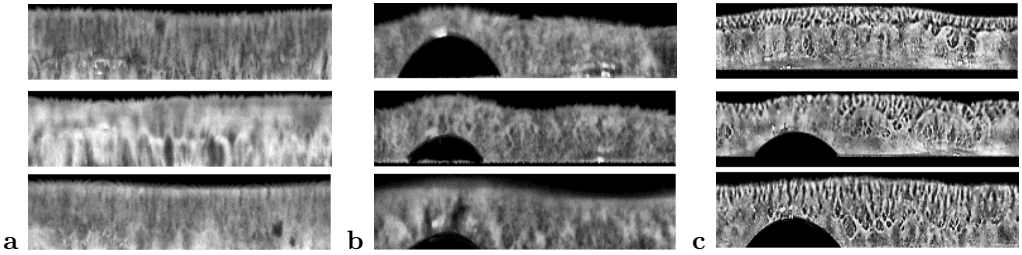


Fig. 2. Modified iris image of the eye – (a) UBIRIS v1 [33], (b) MMU.v2 database [32], (c) IIT Delhi Iris Database (Version 1.0) [31].

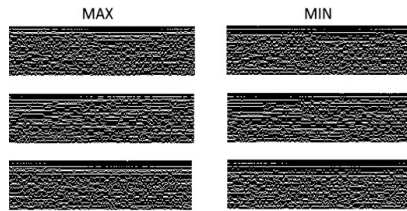


Fig. 3. Examples of extraction of paths of the variability of point values – maximum, minimum.

according to the formula (5) as in the work of Rathgeb et al. [12]:

$$I \rightarrow \{I_1, I_2, \dots, I_{15}\}. \tag{5}$$

From each iris image of the eye, 15 paths are extracted for points with maximum and minimum values in each of the strips P_L and P_H (6).

$$P_L = \left\{ \begin{matrix} P_{L1} \\ P_{L2} \\ P_{L3} \\ \dots \\ P_{L15} \end{matrix} \right\}, \quad P_H = \left\{ \begin{matrix} P_{H1} \\ P_{H2} \\ P_{H3} \\ \dots \\ P_{H15} \end{matrix} \right\}. \tag{6}$$

The detected paths are shown in Figure 3. Black marker without texture corresponds to the area covered by eyelashes, eyelids – these are areas of complete blackness or areas of white colour.

In the described method, points extracted from the paths of maximum and minimum values from the iris image are analysed. The path is formed by the local extremes of grey shade values, excluding the maximum – white colour and minimum – black colour. Thus, the analysis is carried out in relation to the extremes of grey shades. Selected conventional techniques support the process of minimizing the elements to be analysed,

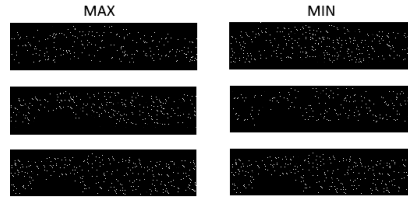


Fig. 4. Examples of key point extraction prior to RANSAC application.

making the process more efficient and comparable in effectiveness to the other techniques discussed in the manuscript.

Proposed method is based on the search for key points in the images shown in Fig. 3. Classic feature extraction methods by means of a key point detector use a region descriptor around each of the detected points. The main purpose of the descriptors is to isolate the characteristics of the information near each key point. Based on previous studies [38, 39], experiments were carried out to select the best feature descriptor from among the descriptors (SIFT – Scale-Invariant Feature Transform, Principal Component Analysis PCA-SIFT, GLOH – Gradient Location and Orientation Histogram, SURF – Speeded Up Robust Features).

In this study, we used a method based on Harris-Laplace [40] (7) and SIFT keypoint descriptor [41] algorithms. The features obtained using SIFT are constant in scale and rotation of the image. The SIFT descriptor creates a vector of the values of the orientation histogram in the region of each key point. These quantities are determined by the gradient and the orientation around the key points. The use of Laplacian-of-Gaussian makes the detected points resistant to changes of scale σ_1 and can be detected in the image after resizing it. The combination of the above algorithms ensures repeatability of features and scale invariant fit.

$$R = g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha [g(I_x^2) + g(I_y^2)]^2, \quad (7)$$

where α was experimentally set at 0.42. Meanwhile, R values greater than 4.8 indicate a detected corner. I_x and I_y are the respective derivatives in the x and y direction applied to the smoothed image and calculated using a Laplacian-of-Gaussian (LoG) filter g with scale $\sigma_D = 8\sigma_1$. The σ_1 parameter determines the current scale at which the Harris corner points are detected.

Only the key points in the images of the extracted paths are subjected to further analysis. Figure 4 shows the key points for each path.

The feature matching process is to find matching points on the recorded image and the pattern in the database. Once the points and their descriptors have been extracted, the goal is to find consistent matches across all the iris images. We introduced location restrictions [39]. Applying localization constraints reduces the time needed to process

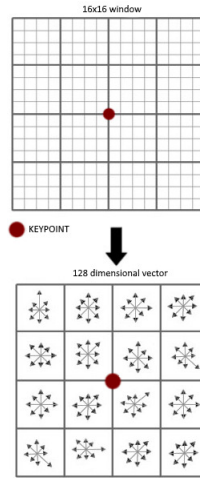


Fig. 5. Localization of a key-point.

the descriptors and prevents false matches. The iris area has been divided into four sections (Fig. 8). The point descriptors are in the same areas regardless of possible scale differences. We used the RANSAC algorithm to eliminate points with erroneous features. Experimentally and through the analysis of research on descriptor comparison methods [42], we decided to use the Brute Force method.

In the Brute Force method, the descriptors from all features must be matched to the descriptors of all features in another image. This is an extremely time-consuming solution. The method guarantees obtaining a solution without any guarantee that the solution is optimal. The Brute Force method uses the Euclidean distance between two descriptors. A smaller distance d_v indicates greater similarity between two points (8).

$$d_v(v_1, v_2) = \sqrt{\sum (v_1 - v_2)^2}, \tag{8}$$

where v_1, v_2 – two feature description, SIFT feature descriptor will be a vector of 128 elements (16 blocks \times 8 values from each block – Figures 5-6).

Although previous studies have shown that feature extraction methods are resistant to cluttered images [40], we decided to remove areas of the iris obscured by the eyelids using a method developed by us [36]. This procedure allows us to increase the quality of detected key points. Figure 7 shows a block diagram of the method we propose.

The percentage of similarity P between two images can be calculated using formula (9):

$$P = |CF|/|TF|, \tag{9}$$

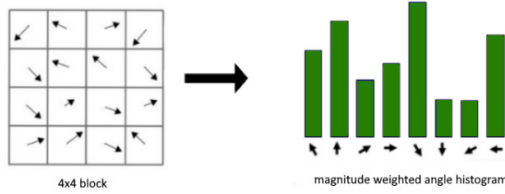


Fig. 6. Structure of a single block 4 × 4.

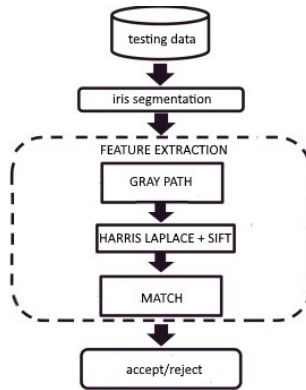


Fig. 7. A block diagram of the proposed method.

where CF is the correctly matched features after applying the RANSAC algorithm, TF is the total number of matches. A P value closer to 1 indicates a high degree of similarity between the analysed irises.

The problem of correctly matching the iris pattern to the correct person can be generalized to the multiclassification problem known from deep learning methods. Considering the drawbacks of the Brute Force method, we decided to use the Siamese Network [43] for iris pattern classification. The structure of a Siamese network can be compared to two other neural networks working side by side. Both networks have the same structure



Fig. 8. Division of key points into four areas of equal width after applying RANSAC.

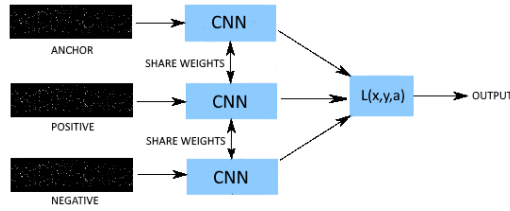


Fig. 9. Siamese Network structure used to compare SIFT descriptors.

and the same weights. These networks are then combined using a function that calculates a measure of similarity or distance. The structure of our Siamese network is shown in Figure 9.

The Siamese Network output aims to measure the similarity between two feature vectors obtained from CNNs. We were guided by the hypothesis that descriptors describing extracted points of the same iris will have similar feature vectors, which is equivalent to a small distance between them. Similarity between feature vectors can be measured using multiple distance metrics. During the training phase of the convolutional Siamese Network, we used the Triplet Loss Function:

$$L(x, y, a) = \max(0, d(a, x) - d(a, y) + m), \quad (10)$$

where two iris descriptor vectors of the same person and an iris descriptor vector of another person are selected randomly. The vectors of iris descriptors belonging to the same person are considered similar, so one is used as an anchor a and the other as a positive x , while the vector of iris descriptors of another person is considered negative, m is a margin value to keep negative samples far apart. In this paper, we used the CNN network architecture proposed in [44] shown in Fig. 10. The neural network architecture was chosen because of the high similarity of our input signal to the one used in the aforementioned work. The magnitude weighted angle histogram obtained from each point can be written in the form of a one-dimensional vector, which in turn is a kind of equivalent of recording the signal path – wave (Fig. 11). The input vector is created by starting from the point closest to the upper-left corner of the image, and then adding points located on the same path towards the right edge. This process is applied to each path.

4. Experimental result

The aim of the experiments was to achieve accurate iris pattern classification results using descriptors of points extracted from paths of extreme value for greyscale. The selection of the similarity value is crucial for the correct decision to confirm or reject

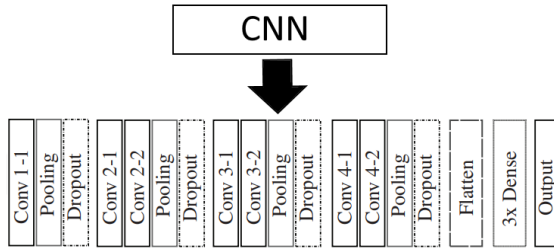


Fig. 10. Diagram of a single CNN network structure.

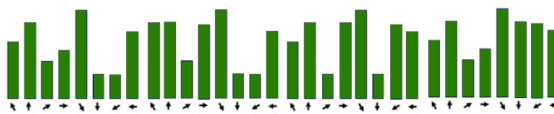


Fig. 11. Graph of SIFT descriptors points as a one-dimensional vector.

the user’s identity. The estimated similarity threshold separating the two results of the biometric verification has been considered. The verification result may or may not match the pattern.

Determining the optimal value for the similarity threshold required image analysis for two types of collections of the iris of the eye. The trials were made on the irises of the same people (mated-comparison) and on the irises of different people (non-mated comparison). Images of the left and right eyes of the same person were treated as if they belonged to two different people. The value of the similarity P threshold of which the images of irises are considered to be from the same person was experimentally set at 0.38. Data from all the databases were used to determine the P threshold. The choice of the P threshold is illustrated by the graph shown in Fig. 12.

For experimental purposes, we made our own implementation of the algorithms: Wang et al. [22], Yao et al. [30], Rathgeb et al. [12]. All these algorithms were tested under the same experimental conditions.

We first analysed the method using the Brute Force technique.

Figure 13 shows the result of comparing the irises of the same people (mated-comparison) with the calculated similarity coefficient. On the other hand, Figure 14 shows the result of comparing the irises of different people (non-mated comparison). Figures show the key points detected. All images from each of the IIT Delhi Iris Database (Version 1.0) [31], MMU.v2 database [32], UBIRIS v1 [33], UBIRIS v2 [34], databases were selected for the experiments.

To evaluate the performance of the proposed system, the recognition (accuracy)

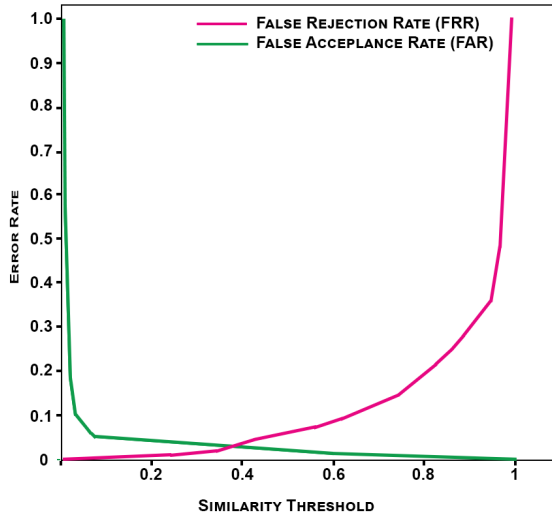


Fig. 12. Diagram illustrating an experiment to select a P threshold.

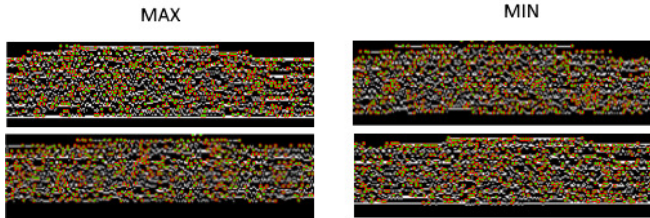


Fig. 13. Comparison of key points for the paths of maximum values ($P = 0.6072$) and minimum values ($P = 0.731$) – mated-comparison.

coefficient was used as an evaluation parameter (11).

$$ACC = (TP + TN) / (TP + TN + FN + FP), \tag{11}$$

where TP – true positive recognition, TN – true negative recognition, FP – false positive recognition, FN – false negative recognition. The above-mentioned AAC parameter ranges from 0, (meaning perfectly correct recognition) to 1, meaning error.

In the second part of our experiment, we used the Siamese Network. The images from each base were divided into a training set and a test set at a ratio of 80% to 20%. The division was applied to each of the classes present in the test sets. In addition, transformations of the source images such as rotation, vertical, and horizontal reflection, zoom and shift along the X or Y axis were used in the testing phase.

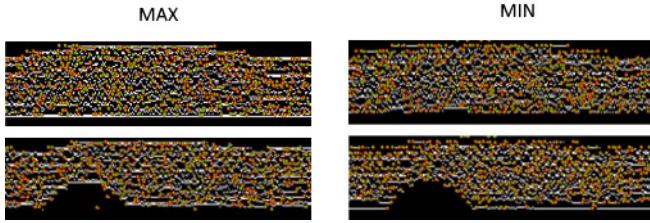


Fig. 14. Comparison of key points for the paths of maximum values ($P = 0.0273$) and minimum values ($P = 0.0911$) – non mated-comparison.

Each class thus contained 15 iris images – 5 images for each input. The Siamese Network uses data augmentation. For this purpose, simple geometric transformations were used – shift, reflection, tilt. The crucial element is the error of validation and training of the network. If the error decreases, the training should continue. If the validation error starts to increase, there is a high probability of over-fitting. It is therefore necessary to set the highest possible number of epochs (e.g., 100 epochs) and, based on the error rates, terminate the training. An epoch is one learning cycle in which the entire training data set is visible. A large number of epochs can result in improved precision up to a certain limit, beyond which the model becomes over-fitted to the data. A small number of epochs, on the other hand, can result in an inappropriate fit to the data. We observed that above 40 epochs, the model does not improve.

The value of m in Eq. (10) must be chosen experimentally and depends on the domain of application. The value of $m = 1$ was experimentally determined. In Table 1, we have presented the results of experiments to determine the optimal m parameter.

The neural network must have correctly prepared data. One of the most important rules is that the input data must have the same size.

In our neural network proposal, we used a 4×1 kernel with 6 to 64 filters. Small kernels can extract much more information from the input data containing highly local functions. The smaller kernel size also leads to a smaller reduction in the dimensions of the layers, allowing for deeper architecture. Other parameters of our network – max pooling with a pool size of 2 and stride 2 and utilize dropout of value 0.23 between the pooling and convolutional layers. The dropout method is very efficient, because in every pass the connections are randomly turned off. This ensures that the neural

Tab. 1. Values of parameter m with corresponding accuracy of Siamese Network.

| m | 0.5 | 0.75 | 1 | 1.25 | 1.5 |
|--|-----|------|----|------|-----|
| Accuracy of the Siamese Network (%) | 88 | 92 | 97 | 95 | 89 |

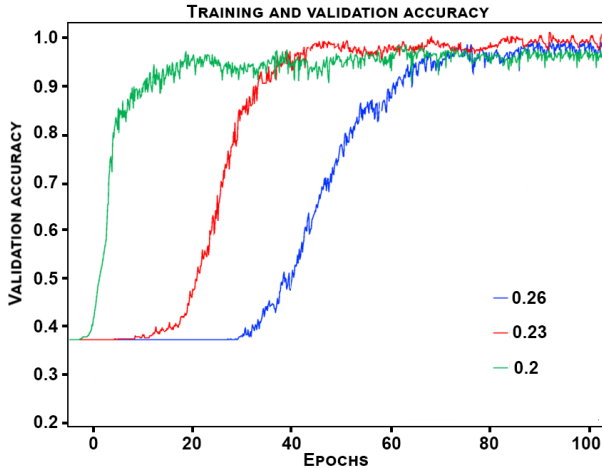


Fig. 15. Experiment results for different values of the dropout parameter.

network does not learn “by heart” too quickly, because the architecture changes a little bit every recalculation by resetting the random connections of the neurons. During the experiments, we tested the parameter dropout in ranges of 0.20 to 0.26, with a step of 0.03. The results of these experiments can be seen in Figure 15.

Table 2 presents the results of measuring the execution time of individual stages of the algorithm we developed. Table 3 presents a comparison of the proposed method to the three methods described at the beginning of this article. To measure execution time, we used BenchmarkDotNet [45]. The test platform was a computer equipped i7-11700K CPU (central processing unit) with NVIDIA TITAN RTX GPU (graphics processing unit) was used.

The decision on the quality of the biometric verification method is assessed by the values of the FAR, FRR and EER coefficients. A wrong acceptance may indicate a security hole, while an unfair rejection becomes embarrassing for the legitimate user. The compromise to the above is the EER factor. The ROC (receiver operating characteristic) curve allows us to determine and indicate the efficiency of biometric comparators, maintaining a compromise between the FAR and FRR coefficients.

We divided our experiments into two groups. In the first experimental group, we analysed and compared the performance of our chosen algorithms for images recorded in the IIT Delhi Iris Database (Version 1.0) [31], MMU.v2 database [32], UBIRIS v1 [33]. These are collections of images recorded in a limited environment. On the other hand, in the second experimental group, we used UBIRIS v1 database [33] comparing the effectiveness of the algorithm with UniNet [3], DRFNet [5] and GraphNet [6]. The

results of the first phase of our experiments are presented below. The authors repeated the experiments 30 times, and the reported results are the arithmetic means.

FRR is the ratio of the false negatives to the sum of the true positive and false negative. The test allowed the FRR ratio at the level of 1.80% (Brute Force) and 0.70% (Siamese Network) to be determined. The FAR coefficient for the implemented algorithm was determined at the level of 2.80% (Brute Force) and 2.10% (Siamese Network). FAR is the ratio of the false positive to the sum of the false positive and true negative.

In contrast, Table 4 shows a comparison of the accuracy of the proposed method for each database. A similar comparison for each of the tested bases for other methods is presented in Table 5.

The maximum execution time of the algorithm is just over two seconds (2229.10 ms, 2178.30 ms), the shortest time was less than a second (125.50 ms, 178.30 ms). This made it possible to obtain an average time of one second (1077.85 ms, 1015.30 ms) for the Brute Force and Siamese Network methods, respectively.

In the second phase of testing, we used the previously discussed neural networks comparing the results they obtained for images from the UBIRIS v2 database [34].

In a paper by Zao et al. [3] investigated the effectiveness of the UniNet neural network in dissecting the iris of the eye using on images recorded in infrared light – in our test we

Tab. 2. Time complexity of the proposed method.

| Algorithm step | Time [ms] | | |
|--------------------------------|---------------|----------------|----------------|
| | Minimum | Maximum | Average |
| Locating and Segment Iris | 104.00 | 2161.00 | 1000.00 |
| Normalization | 2.00 | 3.00 | 2.50 |
| Encoding | 8.20 | 10.30 | 9.25 |
| Match (Brute Force) | 11.30 | 54.80 | 66.10 |
| Match (Siamese Network) | 3.10 | 4.00 | 3.55 |
| Total (Brute Force) | 125.50 | 2229.10 | 1077.85 |
| Total (Siamese Network) | 117.30 | 2178.30 | 1015.30 |

Tab. 3. Time complexity, accuracy, EER of the proposed method with other known algorithms (average value) – IIT Delhi Iris Database (Version 1.0) [31], MMU.v2 database [32], UBIRIS v1 [33].

| Algorithm | Time (s) | Accuracy (%) | EER (%) |
|-----------------------------|----------------|--------------|-------------|
| Wang et al. [22] | 1.13700 | 95.30 | 0.60 |
| Yao et al. [30] | 1.08820 | 91.00 | 0.91 |
| Rathgeb et al. [12] | 1.14990 | 86.00 | 1.17 |
| Proposed method (BF) | 1.07785 | 97.74 | 0.26 |
| Proposed method (SN) | 1.01530 | 98.70 | 0.17 |

used the UBIRIS v2 database [34]. For iris recognition, UniNet can use various image processing techniques such as edge detection, segmentation and normalization of iris images. The network can also take into account different lighting conditions and iris positions to ensure recognition performance.

Tab. 4. Comparison of accuracy, EER, FAR, FRR of the proposed method for each used database.

| Database | Accuracy (%) | EER (%) | FAR (%) | FRR (%) |
|---|--------------|---------|---------|---------|
| Brute Force | | | | |
| MMU.v2 database [32] | 97.31 | 0.34 | 2.60 | 2.90 |
| IIT Delhi Iris Database (Version 1.0) [31] | 98.70 | 0.19 | 2.40 | 0.60 |
| UBIRIS v1 [33] | 97.20 | 0.26 | 3.40 | 1.80 |
| Siamese Network | | | | |
| MMU.v2 database [32] | 98.60 | 0.20 | 2.40 | 1.30 |
| IIT Delhi Iris Database (Version 1.0) [31] | 99.20 | 0.13 | 1.30 | 0.30 |
| UBIRIS v1 [33] | 98.20 | 0.20 | 2.50 | 0.60 |

Tab. 5. Comparison of accuracy and EER of Wang et al. [22], Yao et al. [30], Rathgeb et al. [11] method for each used database (average value).

| Database | Accuracy (%) | EER (%) | FAR (%) | FRR (%) |
|---|--------------|---------|---------|---------|
| Wang et al. [22] | | | | |
| MMU.v2 database [32] | 95.10 | 0.20 | 2.40 | 0.50 |
| IIT Delhi Iris Database (Version 1.0) [31] | 97.80 | 0.23 | 2.20 | 0.64 |
| UBIRIS v1 [33] | 93.00 | 0.17 | 2.80 | 1.40 |
| Yao et al. [30] | | | | |
| MMU.v2 database [32] | 86.50 | 1.70 | 2.60 | 2.10 |
| IIT Delhi Iris Database (Version 1.0) [31] | 98.50 | 0.12 | 1.80 | 0.30 |
| UBIRIS v1 [33] | 88.00 | 0.91 | 2.30 | 1.50 |
| Rathgeb et al. [12] | | | | |
| MMU.v2 database [32] | 88.20 | 0.71 | 1.60 | 1.10 |
| IIT Delhi Iris Database (Version 1.0) [31] | 91.60 | 0.67 | 0.90 | 1.00 |
| UBIRIS v1 [33] | 78.20 | 2.13 | 1.70 | 3.10 |

DRFNet [5] is another neural network model used for iris recognition. This network consists of several blocks with convolution layers, ReLU (Rectified Linear Unit), Batch Normalization, Pooling layers and Global Average Pooling. The entire network is also based on recursive layer pooling.

The latest GraphNet neural network model for iris recognition [6] is based on a graph-based data structure. It consists of two main blocks: a feature extraction block and a classification block. The feature extraction block uses convolutional neural networks to extract important iris features. The classification block uses graph data structure for accurate classification.

In Table 6 we have presented a comparison of the proposed method to the methods presented above for the UBIRIS v2 database [34]. The tests were performed using pre-trained neural networks.

Figures 16-17 show the relation between FPR as the False Positive Rate against the TPR as the True Positive Rate (12).

$$TPR = TP/(TP + FN), \quad FPR = FP/(TN + FP) \quad (12)$$

Images in which the iris was more obscured by eyelids or eyelashes gave the algorithm [12] more problems. The algorithm [12] was able to correctly extract the correct points needed to create paths with extreme values. Our proposed modification is resistant to the above-mentioned problems. Examples of these images are shown in Figure 18. Our CNN model reaches almost 100% accuracy and as one can see the network training should finish at the 40th epoch (Fig. 19, Fig. 20); increasing the training period does not significantly affect the network quality. Fig. 20 shows the training curves for all three networks, which are part of our Siamese Network.

5. Discussion

The iris extraction algorithm proposed by [12] is sensitive to light reflections occurring near the centre of the pupil which causes inaccurate segmentation of the iris. In addition,

Tab. 6. Time complexity, accuracy of the proposed method with other known algorithms UniNet [3], DRFNet [5], GraphNet [6] – UBIRIS v2 [33].

| Algorithm | Time (ms) | Accuracy (%) | EER (%) |
|-----------------------------|-------------|--------------|-------------|
| UniNet [3] | 6.10 | 99.32 | 0.08 |
| DRFNet [5] | 5.80 | 99.36 | 0.06 |
| GraphNet [6] | 6.70 | 99.24 | 0.11 |
| Proposed method (SN) | 5.90 | 99.15 | 0.18 |

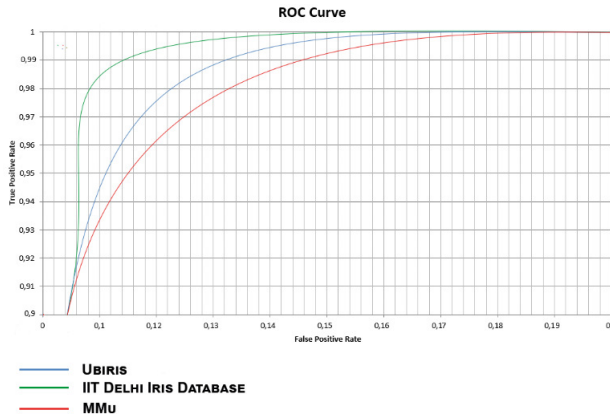


Fig. 16. ROC curve (Brute Force) – UBIRIS v1 [33], MMU.v2 database [32], IIT Delhi Iris Database (Version 1.0) [31].

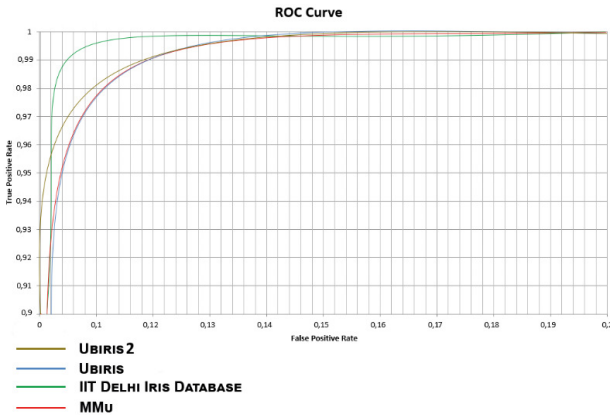


Fig. 17. ROC curve (Siamese Network) – UBIRIS v2 [34], UBIRIS v1 [33], MMU.v2 database [32], IIT Delhi Iris Database (Version 1.0) [31].

the path extraction method does not eliminate all extreme values, which causes large inaccuracies when comparing iris patterns. The elimination of noisy areas in the paper [12] is based on the elimination of the area where noise, due to eyelashes and eyelids, is most likely to occur, without considering noise in other areas of the iris. The extracted iris areas are subjected to the Gaussian blur algorithm, which also does not guarantee getting rid of extreme values from the extracted area. The experiments prove that, in the case of analysing the paths of extremes of grey point values, it is enough to analyse the

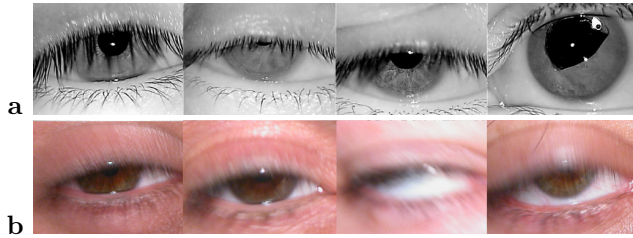


Fig. 18. Example of images which are discarded in our experimentations for our segmentation method – (a) UBIRIS v1 [33], (b) IIT Delhi Iris Database (Version 1.0) [31].

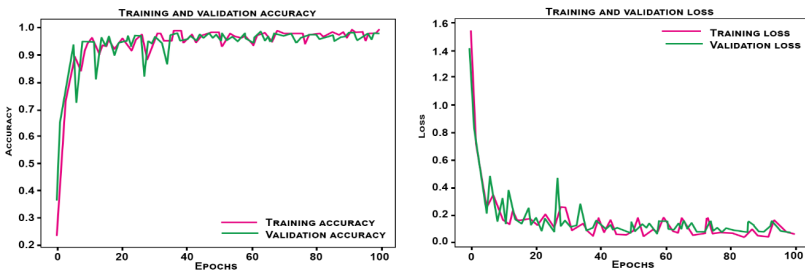


Fig. 19. Accuracy and loss curves for all databases.

appropriately extracted points together with their descriptors. This procedure reduces the amount of data required for analysis.

In the second method we reproduced [30], feature extraction was performed using the Haar wavelet transform, and classification was done by clustering the wavelet feature data using the K-means method. Local iris texture features were extracted using a Log-Gabor filter. The method produced similar results in all tested databases and proved inferior to the proposed method.

In both algorithms [30] and [12], a small fragment of the iris area is analysed, which distinguishes the two approaches from the one proposed here and appears to be an inferior approach to solving the iris recognition problem.

The proposal to eliminate areas obscured by eyelids and eyelashes using rigidly chosen parameters [30] [12] is less effective for highly noisy images. The experiments we conducted prove that using the entire iris area gives better results [22].

The EER was also studied, as it is a compromise between the convenience and effectiveness of the biometric recognition system. The EER measure is determined using the FRR and FAR ratios discussed above. A system with a lower EER is more accurate. The EER value indicates that the proportion of false acceptances is equal to the proportion of false rejections. The average value of the EER 0.26 (Brute Force) coefficient achieved

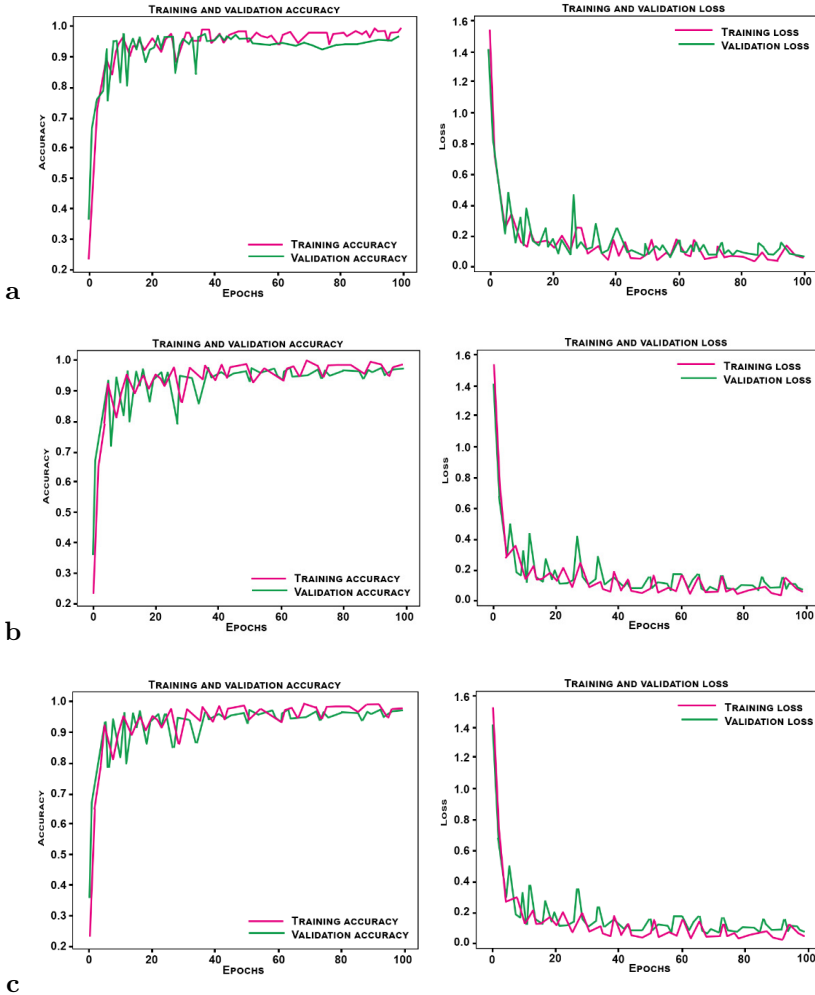


Fig. 20. Accuracy and loss curves – (a) anchor, (b) positive, (c) negative.

by us is comparable to other methods described in the introduction. If we look at the individual iris databases in more detail, it can be noticed that in the case of IIT Delhi Iris Database (Version 1.0) [31], we managed to achieve an EER value of 0.24. This is the best result of all the previously discussed works.

However, the obtained mean accuracy value of proposed method (Brute Force), equal

to 97.74% and a variant of proposed method using the Siamese Network achieved efficiency of 98.70% with an EER of 0.17. The Siamese Network proved to be almost 20 times faster than the Brute Force.

Overall, the IIT Delhi Iris Database (Version 1.0) [31] shows the best results. This is mainly because the IIT Delhi Iris Database (Version 1.0) [31] presents well-contrasted images with appropriate resolution for feature extraction methods based on key points, although this database is strongly disrupted by eyelid and lash occlusions. The MMU.v2 database [32] has a lower image resolution. The UBIRIS v1 [33] database is noisier in terms of lighting, motion blur, tilt angle and viewing direction. We therefore can claim that the proposed method is the most stable and has the highest performance in the three databases considered. Moreover, by comparing the results with those presented in Tables 2, 3, 4 one can see the benefits of using our suggested modification of the algorithm in [12] and indicates the superiority of image analysis methods using CNNs over methods using traditional image processing. The proposed method is better than the methods proposed by Wang et al. [22], Yao et al. [30], Rathgeb et al. [12].

The UBIRIS v2 database [34] is one of the largest and most diverse databases of irises from different individuals. This diversity allows testing the performance of iris recognition under different conditions, such as varying lighting, different cameras, and different iris positions. In addition, it is one of the most popular and widely used iris databases for testing recognition algorithms. Tests on the UBIRIS v2 database [34] have shown the great potential of proposed algorithm. Compared to other neural networks, the result obtained is minimally inferior. The improvement over previously tested databases may be due to a different approach to the iris segmentation problem. On the other hand, a better algorithm execution time was obtained from UniNet [3] and GraphNet [6] algorithms. The execution time of proposed algorithm was 5.90 ms, which was only 0.10 ms worse than that of the DRFNet algorithm [5]. The EER parameter of 0.18 achieved in the test demonstrates the high quality of proposed algorithm.

6. Conclusions

The use of the method of recognizing the iris of the eye with the use of encoding with extreme values of shades of grey and the use of the Harris-Laplace algorithm [40] and SIFT keypoint descriptor [41] Siamese Network [43] gave promising results. The achieved EER, FRR and FAR coefficients allow us to conclude that the proposed method retained a compromise between the efficiency and the speed of comparison of patterns. Through our verification process, we have determined that the utilization of Siamese neural networks in combination with SIFT descriptors serves as a viable alternative to other existing methods, as described in the literature, which rely on point descriptors and neural networks for iris recognition.

Our research shows that the introduction of simple components to methods developed

by other authors [12] allows to significantly improve the quality of these algorithms, providing modern results in the field of iris biometrics. We provide conclusive evidence that valuable information pertaining to the structure and characteristics of the iris can indeed be successfully extracted from the paths of extreme values for different shades of grey. This approach can be considered as an extension of the previously described iris recognition methods, with the added benefit of significantly enhancing their efficiency and effectiveness.

Unfortunately, the method turned out to be less effective in the case of images recorded in visible light and heavily noisy. The method achieves the best results with well-contrasted images. Proposed algorithm can be implemented on more bases. Experiments have proved the effectiveness of the method on images captured in visible light and on images captured in infrared light (weakly and strongly contrasted). The algorithm is unable to properly extract paths if there are large areas with similar values. In this case, the paths overlap, causing distortions in the final stage of extracting these paths.

Future work will focus on the use of artificial intelligence to dynamically determine the degree of similarity and to extract high and low value paths, from noisy images in particular. The described method is applicable in the conditions of the tested image databases. Subsequent studies will focus on the possibilities of applying the method in different environments and image capturing conditions. In addition, the authors intend to eliminate the drawbacks of the developed method, so that it is effective for iris images captured with, for example, smartphones [47] with account eye iris rotation (use more challenging databases). Images captured with an SLR (Single-Lens Reflex) camera that is several years old [32] are characterized by minimal chromatic aberrations and provide sharp, crisp images even from a very short distance compared to even the latest smartphones [46]. Images captured with a DSLR (Digital Single-Lens Reflex) camera are usually deliberately underexposed, lacking saturation and digital overexposure, so that later in further processing we can adequately stretch the tonal space and enhance what we care most about, highlighting in this case the subtle differences between iris points. Images of the iris, taken in infrared light, have additional information about the iris pattern. In the work of Hosseini, et al. [47], an extensive comparison was made between the visible-light and infrared iris registration methods.

References

- [1] Willoughby, C. E., Ponzin, D., Ferrari, S., Lobo, A., Landau, K., Omid, Y. (2010). Anatomy and physiology of the human eye: effects of mucopolysaccharidoses disease on structure and function – A review. *Clinical & Experimental Ophthalmology*, 38:2–11. doi:10.1111/j.1442-9071.2010.02363.x
- [2] Das, P., Holsopple, L., Rissacher, D., Schuckers, M., Schuckers, S. (2021). Iris Recognition Performance in Children: A Longitudinal Study. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(1):138–151. doi:10.1109/TBIOM.2021.3050094
- [3] Zhao, Z., Kumar, A. (2017). Towards more accurate iris recognition using deeply learned spatially

- corresponding features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3809–3818. doi:10.1109/ICCV.2017.411
- [4] Hajari, K., Gawande, U., Golhar, Y. (2016). Neural network approach to iris recognition in noisy environment. *Procedia Computer Science*, 78:675–682. doi:10.1016/j.procs.2016.02.116
- [5] Wang, K., Kumar, A. (2019). Toward more accurate iris recognition using dilated residual features. *IEEE Transactions on Information Forensics and Security*, 14(12):3233–3245. doi:10.1109/TIFS.2019.2913234
- [6] Ren, M., Wang, Y., Sun, Z., Tan, T. (2020, April). Dynamic graph representation for occlusion handling in biometrics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, No. 07, pp. 11940–11947. doi:10.1609/aaai.v34i07.6869
- [7] Chicco, D. (2021). Siamese Neural Networks: An Overview. In: Cartwright, H. (eds). *Artificial Neural Networks. Methods in Molecular Biology*, Vol. 2190. Humana, New York, NY. https://doi.org/10.1007/978-1-0716-0826-5_3
- [8] Rathgeb, C., Wagner, J., Busch, C. (2019). SIFT-based iris recognition revisited: prerequisites, advantages and improvements. *Pattern Analysis and Applications*, 22:889–906. doi:10.1007/s10044-018-0719-y
- [9] Tareen, S. A. K., Raza, R. H. (2023, March). Potential of SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, and 7 More Algorithms for Matching Extremely Variant Image Pairs. In *Proceedings of the 2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pp. 1–6. IEEE. doi:10.1109/iCoMET57998.2023.10099250
- [10] Zhao, Y., Zhai, Y., Dubois, E., Wang, S. (2016). Image matching algorithm based on SIFT using color and exposure information. *Journal of Systems Engineering and Electronics*, 27(3), 691–699. doi:10.1109/JSEE.2016.00072
- [11] Liu, C., Xu, J., Wang, F. (2021). A review of keypoints’ detection and feature description in image registration. *Scientific Programming*, 1–25, 2021. doi:10.1155/2021/8509164
- [12] Rathgeb, C., Uhl, A. (2010, June). Secure iris recognition based on local intensity variations. In *Proceedings of the International Conference Image Analysis and Recognition (ICIAR)*, pp. 266–275. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-13775-4_27
- [13] Lee, M. B., Kang, J. K., Yoon, H. S., Park, K. R. (2021). Enhanced iris recognition method by generative adversarial network-based image reconstruction. *IEEE Access*, 9:10120–10135. doi:10.1109/ACCESS.2021.3050788
- [14] Yang, K., Xu, Z., Fei, J. (2021). Dualsanet: Dual spatial attention network for iris recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 889–897. doi:10.1109/WACV48630.2021.00093
- [15] Chen, Y., Zeng, Z., Gan, H., Zeng, Y., Wu, W. (2021). Non-segmentation frameworks for accurate and robust iris recognition. *Journal of Electronic Imaging*, 30(3):033002. doi:10.1117/1.JEI.30.3.033002
- [16] Winston, J. J., Hemanth, D. J., Angelopoulou, A., Kapetanios, E. (2021). Hybrid deep convolutional neural models for iris image recognition. *Multimedia Tools and Applications*, 81(7):9481–9503. doi:10.1007/s11042-021-11482-y
- [17] Liu, M., Zhou, Z., Shang, P., Xu, D. (2019). Fuzzified image enhancement for deep learning in iris recognition. *IEEE Transactions on Fuzzy Systems*, 28(1):92–99. doi:10.1109/TFUZZ.2019.2912576
- [18] Chen, Y., Wu, C., Wang, Y. (2020). T-center: A novel feature extraction approach towards large-scale iris recognition. *IEEE Access*, 8:32365–32375. doi:10.1109/ACCESS.2020.2973433

- [19] Liu, G., Zhou, W., Tian, L., Liu, W., Liu, Y., Xu, H. (2021). An efficient and accurate iris recognition algorithm based on a novel condensed 2-ch deep convolutional neural network. *Sensors*, 21(11):3721. doi:10.3390/s21113721
- [20] Ahmadi, N., Akbarizadeh, G. (2018). Hybrid robust iris recognition approach using iris image pre-processing, two-dimensional gabor features and multi-layer perceptron neural network/PSO. *IET Biometrics*, 7(2):153–162. doi:10.1049/iet-bmt.2017.0041
- [21] Ahmadi, N., Nilashi, M., Samad, S., Rashid, T. A., Ahmadi, H. (2019). An intelligent method for iris recognition using supervised machine learning techniques. *Optics & Laser Technology*, 120:105701. doi:10.1016/j.optlastec.2019.105701
- [22] Wang, Y., Zheng, H. (2021, February). An improved Iris recognition method based on wavelet packet transform. *Journal of Physics: Conference Series*, Vol. 1744, No. 4, p. 042239. IOP Publishing. doi:10.1088/1742-6596/1744/4/042239
- [23] Bala, N., Vyas, R., Gupta, R., Kumar, A. (2021). Iris Recognition Using Improved Xor-Sum Code. In *Proceedings of the Conference on Security and Privacy*, pp. 107–117. Springer, Singapore. doi:10.1007/978-981-33-6781-4_9
- [24] Galdi, C., Dugelay, J. L. (2017). FIRE: Fast Iris REcognition on mobile phones by combining colour and texture features. *Pattern Recognition Letters*, 91:44–51. doi:10.1016/j.patrec.2017.01.023
- [25] Lv, L., Yuan, Q., Li, Z. (2019). An algorithm of Iris feature-extracting based on 2D Log-Gabor. *Multimedia Tools and Applications*, 78(16):22643–22666. doi:10.1007/s11042-019-7551-2
- [26] Abbasi, M. (2019). Improving identification performance in iris recognition systems through combined feature extraction based on binary genetics. *SN Applied Sciences*, 1(7):1–14. doi:10.1007/s42452-019-0777-9
- [27] Barpanda, S. S., Sa, P. K., Marques, O., Majhi, B., Bakshi, S. (2018). Iris recognition with tunable filter bank based feature. *Multimedia Tools and Applications*, 77(6):7637–7674. doi:10.1007/s11042-017-4668-z
- [28] Barpanda, S. S., Majhi, B., Sa, P. K., Sangaiah, A. K., Bakshi, S. (2019). Iris feature extraction through wavelet mel-frequency cepstrum coefficients. *Optics & Laser Technology*, 110:13–23. doi:10.1016/j.optlastec.2018.03.002
- [29] Gad, R., Talha, M., Abd El-Latif, A. A., Zorkany, M., Ayman, E. S., Nawal, E. F., Muhammad, G. (2018). Iris recognition using multi-algorithmic approaches for cognitive internet of things (CIoT) framework. *Future Generation Computer Systems*, 89:178–191. doi:10.1016/j.future.2018.06.020
- [30] Liping, Y., Zhongliang, P. (2019). Iris recognition method based on Harr wavelet and Log-Gabor transform [J]. *Application of Electronic Technique*, 45(4):113–117. doi:10.16157/j.issn.0258-7998.183173
- [31] Kumar, A. and Passi, A (2010). Comparison and combination of iris matchers for reliable personal authentication, *Pattern Recognition*, 43(3):1016–1026. doi:10.1016/j.patcog.2009.08.016
- [32] MMU v2 MMU Iris Database, Malaysia Multimedia University, <http://andyzeng.github.io/downloads/MMU2IrisDatabase.zip>. [Dataset, accessed 1 November 2021]
- [33] Proença, H. and Alexandre, L. A., 2005, September. UBIRIS: A noisy iris image database. In *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, pp. 970–977. Springer, Berlin, Heidelberg. doi:10.1007/11553595_119
- [34] [dataset] Proença, H., Filipe, S., Santos, R., Oliveira, J. and Alexandre, L. A. (2010). The UBIRIS.v2: A Database of Visible Wavelength Iris Images Captured On-The-Move and At-A-Distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):1529–1535. doi:10.1109/TPAMI.2009.66

- [35] Malinowski, K., Saeed, K. (2022). An Efficient Algorithm for Boundary Detection of Noisy or Distorted Eye Pupil. In *Proceedings of the Conference on Advanced Computing and Systems for Security*, Vol. 13, pp. 51–59. Springer, Singapore. doi:10.1007/978-981-16-4287-6_4
- [36] Malinowski, K., Saeed, K. (2022). An iris segmentation using harmony search algorithm and fast circle fitting with blob detection. *Biocybernetics and Biomedical Engineering*, 42(1):391–403. doi:10.1016/j.bbe.2022.02.010
- [37] Moslhi, O. M. (2020). New full Iris Recognition System and Iris Segmentation Technique Using Image Processing and Deep Convolutional Neural Network. *International Journal of Scientific Research in Multidisciplinary Studies*, 6(3):20–27. https://www.isroset.org/journal/IJSRMS/full_paper_view.php?paper_id=1775
- [38] Malgheet, J. R., Manshor, N. B., Affendey, L. S., Abdul Halin, A. B. (2021). Iris Recognition Development Techniques: A Comprehensive Review. *Complexity*, 2021. doi:10.1155/2021/6641247
- [39] Alvarez-Betancourt, Y., Garcia-Silvente, M. (2016). A keypoints-based feature extraction method for iris recognition under variable image quality conditions. *Knowledge-Based Systems*, 92:169–182. doi:10.1016/j.knosys.2015.10.024
- [40] Tuytelaars, T., Mikolajczyk, K. (2008). *Local invariant feature detectors: A survey*. Now Publishers Inc. doi:10.1561/9781601981394
- [41] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. doi:10.1023/B:VISI.0000029664.99615.94
- [42] Noble, F. K. (2016, November). Comparison of OpenCV’s feature detectors and feature matchers. In *Proceedings of the 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pp. 1-6. IEEE. doi:10.1109/M2VIP.2016.7827292
- [43] Chopra, S., Hadsell, R., LeCun, Y. (2005, June). Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, Vol. 1, pp. 539–546. IEEE. doi:10.1109/CVPR.2005.202
- [44] Schuetzke, J., Benedix, A., Mikut, R., Reischl, M. (2020, November). Siamese Networks for 1D Signal Identification. In *Proceedings of the 30. Workshop on Computational Intelligence*, Berlin, 26-27 November, 2020. Vol. 26, p. 17. KIT Scientific Publishing. doi:10.5445/KSP/1000124139
- [45] .NET Foundation and contributors. BenchmarkDotNet version 0.13.1, Copyright (c) 2013-2021, <https://benchmarkdotnet.org/>
- [46] De Marsico, M., Nappi, M., Riccio, D., Wechsler, H. (2015). Mobile iris challenge evaluation (MICHE)-I, biometric iris dataset and protocols. *Pattern Recognition Letters*, 57:17–23. doi:10.1016/j.patrec.2015.02.009
- [47] Hosseini, M. S., Araabi, B. N., Soltanian-Zadeh, H. (2010). Pigment melanin: Pattern for iris recognition. *IEEE Transactions on Instrumentation and Measurement*, 59(4):792–804. doi:10.1109/TIM.2009.2037996




Prof. Khalid Saeed is a full Professor of Computer Science at Bialystok University of Technology and a half-time visiting professor at Universidad de La Costa, Barranquilla, Colombia. He was with Warsaw University of Technology in 2014-2019 and with AGH Krakow in 2008-2014. He received his B.Sc. Degree from Baghdad University in 1976, M.Sc. and Ph.D. (distinguished) Degrees from Wroclaw University of Technology in Poland in 1978 and 1981, respectively. He received his D.Sc. Degree (Habilitation) in Computer Science from the Polish Academy of Sciences in Warsaw in 2007. He was nominated by the President of Poland for the title of Professor in 2014. He has published more than 250 publications including about 120 journal papers and book chapters, about 100 peer reviewed conference papers, edited 50 books, journals and Conference Proceedings, written 13 text and reference books (h-index 19 in WoS and 15 in SCOPUS base). He supervised more than 15 Ph.D. and 150 M.Sc. theses. He gave more than 50 invited lectures and keynotes in different universities in Europe, China, India, South Korea, Serbia, Germany, Japan, and Canada, on biometric image processing and analysis. He received more than 30 academic awards. He is also a member of more than 15 editorial boards of international journals and conferences. He was selected as the IEEE Distinguished Speaker, from 2011 to 2016. He is also the Editor-in-Chief of *International Journal of Biometrics* with Inderscience Publishers.



Kamil Malinowski is a Ph.D. student in Computer Science at Bialystok University of Technology. His research focuses on bi-modal biometric systems, where he explores the intersection of computer science and biometrics. With a strong background in both theory and practice, is well-versed in various aspects of biometric data acquisition, image processing, signal processing, biometric feature matching algorithms, performance evaluation of biometric systems, as well as privacy and security concerns related to biometric data. Aside from his academic pursuits, also actively contributes to the field of cybersecurity. He works as a trainer at Altkom Akademia, where he shares his expertise in attack protection, risk management, monitoring, detection, and response to cyber threats. His practical experience in the industry allows him to provide valuable insights and knowledge to trainees, equipping them with the necessary skills to safeguard against cyber threats on a daily basis.

EVENT DETECTION SYSTEM FOR THE PENITENTIARY INSTITUTIONS USING MULTIMODAL DATA AND DEEP NETWORKS

Piotr Bilski ¹, Marcin Lewandowski ¹, Adrian Bilski ^{2,*}, Andrzej Buchowicz ¹,
Jacek Olejnik ³, Paweł Mazurek ¹ and Konrad Jędrzejewski ⁴

¹*Institute of Radioelectronics and Multimedia Technology, Warsaw University of Technology, Warsaw, Poland*

²*Institute of Information Technology, Warsaw University of Life Sciences – SGGW, Warsaw, Poland*

³*JAS Technologie Sp. z o.o., Warsaw, Poland*

⁴*Institute of Electronic Systems, Warsaw University of Technology, Warsaw, Poland*

*Corresponding author: adrian.bilski@sggw.pl

Abstract The aim of the paper is to present the distributed system for the unwanted event detection regarding inmates in the closed penitentiary facilities. The system processes large number of data streams from IP cameras (up to 180) and performs the event detection using Deep Learning neural networks. Both audio and video streams are processed to produce the classification outcome. The application-specific data set has been prepared for training the neural models. For the particular event types 3DCNN and YOLO architectures have been used. The system was thoroughly tested both in the laboratory conditions and in the actual facility. Accuracy of the particular event detection is on the satisfactory level, though problems with the particular events have been reported and will be dealt with in the future.

Keywords: deep learning, posture-based event detection, multimodal analysis

1. Introduction

The contemporary society faces multiple challenges related to the internal and external security. The efficiency of the police and internal affairs agencies influences the national security level, as perceived by the citizens. Modern technologies come with help regarding tasks of monitoring and detecting dangerous or unwanted events (such as robbery, vandalism, or civil unrest) based on the input from the surveillance cameras. The most advanced cities are already supervised by the multiple types of sensors and systems (including the infrared or thermal imaging devices). They are supported at the increasing rate by the Artificial Intelligence (AI), which assists human operators in the proper situation assessment. Image processing coupled with Deep Learning (DL) are able to automatically detect and isolate objects, persons or their specific behavioural traits.

Though such monitoring systems belong to the state-of-the art and are introduced commercially, there are many specific applications where the closed-architecture systems fail. For instance, in the case of the closed penitentiary institutions (such as prisons) the set of the detectable events is very different from the typical scenarios encountered in the outside world. For instance, besides the typical aggressive behaviour (with fights being the most obvious), there are also multiple location-specific events, including the

suicide or escape attempts. In such a location the already existing systems are relatively simple, with most of the responsibility put on the human operator. Introduction of the AI-based modules faces multiple challenges, making the whole endeavour complex and difficult to accomplish.

The aim of the paper is to present the AI-based system for the unwanted (anomalous) event detection based on many multimedia streams delivered to the computing platform. Characteristic features and parameters of the system (considering the predefined requirements) are presented, followed by the detailed description of the solution, both from the hardware and software perspective. The AI part is based on the well-known Artificial Neural Network (ANN) architectures, but manages them in the specific way, applicable to the location it is implemented in. Experimental results (both in the laboratory conditions and in the field) show the system can operate in the real-world conditions and ensures accuracy high enough to support the human operator.

The content of the paper is as follows. In Section 2 the current knowledge about the event-detection technologies is presented. Section 3 contains specific requirements for such a system working in the closed facilities (such as prisons). In Section 4 the architecture of the system and the used DL algorithms are discussed. Experiments and their results are in Section 5. Conclusions about the system's performance and future prospects are in Section 6.

2. State-of-the-Art

Human action recognition in CCTV surveillance systems is a critical aspect for security and monitoring applications [5, 19, 31]. Techniques based on Deep Learning have gained particular importance for video action recognition in recent years [38]. Progress in this field has been significant, although less spectacular than in 2D image analysis. Currently DL are the leading solution due to their high accuracy and ability to mimic visual cortex [6]. Three-dimensional Convolutional Neural Networks (3D CNN), being the extension of the classic 2D CNNs into the third dimension was introduced in [21]. This extension is a basis on which the Convolutional 3D (C3D) neural network [35] was created. Another important implementations of the 3D CNN concept are the Inception 3D (I3D) network [11] and the Squeeze-and-Excitation Layer C3D (SELayer-C3D) model [22]. Various structures of Convolutional Neural Networks (CNN) and Long-Short Term Memory (LSTM) Networks have been proposed to enhance the accuracy of identifying human actions in surveillance footage [13, 27]. Hybrid Deep Neural Networks (DNN) like Convolutional LSTM (ConvLSTM) Networks and Long-term Recurrent Convolutional Networks (LRCN) have been developed to classify actions efficiently, with models utilizing different architectures for spatial-temporal feature extraction. Additionally, the use of Self-Organizing Maps (SOM) based on time-series inference skeletons extracted from the CCTV footage has shown promise in behaviour recognition, especially when

implemented on edge AI processors, demonstrating the potential for real-time action recognition in surveillance systems [27].

Another widely used mechanisms for human action recognition in CCTV surveillance systems are the two-stream networks. They are composed of two processing blocks: the first one is for analyzing the stream of video frames (2D images) while the second one is for analyzing the optical flow [17,20,25] calculated for each video frame. The video action is recognized by combining the texture and motion analysis results [18,33,37]. Recurrent Neural Networks (RNN) are used for temporal modeling of video sequences and determining the dependencies between consecutive video frames [15,28], while Transformer-Based Networks (TBNs) were developed to solve the problem of sequence transduction (any task that transforms an input sequence to an output sequence) [8,16].

The range of events to be detected mainly covers the violent behaviour. In [14] the crowd analysis framework has been discussed, including the large volume of data processing. Here Histogram of Oriented Gradients was used to improve performance of the detection close to the Real-Time conditions. In [4] the anomaly detection scheme is applied for the combination of the ANN and Gaussian distribution. Similarly, in [3] the anomalous behaviour is detected through the YOLO v2 network. In the presented cases the binary classification scheme is employed to distinguish the *nominal* behaviour of monitored persons from anything else. In most cases, the image or videos sequence analysis is based on the complete scene processing in search for anomalies. Also, skeletal structures are used to extract pose estimation, which is characteristic for some events.

In most presented solutions the number of events to detect is strictly limited to single type events. Also, the presented works omit the problem of processing large number of streams. Assuming that the presented architectures are effective enough to be used in practice, this paper shifts towards the ensemble of models to cooperate during the detection of multiple events at the same time. Also, multiple types of media are assumed (visible range video streams, infrared streams, audio analysis).

3. Problem statement

The task of the on-line monitoring of large number of inmates located in the penitentiary institution (or similar location, where, potentially dangerous, individuals are being held captive) creates multiple problems, not critical for the typical visual event-detection framework (like the ones proposed for stadiums). These are issued by the prison facility officers, being the main beneficiaries of the implemented system. They see it as the useful tool supporting officers responsible for monitoring the specific location, by drawing their attention to the particular camera image in the control room. The most important requirements include the following.

- The predefined set of 18 event types for detection, which are difficult or impossible to observe in the outside world. These specifically include fights between inmates, arson,

assault on the officer, riots, escape or suicide attempts, death of the inmate, illegal verbal or non-verbal communication between inmates, or with the outside world, or passing the forbidden object (such as a weapon) into the cell.

- The ability to operate with the predefined time-delays (near real-time mode), allowing for the fast reaction to the dangerous events. Based on the discussion with the officers and criminologists the threshold was set to 5 s.
- The ability to process large number of streams from up to 150 cameras, which is the significant challenge regarding the assignment of computing resources.
- Architecture of the system must apply to the technical infrastructure of the location, regarding the computer network topology, types of wiring and separation from the Internet (which eliminates the usage of the cloud services).
- Specificity of the location making usage of the distributed intelligence (with the video sequences processing next to the cameras) impossible to use (as the inmates often attack and damage the sensors).

These requirements make the construction of the system a difficult task, from both the research and technical perspective (for example, verification of the hardware capable of processing the mentioned number of streams).

The focus of the developed solution was at the detection of undesirable behaviour based on the extraction of features from data streams provided by multiple sensors in the on-line mode (Fig. 2). Each data source is a separate stream of information analyzed at the software level that models the AI-based classifier. The typical camera is capable of working in the visible light and infrared, making possible detection of events during the night (but requires separate training data for AI models). Also, some of them are equipped with microphones, which facilitates events related with sound. In specific locations two cameras are installed. Especially inside the cells, they are located opposite to each other to maximize the chance of detecting the event and eliminate blind spots. This leads to the multimodal analysis of the single scene, as presented in Fig. 1.

For such a task DNN are usually applied to obtain critical features from the image and classify them simultaneously, combined with a specific algorithm for describing the observed scene.

The effectiveness of the AI-based module depends on the number and diversity of available learning patterns (multiple-instance learning). In particular, DNN require significant amounts of versatile information (patterns) to correctly operate on the available data. The system analyzes each event as a set of sequences of numerical values representing one of the assumed anomalous behaviours. In the presented case the input video streams are processed by the classifier, which makes a decision about raising the alarm. Each sequence is a set of image frames provided by the camera with the constant speed, so the architecture of the particular network must be able to operate on the time series patterns. To make the DNN useful, it must be trained specifically to the solved tasks, which requires the individual data set preparation. All video sequences must then

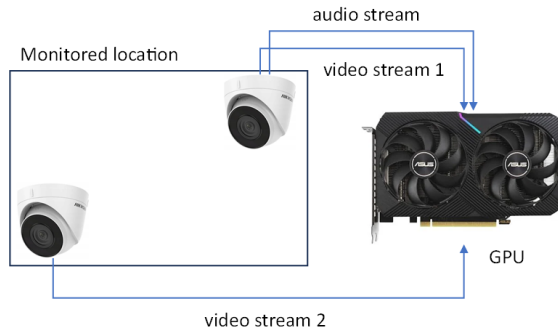


Fig. 1. Typical single scene analysis module, including video and audio streams (where applicable).

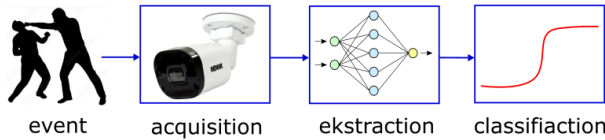


Fig. 2. Event detection by an artificial intelligence algorithm based on images obtained from a video camera.

be divided into those showing the unwanted events and the ones representing normal behaviour. Labeling of the sequences is therefore the crucial step in the proper development of the classification software. Processing schemes for traditional image processing based on feature analysis and processing using DL are presented in Fig. 3.

The presented problem may be treated as the binary or multi-category detection task. The latter is the more complex and from the practical point of would not be relevant,

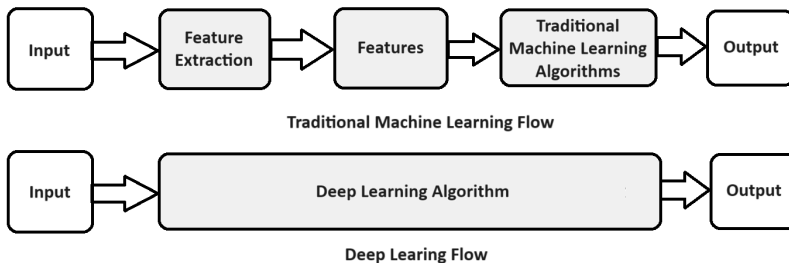


Fig. 3. The difference between the traditional machine learning-based input classification model and the deep learning version.

therefore the former was selected. In this case all data are labeled with two categories: positive (unwanted event) or negative (nominal situation). The actual nature or cause of the event are not relevant here, as the main task would be to focus the officers' attention to the particular location (while the subsequent code of conduct will depend on the nature of the actual event). This way the classifier would be only responsible for detecting any of the predefined events with the maximum possible accuracy. Consequences of the incorrect classification are represented by the specificity and sensitivity. In this project these have the special meaning: the false positive ratio will determine the number of false alarms, which should be suppressed (to avoid tiring the officers); the false negative ratio should be brought down to zero, as missing the suicide attempt or the aggressive behaviour may lead to the death of the inmate.

The special scope of the project is also driven by the strictly limited training material that is available to train the DNN. There are video databases suitable for training models analyzing typical human behaviours, such as practicing sports, work activities, etc. The most popular ones are HMDB51 [24], UCF101 [34], YouTube8M [2], and Kinetics [23]. The number of databases representing violent actions (such as fights) is much smaller. The largest database is RWF-2000 [12], which contains 2,000 video sequences recorded by video surveillance cameras. Other databases, such as MoviesFight and HockeyFight [7], AIRTLab [9] contain fewer recordings. So far, no databases containing recordings regarding specific events related to penitentiary institutions are publicly available.

This problem would have to be solved by employing public databases to train the implemented networks with the additional support from application-specific dataset created in parallel with other activities in the project. This specifically refers to the events that are not present in the public sets.

4. Proposed solution

Approaching all events to be detected required the analysis of the specific scenarios of the sequences captured by the sensors. After the discussion with the experts in the psychology and criminology it became clear that it is not possible to use the single event detector for all of them. They are represented by the large variety of behavioural patterns expressed by the human beings. Therefore the events were divided into groups and for each the separate approach was designed. The latter was also dependent on the type of the detected event (video vs. audio stream).

This section contains the description of the technological solution of the presented problem. Because it refers to the specific implementation of the DNN, details of this unique implementation are given. First, the general architecture of the system is presented. Next, the particular algorithms applied for the event detection are described. Finally, preparation of the data set is outlined. These steps make the data processing framework complete and ready for testing.

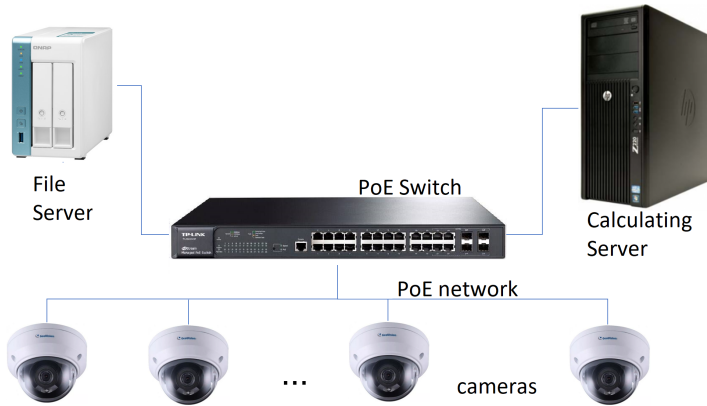


Fig. 4. Diagram of connecting IP cameras to a PoE switch and a calculating graphics server.

4.1. System architecture

The proposed system architecture is presented in Fig. 4. The most important element of the hardware solution is the central processing server. The DNN architectures were implemented there, using NVidia CUDA-compliant accelerators [29]. Its main characteristics covers the number of cores, determining capabilities of processing multiple video streams (as stated in the documentation). The open question was if this parameter holds after adding to the card the DNN functionality. Each camera generates a separate video pipeline, which is then processed by the DNN (one network per pipeline). The computing server operates based on the signal from cameras connected to the system via the POE switches. Cameras that are connected to the system should be properly configured, as each generates two streams (primary and secondary). The former is directed to the recorder so it can archive video materials in full resolution and at full transmission speed (bitrate – Kb/s). The secondary stream is provided to the computing server. Due to the need to save memory, the additional stream should have its bit rate reduced by half, e.g. if the camera transmits the image at a default rate of 2048 Kb/s, then in the additional stream, it should be modified to 1024 Kb/s.

Due to the requirements of the project, the distributed system has the graphical user interface in the form of the web application, written in JavaScript and allowing the for configuration of the cameras and the detection thresholds. Also, this interface is intended to raise alarms in case of event detection. The web server is the separate node, communicating with the processing server through the asynchronous WebSocket interface. The detection module was written in Python and prepare to work on the selected graphic cards.

4.2. Selected DNN architectures

The networks used for the presented tasks have a specific, top-down structure, which makes them useful for event recognition under certain limitations (including, for example, the number of different events and, therefore, sequences that should be remembered). If the network capacity is exceeded, it will not be able to learn subsequent sequences, which may require the use of a larger and more complicated architecture characterized by greater computational requirements. The important assumption was that the functionality of the event detection module relies on the location of cameras delivering the data. For instance, if the device is located outside the building, it does not have to detect suicide or arson, aiming mainly at the escape attempt. Similarly, the most important events detected inside the cell are violent behaviour or suicide attempts. This allows for connecting the particular DNN model for the selected streams, suppressing the amount of computations.

All the network architectures have been implemented in Python language, which is currently the reasonable choice for data science applications. Due to the usage of the DL, the proper library (with the proper network models) had to be selected. In this case, TensorFlow and PyTorch [1, 30], which use hardware acceleration of calculations in a graphics processor with the CUDA architecture [29], were selected to implement the neural networks. The PyTorch and TensorFlow libraries include classes and data structures enabling implementation of all stages of DL, in particular, defining the architecture, reading training data, optimizing the network structure, and evaluating it after the training is complete.

4.2.1. Convolutional Neural Networks - CNN

The 3D CNN are widely used for violence detection in surveillance videos [32]. They process the batch of frames, defined by the height, width (single frame dimension), and depth (color depth). In the project the SELayer-C3D (Squeeze-and-Excitation) [22] mechanism was employed. It is an extension of the C3D model [35], i.e., a deep three-dimensional CNN with a homogeneous architecture based on convolution through a $3 \times 3 \times 3$ kernel function. Here it is used to provide weights to each frame of the video. The SELayer has been used in relation to the attention mechanism. It extracts the importance of different parts of the analyzed image and combines weights with the original image. Thus the attention mechanism grasps the importance of each image frame and weights it. This way, only the most significant frames of the video are processed.

The network consists of 8 convolutional layers (Conv1a-Conv5b), 5 pooling layers (Pool1-Pool5), 2 fully connected layers (fc6 and fc7), and one softmax layer, which transforms outputs into a probability distribution over the input classes (Fig. 5).

The pooling mechanism takes an average of each frame. It is higher for frames considered important by the algorithm and lower for not so useful ones.

The model was trained to detect the following unwanted, anomalous events:

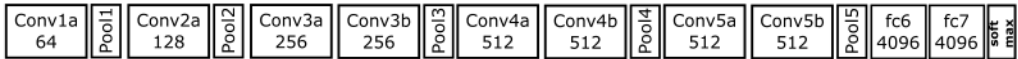


Fig. 5. Architecture of the C3D network.

- events involving violent behaviour of inmates (fight, attack on an officer, rebellion, rape, abuse of power by an officer);
- suicide and suicide attempt;
- death of an inmate;
- unauthorized verbal contact;
- escape attempts (finding an inmate in an area where no one should be);
- attempts to transfer an unauthorized item.

The network is supposed to process the batch of the colour images (RGB scale), extracted from the 5-second sequence, further called a *scene* (every fifth frame is extracted for the single batch). For such a file the softmax layer provides the real number representing the confidence about the detection of the specific event. Now, the crucial element is setting the threshold, above which the processed event is considered anomalous. This is actually the main challenge during application of this network to the described project. The problem is the same event will be seen differently in variety of environmental conditions (day or night, sunny or rainy aura). Also, positions of the cameras vary (the inmates' cell, the corridor, or in the outside perimeter). These conditions enforce the adaptive threshold adjustment, which at this point is a separate task, requiring first finding the optimal values for particular conditions. Due to the potential instability of the system after changing the environment (for example implementing it in another prison), it is recommended to retrain the system on the location-specific data (recorded on-site).

The best predictions were achieved for events involving violent behaviour of inmates and suicide attempts. A sample of detecting violent behaviour is shown in Fig. 6 where a low probability value of detecting abnormal behaviour is shown on the left side of the image (prediction value of 0.39) and a high probability value of detecting abnormal behaviour is shown on the right (prediction value of 0.86).

Experiments performed with the C3D model in laboratory conditions showed that some of the anomalies were not properly detected and algorithms needed to be improved. Unauthorized physical contact, escape attempts, and transfer of an unauthorized item are detected based on C3D-trained network predictions combined with contour detection and its intersection with specific lines and bounding boxes (regions of interest, ROIs) for each group of cameras placed at specific areas in buildings. These shapes are different for the interior of the building and outside the building and are fitted to each camera's video resolution [10]. In Fig. 7, there is an example of ROI for supporting the detection of escape attempts.



Fig. 6. Example of abnormal behaviour's detection and corresponding prediction value. *Nominalnie* – nominal; *ramka* – frame, *Wysoki* – high.



Fig. 7. Example of ROI line for detecting escape attempts outdoors and alarm notification.

4.2.2. YOLO architecture

A separate problem is detection of an inmate's death. In this case, the pose estimation and specific relationship between the head, hips, and feet, which are key points in the human body, were the most important parameters to analyze and support the main

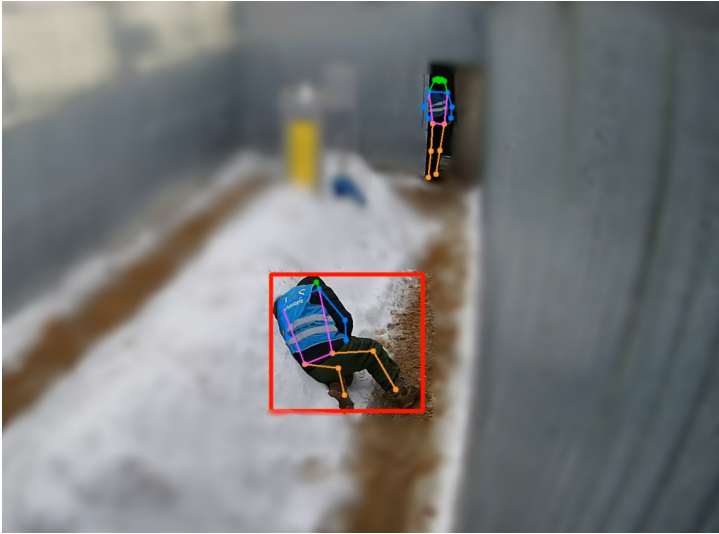


Fig. 8. Example of YOLOv7 pose estimation and prediction of inmate's death.

C3D network predictions. Pose estimation and key point detection were implemented using YOLOv7 (Your Only Look Once, version 7) Fully Connected Neural Network model [36] and in Fig. 8 there is an example of two detected persons, their body key points and alarm notification (red box around a person falling on the ground). The YOLO framework consists of three components:

- Backbone, which extracts essential features (characteristic points) of an image and feeds them to the Head of the network through Neck;
- Neck, which collects feature maps and creates so-called feature pyramids;
- Head, which consists of output layers that predict the locations and classes of objects around which bounding boxes should be drawn.

YOLOv7 is a multi-head framework (Fig. 9). The head responsible for final output is called the Lead Head, while the one used to assist training in the intermediate layers is the Auxiliary Head. The weights of the latter are updated with the help of an assistant loss. These auxiliary classifiers provide direct supervision on the hidden layers in addition to the overall network output. They attach a specific loss function to the intermediate layers, enabling the gradient to be directly propagated back to earlier layers in the network. This helps with gradient flow and facilitates training deep networks.

The final layer aggregation is done by the Extended Efficient Layer Aggregation Network (E-ELAN), which enables the YOLOv7 framework to improve training process. To increase the performance of a model without increasing the training cost, YOLOv7 utilizes so-called Planned Re-parameterized Convolution. Two types of re-parameterization

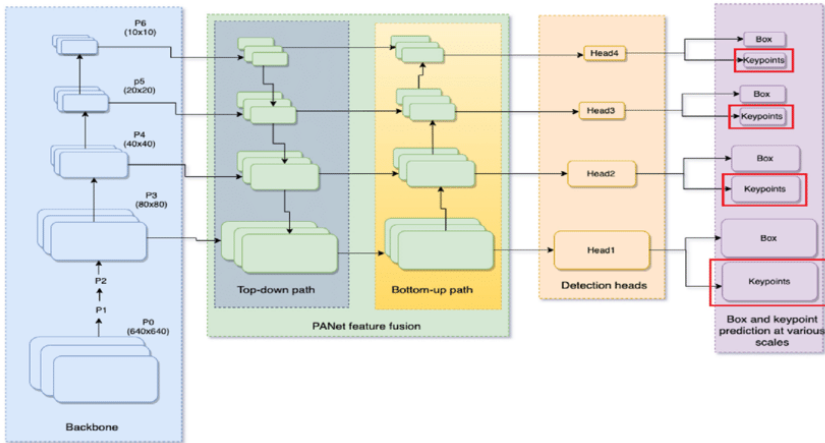


Fig. 9. Architecture of the YOLO network [26] (license: CC BY 4.0).

are used to finalize models: model level and module level ensemble. The first method uses different training data and identical settings to train multiple models. Then averaged weights are used to obtain the final model. The second model training is split into multiple modules. The outputs are combined to obtain the final model.

Joint detection and 2D multi-person detection in an image is conducted by the YOLO-Pose. It is a modified version of YOLOv5 model that learns to jointly detect bounding boxes for multiple persons and their corresponding 2D poses by associating all keypoints of a person (so-called anchors). They are matched with the ground truth box stores forming the 2D pose along with the bounding box location. Keypoints associated with an anchor are already grouped. In case of human pose estimation, each person has 17 associated keypoints, identified by a particular location on a 2D figure with a certain measure of confidence. The keypoint head predicts 51 elements, while the box head predicts six elements. The output of this pose estimation model is a set of points that represent the keypoints on an object in the image, usually along with the confidence scores for each point.

YOLO assigns confidence scores to each predicted bounding box. These scores represent the model's confidence in the accuracy of the prediction. High confidence scores indicate a high probability of the bounding box containing a valid object. If the keypoint is either visible or occluded, the ground truth confidence score is set to 1. Otherwise, if it is outside the field of view, confidence is set to zero. Keypoints outside of the field of view are disregarded.

4.2.3. Line crossing

This method is used to detect events related with objects crossing the line defined inside the field visible by the camera. The user must set the particular coordinates to define where the exactly the crossing should be detected. This approach is applicable to areas with no nominal movement at all, like in the perimeter around the prison block buildings, or on the roof with the view on the windows of cells. The events detected this way include the escape attempts and passing forbidden objects through the outside environment. This process is based on extracting detected moving objects' contours in consecutive video frames and then finding the intersections with ROIs (Region of Interest) defined as lines or specific bounding boxes customized to each camera. All processing is performed with OpenCV library functions [10].

4.2.4. Sound-based events detection

This task was relatively simple, as the number of events related with uttering sound was strictly limited to detecting the unwanted verbal contact between inmates or the verbal contact with the outside world. Also, the number of cameras equipped with the microphones is relatively small (up to 5 units). Because the task was only to identify the contact and not its details (like spoken/shouted words), the simple batch processing of the audio stream was used, consisting in computing energy of the 250 ms time frames. The alarm was risen for the energy above 18 dB.

4.3. Training data preparation and preprocessing

The fundamental importance for the effectiveness of the DL-based system is to provide it with the appropriate amount of data in the form of recorded scenarios of various situations containing undesirable behaviour. The network should obtain a large number of scenes in which anomalous behaviours appear in as many environments as possible (knowing that the network must learn how to ignore the background and focus on the scene). Metadata that may help in describing the particular scenario (the number of participants, recognized behaviour, and place of the event) are also added to each sample. Multiplication of such scenes (by creating different versions of scenarios) helps in reaching the generalization, essential while implementing the system in the Real-World environment.

First, the proper data sets had to be prepared. The process of acquiring information important to train the network was divided into three phases (Fig. 10). The first one used the preliminary data D_1 from the publicly available sets. This did not allow for preparing the system to detect all events, but was enough to test the algorithms and implement them in the first version of the framework. In the second phase (also executed in the laboratory conditions) the original data set D_2 was prepared to supplement and optimize already existing system. It contained scenarios of unique events that could not be extracted from the set D_1 (for instance, suicide attempt). Also, the more general

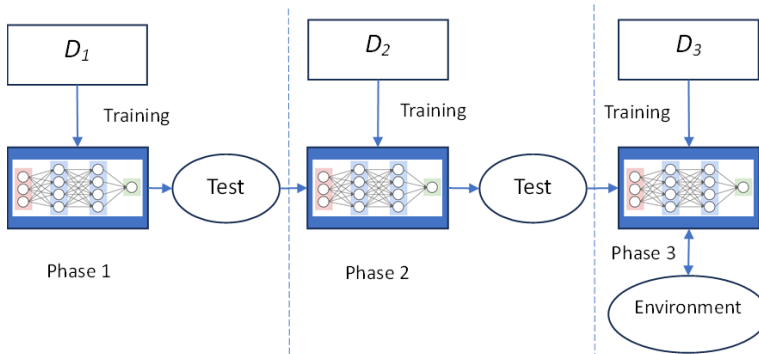


Fig. 10. Training framework for the constructed system.

events (like fights) were added to consider specific locations (corridors, narrow passages, small rooms) in hope of increasing the accuracy of their identification. The final phase was executed in the premises of the facility, where additional scenes have been recorded to form the set D_3 . After each phase the testing was conducted to verify the overall accuracy of the system. From this step all incorrectly detected scenes were also added to the next version of the set. After stabilizing the system on-site, it was run in the on-line mode, with the intention of stabilizing its behaviour in the long term (especially after detecting flaws in the real-world conditions).

The process of preparing the algorithm for identifying individual patterns of undesirable behaviours begins with extracting them from available training scenes. An automatic frame extraction algorithm was developed to create training examples. Each data source is a separate stream of information analyzed at the software level, based on open-source libraries prepared in Python. Each recording was divided into 32 segments, constituting a separate set of patterns during the training process. The C3D features were extracted from each example, forming a training set.

Recordings of anomalous behaviour (intended to trigger an alarm) and neutral situations (not triggering an alarm) must be in MP4 format with $\text{fps} = 25$ frames/second. Each recording of anomalous behaviour should include 1 second before the event and 1 second after the event (approximately 5-6 seconds in total). Recordings of neutral behaviours may be longer (up to 30 seconds). Recordings are recommended to be in a resolution of at least 640×480 pixels (which facilitates scaling down the original image to the input of the network, which is a map of the size of 224×224). The AI models utilized here were trained on the basis of approximately 2500 recordings of anomalous sequences intended to trigger an alarm and 4000 recordings of nominal sequences (not causing an alarm).

5. Results and discussion

This section covers the verification details of the systems efficiency in the binary classification task (though driven by different events). First, efficiency measures are presented. Then, they are used to verify the DNNs performance. Finally, the time efficiency analysis is performed on the system operating in the on-line and off-line mode.

5.1. Accuracy measure

The basic measure of the system's effectiveness is accuracy, which is measured as the percentage of correctly detected events. Since calculating such a coefficient for an online system is difficult to implement (among others because, most of the time, video streams provide data that does not contain any abnormal behaviour), the system's accuracy is determined on the basis of the sampling error e_s . It is calculated for individual events x_i separately (each has different details that constitute individual level of difficulty for the system) as a percentage of correctly detected events (i.e. having the response $h(x_i)$ identical to the actual category c_i) based on a selected set of video sequences T presented to the system.

$$\text{acc} = 1 - e_s = \frac{|x_i \in T : h(x_i) == c_i|}{|T|} \cdot 100\% \quad (1)$$

To check the accuracy in the laboratory conditions (off-line mode), the specific testing sets were created. They contain sequences of the same type as for the training set, but with different actual scenes. Due to the high time consumption of the testing process, each event was represented by 20 scenes of unwanted events and the same number of nominal sequences. This way the accuracy for detecting the particular type of event is between 0 and 100 percent (with the step of 2.5 percent). It is possible to determine accuracy of the detecting individual events and the average accuracy of the system (for all events).

5.2. Off-line accuracy evaluation

In laboratory conditions (locations of the Warsaw University of Technology), the accuracy was on average 90%. Relatively the easiest to detect were violent behaviours such as fights, assaults, rebellion, etc. A very high level (close to 100%) was achieved for events such as fights, escape, and non-verbal contact (passing an object through the windows of the prison pavilion). The transfer of prohibited items is relatively the most difficult to detect because cameras make have problems with capturing the particular behaviour of inmates.

The selected detection threshold values significantly influence the accuracy of the detection system. This is one of the critical parameters requiring individual tuning

Tab. 1. Confusion matrix for the violent behaviour of the inmates.

| | A_{actual} | N_{actual} |
|------------------------|---------------------|---------------------|
| $A_{\text{predicted}}$ | 19 | 0 |
| $N_{\text{predicted}}$ | 1 | 20 |

Tab. 2. Confusion matrix for the suicide attempt.

| | A_{actual} | N_{actual} |
|------------------------|---------------------|---------------------|
| $A_{\text{predicted}}$ | 15 | 0 |
| $N_{\text{predicted}}$ | 5 | 20 |

when installing the system in a new location. The detection threshold (provided by the DNN) is a coefficient with values in the range (0, 1) and should be carefully adjusted to environmental conditions. For values close to 1, false alarms are rarely detected, but the same applies to the detection of undesirable events. For values close to 0, all events will be detected, but with a large number of false alarms. In practice, the threshold should be set in the range of 0.8-0.98, depending on the specific location.

The accuracy described in this way does not include the system's sensitivity to false alarms. These can be deduced from the confusion matrices, such as in Tab. 1. Here A stands for the anomalous event, while N is the nominal event. The violent behaviour of the inmates is relatively easy to detect, as all abrupt changes in the scene were detectable in various locations. On the other hand, suicide attempts were more challenging, and detection is successful only if the actor performs the scenario to the point (Tab. 2). In almost all situations the main problem was missing the event, not the false alarm.

Based on the available results it is assumed the number of false alarms is reciprocal to the amount of training data. Therefore it is expected that the system's robustness will increase with the duration of its operation in the specific facility. To obtain the required immunity to false alarms, additional video sequences must be provided for re-training, especially with scenes that have incorrectly classified as alerts (e.g., meals being delivered or an inmate entering the toilet, as a result of which the light turns on and the camera switches from night to day mode).

On the other hand, the ability to detect all positive cases depends on the nature of such an event. In some cases (for instance, suicide) the event is very difficult to detect, especially if it is not commenced exactly as it was assumed during the scenario creation. Therefore the system is able to detect only the event presented in the specific form. To extend the capabilities of the system, probably additional sensors would be required (such as thermal imaging).

The approach taken to reach the maturity in the presented technology depends on

the constant availability of the new training data (provided that deep networks were selected correctly and are able to extract all required features). Though initially widely available sets are enough to verify the desired detection accuracy, in the repeated cycles the main focus is on the suppression of the false alarms with the zero omissions of the actual events. Unfortunately, only the application-oriented data may help to improve the accuracy. In the presented project the data must come from the penitentiary institutions, as no other source allows for extracting any relevant information regarding the events of interest.

5.3. On-line accuracy evaluation

The on-line accuracy evaluation is significantly different from estimating the sample error in the laboratory conditions. These do not reflect the actual operation of the system in the near Real-Time mode. In the practical application, the system works uninterrupted with multiple data streams incoming all the time. Therefore the evaluation of its efficiency should follow the formula (2), with T_1 being the number of actual events that should be detected (from all streams) and T_0 – the number of normal situations. It reflects the continuous flow of data requiring the assistance from the central system. The processing module makes decision in discrete steps, after collecting the batch of frames from the particular stream. During each time interval the constant number of batches are generated, so based on the duration of the system operation, it is possible to estimate the number of analyzed events (most of them being nominal).

$$\text{acc} = \frac{|\{x \in |T_0| \cup |T_1| : h(x) == c(x)\}|}{|T_0| + |T_1|} \cdot 100\% \quad (2)$$

As a result of the experiments and the validation procedure, a detection accuracy of 84.7% was obtained. The experiments were performed on the premises of the actual penitentiary institution “ZK Chełm”, also with the participation of officers (staff of this facility). The tests consisted of extracting selected scenes in specific locations: a cell, a corridor, a walking area and observing whether the system reacted to the event, and the user interface on the computer in the monitoring center informed appropriately about the event. This accuracy was achieved after the system’s initial configuration, which enabled the determination of optimal thresholds that ensure the detection of typical events without generating false alarms. The “ZK Chełm” employees confirmed the operation of the tool and its effectiveness. Results presented in Tab. 3 were obtained after running the system for 30 minutes, during which 6 cameras remained operational and the processing module was aiming at the violent behaviour detection. This allowed for producing 1872 events evaluated by the system, as presented in the log files. After the event was detected in the particular stream, it was switched off to avoid detecting the same event twice,

Tab. 3. Confusion matrix for the violent behaviour of the inmates in the on-site experiments.

| | A_{actual} | N_{actual} |
|------------------------|---------------------|---------------------|
| $A_{\text{predicted}}$ | 17 | 4 |
| $N_{\text{predicted}}$ | 3 | 1848 |

The main difference between the laboratory and on-site testing is the increasing number of false alarms, generated by the events not considered during the DNN training. This is because in the facility many activities are performed by the inmates the design team was not aware of. Some specific locations caused additional problems. For instance, the cameras installed outside could record the birds, which would cause false alarms of crossing the predefined line. Attempts to suppress these were made during the maintenance stage of the system's life cycle.

5.4. Time efficiency evaluation

This section presents three time aspects of the system's time efficiency. The first one refers to the delays related with the video streams propagation inside the network. The second is the DNN training duration (performed in the offline mode), while the last one is the reasoning duration for the single video stream.

The speed of the system's reaction to the occurrence of an event consists of two components. The first one is the speed of streaming transmission inside the computer network (with the introduced delay is of order of 50-100 ms – in some cases, it may be slower and therefore noticeable for the human operator). the delay associated with calculating the prediction of anomalous situation. In the latter case, the delays are an average of 4 s (calculated based on 180 trials), which is the time needed to acquire the required number of frames, preprocess them, and send them to the detection algorithms. Such a delay may be extended depending on the current load inside the computer network and the number of simultaneously supported camera streams. This is optimized by spreading the prediction callbacks over several cameras at once using proper fork mechanisms for managing multiple processes run in a Linux system.

Duration of the learning process is of secondary importance for the practical use of the system. It determines the operations carried out besides the nominal operation of the system (probably performed on the separate machine). This time depends on the size of the training data set and the speed of available hardware (CPU and graphics cards clocking). Because checking the latter would require using multiple different models, the card configuration was assumed constant (two nVidia Telsa M10 cards with 24MiB of VRAM each) and only the influence of the size of the data set on the training duration was checked. Results of time measurements are in Tab. 4.

Tab. 4. Algorithms learning speed depending on the size of the data set.

| Learning set [GB] | Validating set [GB] | Training time for 50 epochs [h] |
|-------------------|---------------------|---------------------------------|
| 1 | 0.08 | 6 |
| 2 | 0.16 | 15 |

The additional delay is related with acquiring images from IP cameras for the prediction algorithms. Experiments carried out in the laboratory conditions (with small LAN) have shown that the times associated with transmitting signals from sensors is up to 1 second. During the field tests (with much larger network inside the facility) this delay increased and is up to 3 s.

6. Conclusions

The system presented in the paper is capable of detecting the selected set of unwanted events recorded by the IP cameras inside the penitentiary institution. It has the distributed architecture with the separate computing server and the www server presenting the user interface for the human operator. The processing part is based on the nVidia CUDA platforms, where the range of DNN implemented.

Performance of the system verified both in the laboratory and on-site conditions shows its usefulness to the defined task. Accuracy of the system tested in the facility is acceptable from both the research and practical perspectives. The main problem with the implementation of the system in the actual conditions. The significant number of events is classified incorrectly as the false alarms. This is caused by the events present on location, but not considered during the training. They should be suppressed in the future during stabilizing the detection modules.

The evolution of the project is focused on the increase of the accuracy with the suppression of the false alarms, which requires the constant iterative training of the system on the newly delivered data. This, however, can be done only with the video sequences extracted on-site, i.e., from the actual events (which, for instance, in case of the suicide attempts, are rare and difficult to collect). Currently works on the data set extensions in cooperation with penitentiary institutions are carried out.

The possible extensions of the project would include the further development of the classification module. As there are multiple DNNs developed every year, the potential accuracy and sensitivity may be increased by the classifier with better discrimination abilities. As the system has the open architecture, it can be also applied to different locations.

Acknowledgment

This work was supported by the Polish National Centre for Research and Development, grant no. DOB-BIO10/16/02/2019, “Intelligent decision support system based on the algorithmic image analysis in the operations of the justice services”, project value: 6 311 438 PLN.

References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. <https://www.tensorflow.org/>, Software available from tensorflow.org.
- [2] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, et al. YouTube-8M: A large-scale video classification benchmark. *arXiv*, 2016. ArXiv.1609.08675. doi:10.48550/arXiv.1609.08675.
- [3] A. Al Ibrahim, G. Abosamra, and M. Dahab. Deep convolutional framework for abnormal behaviour detection in a smart surveillance system. *Engineering Applications of Artificial Intelligence*, 67:226–234, 2018. doi:10.1016/j.engappai.2017.10.001.
- [4] A. Al Ibrahim, G. Abosamra, and M. Dahab. Real-time anomalous behavior detection of students in examination rooms using neural networks and Gaussian distribution. *International Journal of Scientific and Engineering Research*, 9(10):1716–1724, 2018. doi:10.14299/ijser.2018.10.15.
- [5] A. S. Alturki, A. H. Ibrahim, and F. H. Shaik. Real time action recognition in surveillance video using machine learning. *International Journal of Engineering Research and Technology*, 13(8):1874–1879, 2020. doi:10.37624/IJERT/13.8.2020.1874-1879.
- [6] C. Amrutha, C. Jyotsna, and J. Amudha. Deep learning approach for suspicious activity detection from surveillance video. In: *Proc. 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 335–339, 2020. doi:10.1109/ICIMIA48430.2020.9074920.
- [7] E. Bermejo Nievas, O. Deniz Suarez, G. Bueno García, and R. Sukthankar. Violence detection in video using computer vision techniques. In: P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, and W. Kropatsch, eds., *Proc. Conf. Computer Analysis of Images and Patterns (CAIP)*, vol. 6855 of *Lecture Notes in Computer Science*, pp. 332–339. Springer Berlin Heidelberg, 2011.
- [8] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In: M. Meila and T. Zhang, eds., *Proc. 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, 2021. <https://proceedings.mlr.press/v139/bertasius21a.html>.
- [9] M. Bianculli, N. Falcionelli, P. Sernani, S. Tomassini, P. Contardo, et al. A dataset for automatic violence detection in videos. *Data in Brief*, 33:106587, 2020. doi:10.1016/j.dib.2020.106587.
- [10] G. Bradski. The OpenCV library. *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.
- [11] J. Carreira and A. Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In: *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017. doi:10.1109/CVPR.2017.502.
- [12] M. Cheng, K. Cai, and M. Li. RWF-2000: An open large scale video database for violence detection. In: *Proc. 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 4183–4190, 2021. doi:10.1109/ICPR48806.2021.9412502.

- [13] P. Dasari, L. Zhang, Y. Yu, H. Huang, and R. Gao. Human action recognition using hybrid deep evolving neural networks. In: *Proc. 2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2022. doi:10.1109/IJCNN55064.2022.9892025.
- [14] S. R. Dinesh Jackson, E. Fenil, M. Gunasekaran, G. Vivekananda, T. Thanjaivadivel, et al. Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM. *Computer Networks*, 151:191–200, 2019. doi:10.1016/j.comnet.2019.01.028.
- [15] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, et al. Long-term recurrent convolutional networks for visual recognition and description. In: *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634, 2015. doi:10.1109/CVPR.2015.7298878.
- [16] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, et al. Multiscale vision transformers. In: *Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6804–6815, 2021. doi:10.1109/ICCV48922.2021.00675.
- [17] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In: J. Bigun and T. Gustavsson, eds., *Image Analysis. Proc. 13th Scandinavian Conference (SCIA) 2003*, vol. 2749 of *Lecture Notes in Computer Science*, pp. 363–370. Springer Berlin Heidelberg, 2003. doi:10.1007/3-540-45103-X_50.
- [18] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In: *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941, 2016. doi:10.1109/CVPR.2016.213.
- [19] S. Ganta, D. S. Desu, A. Golla, and M. A. Kumar. Human action recognition using computer vision and deep learning techniques. In: *Proc. 2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, pp. 1–5, 2023. doi:10.1109/ACCTHPA57160.2023.10083351.
- [20] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981. doi:10.1016/0004-3702(81)90024-2.
- [21] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013. doi:10.1109/TPAMI.2012.59.
- [22] B. Jiang, F. Xu, W. Tu, and C. Yang. Channel-wise attention in 3D convolutional networks for violence detection. In: *Proc. 2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, pp. 59–64, 2019. doi:10.1109/ICEA.2019.8858306.
- [23] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, et al. The kinetics human action video dataset. *arXiv*, 2017. ArXiv.1705.06950. doi:10.48550/arXiv.1705.06950.
- [24] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: A large video database for human motion recognition. In: *Proc. 2011 International Conference on Computer Vision (ICCV)*, pp. 2556–2563, 2011. doi:10.1109/ICCV.2011.6126543.
- [25] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In: *Proc. 7th Int. Joint Conf. Artificial Intelligence (IJCAI) 1981*, pp. 674–679, 24–28 Aug 1981. <https://hal.science/hal-03697340>.
- [26] D. Maji, S. Nagori, M. Mathew, and D. Poddar. YOLO-Pose: Enhancing YOLO for multi person pose estimation using object keypoint similarity loss. *arXiv*, 2022. ArXiv.2204.06806. doi:10.48550/arXiv.2204.06806.
- [27] A. Nakajima, Y. Hoshino, K. Motegi, and Y. Shiraishi. Human action recognition based on self-organizing map in surveillance cameras. In: *Proc. 2020 59th Annual Conference*

- of the Society of Instrument and Control Engineers of Japan (SICE), pp. 1610–1615, 2020. doi:10.23919/SICE48898.2020.9240260.
- [28] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, et al. Beyond short snippets: Deep networks for video classification. In: *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702, 2015. doi:10.1109/CVPR.2015.7299101.
- [29] NVIDIA, P. Vingelmann, and F. H. P. Fitzek. CUDA, release: 10.2.89, 2020. <https://developer.nvidia.com/cuda-toolkit>.
- [30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, et al. PyTorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32 – Proc. 33rd Conf. Neural Information Processing Systems (NeurIPS 2019)*, vol. 11, pp. 8024–8035. Vancouver, Canada, 8–14 Dec 2019. Accessible in arXiv. doi:10.48550/arXiv.1912.01703.
- [31] N. S. Rao, G. Shanmugapriya, S. Vinod, R. S, S. P. Mallick, et al. Detecting human behavior from a silhouette using convolutional neural networks. In: *Proc. 2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, pp. 943–948, 2023. doi:10.1109/ICEARS56392.2023.10085686.
- [32] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, and A. F. Dragoni. Deep learning for automatic violence detection: Tests on the AIRTLab dataset. *IEEE Access*, 9:160580–160595, 2021. doi:10.1109/ACCESS.2021.3131315.
- [33] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In: *Proc. 27th International Conference on Neural Information Processing Systems*, vol. 27 of *NIPS Proceedings*, p. 568–576, 2014. <https://ora.ox.ac.uk/objects/uuid:1dd0bcd0-39ca-48a1-9c20-5341d6c49251>.
- [34] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. ArXiv.1212.0402. doi:10.48550/arXiv.1212.0402.
- [35] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3D convolutional networks. In: *Proc. 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497, 2015. doi:10.1109/ICCV.2015.510.
- [36] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: *Proc. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7475, 2023. doi:10.1109/CVPR52729.2023.00721.
- [37] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream ConvNets. *arXiv*, 2015. ArXiv.1507.02159. doi:10.48550/arXiv.1507.02159.
- [38] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, et al. A comprehensive study of deep video action recognition. *arXiv*, 2020. ArXiv.2012.06567. doi:10.48550/arXiv.2012.06567.