Vol. 33, No. 1, 2024

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

An Improved Generative Design Approach based on Graph Grammar for Pattern Drawing

Yufeng Liu*, Yangchen Zhou, Fan Yang, Song Li and Jun Wu
College of Information Engineering, Nanjing University of Finance and Economics, Nanjing, China
**Corresponding author: Yufeng Liu (yfengliu28@126.com)*

Abstract Generative design is used to efficiently generate design solutions with powerful computational methods. Generative design based on shape grammar is currently the most commonly used approach, but it is difficult for shape grammar to formally analyze the generated pattern. Graph grammar derived from one-dimensional character grammar is mainly used for generating and analyzing abstract models of visual languages. However, there is a significant gap between the generated node-edge graphs and the representation of shape appearance. To address these problems, we propose an improved generative design approach based on virtual-node based continuous Coordinate Graph Grammar (vcCGG). This approach defines a new type of grammatical rule named node transformation rules to convert nodes into shapes with node transformation applications. By combining node transformation applications and L-applications in vcCGG, we can generate a node-edge graph as the structure of the pattern through L-applications, and then draw the shape outline, next adjust the positions of these shapes, thus relating abstract structures and the physical layouts of visual languages. At the end of the paper, we provide an example application of this approach: generating an illustration from Emma Talbot using a combination of node transformation applications and L-applications.

Keywords: generative design; graph grammar; shape grammar; node transformation rules; pattern drawing.

1. Introduction

Design is a complex solution process that involves professional knowledge, innovative ability, comprehensive experience, aesthetic literacy, and use of scientific technology. With the rapid development and popularization of new intelligent design automation technologies such as machine learning, additive manufacturing, artificial intelligence, and cloud computing, design approaches are constantly expanding. As a developing design approach, generative design has been extensively studied in academia. Since the introduction of generative design based on shape grammar, as proposed by G. Stiny and J. Gips in 1971 [18], generative design has been introduced into different fields such as architectural design [5], product customization design [9], and visual communication design [14].

Shape grammar is a generation system oriented toward design. It is a design inference approach based on rules, using simple shapes as basic elements to establish the rules for the generation of complex shapes. The foundational rules involve spatial transformations such as translation, scaling, rotation and mirroring, which make one shape part of another shape. With limited predefined rules, there can be an infinite number of

designs generated through shape grammar. Following predefined rules, shape grammar can iteratively replace shapes to generate various patterns. However, shape grammar can generate only the shapes that consist of simple shapes such as lines, points and rectangles. Therefore, it is not yet widely used in computer-aided architectural design (CAAD) applications. Most designers design buildings manually or semi-automatically on CAD platforms, e.g. Revit and AutoCAD.

Shape grammar focuses on generative design, while graph grammar derived from one-dimensional character grammar focuses on modeling and analyzing the syntax and semantics of visual languages. Shape grammar supports only unidirectional workflows. It takes the initial shape and transformation rules as inputs to generate a preliminary design and then adjusts the preliminary design by the rules to generate the final design. In contrast, graph grammars have a bidirectional workflow across derivation and specification. Similarly, the graph grammar derivation process derives graphs by repeatedly applying given productions. The graph grammar reduction process, on the other hand, takes graphs and productions as inputs to parse the graphs by applying productions in a bottom-up fashion. However, there is a significant gap between the generated node-edge graphs and the representation of shape appearance for graph grammar.

In our previous work, we proposed an enhanced grammar system for shape generation [12]. This system defines shape rules to transform edges into shapes by shape applications, which builds an inherent relation between abstract structures and physical layouts of visual languages. The main weakness of this system is the position invariance that reduces the flexibility of design. To address the aforementioned issue, our research focuses on an analysis of semantic relations among shapes that make up a pattern. We propose a generative design approach based on vCGG (virtual-node based Coordinate Graph Grammar) [10]. Our approach defines a new type of grammatical rule named node transformation rules to convert nodes into shapes with node transformation applications. By combining node transformation applications and L-applications in vCGG, we can generate a node-edge graph as the structure of the pattern through L-applications, and then draw the outlines of shapes with node transformation applications, next adjusting the positions of these shapes.

In summary, this paper presents an improved generative design approach that automatically generates or validates patterns conforming to the specified rules. First, the structure of the target pattern is generated through vCGG, and then the nodes are converted into shapes according to the node transformation rules. Finally, the position of the shape is adjusted based on the edge attributes, and the target pattern is generated. This approach can set L-applications and node transformation rules in advance for drawing patterns, and can also formally validate a target pattern to determine whether it belongs to the pattern generated by the specified rules.

This paper addresses the aforementioned problems and makes the following contributions:

- An improved approach for grammar specification, grammar induction, generation and validation of pattern based on the vCGG formalism.
- A complete graph grammar for specifying and analyzing patterns that are composed of multiple geometric shapes.
- According to the concrete requirements, productions and transformation rules are designed to achieve customized designs.

The rest of this paper is organized as follows. Section 2 reviews the related works, including patterns generated by shape grammars, several typical graph grammars and our approach. Section 3 introduces the approach framework, including vCGG and node transformation rules. Next, Section 4 gives an example of the Cloud & Bunny rabbit pattern from Emma Talbot. Section 5 compares our approach and other generative design approaches. Finally, Section 6 concludes the paper and mentions future work.

2. Related works

In 1971, G. Stiny and J. Gips proposed that shape grammar is a generative system oriented toward design. G.Stiny detailed the concept and entire application process of shape grammar in 1980 [17]. Design based on shape grammar was first applied in the field of architectural design. M. Agarwal and J. Cagan [1] proposed the coffee machine shape grammar as the first application of shape grammar in product design, demonstrating its use for generating single products before gradually being applied to product design more broadly. The coffee machine grammar is a parametric grammar consisting of 100 manually created rules and labeled two-dimensional shape grammar implemented through a Java-based application program. Its objective is to provide designers with selectable design inspirations during the conceptual exploration phase. However, this method has limitations because its conceptual nature lacks practical production benefits, resulting in visual operational difficulties due to numerous labels.

H. H. Chau [3] concluded, through analysis of various electronic and fast-moving consumer products, that the appearance of these products is largely determined by straight lines, arcs, and their orthogonal projections. M. Pugliese and J. Cagan [13] summarized previous research methods and found that grammar has become a design tool for creating structures and functional requirements. However, there is no specific method for establishing and maintaining product brand characteristics in the field of product generation design. The field faces two challenges: engineers and designers need tools to help understand, express, and maintain product brands, and engineers, designers, and brand strategists need a common platform to discuss product brands. X. Chen et al. [4] focused on geometric shape in packaging design, proposing an application of shape grammar for packaging design research with personal care bottles as an example in experimentation. S. Wannarumon et al. [20] proposed a method for generating jewelry designs using shape grammar to support designers in exploring shapes as inspiration

sources with ring design as practice examples. S. Garcia and L.Romao [7] coded various types embedded in multifunctional chair classes to develop generative design tools usable during the chair concept design stage. Y. Yu et al. [21] proposed a method of generating origami pattern based on shape grammar recursive applications of shape rewriting rules. In addition, shape grammar provides a perspective and modeling technique for creating origami tessellation patterns.

Compared to shape grammar in the field of design, graph grammar has the characteristics of automated generation and specification. Designers can explore different design options by defining symbols, rules, and parameters, quickly generate a large number of design schemes, and make adjustments and modifications when necessary to improve design efficiency and innovation. H. Bunke [2] proposed attributed programmed graph grammars as a generative tool in image understanding. Based on that, an image understanding system was built to extract descriptions from input images, where a system consists of two major subsystems for preprocessing and segmentation, and understanding, respectively. H. Göttler et al. [8] described the data structures in terms of attributed graphs and their changes in terms of attributed graph productions in an object-oriented manner, applying Graph Grammar to CAD systems.

In the field of architectural design, X. Wang et al. [19] presented a generic approach for grammar specification, grammar induction, validation, and design generation of house floor plans using their path graphs based on the reserved graph grammar formalism (RGG). This approach validates floor plans in different styles with user-specified graph productions and the derivation process is capable of generating floor plan designs. G. Ślusarczyk [23] proposed a framework for supporting the design process by defining design requirements over graph-based representations of designs. First, hierarchical layout graph grammars are used to generate hierarchical layout hypergraphs (HL-graphs) that represent designs; then, local and global graph requirements are defined over HL-graphs, which correspond to design constraints. The proposed ontological interpretations transform first-order and monadic second-order logic formulas expressing design criteria into equivalent local and global graph requirements. The satisfiability of graph requirements by representations of designs allows for checking correctness of design solutions. In subsequent research, G. Ślusarczyk et al. [24] proposed CP-graph grammars to support building layout design, where the grammar rules are combined with semantic-driven embedding transformations and the derivations in this type of grammars are defined. The possibility of relating attributes of right-hand sides to that of the left-hand sides enables the system to capture parametric modelling knowledge. The proposed generative method allows the system to automatically model alternative floor layouts with similar structures but different geometry and parameters, which can be easily adapted to different use case scenarios and environmental conditions.

Apart from the architectural design, graph grammar has been applied to different

fields, including mechanical parts description [6], XML validation [16], cluster analysis [22], entity-relationship (E-R) diagram validation [11], and Web pattern recognition and validation [15]. Overall, graph grammar is a powerful tool for defining and validating graph models, hence the generative design method in this paper is proposed within the framework of graph grammar.

Because patterns are composed of various styles of shapes, there is a positional correlation between each shape. The structure of patterns is generated through graph grammar, which abstracts the positional relationships between various shapes. Then we convert the node-edge graph generated by graph grammar into shapes through node transformation rules, enabling graph grammar to generate shapes and draw patterns. Moreover, graph grammar parsing can check whether a target pattern belongs to the pattern set defined by the rules.

3. Improved generative design approach framework

VCGG is divided into virtual-node based discrete Coordinate Graph Grammar (vdCGG) and virtual-node based continuous Coordinate Graph Grammar (vcCGG) based on different granularity descriptions of spatial semantics. Due to the strict coordinate matching mechanism required in this approach, we choose vcCGG as the basic framework. Below is the theoretical framework of the improved approach.

Definition 3.1. *A **directed graph** G on a given label set L is a 2-tuple (N, E). L consists of a virtual label set L_v and a real label set L_r, where L_r consists of a nonterminal label set L_{NT} and a terminal label set L_T. N is a node set and consists of a virtual node set N_V and a real node set N_r, where N_r consists of a nonterminal node set N_{NT} and a terminal node set N_T. E is a directed edge set.*

Mapping for G includes the following:

- $f_{NL} : N \to L$ is a mapping that assigns a label $l \in L$ to node $n \in N$;
- $f_{NC} : N \to R \times R$ is a mapping that assigns a 2D coordinate $c \in R \times R$ to node $n \in N$;
- $f_{EN_s} : E \to N$ is a mapping that assigns the start node to directed edge $e \in E$;
- $f_{EN_e} : E \to N$ is a mapping that assigns the end node to directed edge $e \in E$.

Definition 3.2. *A **production** p: $G_L := G_R$ is made up of a left-hand-side (or left graph) G_L and a right-hand-side (or right graph) G_R. For a production, there exists a bijection $f_{NN} : G_L.N_v \leftrightarrow G_R.N_v$ between $N_v \in G_L$ and $N_v \in G_R$, where $G_L.N_v$ is a virtual node set N_v of G_L and $G_R.N_v$ is a virtual node set N_v of G_R.*

A production also satisfies the following conditions:

- $\forall n((n \in G_L.N_v) \Rightarrow (f_{NC}(n) = f'_{NC}(f_{NN}(n))))$, where f_{NC} is a mapping that assigns a coordinate to node $n \in G_L$ and f'_{NC} is a mapping that assigns a coordinate to $n \in G_R$;

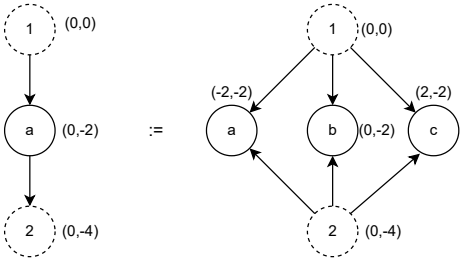

Fig. 1. vcCGG production.

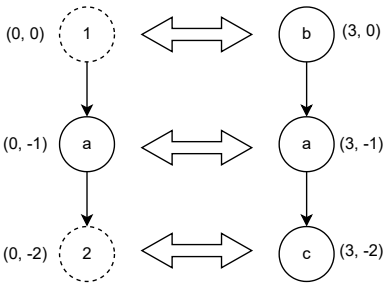

Fig. 2. The isomorphic graphs in vcCGG.

- $\forall n((n \in G_L.N_v) \Rightarrow (f_{NL}(n) = f'_{NL}(f_{NN}(n))))$, where f_{NL} is a mapping that assigns a label to node $n \in G_L$ and f'_{NL} is a mapping that assigns a label to $n \in G_R$;
- $\forall n_1, n_2((n_1, n_2 \in G_L.N_v) \wedge (n_1 \neq n_2) \Rightarrow (f_{NL}(n_1) \neq f_{NL}(n_2)))$;
- $\forall n_1, n_2((n_1, n_2 \in G_R.N_v) \wedge (n_1 \neq n_2) \Rightarrow (f'_{NL}(n_1) \neq f'_{NL}(n_2)))$.

VcCGG stipulates that there is a bijection between the virtual node sets at G_L and G_R, and the corresponding nodes have the same labels and coordinates. In addition, to avoid ambiguity during graph embedding, each virtual node in the same graph must have a unique label, which can be represented by a unique integer.

For example, Fig. 1 is a legal vcCGG production, where the dashed circle represents the virtual nodes and the solid circle represents the real nodes. There is a bijection between the left and right graphs of the production, and the corresponding nodes have the same labels '1', '2' and equal coordinates (0, 0) and (0, 4).

Definition 3.3. *Let G and Q be directed graphs. G and Q are* **isomorphic**, *denoted as $G \approx Q$, if and only if the following conditions hold:*

- *There exists a bijection between the nodes of G and Q, namely, $f_{NN} : G.N \leftrightarrow Q.N$;*

- *There exists a bijection between the edges of G and Q, namely,* $f_{EE} : G.E \leftrightarrow Q.E$*;*
- $\forall n((n \in G.N) \vee (n \in Q.N) \Rightarrow (f_{NL}(n) \in L_v) \vee (f'_{NL}(f_{NN}(n)) \in L_v) \vee (f_{NL}(n) = f'_{NL}(f_{NN}(n))))$*, where* f_{NL} *is a mapping that assigns a label to node* $n \in G$*;* f'_{NL} *is a mapping that assigns a label to* $n \in Q$*;*
- $\forall e((e \in G.E) \vee (e \in Q.E) \Rightarrow (f_{NN}(f_{EN_s}(e)) = f_{EN_s}(f_{EE}(e))))$*;*
- $\forall e((e \in G.E) \vee (e \in Q.E) \Rightarrow (f_{NN}(f_{EN_e}(e)) = f_{EN_e}(f_{EE}(e))))$.

When determining whether a pair of graphs satisfies the isomorphic condition, virtual nodes have a higher abstract degree than real nodes and can match any labeled node. Fig. 2 is an example of graph isomorphism in vcCGG, where all nodes and edges satisfy a bijective relationship. Real node 'a' and the corresponding nodes must have the same label, while virtual nodes '1' and '2' can match any labeled node. In Fig. 2, node '1' matches 'b' and node '2' matches node 'e'.

Definition 3.4. *Let G be a directed graph referred to as the host graph and Q be the subgraph of G. Let* $G_{L|R}$ *be the left or the right hand-side of a production. Q is called a* **redex** *of G with respect to* $G_{L|R}$*, denoted as* $Q \in \text{redex}(G, G_{L|R})$ *if and only if the following conditions hold:*

- $Q \approx G_{L|R}$*;*
- $\forall n((n \in Q.N \wedge ((f'_{NL}(f_{NN}(n)) \in L_r)) \Rightarrow (d_s(n) = d_s(f_{NN}(n))) \wedge (d_e(n) = d_e(f_{NN}(n))))$*;*
- $\forall n_1, n_2((n_1, n_2 \in Q.N) \Rightarrow (f_{NC}(n_1) - f_{NC}(n_2) = f'_{NC}(f_{NN}(n_1)) - f'_{NC}(f_{NN}(n_2))))$.

The nodes of a redex could be divided into two types: the nodes matched by the virtual nodes (context nodes) of the production, and the nodes matched by the non-virtual nodes (inner nodes) of the production. All the edges between the redex and the rest host graph are only allowed to be connected with the former type of nodes.

Definition 3.5. *A* **L/R application** *to graph G is a process that generates graph G′ using production p:* $G_L := G_R$*, denoted as* $G \rightarrow^p G'$*(L-application) or* $G \rightarrow^p G'$*(R-application).*

The L-application in vcCGG is as follows:

1. Generate an instance of the production as a copy of the production.
2. Translate the coordinates of the instance's G_R by the offset between any matched nodes in the redex Q and G_L.
3. Delete edges in the redex Q and nodes that match the real nodes in G_L from the host graph.
4. According to the mapping between the virtual node of G_L and the redex Q, glue the virtual node of G_R to the corresponding node in the redex Q and remove the virtual label from the host graph.

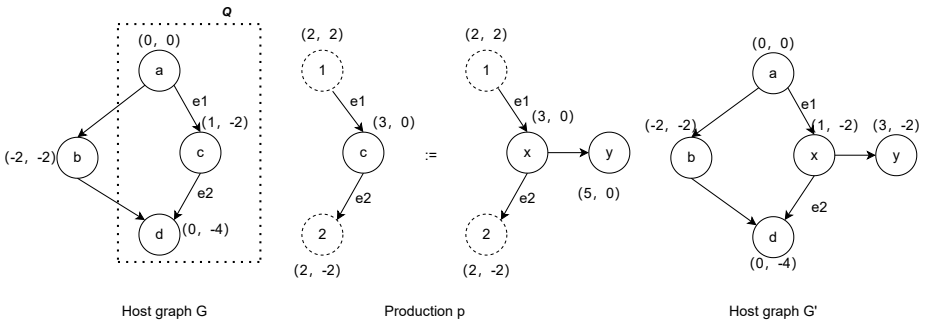

Fig. 3. New host graphs generated by a production.

Fig. 3 depicts an L-application process that generates new host graph G' using production p: $G_L := G_R$.

1. Generate an instance of production p.
2. Find a redex of G with respect to G_L: In the host graph G, we denote a graph in the dashed box as graph Q. $Q \approx G_L$ and the coordinate differences of the corresponding nodes are (2, 2), so $Q \in \text{redex}(G, G_{L|R})$.
3. Subtract all node coordinates of G_R (2, 2).
4. Delete edge 'e1', 'e2' and node 'c' from G.
5. Glue virtual node '1' of G_R to real node 'a' of G and virtual node '2' of G_R to real node 'd' of G; and remove the virtual label from the host graph.

Definition 3.6. *A* **node transformation rule** *is a 4-tuple(cset, cpoint, ops, parm), where*

- *cset is a set of coordinates as the points to represent a shape;*
- *cpoint is the mean point of cset;*
- *ops is the operations performed on the cset, such as translation, rotation, scaling, etc.;*
- *parm is the parameter of the ops, such as the offset of translation or the angle of rotation.*

Given a node transformation rule, the node transformation application is a process that draws the outline of a shape from the perspective of the user using node transformation rules. Below are the steps for a node transformation application:

1. Draw a shape based on the outline described by a node's cset, and make the cpoint coincide with the node. As shown in Fig. 4, a node transformation rule is to transform a node into a rectangle. Use this node transformation rule for node A and B: make the cpoint of this rectangle coincide with node A and B, and transform edge e_1 connecting A and B to line segment l_1;

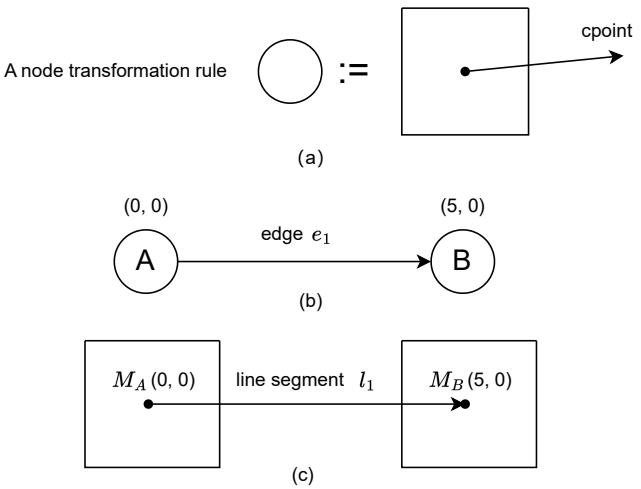

Fig. 4. Demonstration figure of step 1.

2. As shown in Fig. 5, deform the shape by the following methods according to ops and parm:
   (a) Translation: Let A be a shape, and the position of A can change along the X and Y axes, i.e.,
   $\forall(x, y) \in A, (x', y') = (x + a, y + b)$,
   where a is the distance that the position of A changes on the X axes and b is the distance that the position of A changes on the Y axes.
   (b) Scale: Let A be a shape that can expand or shrink in a certain proportion, i.e.,
   $\forall(x, y) \in A, \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$, where S is the factor by which shape A expands or shrinks.
   (c) Stretch: Let A be a shape that can be elongated or shortened along the X and Y axes. Specifically, if the factors of elongation or shortening along the X and Y axes are equal, A can be considered to be scaled, i.e.,
   $\forall (x, y) \in A, \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} Sx & 0 \\ 0 & Sy \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$,
   where Sx is the factor by which A is elongated or shortened along the X axes and Sy is the factor by which A is elongated or shortened along the Y axes.
   (d) Rotate:Let A be a shape that can rotate $\theta\,(0 < \theta < 2\pi)$ counterclockwise around the cpoint $M_A(X_A, Y_A)$, i.e.,

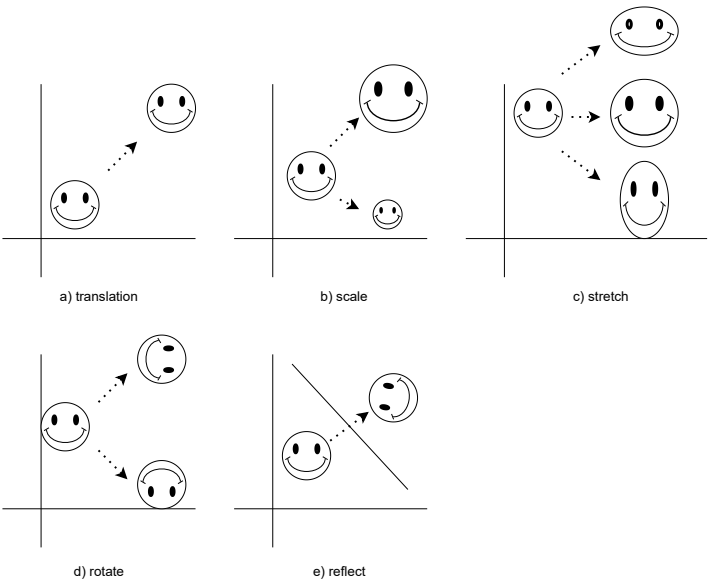

Fig. 5. A new shape formed by 5 operations.

$\forall\,(x,y) \in A,\ (x',y') = ((X_A - x)\,cos\theta - (Y_A - y)\sin\theta + X_A,\ (X_A - x)\sin\theta + (Y_A - y)\cos\theta + Y_A)$.

(e) Reflect: Let A be a shape. $\forall l : PX + QY + M = 0(P^2 + Q^2 > 0)$, new shape A' is a mirror image of A across line l, i.e.,
$\forall\,(x,y) \in A, (x',y') = \left(x - \frac{2P(Px+Qy+M)}{P^2+Q^2}, y - \frac{2Q(Px+Qy+M)}{P^2+Q^2}\right)$.

3. Render the shape from the user's perspective based on the outline described by the cset through its own operations.
4. Adjust the position of the shape based on the attributes of the line segment l_1.

Definition 3.7. *For shape A and shape B, A and B are* **separated** *if and only if* $\exists l : Px + Qy + M = 0(P^2 + Q^2 > 0)$*, A and B are on both sides of line l, as shown in Fig. 6.*

As shown in Fig. 7, for shape A and B, M_A is the cpoint of A and M_B is the cpoint of B. M_A and M_B are connected through a directed line segment l_{AB}, where M_A is the start point of l_{AB} and M_B is the end point of l_{AB}. The position of M_A will change according to the attribute of l_{AB}, and the position of A will be changed following the changes in M_A position. The attribute of l_{AB} is 'far from d' or 'near d', where d is the distance at which the M_A position changes. When using node transformation rules to transform

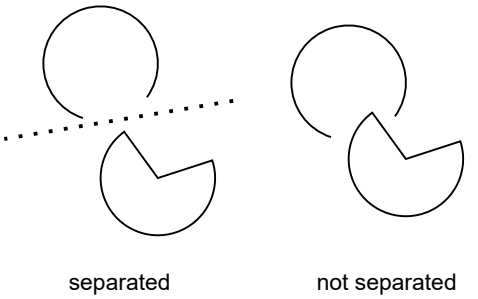

Fig. 6. The two shapes are separated or not.

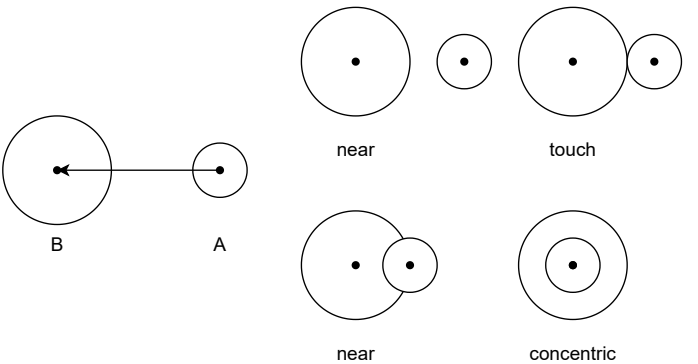

Fig. 7. A is near B; A touches B; A is concentric to B.

node A and B into shape A and B, it is necessary to ensure that they are separated. Therefore, if the attribute of l_{AB} is 'far from d', regardless of the value of d, A and B are still separated. So, we won't limit the value of d when the attribute of l_{AB} is 'far from d'.

Definition 3.8. *If the attribute of l_{AB} is 'near d', A may* **touch** *B or be* **concentric** *with B during the process of changing the position of A.*

- *Touch: $A.cset \wedge B.cset \neq \emptyset$ for the first time;*
- *Concentric: M_A coincides with M_B.*

For convenience, when users want A to touch B or be concentric with B, they can

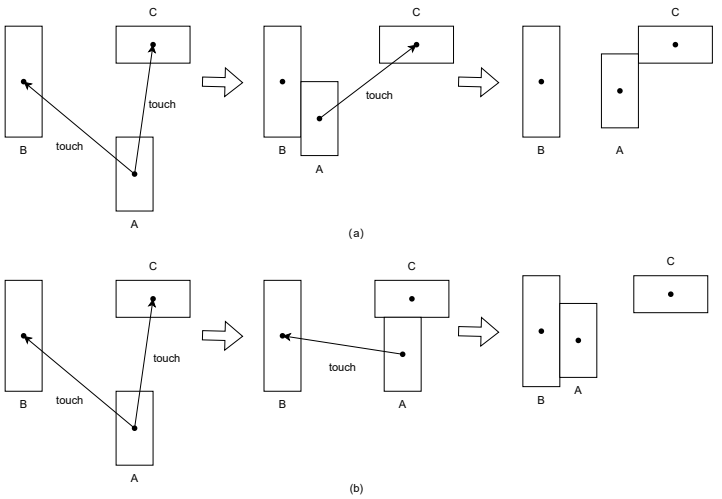

Fig. 8. The final position of A will change due to the order of touch B or C.

set the l_{AB} attribute to 'touch' or 'concentric'. Before the position of M_A changes, make $D_{\max} = |M_A - M_B|$. So, $0 < d \leq D_{\max}$ when the attribute of l_{AB} is 'near d'.

As shown in Fig. 8, for shape A, when M_A is the starting point of two or more directed line segments, the position of A must to be changed at least twice, and different changing sequences can lead to different positions. As shown in the Fig. 8, A needs to touch both B and C, and the final position of A will change based on the order of it touches B or C. Therefore:

- When the X coordinate of the end nodes is different, the position of start node first changes toward the end node with a smaller X coordinate;
- When the X coordinate of the end nodes is the same, the position of start node first changes toward the end node with a smaller Y coordinate.

4. An example on rabbit pattern

This section gives an example to illustrating an application of the improved approach, where a set of designed productions and node transformation rules are used to generate a section of the Cloud & Bunny rabbit pattern from Emma Talbot. Emma is passionate about mixed media research and enjoys using various media to create textures, patterns, and collages to integrate into her artistic creations. The Cloud & Bunny rabbit pattern is composed of simple geometric shapes such as arcs, rectangles, triangles, etc., forming

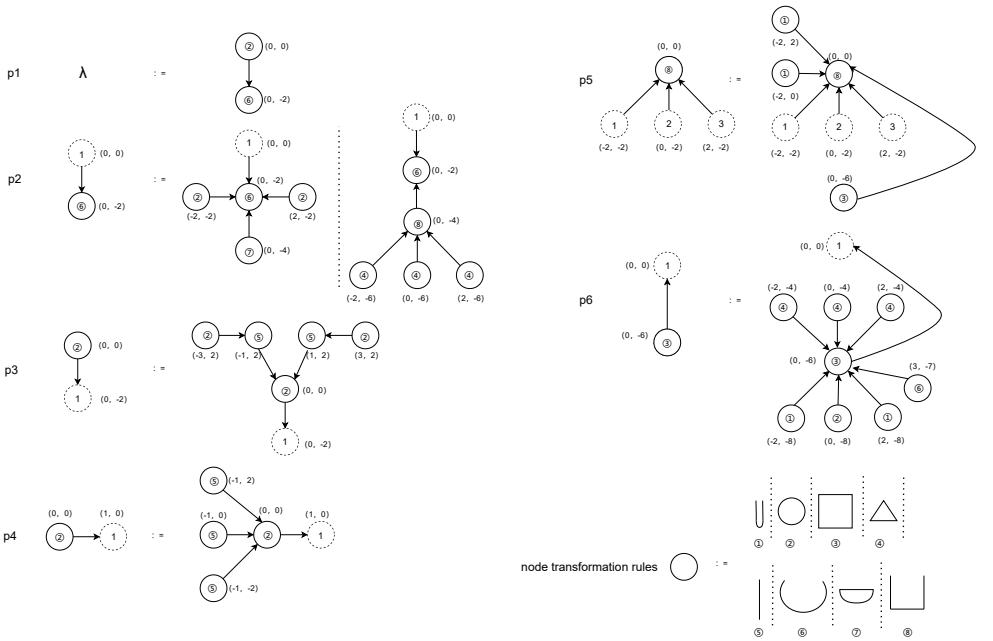

Fig. 9. Productions for a bunny rabbit.

patterns of rabbits, flowers, and clouds. In this paper, a rabbit pattern is selected as the generated pattern. Fig. 9 shows a set of vcCGG productions and eight node transformation rules as a grammar set for the rabbit pattern, where the vcCGG productions are used for the abstract models of pattern and node transformation rules describe physical layouts. For the vcCGG productions, the initial symbol 'λ' denotes the beginning of graph grammar. 'λ' is used to generate the right graph of p1 through production p1 and then generate the target structure of the pattern based on the remaining productions p2-p6. For the productions in Fig. 9, virtual nodes, which are represented by a dashed circle and labeled '1', '2', and '3', are used to match coordinates; real nodes, which are represented by a solid circle and labelled '①', '②', and '③', are converted into shapes. For the node transformation rules in Fig. 9, we set eight shapes to generate the final pattern, including circle, rectangle, triangle, etc.

Fig. 10 shows a process of generating a rabbit pattern using the productions and node transformation rules above. When using an L-application to generate the structure of the target pattern, an attribute is assigned to each generated edge. The attribute can be 'near', 'touch' or 'concentric'. If the attribute is 'near', the distance needs to be

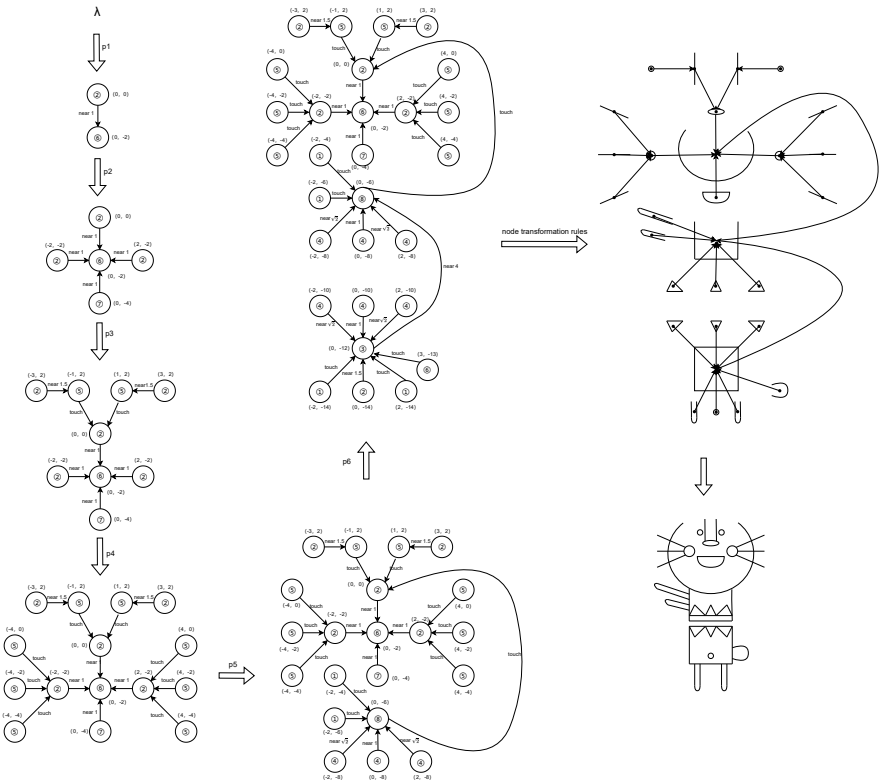

Fig. 10. Generation of a bunny rabbit.

given as parameter. When using node transformation rules for the final generated node-edge graph, each node is traversed and converted into a shape based on the associated label. Then, each edge is traversed, the position of each shape is adjusted based on the attribute of each edge, and the target pattern is ultimately obtained.

5. Comparisons with other generative design approaches

In this section, we compare our approach proposed in this paper with shape grammar, edge transformation grammar [12] and CP-graph grammar [24]. Shape grammar is a design inference approach based on rules, using simple shapes as basic elements to establish the rules for the generation of complex shapes. Edge transformation grammar

defines shape rules to transform edges into shapes by shape applications. The CP-graph grammar is used to automatically generate CP-graphs corresponding to new layout designs with non-geometrical properties (like sizes, areas) specified by graph attributes.

As Table 1 shows, these approaches can all design shapes through derivation. However, when drawing patterns using shape grammar, different shapes of a pattern are related only in terms of position and have no semantic relations. Therefore, it is difficult to formally analyze the generated pattern. Our approach based on vcCGG can formally validate a target pattern to determine whether it belongs to the pattern generated by the specified rules by combining node transformation applications and L-applications. Moreover, after designing the transformation rules for shape grammar, edge transformation grammar and CP-graph grammar, they are unable to adjust the size and position of the shape, resulting in a lack of position and size variability. However, for our approach, after generating the structure of the target pattern through vCGG, the nodes which are converted into shapes according to the node transformation rules can adjust the size and position of themselves. Therefore, in terms of position and size variability, our approach is superior to shape grammar and edge transformation grammar.

Tab. 1. Comparison between approach in this paper, shape grammar, edge transformation grammar and CP-graph grammar.

Approach	Derivation	Parsing	Positional and size variability
Our approach	✓	✓	✓
Shape grammar	✓	×	×
Edge transformation system	✓	✓	×
CP-graph grammar	✓	✓	×

6. Conclusions

When designers use shape grammar to generate patterns, there are no semantic relations among the various shapes that make up the pattern or the small patterns that make up the large patterns. Therefore, it is difficult to formally analyze the generated patterns. In addition, graph grammar is primarily used for generating and analyzing abstract models of visual languages. There is a significant gap between the generated node-edge graphs and the visual representation of shapes, so few researchers have applied these concepts in the design field.

This paper proposes an improved generative design approach for pattern drawing, which introduces node transformation rules in the framework of vcCGG. First, the structure of the target pattern is generated through vcCGG, and then the nodes are converted into shapes according to the node transformation rules. Finally, the position of each

shape is adjusted based on the edge attributes, and the target pattern is generated. In this approach, L-applications and node transformation rules are set in advance for drawing patterns, and a target pattern can be formally analyzed to determine whether it is a pattern generated based on the specified rules.

In the future, we plan to improve the theoretical framework of the improved approach and consider adding gray values to the node transformation rules. If it goes well, we plan to add RGB to it so that the improved approach can be used to design the colored patterns. Moreover, we plan to develop a support system for this approach with a friendly GUI for end users to design graph productions and node transformation rules. The system platform will provide support for grammatical operations and the implementation of related applications.

Acknowledgement

This work was supported by the National Natural Science Foundation of China within the grant No. 62002155 and the National Key Research and Development Program of China within the grant No. 2022YFB3305504.

References

[1] M. Agarwal and J. Cagan. A blend of different tastes: The language of coffeemakers. *Environment and Planning B: Planning and Design*, 25:205–226, 1998. doi:10.1068/b250205.

[2] H. Bunke. Graph grammars as a generative tool in image understanding. In: *Graph-Grammars and Their Application to Computer Science*, pp. 8–19, 1983. doi:10.1007/BFb0000096.

[3] H. H. Chau. *Preserving Brand Identity in Engineering Design Using a Grammatical Approach*. Ph.D. thesis, The University of Leeds, School of Mechanical Engineering, and Keyworth Institute of Manufacturing and Information Systems, 2002. https://www.researchgate.net/publication/286452884_Preserving_brand_identity_in_engineering_design_using_a_grammatical_approach.

[4] X. Chen, A. McKay, A. de Pennington, and H. H. Chau. Package shape design principles to support brand identity. *Proc. 14th IAPRI World Conference on Packaging*, pp. 1–14, 2004. https://www.researchgate.net/publication/267797410_PACKAGE_SHAPE_DESIGN_PRINCIPLES_TO_SUPPORT_BRAND_IDENTITY.

[5] G. Díaz, R. F. Herrera, F. Muñoz-La Rivera, and E. Atencio. Generative design for dimensioning of retaining walls. *Mathematics*, 9(16):1918, 2021. doi:10.3390/math9161918.

[6] M. Flasiński. Use of graph grammars for the description of mechanical parts. *Computer-Aided Design*, 27:403–433, 1995. doi:10.1016/0010-4485(94)00015-6.

[7] S. Garcia and L. Romao. A design tool for generic multipurpose chair design. In: *Proc. Computer-Aided Architectural Design Futures*, pp. 600–619, 2015. doi:10.1007/978-3-662-47386-3_33.

[8] H. Göttler, J. Günther, and G. Nieskens. Use graph grammars to design CAD-systems! In: *Graph-Grammars and Their Application to Computer Science*, pp. 396–410, 1990. doi:10.1007/BFb0017402.

[9] M. Lee, Y. Park, H. Jo, K. Kim, S. Lee, et al. Deep generative tread pattern design framework for efficient conceptual design. *Journal of Mechanical Design*, 144(7):011703, 2022. doi:10.1115/1.4053469.

[10] Y. Liu and Y. Fan. VCGG: Virtual-node based spatial graph grammar formalism. *Journal of Software*, 32:3669–3683, 2021. doi:10.13328/j.cnki.jos.006164.

[11] Y. Liu, X.-Q. Zeng, and Y. Zhu. Application of graph grammar EGG to design of ER diagrams. *Computer Engineering and Design*, 2014(3):1071–1075, 2014. https://api.semanticscholar.org/CorpusID:63040768.

[12] Y. Liu, Y. Zhou, F. Yang, and H. Sun. An enhanced grammatical approach for graph drawing. In: *Conf. International Conference on Artificial Intelligence, Virtual Reality, and Visualization AIVRV 2022*, p. 1258803, 2023. doi:10.1117/12.2667201.

[13] M. Pugliese and J. Cagan. Capturing a rebel: modeling the Harley-Davidson brand through a motorcycle shape grammar. *Research in Engineering Design*, 13:139–156, 2002. doi:10.1007/s00163-002-0013-1.

[14] C. Qian, R. Tan, and W. Ye. An adaptive artificial neural network-based generative design method for layout designs. *International Journal of Heat and Mass Transfer*, 184:122313, 2022. doi:10.1016/j.ijheatmasstransfer.2021.122313.

[15] A. Roudaki, J. Kong, and K. Zhang. Specification and discovery of web patterns: a graph grammar approach. *Information Sciences*, 328:528–545, 2016. doi:10.1016/j.ins.2015.08.052.

[16] G. Song and K. Zhang. Visual xml schemas based on reserved graph grammars. In: *Conf. International Conference on Information Technology: Coding and Computing. ITCC 2004*, pp. 687–691, 2004. doi:10.1109/ITCC.2004.1286546.

[17] G. Stiny. Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design*, 7(3):343–351, 1980. doi:10.1068/b070343.

[18] G. Stiny and J. Gips. Shape grammars and the generative specification of painting and sculpture. In: *Proc. Conf. International Federation for Information Processing IFIP 1971*, pp. 125–135, 1971. https://api.semanticscholar.org/CorpusID:36431081.

[19] X. Wang, Y. Liu, and K. Zhang. A graph grammar approach to the design and validation of floor plans. *The Computer Journal*, 63:137–150, 2019. doi:10.1093/comjnl/bxz002.

[20] S. Wannarumon, P. Pradujphongphet, and I. Bohez. An approach of generative design system: Jewelry design application. *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1329–1333, 2014. doi:10.1109/IEEM.2013.6962626.

[21] Y. Yu, T.-C. Hong, A. Economou, and G. Paulino. Rethinking origami: A generative specification of origami patterns with shape grammars. *Computer-Aided Design*, 137:103029, 2021. doi:10.1016/j.cad.2021.103029.

[22] K.-B. Zhang, M. A. Orgun, and K. Zhang. A prediction-based visual approach for cluster exploration and cluster validation by HOV3. In: *Proc. Knowledge Discovery in Databases. PKDD 2007*, pp. 336–349, 2007. doi:10.1007/978-3-540-74976-9_32.

[23] G. Ślusarczyk. A graph grammar approach to the design and validation of floor plans. *Computer-Aided Design*, 95:24–39, 2017. doi:10.1016/j.cad.2017.09.004.

[24] G. Ślusarczyk, B. Strug, A. Paszyńska, E. Grabska, and W. Palacz. Semantic-driven graph transformations in floor plan design. *Computer-Aided Design*, 158:103480, 2023. doi:10.1016/j.cad.2023.103480.

Yufeng Liu received the Ph.D. degree from Hehai University, Nanjing, China. He is now an associate professor in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include software engineering, machine learning and visual language.

Yangchen Zhou received the bachelor degree from Nanjing University of Finance and Economics, China. He is now studying for a master's degree at College of Information Engineering, Nanjing University of Finance and Economics. His main research interests include graph grammar and generative design.

Fan Yang received the Ph.D. degree from Changchun University of Science and Technology, Changchun, China. He is now an associate professor in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include multimodal data fusion, machine learning, and deep learning.

Song Li received the Ph.D. degree from Anhui University, Hefei, China. He is now a lecturer in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include cloud security and applied cryptography.

Jun Wu received the Ph.D. degree from Nanjing University, Nanjing, China. He is now a lecturer in College of Information Engineering, Nanjing University of Finance and Economics, China. His main research interests include computational economics, algorithmic game theory, and mechanism design.

An Age-Group Ranking Model for Facial Age Estimation

Joseph D. Akinyemi[1,*] and Olufade F. W. Onifade[2]
[1] *Department of Computer Science, University of York, York, United Kingdom*
[2] *Department of Computer Science, University of Ibadan, Ibadan, Nigeria*
[*] *Corresponding author: Joseph D. Akinyemi (akinyemijd@gmail.com)*

Abstract Age prediction has become an important Computer Vision task. Although this task requires the age of an individual to be predicted from a given face, research has shown that it is more intuitive and easier for humans to decide which of two individuals is older than to decide how old an individual is. This work follows this intuition to aid the age prediction of a face by exploiting the age information available from other faces. It goes further to explore the statistical relationships between facial features within age groups to compute age-group ranks for a given face. The resulting age-group rank is low-dimensional and age-discriminatory, thus improving age prediction accuracy when fed into an age predictor. Experiments on publicly available facial ageing datasets (FGnet, PAL, and Adience) reveal the effectiveness of the proposed age-group ranking model when used with traditional Machine learning algorithms as well as Deep Learning algorithms. Cross-dataset validation, a method of training and testing on entirely different datasets, was also employed to further investigate the effectiveness of this method.

Keywords: age estimation, age-group ranking, cross-dataset validation, dimensionality reduction, face processing, facial features.

1. Introduction

Ageing is a spontaneous and irreversible process of human life. This spontaneous and irreversible nature makes the ageing process non-linear and therefore difficult to predict. Thus, judging human age via facial appearance or other physical evaluations is difficult. Humans develop an innate ability, early in life to predict age to a reasonable degree of accuracy [18, 20], but this task still seems difficult for computers. The task of predicting or determining the age of an individual, given his/her facial image, is referred to in the Computer Vision and Image Processing research community as age estimation or age prediction. Automated age estimation has proven to have many interesting applications in security and surveillance, age-specific human-computer interaction, preventing age falsification, age-specific advertising etc. [2, 18].

Despite the success of deep learning for facial age estimation, the bulk of features are mostly learned directly from individual images without considering feature correlations across other images, especially with respect to the ages of those other images. This limits the relevance of learned features to the required discriminatory factor of ageing.

In this work, an age-group ranking approach is proposed, which exploits the relationships between faces across several age groups to enrich the extracted facial features for age estimation. The intuition behind this method is the observation that humans estimate ages by instinctively making comparisons between a given face with an unknown

age and other faces whose ages are known. This process is usually implicit and very fast with humans and it happens almost unconsciously. However, this process is influenced by the amount of exposure or experience of the person trying to estimate the age of another person. It could also involve scanning through faces in certain known age groups and trying to fix the questioned face in one of those age groups. Although it is difficult to completely model this process in a machine, we take intuition from this to develop an age group ranking model through which a questioned face is passed, compared with several age groups, and ranked accordingly. The resulting age-group rank is then used to embellish facial features to enhance the age-learning and prediction processes. The idea is to develop a model for extracting facial features that are age-discriminatory yet low-dimensional such that they can be used to predict ages from input face images. Experiments were performed on three publicly available facial ageing datasets FGnet [12], PAL [32] and Adience [17, 22] and a new dataset, FAGE, and the results obtained compete significantly with the state-of-the-art facial age estimation methods.

The specific contributions of this work include:

1. An age-group ranking model that produces age-discriminatory yet low-dimensional facial features from learned correlations between faces and age groups.
2. Deviation of Feature Values (DoFV) which allows age group ranks to be computed without requiring training or prior knowledge of the age of an input image.
3. An indigenous dataset (FAGE) of age-labelled facial images.
4. Cross-dataset validation to demonstrate the generalisation of the age-group ranking model.

The rest of the paper is organized as follows: Section 2 discusses related previous works in the field of facial age estimation, Section 4 discusses the methodology, Section 5 presents the experiments, results and discussion and Section 6 concludes the paper.

2. Related previous works

2.1. Using direct facial features for age estimation

One of the earliest works on facial age estimation was the work of Kwon and Lobo [24] which used face anthropometry and face wrinkles to describe the face and reported 100% accuracy on a set of 47 high-resolution face images classified as 'Babies', 'Young Adults' or 'Seniors'. Research has since continued to produce several methods for improving facial age estimation using different face descriptors, different age representation methods, and various machine learning algorithms.

In [25], the Active Appearance Model (AAM) was used to represent the face and Principal Component Analysis (PCA) was used to obtain the deviation of each face from the mean AAM face model. In [19], an ageing pattern subspace learning model was proposed for facial age estimation. The authors defined an ageing pattern as a sequence

of personal face images sorted by time. Guo et al. [21] used Biologically Inspired Features (BIF) together with manifold learning techniques to estimate ages using Support Vector Machine (SVM) for age classification and Support Vector Regression (SVR) for age regression. Most of these methods, except for [19], directly used facial features of individuals for age classification or regression without considering possible relationships between faces with respect to age.

2.2. Using age ranking for age estimation

Some works have employed age ranking in various ways. In [8], the authors proposed a ranking approach to age estimation based on the intuition that humans estimate the age of an unknown individual by comparing his/her face to the faces of other individuals whose ages are known, thus resulting in a series of pairwise comparisons across a set of individuals with known ages. Based on this intuition, they proposed an age ranking model which results in binary classification-based comparisons. They used an ordinal ranking algorithm to reduce the ordinal ranking problem to a binary classification problem. [9] also proposed an age estimation algorithm that employed the relative order of ages as well as the classification costs. They maintained ordinal hyperplanes which separated all images into two groups based on the relative order of their age labels and used the cost of classification to find the best-separating hyperplane. In [3], an ethnic-specific age group ranking method was proposed for age estimation. In [7], age ranks were predicted based on a cost-sensitive hyperplane ranking algorithm, facial features were represented in low-dimensional space by a scattering transform so that exact ages are then predicted via category-wise age ranks. In [49], a deep learning model was used to rank faces and to estimate ages from faces. Ranking-CNN was proposed in [10] as a series of basic CNNs with binary outputs which were aggregated to obtain a final age label. Their experiments were conducted by pretraining their basic CNNs on Adience dataset [17] and then fine-tuning and validating it on the MORPH dataset with the best MAE of 2.96 years. While that work employed the ordinal age ranking between face pairs, ours employs ordinal relationships between each face and groups of faces in each age group.

2.3. Using deep learning for age estimation

More recently, deep learning models such as Convolutional Neural Networks (CNN) have been used to determine age from faces. [49] used a Scattering Network (a CNN variant) to develop a deep ranking model from age estimation. [35] used CNN with mean-variance and softmax losses to estimate ages from faces. [15] used CNN in a transfer learning setting to predict apparent as well as biological ages. [48] used CNN to learn the ordinal nature of ages for age estimation. In [47], a group-n age encoding was proposed, a CNN with multiple classifiers was used to learn the several age groups and a Local Age

Decoder was used to predict the exact ages. As accurate as deep learning models can be, they are computationally demanding and often require large amounts of training data.

3. Problem and motivation

Despite the impressive performance of many of these deep learning models, we observed that most of them failed to model the correlation of facial features with age groups as well as the inter-age groups' relationships. This is difficult for many of these models because deep learning architectures learn their features directly from inputs. Those which attempted to capture this relationship to an extent (e.g. [10, 47, 48]) still failed to capture the inter-age group relationships as it concerns facial features.

Also, most age ranking works conducted pairwise comparisons between faces leading to a large set of pairwise comparisons. Although DeepRank [49] does not rely on pairwise ranks, it infers its ranks from single images which still limits the possibility of capturing the correlation of faces within a larger set such as an age group. Secondly, most age-ranking works employed some form of learning to perform the age-ranking on faces. We also observed that in many cases, a reference image set was maintained for age ranking which is a subset of the training set and thus limits the amount of information available for age ranking. In [10], the age ranks were learned by several basic deep-learning networks, the results of which were aggregated to obtain a final age estimate. Considering the computational demand of deep networks, this could even be very expensive.

In this work, we propose an age-group ranking model which ranks face images by comparing an input image with every image in an entire training set and, in an attempt to represent age-group-specific features, derives an age group rank that is representative of each age group. Thus, each input image is ranked with respect to every image in a training set as well as with every age group in the training set. This provides a representation of the correlation of input images with every image in the training set as well as with every age group represented in the training set. Also, instead of learning and predicting age group ranks, we derived the deviation of feature values (DoFV) between compared faces and performed basic statistical computations on these values with respect to age groups, thus reducing the computational overhead that could have been incurred due to learning age ranks prior to learning exact ages.

4. Methodology

When a human is asked to estimate the age of a given facial image, several operations come into play in the mind. Apart from the fact that humans possess an innate ability to recognize age from the face, people generally tend to estimate age by comparing the given face to some other faces whose ages are known. This comparison is part of the innate ability and it is usually very fast and without prior thought or preparation. Thus,

a person's ability to correctly estimate age can be considerably impacted by his/her own age vis-a-vis his/her life experience [20,38]. The more exposed and experienced a person is, the better is his/her age prediction ability. Thus, the age prediction ability of an adult is expected to be better than that of a child because of experience and the extent of development. In developing the proposed age-group ranking model, we leveraged this intuition.

Since a person's age estimation ability is impacted by his/her age and life experience, then the age ranking model can be enriched with more experience by providing more reference images for age ranking. Thus, our proposed age group ranking model employs its entire training image set in a leave-one-out fashion to rank images by their age groups. By using the leave-one-out method it is assured that no face image is ranked by comparison with itself. This is justifiable by the fact that the face whose age is in question should be compared with faces whose ages are known and not with itself, since its age is still unknown. Also, people within an age group tend to exhibit similar ageing features, thus making it easier to rank images by age groups than by exact ages. In fact, the sparse nature of ages in most facial ageing datasets makes it almost impossible to obtain enough images for each exact age rank. Also, unlike most other works, our age group ranking model does not learn age group ranks; rather, it obtains the deviation of feature values (DoFV) from compared faces and obtains the means and standard deviations of these deviation values within age groups which are then used to compute age group ranks for an input image.

However, there is still the challenge that, since the age of the face image in question is not known, it is difficult to decide which age group the image should be compared with in order to obtain an age group rank. To overcome this, the age group ranking model performs an exhaustive comparison of the questioned face with every face in every age group (in a dataset) so that the face is enriched with a representation of its correlation across various age groups. Consequently, the correlation of an input face with its actual age group is also learned from its comparison with several face images in that age group.

4.1. The age learning problem formulation

In this work, age estimation is modelled primarily as a regression problem. Thus, suppose we have a set A of face images and a set B of age labels ordered by the magnitude of the age values, the sets A and B can be represented as follows:

$$A = \{a_i | i = 1, \ldots, p\} \,, \tag{1}$$

$$B = \{b_j | j = 0, \cdots, q \wedge \forall j, b_{j+1} > b_j\} \,, \tag{2}$$

where a_i is face image, b_j is an age value, p is the number of face images and q is the highest age value. The expression $\forall j, b_{j+1} > b_j$ indicates that B is an ordered set, i.e., every age value is greater than the previous age value in the set, since age values are

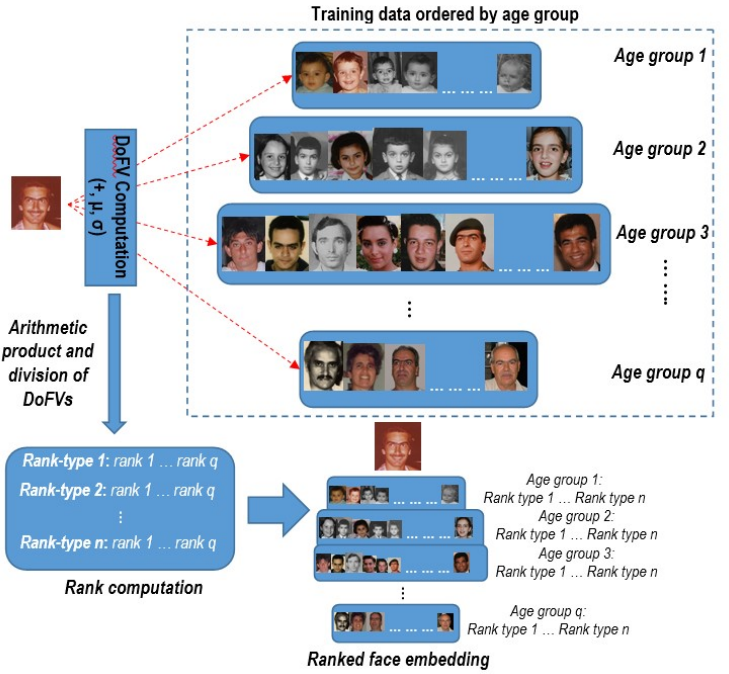

Fig. 1. The age-group ranking model.

ordered in time sequence. This ordering is necessary for age group ranking as we will see in subsection 4.2. Thus, the task of age estimation involves approximating an age learning function, say f_1, which appropriately maps each facial image in A to its age value in B, according to

$$f_1(a_i) = b_j \ , \tag{3}$$

where $a_i \in A$ and $b_j \in B$.

4.2. The age-group ranking model

While age learning explores the relationship between face images and ages, age group ranking explores the relationships between each face image and other images in various age groups. Fig. 1 is a graphical illustration of how the AGR model ranks an input face by an age-group-ordered training set to derive different age group rank-types.

Following the definitions of the sets A and B above, we define a third set C of age

groups, according to

$$C = \{c_\lambda | \lambda = 1, \cdots, w \wedge \forall \lambda, c_{\lambda+1} > c_\lambda\} \, , \tag{4}$$

where c_λ is an age group label and w is the number of age group labels.

Precisely, each $c_\lambda \in C$ is a subset of B. Thus each element of the set C of age groups is itself a set (of age values) contained in the set B and the sets c_λ are disjoint.

Further, the number of age groups in C is definitely less than the number of ages in B, that is $1 < w < q$.

The elements of each c_λ is determined from B by a range parameter, τ. Thus, we write $c_\lambda^\tau \subset B$.

Due to the nature of ageing and the challenge of insufficient data collection for its studies, the range parameter τ could be the same throughout the set C or may change for every $c_\lambda \in C$. This is necessary to ensure that the number of faces available to be mapped to each age group is relatively sizeable. However, as observed in (4), the ordering of B is retained in C as well. In our experiments, the value of τ was empirically determined based on the size of the dataset and the age distribution. This is necessary to ensure that the number of face images and their ages in each age group are sufficient for ranking a face, otherwise, we risk underrepresenting an age group.

Having defined the age learning function f_1 in (3), we further define an age group matching function h which maps faces to age groups, given the age of the face as follows:

$$h(a_i, b_j) = c_\lambda^\tau \, , \tag{5}$$

so that

$$\forall a_i \exists b_j \text{ , such that } f(a_i) = b_j \, , \tag{6}$$

and

$$\forall a_i \exists b_j, c_\lambda \text{ , such that } h(a_i, b_j) = c_\lambda^\tau \, . \tag{7}$$

While the age learning function has to be approximated (by training), the age group matching function simply associates a face (given its age) to its appropriate age group, thus it requires no approximation or training. However, the age group matching function only applies to training images or images whose ages are known and these are the images that make up the reference image set for comparison during age group ranking. As earlier stated, images to which an input image will be compared during age group ranking should be images whose ages or age groups are known, we, therefore, used all training images as the reference image set. The next challenge, however, is how to determine the age group to which an input (test) image belongs and this is where an age group ranking function steps in. It is noteworthy to state, therefore, that while the age group matching function simply assigns a face to an age group given the exact age of the face, the age group ranking function is responsible for capturing and representing the correlation of each

face with each age group. So, the age group matching function requires prior knowledge of the age of a given face so that it can construct the training set as a reference image set organized into age groups, but the age group ranking function requires no prior knowledge of the age of an input face.

Rather than approximating the age group ranking function by training, the function is realized by computing some arithmetic and statistical measures to represent the correlation of each face with each age group. Since the age group of the input (test) image is supposedly unknown, by collecting such measures for all age groups, we are able to capture the correlation of a face with various age groups. This further embellishes each face with relevant information for learning the discriminatory properties of faces in terms of their ages and age groups and reduces the overhead that could have been incurred by learning the age group ranks. The result of this operation is a multivariate age group rank for each face image representing its correlation with every age group.

Given the set A of face images and the set C of age groups as earlier defined, we define a tuple $\vec{A}$ of sets of faces ordered by age groups as follows:

$$\vec{A} = \left(\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_w\right) , \tag{8}$$

and

$$\hat{A}_\lambda = \{a_{\lambda_1}, a_{\lambda_2}, ..., a_{\lambda_g}\} . \tag{9}$$

Each $\hat{A}_\lambda$, $(1 \leq \lambda \leq w)$, is a set of face images matched to the age group c_λ, w is the number of age groups as indicated in equation (5), each a_{λ_j}, $(1 \leq j \leq g)$ is a face image in the set $\hat{A}_\lambda$ and g is the number of face images in a particular age group. Since $\hat{A}_\lambda$ is a set, it means the face images in it are not necessarily ordered by age, but are definitely matched to the age group c_λ.

Given a face image a_i and a tuple $\vec{A}$ of faces ordered by their age groups, the age group ranking function f_2, which assigns an age group rank to image a_i to obtain an age-group-ranked face $\hat{a}_i$, is defined as follows:

$$f_2(a_i, \vec{A}) = \hat{a}_i . \tag{10}$$

At this point, each face image a_i has been transformed into a vector X_i of facial features; therefore, the age group rank $\hat{r}_i$ of each face a_i is a vector obtained by computing the Deviation of Feature Values (DoFV) between each face and every face in the tuple $\vec{A}$ of age grouped faces. The several operations abstracted in $f_2()$ are detailed in the following formulations.

Given a face a_i, with unknown age and age group, the age group rank $\hat{r}_i$ of a_i is obtained as follows:

$$\varsigma(a_i, a_{\lambda_j}) = \Delta_{i\lambda_j} , \tag{11}$$

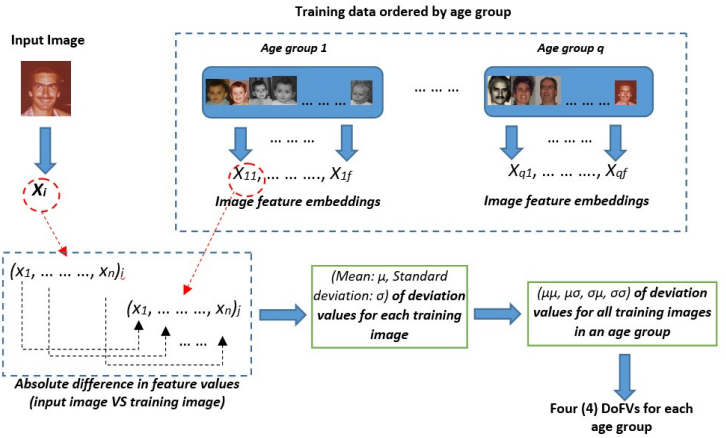

Fig. 2. DoFV computation.

where ς is the DoFV function, a_{λ_j} is the j_{th} face in the set $\hat{A}_\lambda$ of age-grouped faces and $\Delta_{i\lambda_j}$ is the obtained DoFV. DoFV is obtained by taking the absolute difference in feature values between an input image a_i whose age is unknown and an age grouped image a_{λ_j} whose age/age group is known. Then, for each age group, arithmetic and statistical measures of the differences in feature values are obtained for this particular input image and this provides the age group rank for the image at this particular age group. For each input image, this is repeated for all age groups and a vector of ranks is obtained for that input image, by concatenating the arithmetic and statistical measures of the DoFV obtained from all age groups. Therefore, the age group rank contains information about the statistical properties of images at feature, image, and age-group levels. Consequently, the age group rank obtained for each input image corresponds to the correlation of the feature values of the input image with the feature values of the various images in that age group. Hence, the obtained age group rank is actually a measure of the correlation of an input image with images of all age groups. With this information, the age learner (at training) can learn the correlation of each face with every age group, thus being able to better fit faces to their respective ages. Fig. 2 shows the DoFV computation procedure as explained above.

Suppose the facial features of a face image a_i is collected into the vector X_i of size n and each feature value in the vector X_i is indexed by $t, (1 \leq t \leq n)$, then the following formulations can be stated for DoFV for a given face a_i as follows:

$$\Delta_t = |X_{it} - X_{\lambda_{jt}}| \,, \tag{12}$$

Δ_t being the DoFV for the t^{th} feature in the facial feature vector X_i, obtained as the

absolute difference between the t^{th} feature vector in the input face and the t^{th} feature vector in the j^{th} face of the age group $\hat{A}_\lambda$.

Then, for each face feature vector X_i ($1 \leq i \leq p$; p being the number of face images), two arithmetic and statistical measures of the DoFV are taken, namely the arithmetic mean and the standard deviation denoted as $\Delta^{\mu}_{i\lambda_j}$ and $\Delta^{\sigma}_{i\lambda_j}$, respectively.

Subsequently, for each age group, four arithmetic and statistical measures are obtained as mean of means ($\Delta^{\mu\mu}_{i\lambda}$), mean of standard deviations ($\Delta^{\mu\sigma}_{i\lambda}$), standard deviation of means ($\Delta^{\sigma\mu}_{i\lambda}$) and standard deviation of standard deviations ($\Delta^{\sigma\sigma}_{i\lambda}$), as shown in equations (13) to (16), respectively.

$$\Delta^{\mu\mu}_{i\lambda} = \frac{\sum_{j=1}^{g} \Delta^{\mu}_{i\lambda_j}}{g} \tag{13}$$

$$\Delta^{\mu\sigma}_{i\lambda} = \frac{\sum_{j=1}^{g} \Delta^{\sigma}_{i\lambda_j}}{g} \tag{14}$$

$$\Delta^{\sigma\mu}_{i\lambda} = \sqrt{\frac{\sum_{j=1}^{g} \left(\Delta^{\mu}_{i\lambda_j} - \Delta^{\mu+}_{i\lambda}\right)^2}{g-1}} \tag{15}$$

$$\Delta^{\sigma\sigma}_{i\lambda} = \sqrt{\frac{\sum_{j=1}^{g} \left(\Delta^{\sigma}_{i\lambda_j} - \Delta^{\mu+}_{i\lambda}\right)^2}{g-1}} \tag{16}$$

For every face image a_i, these four values are obtained for each age group resulting in $4 \times w$ values (w being the number of age groups), since the age/age group of the query face is supposedly unknown.

The age group rank $\hat{r}_i$ is obtained by performing arithmetic multiplication and division operations between these four values in eight different ways. These eight values are computed for each age group, giving a maximum of $8 \times w$ (w being the number of age groups) values making up the age group rank of each image. The selected eight values, called rank-types, are computed as $\varpi_{i\lambda_1} = \Delta^{\mu\mu}_{i\lambda} \times \Delta^{\sigma\mu}_{i\lambda}$; $\varpi_{i\lambda_2} = \Delta^{\mu\sigma}_{i\lambda} \times \Delta^{\sigma\sigma}_{i\lambda}$; $\varpi_{i\lambda_3} = \Delta^{\mu\mu}_{i\lambda} / \Delta^{\sigma\mu}_{i\lambda}$; $\varpi_{i\lambda_4} = \Delta^{\mu\sigma}_{i\lambda} / \Delta^{\sigma\sigma}_{i\lambda}$; $\varpi_{i\lambda_5} = \Delta^{\mu\mu}_{i\lambda} \times \Delta^{\mu\sigma}_{i\lambda}$; $\varpi_{i\lambda_6} = \Delta^{\sigma\mu}_{i\lambda} \times \Delta^{\sigma\sigma}_{i\lambda}$; $\varpi_{i\lambda_7} = \Delta^{\mu\mu}_{i\lambda} / \Delta^{\mu\sigma}_{i\lambda}$ and $\varpi_{i\lambda_8} = \Delta^{\sigma\mu}_{i\lambda} / \Delta^{\sigma\sigma}_{i\lambda}$, where $\varpi_{i\lambda_1}, \varpi_{i\lambda_2}, ..., \varpi_{i\lambda_8}$ are the eight rank-types. For space constraints, we leave out the equations for these ranks as they can be easily deduced from equations (13)-(16).

Consequently, the rank $\hat{r}_i$ ($1 \leq i \leq p$; p being the number of face images) of each image is made up by concatenating the obtained rank values of all the age groups for

each rank type, as follows:

$$\tilde{r}_{ik} = \varpi_{i1_k} \oplus \varpi_{i2_k} \oplus \ldots \varpi_{iw_k} \ , \tag{17}$$

where $\varpi_{i1_k}, \varpi_{i2_k}, ..., \varpi_{iw_k}$ are the values for rank-type k ($1 \leq k \leq 8$) for each of the w age groups and $\tilde{r}_{ik}$ is the resulting vector for rank-type k for all age groups. Finally, the rank $\hat{r}_i$ of an image a_i for all rank types is given as

$$\hat{r}_i = \tilde{r}_{i1} \oplus \tilde{r}_{i2} \oplus ... \oplus \tilde{r}_{it} \ , \tag{18}$$

where t is the number of different rank-types and in this case, $t = 8$. Eventually, the age group rank obtained for a face image a_i is concatenated with the facial features of a_i to obtain an age-group-ranked face image $\hat{a}_i$ as stated in equation (17). Thus, we can write

$$\hat{X}_i = X_i \oplus \hat{r}_i \ , \tag{19}$$

where $\hat{X}_i$ is the age-group-ranked feature vector of the age-group-ranked face $\hat{a}_i$. Equation (3) can therefore be rewritten as in equation (20) so that a learning algorithm can then approximate this function:

$$f_1(\hat{X}_i) = b_j \ . \tag{20}$$

The effect of this is that the learning algorithm has more age-relevant facial features to learn from in approximating this function and thereby estimating the exact age of a given face. Details of the learning algorithms are given in the next section.

Summarily, the entire process described produces enhanced features (low-dimensional and discriminatory) that can be supplied as input to a learning algorithm to predict the exact age of a given face. Links to the dataset and source code will be made available after acceptance.

5. Experiments, Results, and Discussions

5.1. Experimental Settings

Our age group ranking (AGR) model was implemented in MATLAB R2016a. We used Local Binary Patterns (LBP) [34], raw image pixel features and deep features (VGG16 [45], Inception-V3 [46], Xception [11] and VGGFace [36]) as face descriptors and used Support Vector Regression (SVR) with Radial Basis Function (RBF) kernel (to capture the non-linearity of face ageing) for age learning. Experiments were performed on four different facial ageing datasets, namely FGnet [12], which contains 1002 images of 82 individuals, PAL [32], with 1046 images of 575 individuals and a new dataset, FAGE (Facial expression, Age, Gender and Ethnicity) with 540 images of 328 individuals, and Adience [17]. For Adience dataset, the age labels are not exact ages but age groups, therefore in place of SVR, we used the Discriminant Analysis classifier with a

quadratic kernel, henceforth referred to as Quadratic Discriminant Analysis (QDA), for age group learning. For SVR, the age learning optimization algorithm used was Sequential Minimal Optimization. The estimated Lagrange multipliers for the support vectors as well as the optimization coefficients were initialized to zero and training was done for 1000 iterations. For QDA, the misclassification cost was a square matrix whose values were derived from the distance between the age classes and the prior probabilities were empirically determined from the frequencies of the age classes.

Although our model was originally formulated for regression, in the case of Adience dataset, the model is adapted to classification by using the supplied age groups both for age group ranking and as the responses to be learned in age classification, so Adience does not require the age group matching function of equation (5). As will be seen in Tab. 1, the age groups in Adience are already too wide and too few (only eight of them), so merging two or three age groups into one will only increase the age gap and reduce the number of age groups available for age group ranking. As will be seen in the results, this limitation affected the result of age group ranking on Adience dataset.

Our choice of these datasets is because they are publicly available and have long-standing usage in age estimation research. FAGE was collected for this research, specifically to investigate age estimation on indigenous African faces (a problem rarely studied). To investigate the generalization ability of the trained models, we also performed cross-dataset validation (which is rarely done because of the peculiarities of each dataset) between three of the four datasets studied (Adience was excluded as it does not include exact ages).

For training and validation on FGnet, we adopted the popular subject-exclusive Leave-One-Person-Out (LOPO) cross-validation protocol as described in [19]. For PAL and FAGE datasets, we used 5-fold cross-validation and for Adience, we used the subject-exclusive 5-fold cross-validation as suggested in [17]. The evaluation metrics that have become standards for age estimation are Mean Absolute Error (MAE) and Cumulative Score (CS). MAE is the average of the absolute difference between the actual and predicted ages while CS is the percentage of the dataset whose ages are correctly predicted at a given error level. However, for Adience, the recommended and popular evaluation metric is the percentage classification accuracy (ACC) and is usually divided into exact accuracy and 1-off accuracy (taking as correct, predictions off by one age group). Thus, with MAE, the lower the value, the better the performance, while with ACC and CS, the higher the value, the better the performance.

Each dataset was split into age groups such that each age group spanned about five years (i. e. $\tau \approx 5$) except in cases where there were not enough images to represent an age group. For Adience, we simply used the age group classes that came with the dataset as the age groups for ranking. Tab. 1 shows the division of the age groups within each of the four datasets. Age group ranking was thus performed on each dataset using these age group divisions, thus resulting in 11, 12, 10, and 8 age group ranks for FGnet, PAL,

Tab. 1. Datasets divisions by age group.

Adience Dataset		FAGE Dataset		FGnet Dataset		PAL Dataset	
Age group	# faces	Age group	# faces	Age group	# faces	Age group	# faces
0 – 2	2509	0 – 5	44	0 – 4	194	18 – 20	116
4 – 6	2140	6 – 10	97	5 – 8	153	21 – 25	274
8 – 13	2292	11 – 15	66	9 – 12	135	26 – 30	86
15 – 23	1887	16 – y20	71	13 – 16	130	31 – 35	44
25 – 36	5549	21 – 25	142	17 – 20	118	36 – 40	34
38 – 46	2429	26 – 30	63	21 – 24	64	41 – 45	38
48 – 58	937	31 – 35	27	25 – 28	51	46 – 50	34
60 – 100	872	36 – 40	10	29 – 32	38	51 – 55	40
–	–	41 – 45	13	33 – 36	36	56 – 60	12
–	–	46 – 80	7	37 – 40	23	61 – 70	162
–	–	–	–	41 – 69	60	71 – 80	139
–	–	–	–	–	–	81 – 93	67
Total	18615	Total	540	Total	1002	Total	1046

FAGE, and Adience datasets, respectively. For brevity, AGR refers to age group ranking in all tables and figures where it appears.

A note on Adience dataset

According to [17], the Adience dataset is said to contain 26 580 images of 2 284 subjects. However, the dataset downloadable from the authors' website contains exactly 19 370 images (see Table I of [37]) out of which only 18 615 images are labelled with age groups. This is further confirmed by our observation of the fact that the breakdown provided in Table II in [17] does not in any way add up to 26 580 images. More so, we observed that the age labels in the available dataset (from their website) are somewhat inconsistent with what is provided in the paper. We worked around this to aggregate the scattered pieces of age labels into coarse age groups and we eventually ended up with eight labels similar to the ones indicated in [17], but some of our age groups covered wider ranges.

Face preprocessing and feature extraction

Each face image was preprocessed by converting it into an 8-bit grayscale image (if coloured) resulting in pixel intensity values between 0 and 255. From the grayscale image, the face was detected and aligned using a multi-stage method described in [4]. Before feature extraction, images were resized to various sizes depending on the feature descriptor to be used. For LBP and raw image pixels features, images were resized to 120×100 pixels; for VGG16 and VGGFace features, images were resized to 224×224 pixels; for Inception-V3 and Xception, images were resized to 299×299 pixels. For raw pixels and LBP features, feature histograms were obtained from ten (10) face regions

defined around the forehead, the outer eye corners, the inner eye corners, the area under the eyes, the area between the two eyes, the nose bridge, the nose lines, the cheek area, the cheekbone areas, and the periocular face region. Features histograms from each defined face region were aggregated and compacted using the method in [5]. We selected compaction ranges of 5 and 10 for raw pixels and LBP, respectively. For LBP features, $\text{LBP}_{8,1}$ (8-pixel neighbourhood and pixel distance/radius of 1) was used. The resulting features from each descriptor were then used to rank each face as described in the previous section and to obtain age group ranks for each face for all age groups. The resulting age group ranks were passed into SVR/QDA for age/age-group learning and prediction. We then carried out comparative analyses of the performance of age group ranking on each dataset and each feature descriptor.

5.2. Dataset-specific results

To investigate the impact of our AGR model, we trained SVR/QDA on:

1. the entire features vector before age group ranking (high-dimensional features);
2. the entire features along with the age group ranks (high-dimensional features);
3. the age group ranks alone (low-dimensional features).

Each feature type (before and after age group ranking), was normalized by scaling the feature values to a narrow interval $(0, 1)$ using the standard deviation and means of the feature values. The MAEs obtained in each case are reported in Tab. 2. The value of x in Tab. 2 refers to the number of rank-types multiplied by the number of age groups in each dataset. So, from Tab. 1 and Tab. 2, it can be inferred that $x = 64$, 80, 88, and 96 for Adience, FAGE, FGnet, and PAL datasets respectively. From Tab. 2, it is obvious that the age group ranks significantly reduced the age estimation error in all cases even though it provides significantly low-dimensional features for age learning.

We further investigated the performance of each of the eight (8) rank-types for age estimation and reported the results in Tab. 3. From Tab. 3, it can be observed that rank-types 3, 4, and 6 generally gave the lowest MAE (values in boldface). For all raw pixel features, rank-types 4 and 6 seem to give the best performance, except on PAL dataset where rank-type 8 performed better than the two and that was the only instance where rank-type 8 performed the best in the entire experiment. For LBP features, rank-types 3 and 6 gave the best performances. For both VGG16 and VGGFace features, rank-types 4 and 6 were the best. For Inception and Xception features, rank-types 3 and 6 were the best; in fact, with Xception, rank-type 3 consistently outperformed rank-type 6 on all datasets. On Adience dataset, the best performing rank-types are rank-types 3 and 6; on FAGE dataset, the best performing is rank-type 6; on FGnet dataset, the best performing are rank-types 3, 4, and 6, but predominantly 4; while on PAL dataset, the best performing are rank-types 3, 4, 6 and 8 (but the good performance of rank-type 8 is more like an outlier in the entire set of experiments).

Tab. 2. MAE of age estimation results before and after age group ranking. *Ftr.* stands for *feature(s)* and *dim.* stands for *dimensionality*.

Experiment setting	Ftr. type	Ftr. dim.	ACC [%] Adience	MAE (years) FAGE	FGnet	PAL
Before AGR (features only)	Raw pixel	520	(31.30, 56.79)	7.02	8.43	14.44
	LBP	260	(29.59, 58.09)	6.56	8.36	12.32
	VGG-16	4096	(19.06, 52.12)	6.25	6.94	10.39
	VGGFace	2622	(18.89, 43.31)	5.18	4.65	5.07
	Incep-V3	2048	(22.67, 41.97)	6.49	6.14	12.34
	Xception	2048	(19.82, 36.51)	6.97	6.78	11.96
After AGR (features + ranks)	Raw pixel	$520+x$	(36.93, 59.93)	6.72	8.36	13.23
	LBP	$260+x$	(43.83, 64.54)	4.29	4.99	7.29
	VGG-16	$4096+x$	(19.25, 52.18)	6.10	6.77	10.19
	VGGFace	$2622+x$	(18.71, 42.52)	5.05	4.52	5.00
	Incep-V3	$2048+x$	(25.06, 45.72)	6.44	5.83	12.05
	Xception	$2048+x$	(17.93, 33.83)	6.95	6.26	11.68
After AGR (ranks only)	Raw pixel	x	(60.24, 71.70)	6.22	7.27	12.44
	LBP	x	(61.75, 75.48)	3.11	2.98	5.17
	VGG-16	x	(53.02, 74.94)	3.55	3.51	6.36
	VGGFace	x	(67.90, 90.28)	3.71	2.84	4.52
	Incep-V3	x	(52.47, 75.38)	6.70	3.25	13.37
	Xception	x	(52.68, 75.26)	6.88	3.43	11.60

This is significant as it shows that we can even lower age estimation error by using just one of the rank-types, thereby dropping the dimension of features needed for age learning from x to $x/8$; meaning just 8 feature dimension for Adience, 10 for FAGE, 11 for FGnet and 12 for PAL datasets. One observable similarity in the computation of these three best-performing rank-types is the fact that they all involve either the standard deviation of means ($\sigma\mu$) or the mean of standard deviations ($\mu\sigma$) as described in Subsection 4.2. This shows that the combination of statistical and arithmetic measures of the facial features properly captured the relationship between facial features within and across age groups in low dimensions.

As expected, the performance of these rank-types on Adience is still relatively poor. This is due to the few age groups *vis-a-vis* the dataset size – there are only 8 age groups for ranking over 18 000 images. For this reason, we investigated the combination of the different best-performing rank-types as well as the best-performing feature types on Adience and reported the results in Tab. 4. Interestingly, with the proper combinations of rank-types as well as feature types, the performance improves significantly and the best result was obtained with the combination of rank-types 3 and 6 on the combination

Tab. 3. MAE of age estimation with each rank-type (*rt*). Only exact ACC is shown for Adience.

Ftr. type	ACC [%]	MAE (years)		
rt 1 to 8	Adience	FAGE	FGnet	PAL
Raw pixel	48.79, 52.64, 53.58, 55.63, 28.69, **62.35**, 26.56, 33.45	5.43, 6.17, 4.52, 4.71, 6.11, **4.23**, 7.15, 6.64	6.68, 7.05, 5.51, **4.93**, 7.42, 5.02, 9.25, 8.25	13.61, 15.07, 13.45, 13.35, 10.44, 11.77, 13.3, **9.84**
LBP	51.34, 30.69, **57.05**, 36.47, 28.95, 49.70, 24.58, 34.77	1.93, 2.98, 2.40, 3.18, 4.74, **1.71**, 7.21, 4.88	1.88, 3.35, 2.17, 3.27, 7.63, **1.79**, 9.61, 5.44	4.21, 7.92, 3.53, 7.05, 10.14, **3.29**, 12.99, 8.09
VGG16	38.13, 43.11, 45.72, **48.77**, 36.43, 46.38, 35.64, 38.80	3.13, 2.21, 2.91, 2.57, 5.02, **2.10**, 6.78, 4.45	5.32, 3.71, 3.30, **2.55**, 5.61, 2.67, 6.92, 5.85	9.57, 7.22, 5.72, **5.10**, 7.27, 5.71, 11.11, 8.49
VGGFace	52.48, 54.41, 66.99, **67.82**, 58.04, 63.22, 30.08, 31.82	3.22, 2.95, 3.14, 3.01, 4.36, **2.20**, 7.10, 7.09	3.77, 3.41, 2.05, **1.96**, 3.74, 2.09, 7.67, 6.86	6.27, 5.87, 4.27, **3.99**, 4.47, 4.40, 11.84, 13.66
Incep-V3	42.30, 47.17, 47.02, 45.12, 39.04, **49.65**, 35.41, 38.49	5.76, 6.42, 5.25, 5.48, 5.97, **5.04**, 7.48, 6.36	4.29, 4.27, **2.44**, 2.61, 5.52, 2.76, 7.74, 6.84	14.82, 14.30, **11.03**, 11.34, 12.93, 11.73, 15.23, 14.26
Xception	36.41, 45.20, **46.62**, 45.14, 39.33, 46.61, 36.36, 39.37	7.20, 6.95, **5.96**, 5.95, 6.62, 5.99, 7.10, 7.00	5.02, 4.31, **2.84**, **2.84**, 5.56, 3.37, 7.80, 6.17	12.44, 11.95, **9.72**, 9.97, 10.65, 9.93, 14.03, 13.53

of VGGFace, LBP, Raw Pixel, Inception, and Xception features. Fig. 3 shows sample images from the four datasets for which age prediction with AGR succeeded and those for which it failed using the best-performing features.

Tab. 5 shows some of the most recently reported state-of-the-art results on Adience, FGnet, and PAL datasets (FAGE is a relatively new dataset, so there are no existing methods on it to compare with). In the table, the asterisk (*) in the third column (ftrs. dim.) refers to those in which the exact feature dimension was not explicitly reported in the literature. However, it is common knowledge that most of the deep learning features are in the order of thousands, while our method uses features in the order of tens. From Tab. 5, it is seen that our method competes significantly with the best of these methods achieving the lowest MAEs on FGnet (1.79 years) and PAL (3.29 years) and the best exact accuracy (85.1%) on Adience; VLRIX stands for the combination of VGGFace, LBP, Raw pixel, Inception and Xception features as seen in the third to the last row of Tab. 4. We consider this a significant achievement considering the highly reduced feature dimension generated by our AGR model and the fact that it achieves this even with fairly simple feature extraction techniques (raw pixel and LBP), thus making our

Tab. 4. Different combinations of rank-types and feature types on Adience dataset. Abbreviations: I – Inception, L – LBP, R – Raw pixel, V – VGGFace, V16 – VGG16, X – Xception.

Rank-types	Feature types	Ftr. dim.	ACC (%)	
			Exact$_{\pm \text{ std.}}$	1-off$_{\pm \text{ std.}}$
3, 4	All	96	$83.7_{\pm 2.10}$	$93.2_{\pm 1.07}$
3, 6	All	96	$84.0_{\pm 2.79}$	$93.9_{\pm 1.26}$
4, 6	All	96	$82.1_{\pm 2.91}$	$93.1_{\pm 1.54}$
3, 4, 6	All	144	$83.7_{\pm 2.56}$	$93.6_{\pm 1.22}$
3, 4	X, I	32	$55.8_{\pm 4.31}$	$78.4_{\pm 2.23}$
3, 6	X, I, L, R	64	$79.4_{\pm 2.12}$	$89.5_{\pm 1.18}$
4, 6	V16, V, L, R, I	80	$83.2_{\pm 2.88}$	$93.4_{\pm 1.34}$
4, 6	V16, V, L, R, X	80	$83.4_{\pm 3.02}$	$93.6_{\pm 1.47}$
3, 6	V16, V, L, R, X	80	$84.8_{\pm 3.11}$	$94.2_{\pm 1.21}$
3, 6	V16, V, L, R, I	80	$84.5_{\pm 2.88}$	$94.0_{\pm 1.38}$
3, 6	V, L, R, I, X	80	$85.1_{\pm 2.33}$	$94.6_{\pm 0.88}$
3, 4, 6	V, L, R	72	$85.5_{\pm 3.12}$	$94.3_{\pm 1.15}$
3, 4, 6	V, L, R, V16	96	$84.6_{\pm 2.99}$	$93.7_{\pm 1.39}$

results more easily reproducible. All these results had been achieved with features of relatively low dimension – 80 on Adience, 11 on FGnet, and 12 on PAL.

CS often gives a better picture of the performance of an age estimation algorithm at different levels of the prediction error. We plotted our CS scores along with some of the best results on FGnet for which CS plots were reported and compared the results. Fig. 4 further confirms the significant improvement offered by our AGR model (AGR-LBP-r6 and AGR-VGGFace-r4) on FGnet. At an error level of 0, only EBIF [14] started ahead of the AGR model and AGR overtook it at error level 1. AGR performs at par with GEF up to error level 1 after which AGR significantly overtakes. Generally, from error level 2 upwards, AGR outperforms all the compared methods and finishes far ahead of them with CS of 95% at error level 5 and 99% at error level 10. Previous works on PAL rarely report their CS scores so there will be no basis for such comparisons, thus we leave out the CS curve on PAL. Also, because the FAGE dataset is new, there are no previous results with which we can compare it.

5.3. Cross-Dataset Validation

To better study the generalization of our model, we performed cross-dataset validation in two settings:

1. on FGnet and PAL datasets;
2. on FGnet, PAL, and FAGE datasets.

Tab. 5. Comparison with previous results on Adience, FGnet and PAL. rt: rank-type. Note the 3rd column: filters dimension.

Method	Year	filters dimension (Adience, FGnet, PAL)	ACC [%] Adience (Exact, 1-off)	MAE (years) FGnet	PAL
EBIF [14]	2011	EBIF*	–	3.17	–
W-RS [50]	2013	100–900	–	–	5.99
Joint-Learn [6]	2014	LBP*	–	–	5.26
DeepRank [49]	2015	500	–	–	4.31
GEF [30]	2015	LBP,BIF,HOG*	–	2.81	–
CNN [26]	2015	CNN ftrs.*	(50.7, 84.7)	–	–
DA [39]	2017	VGG-16 ftrs.*	(60.0, 94.5)	–	–
DNN [41]	2017	VGG-16 ftrs.*	(62.8, 95.8)	–	–
ODFL [28]	2017	CNN ftrs.*	–	3.89	–
All-in-one [37]	2017	CNN ftrs.*	–	2.00	–
DEX [40]	2018	VGG-16 ftrs.*	(64.0, 96.6)	3.09	–
Group-n [47]	2018	VGG-16 ftrs.*	–	2.96	–
DRF [42]	2018	VGG-16 ftrs.*	–	3.85	–
CNN2ELM [16]	2018	CNN ftrs.*	(66.49, –)	–	–
Joint-Learn [31]	2018	$LBP_{(8,1)}$	–	–	5.26
MVL [35]	2018	CNN ftrs.*	–	2.68	–
BridgeNet [27]	2019	CNN ftrs.*	–	2.56	–
TransLearn [15]	2019	4096 VGG-16 ftrs.	–	–	3.79
SORD [13]	2019	VGG-16 ftrs.*	(59.6, –)	–	–
ODL [29]	2019	VGGFace ftrs.*	–	2.92	3.99
DDRF [43]	2019	VGG-16 ftrs.*	–	3.47	–
C3AE [51]	2019	*	–	2.95	–
DOEL [48]	2020	ResNet ftrs. *	–	3.44	–
DLC [1]	2020	CNN ftrs.*	(83.1, 93.8)	–	–
SR [33]	2020	CNN ftrs.*	–	–	8.33
DCN [23]	2022	VGG ftrs.*	–	2.13	–
ABC+Swin [44]	2023	Transformer ftrs.*	(56.1, –)	2.52	–
AGR-LBP (rt6)	Ours	[8, 11, 12]	(49.7, 68.9)	**1.79**	**3.29**
AGR-VLRIX (rt3+rt6)	Ours	[80, –, –]	(**85.1**, 94.6)	–	–

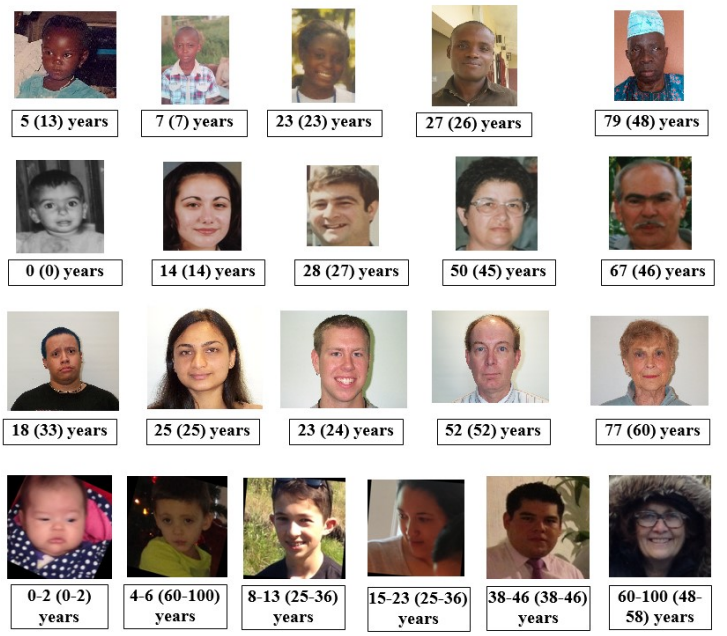

Fig. 3. Sample images and their true/predicted ages. from the 1st to the last row: FAGE, FGnet, PAL and Adience. Predicted ages are in parentheses.

In both settings, we used LBP (rank-type 6) and VGGFace (rank-type 4) features since they were the two best-performing features. In the second setting, we trained and tested the model on a combination of FGnet, PAL, and FAGE datasets. The Adience dataset is not used for Cross-dataset validation because it does not contain exact ages and is therefore unsuitable for a regression task as is the case with the other 3 datasets.

In setting 1, since both datasets cover separate age ranges, we selected the intersection of the age ranges covered (i. e. 18-69 years) and selected all faces falling within this age range. We found 362 FGnet images and 820 PAL images within this age range, making 1182 images altogether. We then ranked this new set of 1182 images on the entire set of FGnet and referred to it as FG-ranked, we also ranked it on the entire set of PAL images and referred to it as PAL-ranked. We trained and tested FG-ranked and PAL-ranked datasets using 5-fold cross-validation and obtained MAEs of 8.86 and 6.27 years with LBP features and 4.55 and 4.32 years with VGGFace features on FG-ranked and PAL-ranked datasets, respectively. As expected, the MAEs are higher in the cross-dataset environment, however, the result is worse when FGnet images are used to rank the data. This is because FGnet has 44 images less than PAL and FGnet contains 7 missing ages,

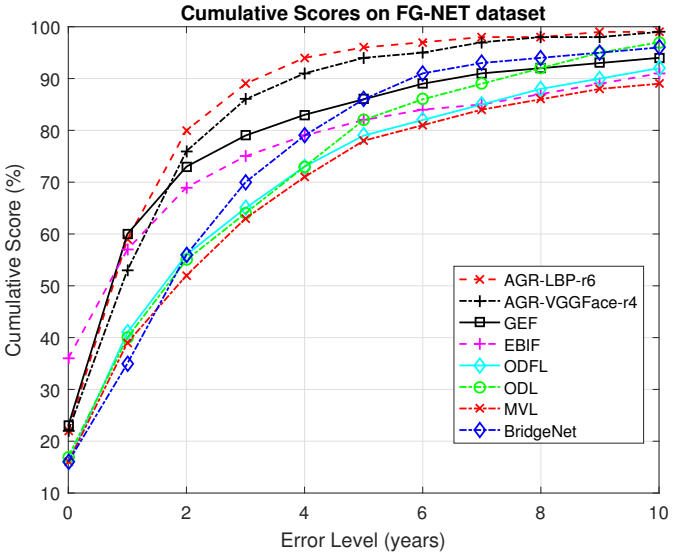

Fig. 4. CS curves of best-reported works on FGnet

while PAL contains only 1 missing age. PAL also covers a wider age range and contains more images for its age groups than FGnet. This goes to show that with more images available for age ranking and more ages represented within each age group, AGR offers better performance.

In the second setting, because of the differences in the number of age groups in each of the combined datasets, we created a new set of 15 age groups covering all the age groups in all three datasets and ranked each image in the combined dataset on this. There are a total of 2715 images in the combined dataset. We trained and tested with 5-fold cross-validation and obtained MAEs of 4.03 years and 4.33 years for VGGFace and LBP, respectively. However, the increased error rate is attributed to the ethnic diversity of the three datasets and the possibility that the age groups have become relatively too much for the dataset size.

The improved performance of VGGFace over LBP is an indication of the expressiveness of deep features in more complicated settings such as cross-dataset validation and with more data (as in setting 2). Generally speaking, the MAEs in both cross-dataset validation settings did not soar beyond expectations despite the wide inter-dataset variations; this is a pointer to the robustness of the AGR model and the intuition of age group ranking.

6. Conclusion

In this work, an age group ranking approach for facial age estimation was developed. The developed model uses the intuition that age can be better estimated from faces when there is sufficient information about other faces in several different age groups to rank a query face. The developed method was tested and validated on four datasets (FAGE, FGnet, PAL, and Adience). Experiments were performed on these datasets using standard protocols and the results compete significantly with the state-of-the-art age estimation methods. We further investigated the generalization of the method using cross-dataset validation and it turned out that the developed AGR method gives relatively good performance even across different datasets. The intuition of age group ranking developed here is superior to the existing age ranking methods in that age group ranking ranks images by age group rather than by exact ages thus making more data available for an image to be ranked. This is done without the need for prior knowledge of a particular age group rank via learning as the age ranking model uses available aging information from all age groups to rank a given face. More interestingly, the AGR model does not depend extensively on deep learning models as in current works but still competes significantly with deep-learning-based age estimation models. The findings from this work show that despite the impressive results of deep learning in recent times, the impact of age group ranking on face-based age estimation is indeed significant and should not be discarded. This work has also shown that age estimation via age-group ranking is more intuitive and gives better performance than direct age estimation from a single face.

The major limitation of the AGR model is that it does not fit directly into a deep learning architecture as it requires features to be extracted and enhanced before it is been passed to a classifier/regressor. However, the AGR model works when on simple features such as raw pixels as well as deep features as the features are further enriched with age group information before they are passed into a classifier/regressor.

Future works could consider building deep learning models that can explore the relationship between faces in terms of their age groups while estimating the age of a given face image. Future works could also consider using more rank-types and different age groupings to understand the impact of the number of age groups *vis-a-vis* the age range and the number of images within each age group. Considering the impact of the statistical measures of variation used in DoFV, there is a need to explore more statistical measures that could improve age estimation accuracy.

Acknowledgments

The authors would like to thank Prof. Andreas Lanitis for providing his copy of the FGnet dataset for this research. We also like to thank Dr. E. J. Dansu for his contribution in reviewing and correcting the mathematical equations.

References

[1] O. Agbo-Ajala and S. Viriri. Deeply learned classifiers for age and gender predictions of unfiltered faces. *Scientific World Journal*, 2020:1289408, 2020. doi:10.1155/2020/1289408.

[2] J. D. Akinyemi. *GWAgeER; A GroupWise Age-Ranking Approach to Age Estimation from Still Facial Image.* Master's thesis, University of Ibadan, Ibadan, 2014. 161 pages. doi:10.13140/RG.2.1.2495.1763, https://ibadan.academia.edu/AkinyemiDamilola/Theses.

[3] J. D. Akinyemi and O. F. W. Onifade. An ethnic-specific age group ranking approach to facial age estimation using raw pixel features. In: *Proc. 2016 IEEE Symposium on Technologies for Homeland Security (HST)*, pp. 1–6. IEEE, Waltham, MA, USA, 10-11 May 2016. doi:10.1109/THS.2016.7819737.

[4] J. D. Akinyemi and O. F. W. Onifade. A computational face alignment method for improved facial age estimation. In: *Proc. 2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, pp. 1–6. IEEE, Abuja, Nigeria, 12 2019. doi:10.1109/ICECCO48375.2019.9043246.

[5] J. D. Akinyemi and O. F. W. Onifade. Facial age estimation using compact facial features. In: *Computer Vision and Graphics: Proc. International Conference on Computer Vision and Graphics (ICCVG) 2020*, vol. 12334 of *Lecture Notes in Computer Science*, pp. 1–12. Springer International Publishing, Warsaw, Poland, Sep 14-16 2020. doi:10.1007/978-3-030-59006-2_1.

[6] F. Alnajar and J. Alvarez. Expression-invariant age estimation. In: *Proc. 25th British Machine Vision Conference (BMVC) 2014*, pp. 28.1–28.11. Nottingham, UK, 1-5 Sep 2014. (doi inoperative). doi:10.5244/C.28.14, https://bmva-archive.org.uk/bmvc/2014/papers/paper081/index.html.

[7] K.-Y. Chang and C.-S. Chen. A learning framework for age rank estimation based on face images with scattering transform. *IEEE Transactions on Image Processing*, 24(3):785–798, 2015. doi:10.1109/TIP.2014.2387379.

[8] K.-Y. Chang, C.-S. Chen, and Y.-P. Hung. A ranking approach for human ages estimation based on face images. In: *Proc. 2010 20th International Conference on Pattern Recognition (ICPR)*, pp. 3396–3399. Istanbul, Turkey, 23-26 Aug 2010. doi:10.1109/ICPR.2010.829.

[9] K.-Y. Chang, C.-S. Chen, and Y. P. Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. In: *Proc. 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 585–592. Colorado Springs, CO, USA, 20-25 Jun 2011. doi:10.1109/CVPR.2011.5995437.

[10] S. Chen, C. Zhang, M. Dong, J. Le, and M. Rao. Using Ranking-CNN for age estimation. In: *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 742–751. Honolulu, HI, USA, 21-26 Jul 2017. doi:10.1109/CVPR.2017.86.

[11] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807. Honolulu, HI, USA, 21-26 Jul 2017. doi:10.1109/CVPR.2017.195.

[12] T. F. Cootes, G. Rigoll, E. Granum, J. L. Crowley, S. Marcel, et al. Face and Gesture Recognition Working group, 2002. http://www-prima.inrialpes.fr/FGnet/, FGnet – Project IST-2000-26434. Original URL is not operative, copy can be accessed at http://crowley-coutaz.fr/FGnet/html/home.html.

[13] R. Diaz and A. Marathe. Soft labels for ordinal regression. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4733–4742. Long Beach, CA, USA, 15-20 Jun 2019. doi:10.1109/CVPR.2019.00487.

[14] M. Y. E. Dib and H. M. Onsi. Human age estimation framework using different facial parts. *Egyptian Informatics Journal*, 12(1):53–59, 2011. doi:10.1016/j.eij.2011.02.002.

[15] F. Dornaika, I. Arganda-Carreras, and C. Belver. Age estimation in facial images through transfer learning. *Machine Vision and Applications*, 30(1):177–187, 2019. doi:10.1007/s00138-018-0976-1.

[16] M. Duan, K. Li, and K. Li. An ensemble cnn2elm for age estimation. *IEEE Transactions on Information Forensics and Security*, 13(3):758–772, 2018. doi:10.1109/TIFS.2017.2766583.

[17] E. Eidinger, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *IEEE Transactions on Information Forensics and Security*, 9(12):2170–2179, 2014. doi:10.1109/TIFS.2014.2359646.

[18] Y. Fu, G. Guo, and T. S. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1955–1976, 2010. doi:10.1109/TPAMI.2010.36.

[19] X. Geng, Z.-H. Zhou, Y. Zhang, G. Li, and H. Dai. Learning from facial aging patterns for automatic age estimation. In: K. Nahrstedt, M. Turk, Y. Rui, W. Klas, and K. Mayer-Patel, eds., *Proc. MM '06: Proc. 14th ACM International Conference on Multimedia*, pp. 307–316. ACM, Santa Barbara, CA USA, 23-27 Oct 2006. doi:10.1145/1180639.1180711.

[20] P. A. George and G. J. Hole. Factors influencing the accuracy of age-estimates of unfamiliar faces. *Perception*, 24(9):1059–1073, 1995. doi:10.1068/p241059.

[21] G. Guo, G. Mu, Y. Fu, and T. S. Huang. Human age estimation using bio-inspired features. In: *Proc. 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pp. 112–119. Miami, FL, USA, 20-25 Jun 2009. doi:10.1109/CVPRW.2009.5206681.

[22] T. Hassner. The OUI-Adinece Face Image project. The Open University of Israel. `https://talhassner.github.io/home/projects/Adience/Adience-data.html`, [Accessed May 2024].

[23] C. Kong, Q. Luo, and G. Chen. Learning deep contrastive network for facial age estimation. In: *Proc. International Joint Conference on Neural Networks (IJCNN)*. IEEE, Padua, Italy, 18-23 Jul 2022. doi:10.1109/IJCNN55064.2022.9892308.

[24] Y. H. Kwon and N. da Vitoria Lobo. Age classification from facial images. In: *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (ICCVPR)*, p. 762–767. Seattle, WA, USA, 21-23 Jun 1994. doi:10.1109/CVPR.1994.323894.

[25] A. Lanitis. On the significance of different facial parts for automatic age estimation. In: *Proc. International Conference on Digital Signal Processing (DSP)*, vol. 2, pp. 1027–1030. Santorini, Greece, 01-03 Jul 2002. doi:10.1109/ICDSP.2002.1028265.

[26] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In: *Proc. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 34–42. Boston, MA, USA, 07-12 Jun 2015. doi:10.1109/CVPRW.2015.7301352.

[27] W. Li, J. Lu, J. Feng, C. Xu, J. Zhou, et al. BridgeNet: A continuity-aware probabilistic network for age estimation. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1145–1154. Long Beach, CA, USA, 15-20 Jun 2019. doi:10.1109/CVPR.2019.00124.

[28] H. Liu, J. Lu, J. Feng, and J. Zhou. Ordinal deep feature learning for facial age estimation. In: *Proc. 12th IEEE International Conference on Automatic Face and Gesture Recognition, (FG)*, pp. 157–164, 30 May – 03 Jun 2017. doi:10.1109/FG.2017.28.

[29] H. Liu, J. Lu, J. Feng, and J. Zhou. Ordinal deep learning for facial age estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):486–501, 2019. doi:10.1109/TCSVT.2017.2782709.

[30] K. H. Liu, S. Yan, and C.-C. J. Kuo. Age estimation via grouping and decision fusion. *IEEE Transactions on Information Forensics and Security*, 10(11):2408–2423, 2015. doi:10.1109/TIFS.2015.2462732.

[31] Z. Lou, F. Alnajar, J. M. Alvarez, N. Hu, and T. Gevers. Expression-invariant age estimation using structured learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(2):365–375, 2018. doi:10.1109/TPAMI.2017.2679739.

[32] M. Minear and D. C. Park. A lifespan database of adult facial stimuli. *Behavior Research Methods, Instruments, and Computers*, 36(4):630–633, 2004. doi:10.3758/BF03206543.

[33] S. H. Nam, Y. H. Kim, N. Q. Truong, J. Choi, and K. R. Park. Age estimation by super-resolution reconstruction based on adversarial networks. *IEEE Access*, 8:17103–17120, 2020. doi:10.1109/ACCESS.2020.2967800.

[34] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. doi:10.1109/TPAMI.2002.1017623.

[35] H. Pan, H. Han, S. Shan, and X. Chen. Mean-variance loss for deep age estimation from a face. In: *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5285–5294. Salt Lake City, UT, USA, 18-23 Jun 2018. doi:10.1109/CVPR.2018.00554.

[36] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In: *Proc. 26th British Machine Vision Conference (BMVC)*, pp. 41.1–41.12. Swansea, UK, 7-10 Sep 2015. (doi inoperative). doi:10.5244/c.29.41, `https://bmva-archive.org.uk/bmvc/2015/papers/paper041/`.

[37] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa. An all-in-one convolutional neural network for face analysis. In: *Proc. 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 17–24. Washington, DC, USA, 30 May – 03 Jun 2017. doi:10.1109/FG.2017.137.

[38] G. Rhodes. Lateralized processes in face recognition. *British Journal of Psychology*, 76(2):249–271, 1985. doi:10.1111/j.2044-8295.1985.tb01949.x.

[39] P. Rodríguez, G. Cucurull, J. M. Gonfaus, F. X. Roca, and J. Gonzàlez. Age and gender recognition in the wild with deep attention. *Pattern Recognition*, 72:563–571, 2017. doi:10.1016/j.patcog.2017.06.028.

[40] R. Rothe, R. Timofte, and L. V. Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2):144–157, 2018. doi:10.1007/s11263-016-0940-3.

[41] W. Samek, A. Binder, S. Lapuschkin, and K.-R. Müller. Understanding and comparing deep neural networks for age and gender classification. In: *Proc. 2017 IEEE International Conference on Computer Vision Workshops, (ICCVW)*, pp. 1629–1638. Venice, Italy, 22-29 Oct 2017. doi:10.1109/ICCVW.2017.191.

[42] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, et al. Deep regression forests for age estimation. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2304–2313. Salt Lake City, UT, USA, 18-23 Jun 2018. doi:10.1109/CVPR.2018.00245.

[43] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, et al. Deep differentiable random forests for age estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):404–419, 2019. doi:10.1109/tpami.2019.2937294.

[44] C. Shi, S. Zhao, K. Zhang, Y. Wang, and L. Liang. Face-based age estimation using improved swin transformer with attention-based convolution. *Frontiers in Neuroscience*, 17, 2023. doi:10.3389/fnins.2023.1136934.

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In: *Proc. 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, 7-9 May 2015. Published only on arXiv. http://arxiv.org/abs/1409.1556.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826. Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.308.

[47] Z. Tan, J. Wan, Z. Lei, R. Zhi, G. Guo, et al. Efficient group-n encoding and decoding for facial age estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(11):2610–2623, 2018. doi:10.1109/TPAMI.2017.2779808.

[48] J. C. Xie and C. M. Pun. Deep and ordinal ensemble learning for human age estimation from facial images. *IEEE Transactions on Information Forensics and Security*, 15(8):2361–2374, 2020. doi:10.1109/TIFS.2020.2965298.

[49] H.-F. Yang, B.-Y. Lin, K.-Y. Chang, and C.-S. Chen. Automatic age estimation from face images via deep ranking. In: *Proc. 26th British Machine Vision Conference (BMVC)*, pp. 55.1–55.11. Swansea, UK, 7-10 Sep 2015. (doi inoperative). doi:10.5244/C.29.55, https://bmva-archive.org.uk/bmvc/2015/papers/paper055/.

[50] C. Zhang and G. Guo. Age estimation with expression changes using multiple aging subspaces. In: *Proc. IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pp. 1–6. Arlington, VA, USA, 29 Sep – 02 Oct 2013. doi:10.1109/BTAS.2013.6712720.

[51] C. Zhang, S. Liu, X. Xu, and C. Zhu. C3AE: Exploring the limits of compact model for age estimation. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12579–12588. Long Beach, CA, USA, 15-20 Jun 2019. doi:10.1109/CVPR.2019.01287.

Joseph D. Akinyemi is currently with the University of York, York, United Kingdom. He received his Bachelor's degree in Computer Science from the University of Ilorin, Ilorin, Nigeria in 2010. He received his Master's degree in Computer Science from the University of Ibadan, Ibadan, Nigeria, in 2014 and a Ph.D. degree in Computer Science from the same institution in 2020. His research spans areas of Computer Vision such as facial and medical image processing as well as aspects of Natural Language Processing such as Sentiment Analysis. He is a 2022 Heidelberg Laureate Forum Fellow in Germany, a recipient of the Google Developers Machine Learning Bootcamp sponsorship for Sub-Saharan Africa and a member of the ACM.

Olufade F. W. Onifade is currently Professor of Computer Science at the University of Ibadan, Ibadan, Nigeria and a Deputy Director at the Open and Distance Learning Center of the University of Ibadan, Ibadan, Nigeria. He received his Bachelors degree in Mathematics and Computer Science at the Federal University of Agriculture, Abeokuta, Nigeria in 1998 and received his Masters degree at the University of Ibadan, Ibadan, Nigeria in 2002. He benefitted from the French government scholarship which led him to receive double doctorate degrees one from the University of Ibadan, Ibadan, Nigeria, and the other from Nancy 2 University, France, in 2010. His research interests are in Information Retrieval, Risk Management, Pattern Recognition and Computer Vision. He is a member of the IEEE, IAENG, ISKO and NCS. He has received a number of grants and awards including the MIT-ETT fellowship for Content Development and Delivery and the CV Raman Fellowship for African researchers in India. He is a well-cited author of over 80 papers in peer-reviewed journals and conferences.

An Attention-Based Deep Network for Plant Disease Classification

Asish Bera[1,*], Debotosh Bhattacharjee[2,3] and Ondrej Krejcar[3,4,5]
[1]*Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani, Rajasthan, India*
[2]*Department of Computer Science and Engineering, Jadavpur University, Kolkata, West Bengal, India*
[3]*Center for Basic and Applied Science, Faculty of Informatics and Management, University of Hradec Králové, Czech Republic*
[4]*Škoda Auto University, Mladá Boleslav, Czech Republic*
[5]*Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia*
[*]*Corresponding author: Asish Bera (asish.bera@pilani.bits-pilani.ac.in)*

Abstract Plant disease classification using machine learning in a real agricultural field environment is a difficult task. Often, an automated plant disease diagnosis method might fail to capture and interpret discriminatory information due to small variations among leaf sub-categories. Yet, modern Convolutional Neural Networks (CNNs) have achieved decent success in discriminating various plant diseases using leave images. A few existing methods have applied additional pre-processing modules or sub-networks to tackle this challenge. Sometimes, the feature maps ignore partial information for holistic description by part-mining. A deep CNN that emphasizes integration of partial descriptiveness of leaf regions is proposed in this work. The efficacious attention mechanism is integrated with high-level feature map of a base CNN for enhancing feature representation. The proposed method focuses on important diseased areas in leaves, and employs an attention weighting scheme for utilizing useful neighborhood information. The proposed Attention-based network for Plant Disease Classification (APDC) method has achieved state-of-the-art performances on four public plant datasets containing visual/thermal images. The best top-1 accuracies attained by the proposed APDC are: PlantPathology 97.74%, PaddyCrop 99.62%, PaddyDoctor 99.65%, and PlantVillage 99.97%. These results justify the suitability of proposed method.

Keywords: agriculture, attention, Convolutional Neural Networks, CNNs, Deep Learning, plant disease classification.

1. Introduction

Modernization in agriculture is reckoned as an emerging research area. Decent growth has been achieved over conventional engineering and laborious farming technologies using artificial intelligence and machine learning [29, 32]. A myriad of diversified applications of computer vision, in conjunction with the plethora of machine learning (ML) techniques, are playing important roles in agricultural development and in supporting the sustainability. Still, agriculture needs to be improved further to meet growing global food demands as envisaged by scientists. Several key challenges are identified in allied areas of

agriculture and related futuristic aspects, which seek more research attention, e.g., early disease prediction, crop yield estimation, crop health monitoring, and others [13, 34].

Automated plant disease prediction from leaf images using computer vision techniques is difficult due to wider variations in visual symptoms [40, 43]. In general, the images of various plants and crops are collected by the users/farmers and pre-processed with image processing techniques, such as noise removal, leaf-area detection, area of interest localization, edge map extraction, scaling, contrast adjustment, and others [27]. Several existing methods have applied pre-processing techniques for image segmentation, especially, segmented the region of interests (RoIs) representing infected regions/spots within the leaves, mask generation, and others [30]. Hence, these conventional pipelines essentially require a well-defined set of tasks to be accomplished before the feature extraction. To alleviate this, many deep learning methods have used actual images of plants and defined a deep network by integrating several sub-modules, such as generative adversarial networks (GAN) for augmentation [11] or U-Net for segmentation [41]. Some works have devised deep convolutional neural networks (CNNs) [10]. Also, lightweight CNNs have been studied for corn disease prediction and other applications due to lesser parametric complexities [13].

In recent years, attention mechanism plays as an indispensable component of modern deep architectures due its superior performance in solving diverse challenges in natural language processing, computer vision, and others [5, 7, 8, 46]. An attention method is effective for crop disease classification too [28]. Its aptness is witnessed for plant disease classification using self-attention [60]. Several prior works have used additional offline pre-processing, GAN-based augmentation, and additional sub-networks for localizing the infected leaf regions, as said above. Also, some methods are developed by transfer learning and ensemble techniques. Often, these existing techniques might overlook part and region based local information for subtle discrimination between infected similar types of leaves. Other than a global feature map, local descriptors are very useful for automated diagnosis and localizing finer details within a leaf. Because, various diseases can infect similar leaves of the same plant category [13, 47]. For example, the same tomato leaf can be infected by several diseases (e.g., mosaic, septoria, curl virus, etc.), and the differences among various plant leaves are naturally subtle. Thus, an efficient feature descriptor is crucial for discriminating and solving this problem.

The proposed Attention-based deep network for Plant Disease Classification (APDC) approach can be divided into three phases, shown in Fig. 1. A high-level feature map of an input leaf image is extracted using a backbone CNN in the first phase. The output feature vector is upsampled to a higher resolution for pooling the features from a set of fixed-size disjoint region proposals. These regions are spatially mapped with the upsampled base CNN's feature vector. Next, a bilinear pooling layer is applied to extract the upsampled convolutional features from each region [6]. The output dimension of these regions are kept the same as the output feature space of a base CNN. Overall, these

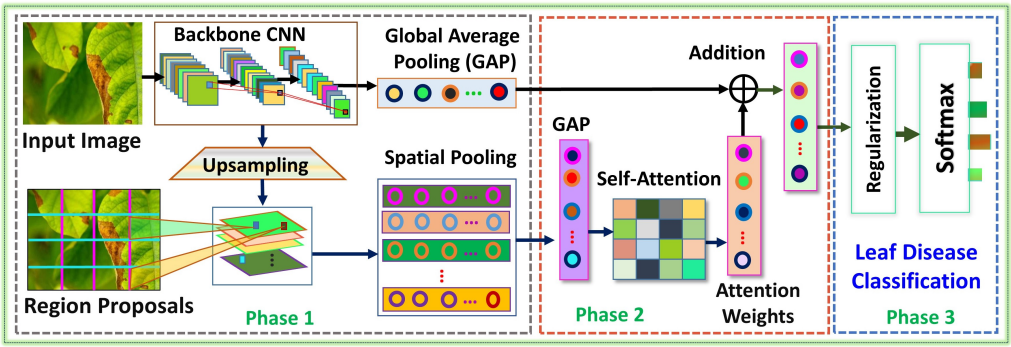

Fig. 1. The proposed APDC framework is divided into three phases: (1) Deep feature extraction from base CNNs and computing region proposals. (2) Attention-based weight computation for the candidate regions across the channel dimension with a residual connection. (3) Regularization of the learning task for plant disease classification using softmax activation.

region-based pooled feature maps are considered as the output of Phase 1. Then, intra-attention is computed for emphasizing the importance of various regions and assigning weights accordingly in Phase 2. The weighted attention score directs at accumulating a precise feature description relevant to classification. A residual path is added as a skip connection which is the output of a global average pooling layer applied to the base CNN's feature map. The added feature map combining the attention scores and skip path defines an efficient feature vector representing the output of Phase 2. A regularization technique is applied for handling the overfitting issues during the training of the proposed network, followed by a softmax layer for classification in the third phase. Experimentally, proposed APDC is found to be an effective and easy solution for leaf disease recognition.

The main contributions of this paper are summarized as:

- An attention-driven deep network integrating three key phases to emphasize the informativeness of complementary regions by weight assignment that represents an efficient feature vector for plant disease recognition.
- The proposed method is end-to-end trainable avoiding additional pre-processing module and bounding-box regression, implying a simple implementation.
- The proposed method has achieved state-of-the-art performance on four public datasets, representing visual and thermal leaf images of various plant classes.
- Rigorous experimental evaluation and ablation studies justify the importance of major components of the proposed deep network.

The rest of this paper is organized as follows: related works are summarized in Section 2. The proposed method is described in Section 3. The experimental results and ablation studies are discussed in Section 4. The conclusion is given in Section 5.

2. Related work

Various deep-learning techniques for plant disease detection and classification have been developed [10,27,38]. Common crop leaves such as the potato, rice, tomato, corn, wheat, etc., have been tested for solving disease identification [31]. A deep network consisting of object detector YoloX and siamese network is described for classifying rice diseases in RiceNet [33]. Multiple pest detection of orchard apples using improved faster R-CNN is presented [15]. A modified GoogLeNet is used for rice disease detection [50]. MobInc-Net is developed by combining MobileNet with the Inception module for disease recognition of 12 rice categories [12]. A dual-stream hierarchical bilinear pooling (DHBP) method is presented in [47]. Bacterial spot detection in the peach leaf images using Convolutional Autoencoders (CAE) and CNN is presented [4]. Six disease classes (e.g., anthracnose, etc.) of the maize crop is tested using NPNet-19 [31]. Pre-trained CNNs (e.g., Inception-v3, etc.) are used for transfer learning to detect 12 types of abnormalities, including huanglongbing of citrus [17].

A CNN is built with the Inception and residual architecture with a convolution block attention module (CBAM) is described in [56]. The method is tested on the epidemiological PlantVillage dataset [22], containing 54.3k images of 14 plant species. Fine-grained classification of infected tomato leaves of the PlantVillage dataset is tested [49]. A lightweight CNN for leaf disease identification is developed and tested on five datasets [45]. A multi-granular feature aggregation approach using self-attention is tested for crop disease classification [60]. A lightweight double fusion block with a coordinate attention network (DFCAnet) is developed [13]. A shuffle attention method and HardSwish function are introduced for recognizing tomato leaf diseases [52]. A cross-attention module, and bidirectional transposed feature pyramid Network is developed for apple disease detection [54]. A Multi-channel recurrent attention network is described for tomato leaf disease prediction [53]. The least important attention pruning algorithm selects the most important attention heads of multi-head self-attention module of each layer in the Transformer model for detecting Cassava leaf disease [43].

A convolutional vision transformer-based lightweight model (ConvViT) is proposed for apple leaf disease identification [26]. A Swin transformer is applied in the path aggregation Swin transformer network (PAST-Net) [48] for detecting and segmenting anthracnose-infected crops, e.g., apple, strawberry, pepper, etc. The Inception convolutional vision transformer (ViT) is developed [51]. The explainable ViT fuses vision transformers with CNN for plant disease identification [44]. A transformer-based with spatial convolutional self-attention transformer is developed for strawberry disease identification [25]. The GANs have been explored to enrich data diversity from small-scale various plant datasets [11]. GrapeGAN [23] follows a U-Net-like generator structure, and the discriminator is built with a convolution block and capsule structure. Four types of

grape leaf images are generated by GrapeGAN. Fine grained-GAN method presents a local spot area data augmentation for grape-leaf disease classification [57]. Double GAN is applied for producing high-quality leaf images, representing five classes of PlantVillage dataset [55]. MergeModel identifies tea-leaf diseases [19]. It has applied the U-Net for segmentation and SinGAN for augmentation.

Thermal imaging is explored for crop yield estimation, disease detection, and classification [34]. Thermal images were tested for disease detection from tomato, wheat, and other leaves [18, 58]. The deep explainable artificial intelligence (PlantDXAI) classified plant diseases using CNN-16 in thermal images [3]. The PlantDXAI could be improved by adopting the class activation map and discriminator network during the training. Blight disease detection in rice plants using thermal images is tested [9]. A fusion of color information with thermal and depth information, could attain better accuracy for detecting diseases [35]. In this work, we have presented an attention-driven deep architecture tested on color and thermal leaf images for disease classification.

3. Proposed method

A global feature descriptor could be extracted from an input image using a backbone CNN. Sometimes, a global descriptor might overlook underlying detailed information and and might summarize an overall feature representation, which is relevant to a general classification problem. In contrast, the detailed and subtle information is essential for categorization of leaf sub-categories. An aggregation of partial feature descriptors extracted from complementary regions could effectively capture finer details. We aim to integrate subtle informativeness of several disjoint regions into a comprehensive feature descriptor. The proposed APDC method combines contextual information from complementary regions by aggregating their overall weighted attention scores, which improves holistic feature representation capability. The proposed APDC method is conceptualized in Fig. 1; it is divided into three phases for easier understanding. The extraction of base feature map, and region proposals are described in Phase 1. The attention module with weight computation from pooled regions and is performed Phase 2. The classification is discussed in Phase 3.

3.1. Disjoint region proposal

A region proposal generation method avoiding object detectors, segmentation modules, or bounding box annotations is devised to capture contextual descriptions from different locations of an input image. Let an input leaf image, $I_y \in \mathbb{R}^{h \times w \times 3}$, is to be fed into a backbone CNN with its class label y representing a leaf category. A backbone CNN extracts deep features $\mathbf{F} \in \mathbb{R}^{h \times w \times c}$, where h, w, and c denote the height, width, and channels, respectively. The feature vector $\mathbf{F}$ represents high-level information of

input I_y. It could also be interpreted that a local region at low-level image representation is summarized within a small window of the high-level feature space $\mathbf{F}$. Thus, a correspondence between a local image-region with its feature map is necessary to correlate their significance holistically. We consider each uniform/regular region as a fixed rectangular dimension of $p \times p$ pixels. The window-size for spatial pooling from different uniform regions requires to be aligned because the spatial dimension of an I_y is squeezed to a lower size at the deeper levels through successive non-linear transformations in bottleneck layers of a standard CNN. Hence, $\mathbf{F}$ is upscaled to a higher spatial size $q \times q$ using a bilinear interpolation. The number of RoIs is $n = (q/p)^2$, generated without additional pixel-level adjustment during spatial pooling. The set of RoIs is denoted as $R = \{r_1, r_2, ..., r_n\}$, and feature map of r_i-th region is denoted as $\mathbf{F}_i$. The feature maps of all regions are $\mathbf{F}_R = \{\mathbf{F}_r\}_{r=1}^{r=n} \in \mathbb{R}^{n\times(h\times w\times c)}$. In addition to these key steps of Phase 1, a global average pooling (GAP) layer is added to optimize the output features of a base CNN across the channel dimension. A GAP layer squeezes the spatial dimension of a base CNN's output feature map. The pooled feature vector is $\mathbf{G}_R = \mathcal{GAP}(\mathbf{F}_R) \in \mathbb{R}^{n\times 1\times c}$ maintaining the same channel dimension of $\mathbf{F}$.

3.2. Attention mechanism

The visual attention mechanism focuses on the most informative region(s) of an input image to improve the learning efficacy of a deep architecture by contriving long-range dependency of partial descriptors. Here, self-attention is applied across the channel dimension of feature maps for all regions [2, 46]. The self-attention captures channel-wise relationships among various regions. It correlates cross-channel feature interactions and explores essential parts, accordingly. The self-attention uses three similar feature vectors to compute attention scores: the *query* $\mathbf{Q}$, *key* $\mathbf{K}$, and *value* $\mathbf{V}$ which are derived from the same feature vector $\mathbf{G}_R$. The attention matrix is considered as a dot product of $\mathbf{Q}$ and $\mathbf{K}$, multiplied by $\mathbf{V}$ to produce a weighted feature vector. Here, intra-attention is applied to the r_n region and its neighbor r_m region such that $n \neq m$. The attention method generates feature vector $\mathbf{V}$ to focus on discriminative regions in I_y. The vectors $\mathbf{G}_n$ and $\mathbf{G}_m$ are computed from the r_n and r_m regions, respectively. The feature map is defined as

$$\phi_{n,m} = \tanh(\mathbf{W}_\phi \mathbf{G}_n + \mathbf{W}_{\phi'} \mathbf{G}_m + \mathbf{b}_\phi), \quad (1)$$

$$\theta_{n,m} = \sigma(\mathbf{W}_\theta \phi_{n,m} + \mathbf{b}_\theta), \quad (2)$$

where weight matrices $\mathbf{W}_\phi$ and $\mathbf{W}_{\phi'}$ compute attention vectors of r_n and r_m, respectively; and $\mathbf{W}_\theta$ is their nonlinear combination. The bias vectors are $\mathbf{b}_\phi$ and $\mathbf{b}_\theta$, and $\sigma(\cdot)$ is an element-wise activation function. The importance of each r_n is computed next using a weighted sum of the attention scores generated from all regions in R. The

attention matrix $\hat{\mathbf{G}}_n$ indicates the importance to be given to a region conditioned on its neighborhood regions.

$$\beta_{n,m} = \text{softmax}(\mathbf{W}_\beta \theta_{n,m} + \mathbf{b}_\beta)\,, \hat{\mathbf{G}}_n = \sum_{m=1}^{n} \beta_{n,m} \mathbf{G}_m\,, \tag{3}$$

where the weight matrix is $\mathbf{W}_\beta$, and b_β is the bias. Next, the feature map $\hat{\mathbf{G}}_n$ is undergone to produce a weighted attention map γ_m using a softmax activation over all regions. The output vector of attention importance scores is considered as attention weights representing a high level encoding of all regions and is denoted as $\mathbf{G}_A$. This overall attention map interprets underlying explanation of a given region by weighting its importance towards decision making, essential for plant disease recognition.

$$\mathbf{G}_A = \sum_{m=1}^{n} \gamma_m \hat{\mathbf{G}}_m\,, \quad \gamma_m = \text{softmax}(\mathbf{W}_\phi \hat{\mathbf{G}}_m + \mathbf{b}_\gamma)\,. \tag{4}$$

A residual path is connected by including a GAP layer to the feature maps of a base CNN. This residual path supports further refinement of attentional weighted feature description by improving the gradient flow from the output layers to early layers during the learning without any additional computational overhead. The GAP layer inherently selects the mean features by scaling down a high dimensional feature map precisely to $(1 \times 1 \times c)$, obtained from a base CNN by neglecting trivial information. Also, the GAP enriches the confidence scores for classification, and is robust to spatial translation. The rendered feature map is denoted as $\mathbf{H} = \mathcal{GAP}\,(\mathbf{F}) \in \mathbb{R}^{1\times1\times c}$ where the feature mapping is $\mathbf{F} \rightarrow \mathbf{F}_{\text{gap}} : \mathbb{R}^{(1\times1\times c)}$. Both $\mathbf{G}_A$ and $\mathbf{H}$ feature vectors are added to represent the final attentional feature vector $\mathbf{F}_A \in \mathbb{R}^{(1\times c)}$.

$$\mathbf{F}_A = \text{Addition}\Big(\mathbf{G}_A, \mathbf{H}\Big)\ ;\ \mathbf{Y}_{\text{pred}} = \text{Softmax}(\mathbf{F}_A)\,. \tag{5}$$

3.3. Image classification

The dropout and batch normalization layers act as regularizers to ease overfitting issues, stabilizes and accelerate the speed of training. Thus, these two layers effective for enhancing the performance during the training. The final feature vector $\mathbf{F}_A$ is passed through a softmax layer to compute the output probability vector representing each class of leaf sub-categories. The categorical cross-entropy loss $\mathcal{L}_{\text{ce}}(Y_{\text{true}}, Y_{\text{pred}})$ optimizes the errors between the true class label (Y_{true}) and predicted class label (Y_{pred}). Overall, the attention technique strengthens the distinctness of feature vectors by capturing finer details without adhering to computational complexities, which is essentially required for leaf disease classification in the proposed APDC method.

Fig. 2. Samples of leaf images of the PlantPathology-22 dataset.

3.4. Model implementation

The standard backbone CNNs are used for deep feature extraction in Phase 1 of the proposed APDC. The input image-size of 224×224 is fed into a deep CNN, e.g., MobileNet-v2 [39], NASNetMobile [59], DenseNet-169 [20], Inception-v3 [42], etc. During the image pre-processing stage, data augmentations of random rotation (±30 degrees), scaling (1±0.30), and random region erasing (within 0.2-0.7 scale) with a fixed RGB value $q = 127$, are applied for data diversity. Though the output feature dimension of various base CNNs are different, the feature maps are rescaled to a higher resolution using a bilinear interpolation for uniform spatial pooling in Phase 1. For example, a feature map of size 7×7 is upsampled to 40×40 and then the features of non-overlapping regions with a fixed size are computed. Three different sets comprised a total of 16 (4×4), 25 (5×5), and 36 (6×6) regions are generated for experiments. The upscaled resolution is 42×42 for 36 regions, and 40×40 for 25 and 16 regions. The purpose of using such resolutions is to maintain proper pixel alignment during spatial pooling with a fixed window size. However, no feature dimension is calibrated in Phase 2. The output dimension of attention and GAP layers are the same as the output channel dimension of a base CNN, e.g., $c = 1280$ for MobileNet-v2. Finally, a batch normalization and a dropout rate of 0.2 are applied for stabilization of input distributions and regularization for improving the training capacity prior to a softmax layer in Phase 3. Our model is trained with pre-trained ImageNet weights for initializing a base CNN, as well as trained from scratch, i.e., random initialization in different experiments to observe performance variation due to weight initialization not altering other parameters.

The Stochastic Gradient Descent (SGD) is used to optimize the categorical cross-entropy loss function ($\mathcal{L}_{\text{ce}}$) with an initial learning rate of 1×10^{-3}, and multiplied by 0.1 after every 75 epochs for smoother convergence of the learning parameters θ. The proposed APDC is trained for 200 epochs with a mini-batch size of 8 using a Tesla M10 GPU of 8 GB. The top-1 accuracy [%] is used for performance evaluation. The model parameter is estimated in million (M).

Fig. 3. Diseased leaves of the PaddyDoctor-13 dataset representing infected leaves of plants and crops collected in a natural field environment.

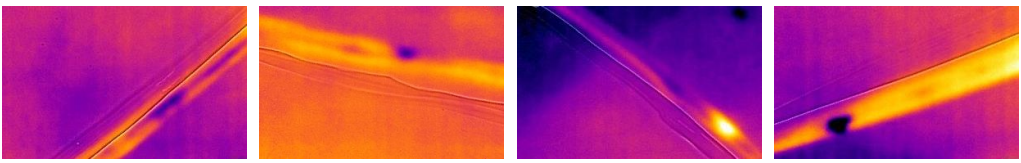

Fig. 4. Examples of diseased leaf images of PaddyCrop-6 thermal dataset.

4. Experimental results and discussions

First, a summary of various datasets tested in this work is briefed. Next, the experimental results, ablation studies, and visualizations are analysed.

4.1. Dataset description

One of the major challenges in agricultural disease diagnosis is the availability of a large realistic dataset of various crops and plants. Since the inception of the PlantVillage dataset, the largest crop dataset to date (to the best of our knowledge), several approaches have been tested for disease recognition and classification. However, this epidemiological dataset is curated in a controlled environment (Fig. 5) and not presented in a realistic manner (e.g., does not consider natural background, leaves are independent and isolated), which is considered as a restriction of this dataset while dealing with a real-world scenarios in agricultural fields. To alleviate this limitation, several other datasets representing various plants/crops are constructed (e.g., Fig. 3). However, most of these recent plant datasets are small-scale, which is further increased in size and image quality by leveraging GAN-based and other augmentations.

A summary of the datasets used in our study is listed in Table 1. Examples of diseased leaves from different datasets, namely PlantVillage-25 [22] (Fig. 5), PlantPathology-22 [14] (Fig. 2), PaddyDoctor-13 [24] (Fig. 3), and PaddyCrop-6 [3] (Fig. 4) are illustrated. The image examples imply that the PlantVillage and PlantPathology datasets are formulated in a simple and clear background condition. On the contrary, PlantDoc and PaddyDoctor represent realistic field environments and complex backgrounds.

Fig. 5. Samples of diseased leaf images of the PlantVillage dataset.

Tab. 1. Summary of the datasets tested in this work.

Dataset Name	Train	Test	Class	Type
PlantVillage-25	24240	16053	25	RGB
PlantPathology-22	2695	1809	22	RGB
PaddyDoctor-13	12980	3245	13	RGB
PaddyCrop-6	397	240	6	Thermal

The PlantPathology-22 dataset represents healthy (2278) and diseased (2225) leaves from 12 different plants, containing a total of 4503 images and categorised into 22 fine-grained classes.

The thermal images of diseased and healthy leaves of paddy crops comprising a total of 636 samples representing 6 classes were collected using a high-resolution FLIR E8 Thermal camera. Details of this dataset are given in PlantDXAI [3].

4.2. Performance analysis

Firstly, the baseline performances on each dataset are evaluated using four base CNNs. Next, the performances of our method using 16 (4×4), 25 (5×5), and 36 (6×6) RoIs are evaluated in different sets of experiments. The results are given in Table 2. The results imply that the accuracy could be improved with a more number of regions. Because, the attention mechanism focuses on the most important regions of leaf images and emphasizes their inter-channel interactions for weighted feature aggregation. The attention scheme enhances overall prediction performances using four base CNNs. The experimental results, given in Table 2, are achieved by training with ImageNet weights for a fair comparison with existing works on diverse datasets. The model parameters (last column, Table 2) of various experiments remain almost the same for different RoIs and differ according to the backbone CNNs.

Next, the performances on these datasets are evaluated by training the networks from scratch, *i.e.*, initializing the APDC with random weights, and the results are reported

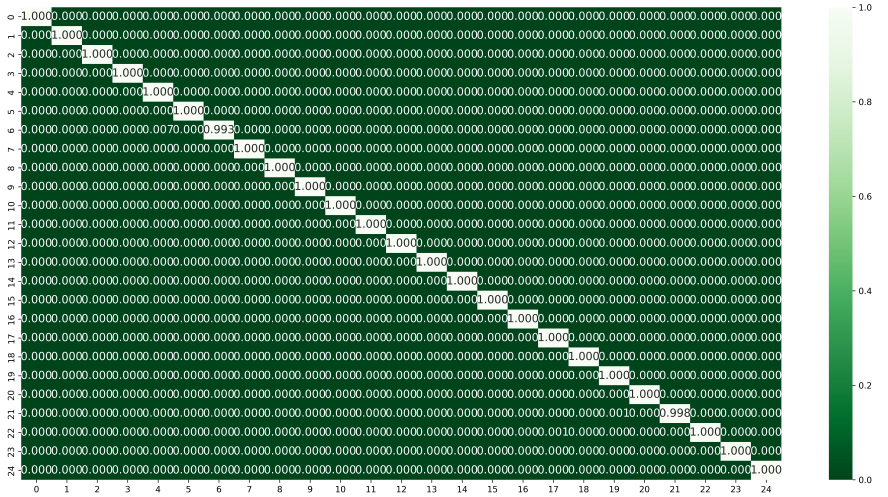

Fig. 6. Confusion Matrix of APDC (36 RoIs) on the PlantVillage-25 dataset.

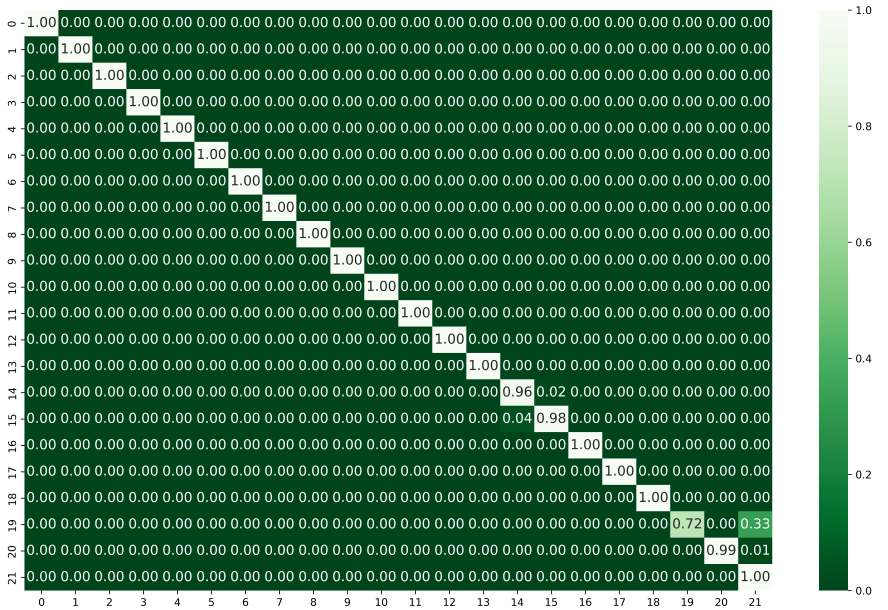

Fig. 7. Confusion Matrix of APDC (36 RoIs) using DenseNet169 on PlantPathology (left) dataset.

Tab. 2. Top-1 accuracy [%] of the proposed APDC using various CNNs backbones trained with ImageNet weights on five plant datasets. The accuracy of similar experiments attained by the CNNs trained from scratch is given in parenthesis. **Bold font** indicates the best performance(s) for each dataset.

Method	PlantVill	PlantPath	Pad'Crop	Pad'Doc	Par
Mob'Net	97.98 (97.69)	94.96 (90.76)	96.66 (83.33)	98.24 (95.82)	2.3
16 RoI	99.32 (98.43)	97.34 (94.79)	97.91 (94.16)	99.02 (96.63)	2.4
25 RoI	99.58 (99.61)	97.45 (95.52)	98.75 (95.41)	99.47 (98.20)	2.4
36 RoI	**99.97** (99.90)	97.62 (97.12)	99.16 (98.25)	99.62 (98.85)	2.4
NasNet	98.49 (95.51)	95.13 (93.58)	95.00 (86.25)	98.14 (95.30)	4.3
16 RoI	99.73 (98.34)	97.46 (96.23)	97.50 (94.58)	99.04 (98.70)	4.4
25 RoI	99.85 (98.76)	97.61 (97.10)	98.33 (95.00)	99.25 (99.21)	4.4
36 RoI	99.93 (99.85)	97.65 (97.24)	99.52 (95.82)	99.60 (99.40)	4.4
DenseNet	99.31 (97.92)	96.73 (92.80)	95.83 (87.50)	98.40 (96.62)	12.7
16 RoI	99.52 (98.55)	97.56 (95.52)	99.16 (93.75)	99.26 (98.45)	12.9
25 RoI	99.67 (98.67)	97.61 (96.72)	99.50 (96.21)	99.58 (99.02)	12.9
36 RoI	99.94 (99.89)	**97.74** (**97.32**)	99.58 (**98.52**)	**99.65** (**99.43**)	12.9
Inception	99.37 (97.65)	96.23 (92.53)	97.00 (86.23)	98.00 (96.72)	21.9
16 RoI	99.91 (98.55)	97.51 (96.23)	98.75 (95.30)	99.41 (98.71)	22.0
25 RoI	99.92 (98.98)	97.60 (97.12)	99.28 (95.81)	99.56 (99.32)	22.1
36 RoI	**99.97** (**99.94**)	97.64 (97.21)	**99.62** (97.50)	99.63 (99.41)	22.1

Tab. 3. Performance Summary of APDC (36 RoI) using various metrics [%].

Dataset	Base CNN	Top-1	Top-5	Precision	Recall	F1-score
PlantPathology	DenseNet169	97.74	99.94	98.0	98.0	98.0
PaddyCrop	MobileNetV2	99.16	100.0	99.0	99.0	99.0
PaddyDoctor	MobileNetV2	99.62	99.97	100.0	100.0	100.0
PlantVillage	MobileNetV2	99.97	100.0	100.0	100.0	100.0

within parenthesis in Table 2. It signifies a clear distinction between the accuracy of APDC while trained with ImageNet weight vis-à-vis random weight initialization which requires more epochs to attain similar accuracy compared to the former. Our model is trained for 300 epochs from scratch in this test, while other hyper-parameters were unaltered. Whereas, 200 epochs are sufficient to attain satisfactory results with the ImageNet weights, which converged quickly. The influence of pre-trained ImageNet weights, compared to random weights, for plant disease prediction accuracy is notable. This accuracy gaps are small on the PlantVillage, and PlantPathology datasets. A reason could be the nature and characteristics of datasets. The samples of these two datasets (Fig. 5-2) were collected in a controlled manner with limited variations by following

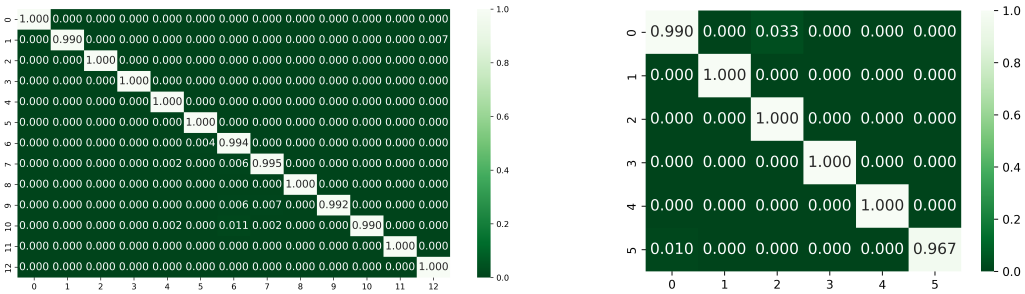

Fig. 8. Confusion Matrix of APDC (36 RoIs) using MoblieNetv2 on the PaddyDoctor (left) and PaddyCrop (right) datasets.

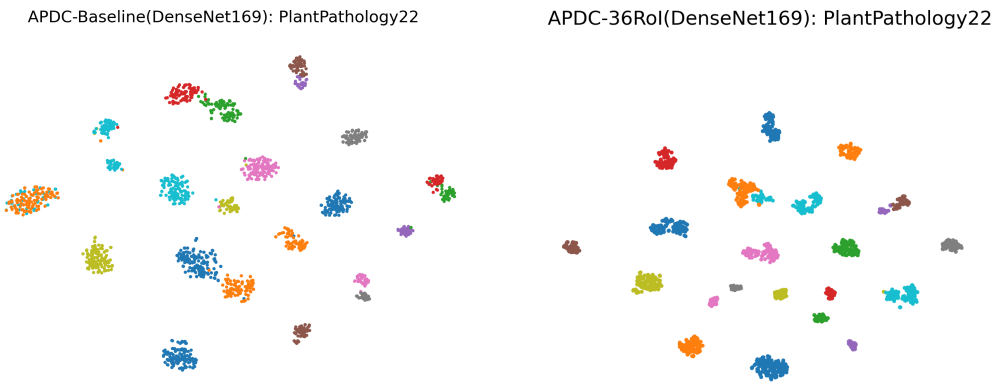

Fig. 9. t-SNE plots of baseline (left) and APDC (36 RoI) using DenseNet-169 (ImageNet) on the PlantPathology dataset.

simple image acquisition scenarios. A summary of the best performances (%) of APDC with 36 RoIs and ImageNet weights on five datasets using standard metrics, namely the top-1 accuracy, top-5 accuracy, precision, recall, and F1-score, are evaluated and reported in Table 3.

Also, one confusion matrix per dataset is shown in Fig. 6-8 for better clarity. In this assessment, MobileNetv2 (MN) is considered for the PlantVillage (Fig. 6), PaddyCrop, and PaddyDoctor datasets (Fig. 8). Whereas, DenseNet169 (DN) is used for generating the confusion matrices on the PlantPathology dataset (Fig. 7) for fair comparison.

Tab. 4. Performance comparison with SOTA on the PlantVillage dataset

Method [Ref]	Plant's Disease / #Class	Dataset Size	Accuracy [%]
GrapeGAN [23]	Grape leaf	4.1 K	96.13
Fine-grained-GAN [57]	Grape leaf-spot disease	1.5 K	96.27
ConvViT [26]	Apple disease	15.8 K	96.85
DenseNet-169 [1]	Corn Foliar disease, 4 cls.	9.1 K	99.50
PCA DeepNet [36]	Tomato, 10 classes	18.1 K	99.60
Double-GAN [55]	10 disease, 5 classes	31.3 K	99.70
PDD271 [27]	38 classes	50.3 K	99.78
FPDR (ResNet50) [16]	38 classes	50.3 K	99.84
APDC: MobileNet-v2	25 classes	40.3 K	**99.97**
DenseNet-169			**99.94**

4.3. Performance comparison

According to our study, many SOTA methods have achieved more than 99.50% accuracy on the PlantVillage dataset [27], and a few recent of them are listed in Table 4 for comparative study. The dataset was created in a controlled laboratory setup with a clear background. Hence, several deep-learning models achieved 99.50% accuracy. The gains in different successive works are competitively very small, e.g., 0.1% only between [36] and [55]. In this work, the average accuracy achieved by APDC with 36 RoIs is 99.95% with a standard deviation of ±0.02, considering four base CNNs trained with ImageNet weights (Table 2). The results on PlantVillage are computed with 25 classes of leaf categories. A brief description of existing disease prediction approaches and their accuracies are summarized in Table 4. The APCD (99.95%) has attained a competitive gain of 0.25% accuracy compared to Double-GAN (99.70%), whereas the accuracy gain over other methods is significant. The PCA DeepNet [36] reported 99.60% accuracy and 98.55% precision. Our APDC has gained 100% precision and F1-score (Table 3). In [27], 99.78% accuracy is obtained by ResNet-152, which is a heavier/deeper base model (≈60.4M) regarding the model parameters compared to lightweight backbones used here. The detailed results are given in Table 2. The IBSA-Net [52] has reported 99.40% accuracy, 98.90% precision, 99.30% and recall. Considering FPDR [16] as the previous best accuracy, 99.84% using ResNet-50 with ImageNet weights, the best 99.97% accuracy of APDC implies a 0.13% margin, with a lesser model parameters of MobileNet-v2. Nevertheless, to analyze the efficiency of our model, the gains on other datasets are significant. We have achieved SOTA performances on recently published public datasets. Rigours experiments have been conducted on the PlantPathology, PaddyDoctor, and PaddyCrop datasets. A fused multi-stream fusion (fsn) with learnable filters scheme [37] has reported 90.02% accuracy on the PlantPathology, curated with a clear background

Tab. 5. Comparison with SOTA on the PlantPathology, PaddyDoctor, and PaddyCrop Datasets [%]. The bottom row-set provides the accuracy of APDC with 36 RoIs using different base CNNs.

PlantPath'y	Acc	PaddyDoc	Acc	PaddyCrop	Acc
Multi-strm fsn [37]	90.02	MobileNet [24]	92.42	CNN16 [3]	88.63
DenseNet201 fsn [21]	96.14	ResNet34 [24]	97.50	PlantDXAI [3]	90.04
MobileNetv2	**97.62**	MobileNetv2	**99.62**	MobileNetv2	**99.16**
DenseNet169	**97.74**	DenseNet169	**99.65**	DenseNet169	**99.58**

like the PlantVillage. An ensemble of CNNs and statistical descriptors has reported an improved classification accuracy of 96.14% using DenseNet-201 [21]. Contrarily, our method has achieved at least 97.62% accuracy using MobileNet-v2 with 36 RoIs. The highest 97.74% accuracy is attained by DenseNet-169. The results are given in Table 5.

PaddyDoctor is a recent dataset on which transfer learning were tested [24]. The best 97.50% accuracy is achieved by ResNet-34, and MobileNet has attained 92.42% accuracy by training with ImageNet weights. The accuracy of APDC underlying on MobileNet-v2 (ImageNet weights) is 99.62%, and training from scratch achieves 98.85% accuracy. Also, APDC based on other CNNs has obtained SOTA results on PaddyDoctor (Table 5) irrespective of training scheme.

The PaddyCrop is a very small dataset containing thermal leaf images of infected rice crops. The PlantDXAI [3] is built with a CNN-16 and trained with class activation map and discriminator network. It has attained 90.04% accuracy on PaddyCrop. The accuracy achieved by our method underlying on DenseNet-169 is 99.58%, and Inception-v3 is 99.62%. Also, more than 99% accuracy is gained by APDC with 36 RoIs, while trained with ImageNet weights. The comparative results are given in Table 5. Overall result analysis evinces that our method outperforms existing works and achieves SOTA performances.

4.4. Ablation study

The necessity of major components of APDC is evaluated through two types of experiments. Firstly, various sets of regions avoiding the attention module are tested to understand their usefulness on different datasets using MobileNet-v2, NASNetMobile, and DenseNet-169 backbones. The results are given in Table 6. The contributions of various RoIs sets are notable using MobileNet-v2. However, in a few other cases, differences between the accuracies of 25 and 36 RoIs using various CNNs are small, e.g., PlantPathology. A reason could be the characteristics of dataset formulation which considered a simple background, such as the PlantPathology (Fig. 2). As a result, a

Tab. 6. Ablation Study I: Top-1 accuracy [%] in proposed APDC (ImageNet Weights) with RoIs Only, excluding attention mechanism.

Base CNN	RoIs	PlantPathology	PaddyCrop	PaddyDoctor
MobileNet-v2	16	96.18	94.58	98.85
	25	96.40	96.66	98.98
	36	96.79	98.75	99.10
NASNetMobile	16	95.06	95.46	98.80
	25	96.02	96.24	99.12
	36	97.01	96.66	99.44
DenseNet-169	16	96.90	98.33	99.16
	25	97.21	99.16	99.44
	36	97.34	99.50	99.56
Inception-v3	16	96.84	98.35	99.19
	25	97.06	99.16	99.41
	36	97.23	99.58	99.63

Tab. 7. Ablation Study II: Top-1 accuracy [%] of using attention on lightweight CNNs (random weight initialization) outputs, neglecting RoIs.

Base CNN	PlantPathology	PaddyyCrop	PddyDoctor
MobileNet	95.56	88.75	96.46
NASNet	95.44	87.25	96.23

few smaller regions may represent trivial information which directs the network to focus on central part of an image where crucial information about an infected leaf exists, neglecting other regions as insignificant.

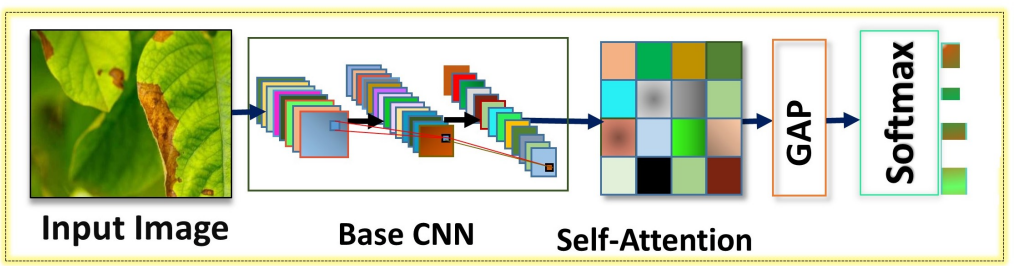

Fig. 10. A generalized CNN-based attention model excluding the regions from proposed APDC.

Next, lightweight MobileNet-v2 and NASNetMobile backbones are considered only and trained with random weight initialization. Here, the candidate regions are neglected from full model, and only intra-attention is applied to the base CNN's output features, followed by a GAP layer before a softmax layer. The deep network is shown in Fig. 10. The base CNN could be replaced by other backbones, e.g., ResNet, DenseNet, and other CNN families. The results are given in Table 7. In this test, the model parameters are reduced slightly, which also causes performance degradation in various datasets. The parameters of considering 36 RoIs for MobileNet-v2 based implementation are 2.46 M. Whereas, excluding the regions, 2.34 M parameters are required using the same MobileNet-v2. Similarly, the parameters for NASNetMobile based implementation are 4.34 M. These results (Table 7) are competitive on various datasets. This study justifies that complementary RoIs are effective to accomplish SOTA results on diverse plant datasets.

5. Conclusion

This paper proposes a deep architecture utilizing a visual attention mechanism, called APDC, for plant disease classification from visual/thermal images of leaves. Experiments were carried out using four plant datasets representing a wider variations in the plant categories, and background conditions. The proposed method follows an end-to-end trainable deep network and simple implementation using class labels only. It avoids extra pre-processing stage or sub-network for data pre-processing compared to existing techniques. The proposed APDC has achieved SOTA performances and emphasized lightweight CNN implementation balancing the accuracy with lower model parameters, unlike the existing ensemble of multiple CNNs-oriented techniques which are heavier models. The lightweight implementation of APDC requires lesser than 5M parameters only. We plan to develop a realistic approach for experimenting on larger and real-world datasets for plant disease classification in the future. A fusion with other sensory information such as soil data of agricultural fields will be another pertinent research direction for sustainable agricultural growth.

Acknowledgement

This work was supported by the New Faculty Seed Grant (NFSG/23-24) and necessary computational infrastructure at the Birla Institute of Technology and Science (BITS) Pilani, Pilani Campus, Rajasthan, 333031, India. This work was also supported in part by the project (2024/2204), Grant Agency of Excellence, University of Hradec Králové, Faculty of Informatics and Management, Czech Republic, and in part by Long-Term Conceptual Development of Research Organization (2024) at Škoda Auto University, Czech Republic.

References

[1] A. Ahmad, A. El Gamal, and D. Saraswat. Towards generalization of deep learning-based plant disease identification under controlled and field conditions. *IEEE Access*, 11, 2023. doi:10.1109/ACCESS.2023.3240100.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In: *Proc. 3rd International Conference on Learning Representations (ICLR)*. San Diego, CA, USA, 7-9 May 2015. doi:10.48550/arXiv.1409.0473.

[3] G. Batchuluun, S. H. Nam, and K. R. Park. Deep learning-based plant classification and crop disease classification by thermal camera. *Journal of King Saud University – Computer and Information Sciences*, 34(10):10474–10486, 2022. doi:10.1016/j.jksuci.2022.11.003.

[4] P. Bedi and P. Gole. Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. *Artificial Intelligence in Agriculture*, 5:90–101, 2021. doi:10.1016/j.aiia.2021.05.002.

[5] A. Bera, D. Bhattacharjee, and O. Krejcar. PND-Net: plant nutrition deficiency and disease classification using graph convolutional network. *Scientific Reports*, 14(1):15537, 2024. doi:10.1038/s41598-024-66543-7.

[6] A. Bera, O. Krejcar, and D. Bhattacharjee. Rafa-net: Region attention network for food items and agricultural stress recognition. *IEEE Transactions on AgriFood Electronics*, pp. 1–13, 2024. Early Access. doi:10.1109/TAFE.2024.3466561.

[7] A. Bera, M. Nasipuri, O. Krejcar, and D. Bhattacharjee. Fine-grained sports, yoga, and dance postures recognition: A benchmark analysis. *IEEE Transactions on Instrumentation and Measurement*, 72:5020613, 2023. doi:10.1109/TIM.2023.3293564.

[8] A. Bera, Z. Wharton, Y. Liu, N. Bessis, and A. Behera. SR-GNN: Spatial Relation-aware Graph Neural Network for fine-grained image categorization. *IEEE Transactions on Image Processing*, 31:6017–6031, 2022. doi:10.1109/TIP.2022.3205215.

[9] I. Bhakta, S. Phadikar, K. Majumder, H. Mukherjee, and A. Sau. A novel plant disease prediction model based on thermal images using modified deep convolutional neural network. *Precision Agriculture*, 24:23–39, 2022. doi:10.1007/s11119-022-09927-x.

[10] A. C. P. Calma, J. D. M. Guillermo, and C. C. Paglinawan. Cassava disease detection using MobileNetV3 algorithm through augmented stem and leaf images. In: *Proc. 17th Int. Conf. Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–6. IEEE, Seoul, Republic of Korea, 3-5 Jan 2023. doi:10.1109/IMCOM56909.2023.10035648.

[11] Q. H. Cap, H. Uga, S. Kagiwada, and H. Iyatomi. LeafGAN: An effective data augmentation method for practical plant disease diagnosis. *IEEE Transactions on Automation Science and Engineering*, 19(2):1258–1267, 2020. doi:10.1109/TASE.2020.3041499.

[12] J. Chen, W. Chen, A. Zeb, S. Yang, and D. Zhang. Lightweight inception networks for the recognition and detection of rice plant diseases. *IEEE Sensors Journal*, 22(14):14628–14638, 2022. doi:10.1109/JSEN.2022.3182304.

[13] Y. Chen, X. Chen, J. Lin, R. Pan, T. Cao, et al. DFCANet: A novel lightweight convolutional neural network model for corn disease identification. *Agriculture*, 12(12):2047, 2022. doi:10.3390/agriculture12122047.

[14] S. S. Chouhan, U. P. Singh, A. Kaul, and S. Jain. A data repository of leaf images: Practice towards plant conservation with plant pathology. In: *Proc. 4th Int. Conf. Information Systems and Computer Networks*, pp. 700–707. IEEE, Mathura, India, 21-22 Nov 2019. doi:10.1109/ISCON47742.2019.9036158.

[15] F. Deng, W. Mao, Z. Zeng, H. Zeng, and B. Wei. Multiple diseases and pests detection based on federated learning and improved faster R-CNN. *IEEE Transactions on Instrumentation and Measurement*, 71:3523811, 2022. doi:10.1109/TIM.2022.3201937.

[16] P. Gui, W. Dang, F. Zhu, and Q. Zhao. Towards automatic field plant disease recognition. *Computers and Electronics in Agriculture*, 191:106523, 2021. doi:10.1016/j.compag.2021.106523.

[17] W. Gómez-Flores, J. J. Garza-Saldaña, and S. E. Varela-Fuentes. A huanglongbing detection method for orange trees based on deep neural networks and transfer learning. *IEEE Access*, 10:116686–116696, 2022. doi:10.1109/ACCESS.2022.3219481.

[18] I. C. Hashim, A. R. M. Shariff, S. K. Bejo, F. M. Muharam, K. Ahmad, et al. Application of thermal imaging for plant disease detection. In: *Proc. 10th IGRSM Int. Conf. and Exhibition on Geospatial & Remote Sensing*, vol. 540 of *IOP Conference Series: Earth and Environmental Science*, p. 012052. IOP Publishing, Kuala Lumpur, Malaysia, 20-21 Oct 2020. doi:10.1088/1755-1315/540/1/012052.

[19] G. Hu and M. Fang. Using a multi-convolutional neural network to automatically identify small-sample tea leaf diseases. *Sustainable Computing: Informatics and Systems*, 35:100696, 2022. doi:10.1016/j.suscom.2022.100696.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. Honolulu, HI, USA, 21-26 Jul 2017. doi:10.1109/CVPR.2017.243.

[21] J. Huertas-Tato, A. Martín, J. Fierrez, and D. Camacho. Fusing CNNs and statistical indicators to improve image classification. *Information Fusion*, 79:174–187, 2022. doi:10.1016/j.inffus.2021.09.012.

[22] D. Hughes, M. Salathé, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv*, 2016. ArXiv:1511.08060v2. doi:10.48550/arXiv.1511.08060.

[23] H. Jin, Y. Li, J. Qi, J. Feng, D. Tian, et al. GrapeGAN: Unsupervised image enhancement for improved grape leaf disease recognition. *Computers and Electronics in Agriculture*, 198:107055, 2022. doi:10.1016/j.compag.2022.107055.

[24] B. Kiruba and P. Arjunan. Paddy Doctor: A visual image dataset for automated paddy disease classification and benchmarking. In: *Proc. 6th Joint Int. Conf. Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, pp. 203–207. Mumbai, India, 4-7 Jan 2023. doi:10.1145/3570991.3570994.

[25] G. Li, L. Jiao, P. Chen, K. Liu, R. Wang, et al. Spatial convolutional self-attention-based transformer module for strawberry disease identification under complex background. *Computers and Electronics in Agriculture*, 212:108121, 2023. doi:10.1016/j.compag.2023.108121.

[26] X. Li and S. Li. Transformer help CNN see better: A lightweight hybrid apple disease identification model based on transformers. *Agriculture*, 12(6):884, 2022. doi:10.3390/agriculture12060884.

[27] J. Liu and X. Wang. Plant diseases and pests detection based on deep learning: A review. *Plant Methods*, 17(1):22, 2021. doi:10.1186/s13007-021-00722-9.

[28] Y. Liu, G. Gao, and Z. Zhang. Crop disease recognition based on modified light-weight CNN with attention mechanism. *IEEE Access*, 10:112066–112075, 2022. doi:10.1109/ACCESS.2022.3216285.

[29] J. Lu, L. Tan, and H. Jiang. Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture*, 11(8):707, 2021. doi:10.3390/agriculture11080707.

[30] O. Mzoughi and I. Yahiaoui. Deep learning-based segmentation for disease identification. *Ecological Informatics*, p. 102000, 2023. doi:10.1016/j.ecoinf.2023.102000.

[31] M. Nagaraju and P. Chawla. Maize crop disease detection using NPNet-19 convolutional neural network. *Neural Computing and Applications*, 22:3075–3099, 2022. doi:10.1007/s00521-022-07722-3.

[32] A. Pal and V. Kumar. AgriDet: Plant leaf disease severity classification using agriculture detection framework. *Engineering Applications of Artificial Intelligence*, 119:105754, 2023. doi:10.1016/j.engappai.2022.105754.

[33] J. Pan, T. Wang, and Q. Wu. RiceNet: A two stage machine learning method for rice disease identification. *Biosystems Engineering*, 225:54–68, 2023. doi:10.1016/j.biosystemseng.2022.11.007.

[34] M. Pineda, M. Barón, and M.-L. Pérez-Bueno. Thermal imaging for plant stress detection and phenotyping. *Remote Sensing*, 13(1):68, 2021. doi:10.3390/rs13010068.

[35] S.-e.-A. Raza, G. Prince, J. P. Clarkson, and N. M. Rajpoot. Automatic detection of diseased tomato plants using thermal and stereo visible light images. *PloS ONE*, 10(4):e0123262, 2015. doi:10.1371/journal.pone.0123262.

[36] K. Roy, S. S. Chaudhuri, J. Frnda, S. Bandopadhyay, I. J. Ray, et al. Detection of tomato leaf diseases for agro-based industries using novel PCA DeepNet. *IEEE Access*, 11:14983–15001, 2023. doi:10.1109/ACCESS.2023.3244499.

[37] N. S. Russel and A. Selvaraj. Leaf species and disease classification using multiscale parallel deep CNN architecture. *Neural Computing and Applications*, 34(21):19217–19237, 2022. doi:10.1007/s00521-022-07521-w.

[38] M. H. Saleem, J. Potgieter, and K. M. Arif. Plant disease detection and classification by deep learning. *Plants*, 8(11):468, 2019. doi:10.3390/plants8110468.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In: *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520. Salt Lake City, UT, USA, 18-23 Jun 2018. doi:10.1109/CVPR.2018.00474.

[40] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, et al. PlantDoc: A dataset for visual plant disease detection. In: *CoDS COMAD 2020: Proc. 7th ACM IKDD CoDS and 25th COMAD*, pp. 249–253. Hyderabad, India, 5-7 Jan 2020. doi:10.1145/3371158.3371196.

[41] C. K. Sunil, C. D. Jaidhar, and N. Patil. Cardamom plant disease detection approach using EfficientNetV2. *IEEE Access*, 10:789–804, 2021. doi:10.1109/ACCESS.2021.3138920.

[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826. Las Vegas, NV, USA, 27-30 Jun 2016. doi:10.1109/CVPR.2016.308.

[43] H.-T. Thai, K.-H. Le, and N. L.-T. Nguyen. Formerleaf: An efficient vision transformer for Cassava Leaf Disease detection. *Computers and Electronics in Agriculture*, 204:107518, 2023. doi:10.1016/j.compag.2022.107518.

[44] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha. Explainable vision transformer enabled convolutional neural network for plant disease identification: PlantXViT. *arXiv*, 2022. ArXiv:2207.07919. doi:10.48550/arXiv.2207.07919.

[45] P. S. Thakur, T. Sheorey, and A. Ojha. VGG-ICNN: A lightweight CNN model for crop disease identification. *Multimedia Tools and Applications*, 82(1):497–520, 2022. doi:10.1007/s11042-022-13144-z.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, et al. Attention is all you need. In: *Advances in Neural Information Processing Systems: Proc. NIPS 2017*, vol. 30. Curran Associates, Inc., 2017. https://papers.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[47] D. Wang, J. Wang, Z. Ren, and W. Li. DHBP: A dual-stream hierarchical bilinear pooling model for plant disease multi-task classification. *Computers and Electronics in Agriculture*, 195:106788, 2022. doi:10.1016/j.compag.2022.106788.

[48] Y. Wang, S. Wang, W. Ni, and Q. Zeng. PAST-net: a swin transformer and path aggregation model for anthracnose instance segmentation. *Multimedia Systems*, 29(3):1011–1023, 2022. doi:10.1007/s00530-022-01033-2.

[49] G. Yang, G. Chen, Y. He, Z. Yan, Y. Guo, et al. Self-supervised collaborative multi-network for fine-grained visual categorization of tomato diseases. *IEEE Access*, 8:211912–211923, 2020. doi:10.1109/ACCESS.2020.3039345.

[50] L. Yang, X. Yu, S. Zhang, H. Long, H. Zhang, et al. GoogLeNet based on residual network and attention mechanism identification of rice leaf diseases. *Computers and Electronics in Agriculture*, 204:107543, 2023. doi:10.1016/j.compag.2022.107543.

[51] S. Yu, L. Xie, and Q. Huang. Inception convolutional vision transformers for plant disease identification. *Internet of Things*, 21:100650, 2023. doi:10.1016/j.iot.2022.100650.

[52] R. Zhang, Y. Wang, P. Jiang, J. Peng, and H. Chen. IBSA_Net: A network for tomato leaf disease identification based on transfer learning with small samples. *Applied Sciences*, 13(7):4348, 2023. doi:10.3390/app13074348.

[53] Y. Zhang, S. Huang, G. Zhou, Y. Hu, and L. Li. Identification of tomato leaf diseases based on multi-channel automatic orientation recurrent attention network. *Computers and Electronics in Agriculture*, 205:107605, 2023. doi:10.1016/j.compag.2022.107605.

[54] Y. Zhang, G. Zhou, A. Chen, M. He, J. Li, et al. A precise apple leaf diseases detection using bctnet under unconstrained environments. *Computers and Electronics in Agriculture*, 212:108132, 2023. doi:10.1016/j.compag.2023.108132.

[55] Y. Zhao, Z. Chen, X. Gao, W. Song, Q. Xiong, et al. Plant disease detection using generated leaves based on DoubleGAN. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(3):1817–1826, 2021. doi:10.1109/TCBB.2021.3056683.

[56] Y. Zhao, C. Sun, X. Xu, and J. Chen. RIC-Net: A plant disease classification model based on the fusion of Inception and residual structure and embedded attention mechanism. *Computers and Electronics in Agriculture*, 193:106644, 2022. doi:10.1016/j.compag.2021.106644.

[57] C. Zhou, Z. Zhang, S. Zhou, J. Xing, Q. Wu, et al. Grape leaf spot identification under limited samples by fine-grained GAN. *IEEE Access*, 9:100480–100489, 2021. doi:10.1109/ACCESS.2021.3097050.

[58] W. Zhu, H. Chen, I. Ciechanowska, and D. Spaner. Application of infrared thermal imaging for the rapid diagnosis of crop disease. *IFAC-PapersOnLine*, 51(17):424–430, 2018. doi:10.1016/j.ifacol.2018.08.184.

[59] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In: *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 8697–8710. Salt Lake City, UT, USA, 18-23 Jun 2018. doi:10.1109/CVPR.2018.00907.

[60] X. Zuo, J. Chu, J. Shen, and J. Sun. Multi-granularity feature aggregation with self-attention and spatial reasoning for fine-grained crop disease classification. *Agriculture*, 12(9):1499, 2022. doi:10.3390/agriculture12091499.

Fracture Fusion: Revolutionizing the Recognition of Bone Fractures with MetaMag Efficiency Approach

S. Rajeashwari*, Dr. K. Arunesh
Department of Computer Science, Sri S. Ramasamy Naidu Memorial College, Affiliated to Madurai Kamaraj University, Sattur, India
**Corresponding author: S. Rajeashwari (rajeeragavan83@gmail.com)*

Abstract Bone fractures are common in diabetic patients and can result in several musculoskeletal conditions. Both type 1 and type 2 diabetes substantially increase the risk and severity of bone fractures. Prompt treatment and management of diabetes and its complications are crucial to mitigate this serious complication. Detection and diagnosis in its early stage can reduce the challenging conditions in treatment. Traditional image processing techniques like digital-geometric analysis, entropy measures, and gray-level co-occurrence matrices have been used for automated bone fracture detection. However, these detection methods rely neither on healthy controls nor diabetic-affected patients. Only few studies focused on detecting fractures in diabetic patients. The rising prevalence of diabetic ankle fractures made the study emphasize the development of a fracture detection model based on the Meta Magnify (MetaMag) efficiency model. The proposed model involves the Lower Extremity Radiographs (LERA) dataset, which consists of image samples of normal and abnormal lower extremities of the body, such as the hip, ankle, knee, and foot. Pre-processing involves a one-hot encoding method that handles the missing data and represents categorical variables as numerical values. Further, the classification is performed using the MetaMag efficiency model, incorporated with MetaMag scaling and unified normalization. Further, the efficiency of the proposed model is analyzed by comparing it with conventional EfficientNet and another model. Finally, the proposed work's performance is analyzed using evaluation measures such as accuracy, precision, recall and F1-score. The results indicate the improved efficiency of the model.

Keywords: fracture, Lower Extremity Radiographs dataset, diabetes, Deep Learning, radiograph images, EfficientNet.

1. Introduction

Among other parts of the body, the knee is considered the most complex joint that involves many daily activities. A high prevalence of knee injuries occurs due to twisting movements and sudden changes of direction [7]. This creates chances of knee damage and other risk factors leading to severe impact on the patient's lifestyle. Approximately one in eight patients has diabetes and undergoes treatment for rotational ankle fractures. With this, complications of ankle fracture fixation in patients with Diabetes Mellitus (DM), after surgery vary between 26% and 47% [20]. Several researchers have also identified that an ankle injury may trigger the process of Charcot neuroarthropathy. These higher complication rates can cause bone deformity, loss, and joint destruction. The most affected areas damaged due to an injury are the patellofemoral, ligaments,

cartilage, and meniscus. In addition, the data analysis has resulted that a large cohort of 58 748 patients who undergo ankle fracture fixation in New York discovered that 12.5% were diabetic, and 14.6% of patients resulted in complicated diabetes [10]. Moreover, the widely used methods involved in detecting lesions in the knee part are Magnetic Resonance Imaging (MRI) and X-ray copies [3]. The results produced by these methods are promising, but there is still a need to develop new equipment and research. So, in recent times, AI has emerged as the significant opinion of specialists that assist in providing non-invasive tools, and low-complexity and low-cost instruments [11]. These methods enable the system to extract the patterns from the input data and map the relationships among the input variables and outcomes. Thus, these new technologies tend to efficiently identify knee abnormalities and diagnosing methods at their early stages to avoid higher consequences of disease in patients. Although these techniques effectively detect and interpret fractures in DM patients, they lack high detection accuracy due to the irregularity and lucidity in the input sample images.

On the contrary, several studies investigated the prediction of knee fractures in diabetes patients by using Machine Learning (ML) and Deep Learning (DL) algorithms [1, 31]. Hence, the considered study [25] implements Convolutional Neural Network (CNN), to perform the detection of abnormality on lower extremity radiographs. The lower extremity includes the range of abnormalities in hip, knee, ankle, and foot radiographs. This study's larger dataset comprises almost 93 455 input samples of lower extremity radiographs of several body parts. These samples are labeled as normal and abnormal at the initial interpretation by the attending radiologist. The CNN is pre-trained with 161-layer densely connected to achieve improved accuracy in the process of classification. The performance of the study was analyzed by using three different models such as pre-trained ResNet-101, DenseNet-161 [30], and ResNet-50. Further, an extensive random hyperparameters search for each model is performed. The motive of the study is to provide increased accuracy in the classification tasks. This is done by augmenting the dataset by using MURA radiographs, this tends to optimize the efficacy of the model. From analysis, it is found that the DenseNet-161 produced better diagnostic accuracy. In the other aspects, the intimated study [2] applies the detection of Anterior Cruciate Ligament (ACL) using the DL model. The model involves the customized 14-layer ResNet-14 structure of CNN and six directions. This is done by involving real-time data augmentation and hybrid class balancing. Three classes are classified: ruptured tears, partial and healthy. Initially, the data pre-processing undergoes three steps and after the steps, the three classes are raised. The original version-I residual ResNet-18 in the classification model is modified into ResNet-14 network architecture. Here, the Batch Normalization (BN) is added after the CNN model and previous to the activation function Rectified Linear Unit (ReLu) [29]. The fine-tuned hyperparameters are being used that provide a huge impact on the effectiveness of the method. The outcomes of the study projected better outcomes in terms of accuracy, specificity, sensitivity, F1-score, precision, and

AUC. However, the techniques failed to produce improved classification accuracy and enhanced input sample images to perfectly interpret the affected region [19].

The present study aims to further optimize the automatic detection of fractures by using a DL model with radiographic images. The MetaMag efficiency model using MetaMag Scaling along with Unify Normalization is proposed in the present study, which tends to significantly increase the detection of fractures and classifies whether the input is fractured or non-fractured. Input from the LERA dataset is first passed into the pre-processing stage, where the one-hot encoding method is applied. This method endeavors to handle missing values and generates efficient features for classification. Then the pre-processed data are fed into the train-test phase, where the train data are used for pre-training the classifier. In the classification process, the MetaMag efficiency model undergoes MetaMag scaling that uniformly scales all the dimensions of resolution, width and depth for procuring improved performance. It systematically analyses the model scaling and identifies the balancing network using a simple yet highly effective compound coefficient. This work focuses on improving the practical efficiency of the traditional EfficientNet model by using the unified normalization that reduces the computational loss and inexpensively fine-tuning at higher resolution. It eventually increases the size of the image and aids in obtaining finer details of the input image. This helps the classifier distinguish the input images into two categories: normal as 0 and abnormal as 1. Thus, the efficiency of the proposed model is evaluated by using performance measures.

1.1. The main contributions of the study

- To efficiently classify the normal and abnormalities in the input LERA dataset, to detect fractures in the lower extremities of the human body.
- To implement MetaMag scaling and Unify Normalization approaches to precisely analyze the attributes and improve classification accuracy.
- To evaluate the model's efficacy by involving performance measures: accuracy, recall, precision, and F1-score.
- To compare the proposed MetaMag efficiency model with other conventional algorithms to project the effectiveness of the proposed system.
- To develop MetaMag efficiency model for improved bone fracture classification accuracy and efficiency, as well as plans to create automated systems to assist clinicians in diagnosis and treatment planning.

1.2. Organization of the paper

The remaining parts of this paper are organized as follows. Section 2 deliberates the review of conventional works with the problems identified by analysis of several studies. Section 3 expounds on the projected procedures with the proposed flow, algorithms, and their mathematical derivations. Subsequently, section 4 presents the results attained by

the proposed and conventional models. The overall study is concluded in Section 5 with future suggestions.

1.3. Motivation of the research

Patients with both type 1 and type 2 diabetes have a significantly increased risk of bone fractures compared to those without diabetes. Diabetes can impair bone quality and fracture healing, leading to a higher risk of complications like delayed union, non-union, or prosthetic joint formation. Identifying and managing bone fragility in diabetic patients is an emerging challenge that requires more attention, as current osteoporosis and diabetes guidelines do not adequately address this issue. Improving the understanding and management of bone health in diabetes is crucial to mitigate this serious complication. Hence, the proposed model utilises LERA for effective classification process.

2. Literature review

The analysis of various studies on fracture detection using different strategies and the methodologies and problem identification for specific studies are also deliberated.

The human knee joints are the main and complex joints present in the human body that maintain weight and offer flexible movements of the body. It bears the excess load and is thus highly prone to injuries. So, detecting knee injuries as early as possible is important to avoid complications and provide appropriate treatments.

The study [14] involved the prediction of Knee Osteoarthritis (KOA) using the ML-based approach. The study has applied a multidisciplinary Osteoarthritis Initiative (OAI) database collected through self-reported data on joint symptoms, physical activity indexes, disability and function, physical examination data, and questionnaire data. Initially, the data pre-processing has been done by implying data imputation to tackle the missing values. Then, the feature selection was done by integrating the output of six feature selection algorithms, three embedded techniques, one wrapper, and two filter algorithms. Whereas, the ML-based techniques like Logistic Regression (LR), k-Nearest Neighbor (KNN), Random Forest (RF), Naive Bayes (NB), Decision Tree (DT), XGBoost and SVM have been evaluated for validating their sustainability been utilized to solve the classification issues. The better accuracies produced by these models have been identified and found that the SVM model has performed better, producing an accuracy of 74.07%. Even though the model has been reliable, the predictive capacity has to be improved predominantly.

Knee abnormalities are mostly due to hard injury or osteoarthritis that greatly impact the patient's health. Generally, the MRI plays a vital part in detecting the biochemical and morphologic features that provide an in-depth understanding of patterns. So, the suggested study [23] has MRI-based studies to conduct the identification of lesion severity

in the ACL, meniscus, bone marrow, and cartilage. A three-dimensional CNN has been developed to identify the Region of Interest (ROI) and then grade the abnormalities. At first, the segmentation was performed by using two V-Net architectures under two consecutive steps. From analysis, it has been consolidated that the study has produced improved specificity, sensitivity, and multiclass lesion severity staging in several tissues of the knee. In addition, the generalizability of the model has to be improved, and the assessment of lateral and medial ligaments has to be considered. On the other hand, the intimated study [13] has relied on the detection of abnormalities and classification automatically using Musculo-Skeletal Disorders Network (MSDNet). These methods have been an ensemble of CNN that integrates the features of several CNN models to improve the performance of abnormality classification. A boundary detection algorithm has been developed to predict the ROI to facilitate enhanced detection of anomalies. The MSDNet is the combination of both AlexNet and ResNet18 structures. Firstly, the global features have been produced from the AlexNet by directly feeding the original input data, whereas the local features have been generated by the ResNet18 model. The overall accuracy produced by the MSDNet model is 82.69%. Among aged people, the main factor for fracture [6,24,29] is due to the reduction of bone density. A low-cost diagnostic technology is important in identifying osteoporosis in its initial stage. So, the suggested study [15] has analyzed osteoporosis using X-ray radiography to predict the essential components and categorize it into osteoporosis, osteopenia, and normal. The study has implemented three CNN architectures namely, ResNet18, Xception, and Inceptionv3 models. This ensemble method has implied a fuzzy rank-based fusion of classifiers by considering the two different factors. A fuzzy ranking-based approach has been applied, which has been exposed to two distinct non-linear processes. After implementation, the study's outcomes have shown that the study has produced a classification accuracy of 93.5%. The accuracy has been hindered due to overlapping cells or insufficient picture quality that made complexity in classifying the images effectively.

The advancements in radiological technologies have improved the treatment of various diseases. But, when compared with a huge number of fractural patients, the number of radiologists is insufficient. This makes radiologists astounded by the large amount of medical image data. Hence, the imitated study [12] has deployed a backbone network by applying dilated convolutions to detect the fractured thigh region. The DL method known as Dilated Convolutional Feature Pyramid Network (DCFPN) has been used, in which stage 1 has been adopted to extract the features from the original image. It has been insisted that the dilated convolutional kernel could gain more information from the extended receptive field. The FPN structure has been comprised of five feature maps. The Region Proposal Network (RPN) has been developed to generate the region proposal that shares the convolutional feature maps. Thus, the output is an image with a predicted bounding box. The radiologists have used the technique of Computer-Aided Diagnosis (CAD) to diagnose the fractures [8,22] on bones, which minimizes their

difficulties. Thus, the suggested study [16] has involved classification using a Crack Sensitive Convolutional Neural Network (Crack-Net) to identify the sensitive fracture lines on human bones. This paper clearly explains the two different stages of discovering the fracture [4, 21, 26]. Initially, Faster R-CNN, which is Faster Region with CNN, was deployed. This method has been performed to identify 20 types of bone regions and fractures [27] using Crack-Net in the collected X-ray copies. The results have shown that from the total of 1052 copies 526 copies are fractured copies.

Further, the study has produced an accuracy rate of 90.11% and an F-measure of 90.14% of the x-ray copies. In radiographs, the method of Guided Anchoring (GA) Faster R-CNN has been used to identify and locate the fractures in hand [28]. This GA method has resulted in improved, accurate, and effective anchor generation. It has eventually increased the network's performance and saved computing energy. In this system, the Feature Pyramid Network (FPN) method has been used to detect small fractures [5, 9] such as knuckles and fingertips joints and others. Additionally, the implementation of balanced loss (L1) has been applied to adapt imbalanced learning tasks. The result of this system has shown that among 3067 HF dataset X-ray copies, 2453 are training data and 614 are testing data. The accuracy of the dataset has been achieved to be 97%-99% with an Average Precision (AP) of 70.7%. This System has accomplished all the other conventional methods for identifying HF.

Problem identification

- The study has involved the detection of fractures using X-ray images. Though the system has produced a better detection rate, the classification accuracy can be prominently improved by applying different algorithms [16].
- The risk factors accompanied by knee osteoarthritis have been involved in the study using DL models. The accuracy produced by the study has been identified to be 74.07%. It can be further improved to support radiologists in finding the complexities [14].
- Binary classification of lower extremity fracture has been performed in the study and produced limitation of producing generalizability in detecting the abnormalities. Efficient methods can be applied to detect the fracture [25].

3. Proposed methodology

DM is a metabolic disorder that increases the chance of interfering with bone formation and fracture risk. This leads to the impairment of fracture healing and several other common features that affect the bone. DL techniques greatly impact the medical domain and lead to advancements in the detection of abnormalities that help in affording early diagnosis of diseases. There is still a lack of studies investigating the association between DM and fracture risk in patients. The possible solutions to fracture risk should

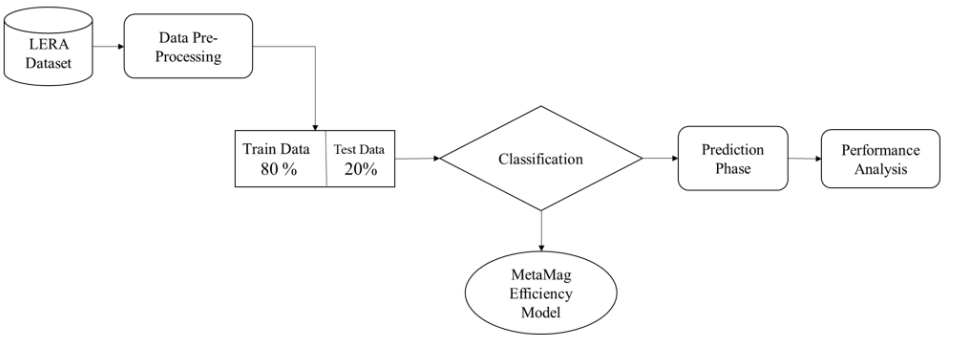

Fig. 1. Illustrative diagram of the overall methodology

be addressed at an early stage to avoid the severity of risk in patients with diabetes. Spontaneous calcaneal fractures without obvious trauma may occur in diabetic patients sometimes accompanied by DFU. With this intention, the initial phase concentrated on detecting the foot ulcer in DM patients. The study implemented a Deep Convolutional Neural Network (DCNN) based on the Xception model to classify healthy and DFU skin images. The DCNN-based Xception classifier was integrated with Residual Linearly Clamped Layers (RLCL) comprising minimum detached convolution layers. Further, the input images are optimized by using image enhancement techniques such as Histogram equalization, Adaptive filter, and Gamma correlation. Then, the efficiency of the proposed system is evaluated based on the performance measures, namely precision, F1-score, recall, and accuracy, to validate the performance of the proposed model with existing algorithms. Though the study has proclaimed improved efficiency. It is noteworthy that patients manifesting systematic signs of diabetic foot infection cause fractures or dislocations of the ankle or foot. With this regard, it is also significant to address the challenges faced by the diabetic patients with lower limb amputations. So, the present work focussed on detecting and classifying the normal and fractured bone classes by using the MetaMag efficiency model. This method tends to reduce the problems related to high-risk factors and efficiently contributes towards risk reduction and management. The overall process involved in the proposed technique is demonstrated in Figure 1.

The input from the LERA dataset (see Section 3.1) is first passed into the pre-processing stage, where the one-hot encoding process is applied. This method tends to handle the missing values and generates efficient features for classification. Then the pre-processed data are fed into the train-test phase, where the train data are used for pre-training the classifier. Further, the classification is performed by a MetaMag efficiency classifier that involves MetaMag scaling and a unified normalization process that supports enhancing the performance of the proposed method. The classifier classifies

Tab. 1. Class Distribution of LERA Dataset.

Samples	Hip	Foot	Ankle	Knee
Abnormal images	3	36	36	99
Normal images	91	12	285	435
Total images	94	348	321	534

the input images into two categories: normal as 0 and abnormal as 1. Wherein the prediction phase validates the classifier's efficiency by using test data and analyses by using performance measures.

Association of diabetes with fractures

- DM type 1 and type 2 affect several people worldwide and are characterized by hyperglycemia. The traditional impediments of DM are microvascular complications like neuropathy, nephropathy, and retinopathy. Whereas the macrovascular complications include CVD (Cardiovascular Disease). The researchers have also found that diabetes affects the bones of DM patients with increased chances of fracture due to impaired bone quality. Further, the fracture risk in diabetes patients can be described by possible cofounders, diabetes type, and fracture site.
- Type 1 DM is related to a modest reduction of bone mineral density. Type 2 DM increases the chance of affecting bone health in its advanced phases of disease. The biomechanical characteristics of bone and bone architecture are negatively impacted by chronic inflammation, Advanced Glycation End products (AGE), hyperglycemia, and insulinopenia.
- Several methods are used to evaluate bone quality in DM, including the diagnosis based on X-ray images, MRI images, Grayscale images, Red Green Blue (RGB) images, and radiography images.

3.1. Dataset description

The dataset used in the proposed method is LERA [17], which covers the broad range of joints and bone abnormalities of lower extremity areas of the human body. The dataset is considered a diverse-natured dataset due to its collection over a wide range of time, from 2003 to 2014. This LERA dataset comprises anomalous and standard image dissemination and sample images of hip, ankle, knee, and foot bones. This dataset has been accumulated by HIPAA complaint that compiled data from almost 182 patients who have undergone radiographic examination at Standard University Medical Centre. A total of 1297 normal and abnormal images of lower extremities have been presented in the dataset. Table 1 shows the class distribution of the LERA dataset.

The LERA dataset is one of the benchmark musculoskeletal radiograph image datasets and has been applied in the proposed approach for producing a relatively improved

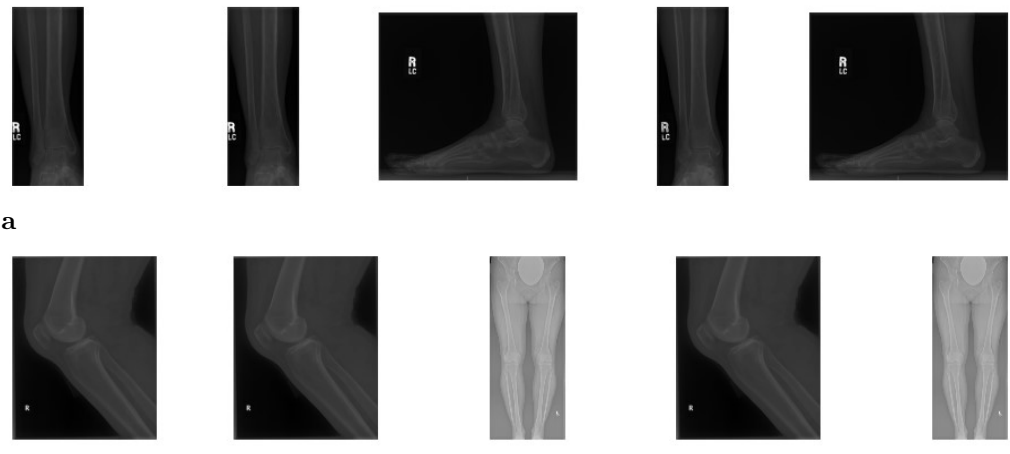

Fig. 2. LERA Dataset – (**a**) normal and (**b**) abnormal image samples.

degree of classification accuracy. Moreover, the interpretation in binary classification is distinguished in a way that abnormal as "1" and normal as "0". The abnormal categorization denotes that the radiograph consists of either fractures or any other abnormalities. Meanwhile, in normal categorization, the radiographs represent that the image is normal. The Figure 2 presents the sample images of the LERA dataset.

3.2. Pre-processing techniques

The image pre-processing method is applied in the input image to predominantly enhance the radiographic image's eminence, the edges that denote the possible fractures. This study's proposed method involves a one-hot encoding-based pre-processing approach to rectify the missing data issues.

One-Hot Encoding The one-hot encoding is a type of encoding method and is considered to be the most popular target encoding technique. The main advantage of this strategy is that it is a sparse vector, which is used in calculating the similarities or distances between the features for efficient classification. Here, one element is set to 1, and all other elements are set to 0. Contradictory to the other existing algorithms, the one-hot encoding method treats all missing values as a new class. This tends to mitigate the interference with data structure in the simulation.

3.3. Train and test split of data

The input LERA dataset consisting of normal and abnormal images of the lower extremities of the human body is split into train and test datasets. The splitting of data is done with 80% of train data and 20% of test data. The splitting of input data is such that training data gains more than two-thirds of the entire data. The training dataset is used in training the classifier employed in classifying the normal and abnormal images. The test data are applied to compute the performance measures.

3.4. Classification

3.4.1. EfficientNet model

The conventional EfficientNet is a kind of NN which uses the compound scaling method to produce better system performance. These existing models target to improve the performance and computational efficiency by subsiding the Floating Point Operations Per Second (FLOPS) and several parameters. Scaling up mechanisms involved in EfficientNet are Neural Architecture Search (NAS) and compound scaling. Initially, the baseline network is designed by performing NAS, a method used to automate the design of neural networks. It efficiently optimizes both efficiency and accuracy as measured on a FLOPS basis. The two parts present in EfficientNet are created using a baseline with NAS and compound scaling to increase the performance. Compared with other state-of-arts models, the EfficientNet significantly reduces the computational resources required to train the classifier. The scaling method involved in EfficientNet has shown uniform scaling across multiple dimensions. This could be more efficient when applied to a highly versatile architecture to improve the effectiveness of the model. When combined with CNN, the EfficientNet involves a scaling approach and achieves significant output in the performance.

3.4.2. MetaMag efficiency classifier

The MetaMag efficiency classifier is deployed in the proposed method, where the network architecture involves a new scaling model known as MetaMag scaling. The other existing CNNs randomly scale the network dimensions like resolution, dimension, and width. The MetaMag efficiency model uniformly scales the entire image with a fixed scaling coefficient. This tends to enhance the efficiency and accuracy of classification. In the classification process, the MetaMag efficiency model undergoes MetaMag scaling that uniformly scales all the dimensions of resolution, width and depth for procuring improved performance. It systematically analyses the model scaling and identifies the balancing network using a simple yet highly effective compound coefficient.

MetaMagnify scaling

The scaling factor, denoted as ϕ, allows for adjustments in the depth of the network. When ϕ is increased, the model becomes deeper and more robust, enhancing its capability to extract complex features. This is advantageous for tasks that demand sophisticated feature extraction, such as intricate pattern recognition in images or nuanced language understanding. Conversely, reducing ϕ results in a shallower model. This can be advantageous for simpler tasks or scenarios where computational resources are restricted. Shallow models are effective for straightforward classification tasks or when rapid inference speed is crucial. Furthermore, smaller values of ϕ facilitate faster training and reduce memory requirements. This makes them particularly suitable for environments where efficiency in model development and deployment is prioritized.

Unify normalization

The use of Unify Normalization offers a way to maintain the benefits of Batch Normalization (BN) while addressing its challenges with large activation memory requirements due to the need for sizable batch sizes. This is particularly relevant in memory-intensive AI accelerators that rely on local memory for enhanced speed and energy efficiency, despite tighter memory constraints. Additionally, our approach aims to preserve BN's normalization advantages while circumventing its regularization effects when they prove counterproductive. To adapt the EfficientNet architecture effectively, it is essential to adjust the initial scaling operations within the network. This ensures that scaling factors play a significant role in shaping the overall network structure. Furthermore, modifying batch normalization layers to accommodate variations in network width and depth is crucial for maintaining effective normalization during training.

Besides, this work focuses on improving the practical efficiency of the traditional EfficientNet model by using the unified normalization that reduces the computational loss and inexpensively fine-tuning at higher resolution. It eventually increases the size of the image and aids in obtaining finer details of the input image. This helps the classifier distinguish the input images into two categories: normal as 0 and abnormal as 1. The input data from the training dataset is fed into the input layer of the MetaMag efficiency classifier and then to the MetaMag scaling layer. By using this layer, the finer details of radiographic images are obtained that precisely classify the abnormalities found in the bone. The process involved in the MetaMag efficiency model is shown in Figure 3.

The MetaMag efficiency model uses the MetaMag scaling method that involves a series of fixed factors to scale the dimension of the network in a uniform manner based on resolution, depth, and width. The building block i is defined as a function of $A_{i+1} = B_i(A_i)$, where B_i denotes the operator and A_i represents the input tensor, and A_{i+1} is the output tensor. Thus, the CNN, denoted symbolically as n, is characterized by different layers as given in equation (1),

$$n = B_m^{q_m} \odot \cdots \odot B_2^{q_2} \odot B_1^{q_1}(A_1) = \odot_{i=1,\ldots,m} B_i(A_1)\,, \tag{1}$$

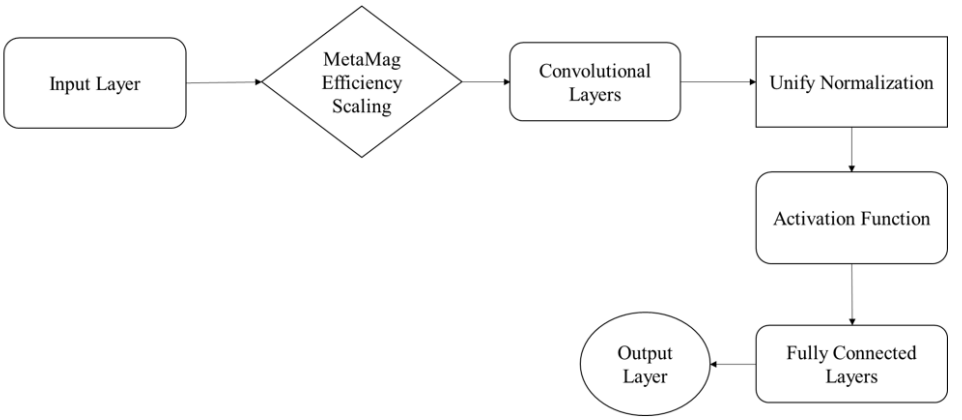

Fig. 3. Flow diagram of the proposed MetaMag efficiency classifier.

where $\odot$ is the Hadamard product, that is, the element-wise multiplication of two matrices, and the superscript q_i denotes the hyperparameter vector of B_i. This epitomizes the architecture of building block i, which is not able to be determined from training. Further, m signifies the number of layers present in the network. Further on, the hyperparameter matrix q with a building block defined in CNN is shown in equation (2),

$$n = \odot_{i=1,\ldots,m} B_i^{q_i}(A_{C_i,D_i,H_i,W_i}) \,. \tag{2}$$

The proposed modified EfficientNet model aims to resolve the optimization problem formulated in equation (3),

$$q_{\text{optimum}} = \arg\max_{q} \text{Accuracy } (n(q)(A_{C_i,D_i,H_i,W_i})) \,, \tag{3}$$

where q is the matrix of hyperparameters of the whole network, formed by vectors q_i of the subsequent operators B_i. The denotation $n(q)$ underlines the dependency of the network on its parameters. Therefore, the result of a search procedure of the modified EfficientNet model is the optimal hyperparameter matrix q. The architecture of the proposed modified EfficientNet model is displayed in Figure 4. In this structure, the convolution pooling layers consist of extracted features from the input radiographic images and `conv` blocks that process the feature maps. Further, the unified normalization is performed at the end of the network.

The modified `conv` layer i is defined by the function $Y_i = B_i(X_i)$, in which B_i is the operator, X_i denotes the input tensor, and Y_i represents the output tensor. The tensor shape for the function is given by $X_i = (H_i, W_i, C_i)$, where W_i and H_i are the spatial

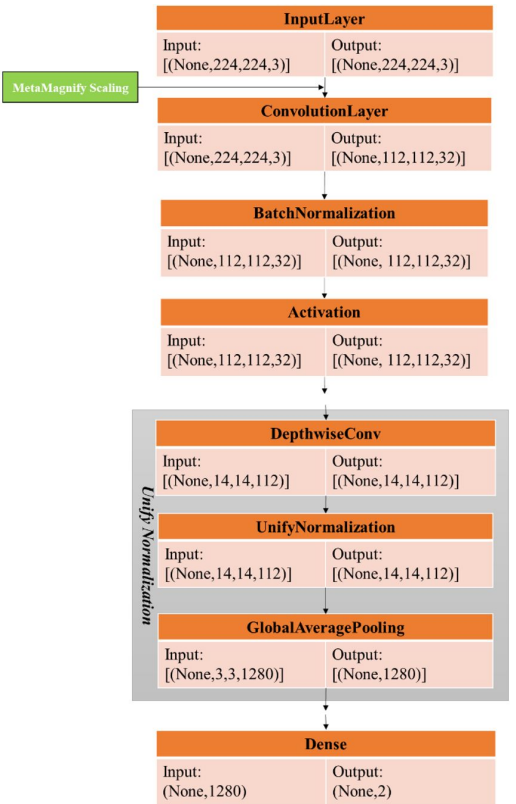

Fig. 4. Model architecture of the proposed MetaMag efficiency model

dimensions. Further, C_i signifies the channel dimension. Moreover, the modified `conv` layer is characterized by a list of composed layers, as shown in equation (4),

$$n = B_k \odot \cdots \odot B_2 \odot B_1(X_1) = \odot_{j=1,\ldots,k} B_j(X_j)\,. \tag{4}$$

All layers in each stage of modified layers possess the same convolutional type, while the first layer alone performs the down-sampling method, and the modified `conv` layer is represented in equation (5),

$$n = \odot_{i=1,\ldots,m} B_i^{p_i} X_{H_i,W_i,C_i}\,, \tag{5}$$

where $\odot_{i=1,\ldots,m} B_i$ is repeated p_i times in stage i, and H_i, W_i, C_i is the shape of input tensor X of layer i. To find the best layer architecture B_i, the model involved MetaMag

scaling that expands the network length p_i, width C_i, and resolution H_i, W_i without altering the predefined B_i in the baseline network. Thus, by fixing the B_i, the MetaMag scaling simplifies the design issues for new resource constraints. However, to improve the accuracy of the proposed model for any resource constraints, an optimization problem is formulated in equation (6),

$$n_{\text{optimum}} = \max_{d,w,r} \text{Accuracy}\left(n(d,w,r)\right) , \tag{6}$$

where

$$n(d,w,r) = \odot B_i^{d.^{L_i}} \left(X_{r\cdot H_i, r.W_i, r.C_i}\right),$$

here, (d, w, r) denote the depth, width and resolution of the scaling network, and L_i is the layer at the stage i. Specifically, the modified `conv` layer captures more complex features and gets generalized better in new tasks. But, this network faces difficulty due to vanishing gradient issues. So, the computation is reduced by lowering the training resolution and thus inexpensively fine-tuning at higher resolution. This method is done by implementing the unifying normalization mechanism to normalize activations throughout the network. It combines statistics from LN (Layer Normalization) and BN (Batch Normalization), adapting different batch sizes and model depths. This ensures stable and efficient training across the proposed MetaMag efficiency model. The unified normalization is applied on X, which denotes the unnormalized pre-activations to generate normalized pre-activations $Q_{..c}$ before a nonlinearity Θ and an affine transform finally produce the post-activation function $P_{..c}$, as follows

$$Q_{..c} = \frac{X_{..c} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} , \tag{7}$$

$$P_{..c} = \Theta(\gamma_c Q_{..c} + \alpha_c) , \tag{8}$$

where c is the index of the channel, $\sqrt{\sigma_c^2}, \mu_c$ denote the standard deviation and mean of X, and α_c, γ_c are the unified normalization's shift parameters and scale in each channel. The ϵ represents the unified normalization's numerical stability constant, and '.' denotes a placeholder for an index. Thus, this foundational principle of unified normalization is significant for successful scaling to deep and large models. Further, the proxy-normalized activation step is applied in equation (8). This step tends to normalize $\Theta(\gamma_c Q_{..c} + \alpha_c)$, where $Q_{..c} \sim N(\alpha_{..c}, (1+\gamma_c)^2)$ is the proxy variable with variance $(1+\gamma_c)^2$ and mean $\alpha_{..c}$. These variables are subjected to weight decay to denote that Q is close to normalized. Hence, the unified normalization for each element and the channel is given by equations (9) and (10) (some index placeholders dropped for simplicity),

$$Q_{..b} = \frac{X_{..b} - \mu_b}{\sqrt{\sigma_b^2 + \epsilon}} , \tag{9}$$

$$P_b = \frac{\Theta(\gamma_c Q_{..c} + \alpha_c) - E_{\gamma_c}\left[\Theta(\gamma_c Q_{..c} + \alpha_c)\right]}{\sqrt{\text{Var}_{\gamma_c}[\Theta(\gamma_c Q_{..c} + \alpha_c)] + \epsilon}} , \tag{10}$$

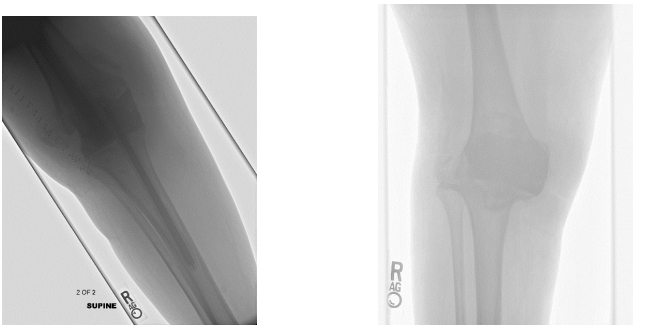

Fig. 5. Examples of original normal images present in the data set.

where b denotes the batch element for the proxy-normalization of P_b; further, $Q_c \sim N(\alpha_c, (1+\gamma_c)^2), \epsilon$ are numerical stability constants of unified and proxy normalizations, γ_c is the Gaussian proxy variable, and E_{γ_c} represents the measures of central tendency for the variable γ_c. On the other hand, the inclusion of unified normalization at the network leads in a full-batch setting to add the following operations as shown in equation (11),

$$y^l_{a,c} = \frac{y^l_{a,c} - \mu_c(X^l)}{\sigma_c(X^l)}, \quad y^l_{a,c} = \gamma^l_c y^l_{a,c} + \alpha^l_c\,, \tag{11}$$

where l is the layer, $\sigma_c(X^l)$ and $\mu_c(X^l)$ are the standard deviations and mean of X^l, and α^l_c, γ^l_c denote the shift parameters and channel-wise scale. Finally, the output is driven to the `avg max` pooling layer and then collected from the dense layer.

4. Results and discussion

The effectiveness of the proposed MetaMag efficiency model has been validated by using four different performance measures based on different lower extremity images from the LERA dataset. The experiment was carried out on the Google Colab Notebook Pro version. In total 50 epochs were used in each fold. This section deliberates the results produced by the proposed method in classifying the image samples.

4.1. Exploratory Data Analysis

The Exploratory Data Analysis (EDA) is specifically used to analyze and examine the LERA dataset and thus summarise the main attributes of the dataset. It also visualizes the distribution of data, discovers patterns, locates outliers, and detects correlations. The figure 5 represents the original images present in the LERA dataset.

4.2. Performance measures

The outcome of the proposed system is attained by evaluating the measures: accuracy, precision, recall, specificity, and F1-score. With this output testing accuracy, the improvement of the system is analyzed. Below, TP, TN, FP and FN denote the numbers of true positive, true negative, false positive, and false negative classifications. The probabilities are estimated by the respective relative frequencies.

Accuracy The accuracy is considered as the primary evaluation index in the classification process, which refers to the proportion of input samples that are classified correctly. The accuracy is evaluated as follows

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \,. \tag{12}$$

Precision Precision denotes the probability of the sample that is truly positive among all the samples that are identified to be positive and is given by

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \,. \tag{13}$$

Sensitivity (Also called **recall**; these two names are used interchangeably in the paper, depending on the convention used in the reference sources.) It is the probability of being identified as a positive sample within the actually positive samples. It is denoted as

$$\text{Sensitivity} = \frac{\text{TP}}{\text{FN} + \text{TP}} \,. \tag{14}$$

Specificity It is the probability of being identified as a negative sample within the actually negative samples. It is denoted as

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \,. \tag{15}$$

F1-score The F1-score is calculated as the harmonic mean of recall and precision and is given by

$$\text{F1-score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \,. \tag{16}$$

The above evaluation metrics, or indexes, are used in analyzing the performance of the proposed MetaMag efficiency model.

4.3. Performance analysis

To better verify the efficiency of the proposed model, the obtained results of abnormalities in the body's lower extremities are shown in Figure 6.

From figure 6 it is projected that the proposed model can segment the abnormal part of the image by visualizing it through contrast enhancement. Thus, generalizability was effectually recognized showing a lack of significant decrement in performance.

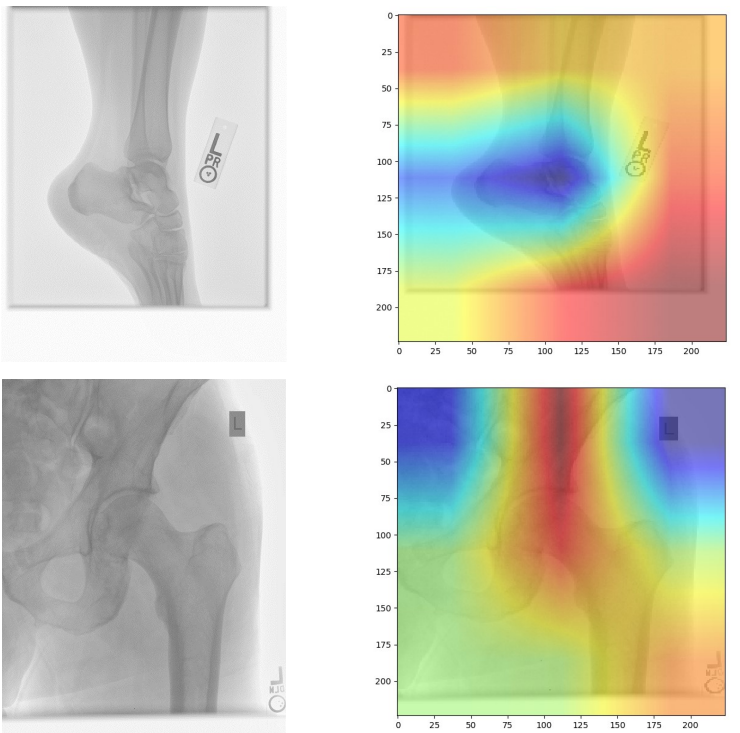

Fig. 6. Abnormalities identified by the proposed MetaMag Efficiency model. Left: original images, right: processed images. In the upper image, the blue color shows high intensity of the abnormality, whereas in the lower image, the red color shows high intensity of the abnormality.

4.4. Internal results

By evaluating the internal test set, the precision, recall, F1-score, and accuracy of the proposed MetaMag Efficiency technique and traditional EfficientNet model are generated. The outcomes are shown in Table 2 and the corresponding graphical representation is displayed in Figure 7.

It is observed that the traditional EfficientNet model produces an accuracy rate of 85%, precision of 94%, recall of 78%, and F1-score of 85%. While, the proposed MetaMag Efficiency model produced an accuracy of 95%, precision of 95%, recall of 97%, and F1-score of 96%. This indicates the improved performance of the proposed method by implementing MetaMag scaling and Unify normalization methods. Figure 8 illustrates the graphical representation of model accuracy and loss.

From figure 8, it can be concluded that the proposed method has produced increased

Tab. 2. Outcome of the proposed and the traditional model.

Model	Precision	Recall	F1-score	Accuracy
Proposed	0.95	0.97	0.96	0.95
EfficientNet	0.94	0.78	0.85	0.85

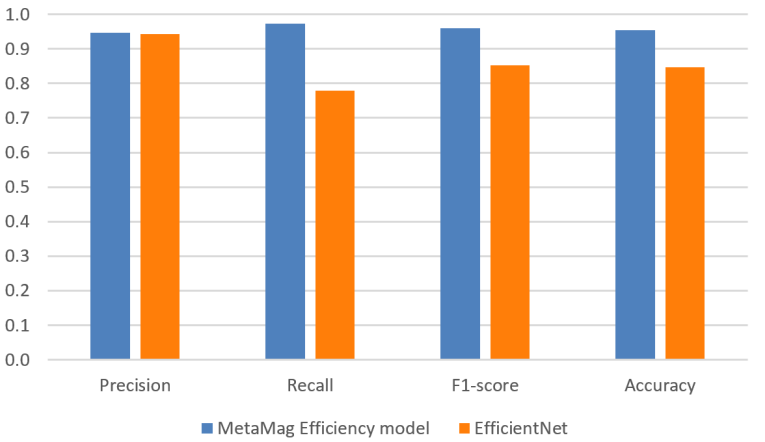

Fig. 7. Graphical representation of performance analysis of the proposed MetaMag Efficiency model and the traditional EfficientNet model.

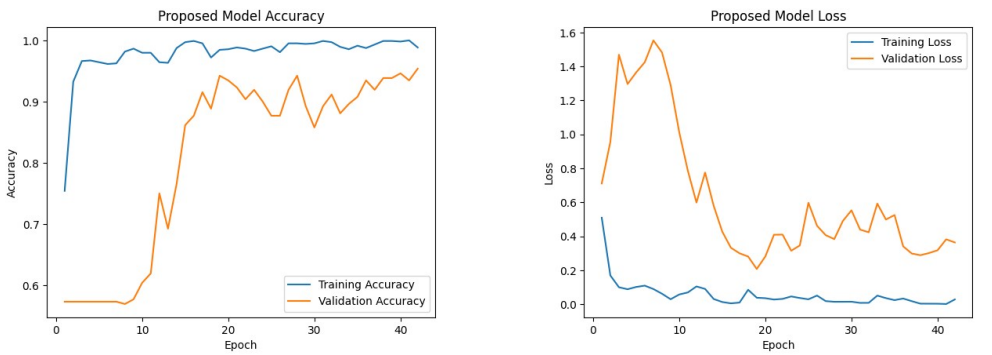

Fig. 8. Accuracy and loss prediction of the proposed MetaMag Efficiency model.

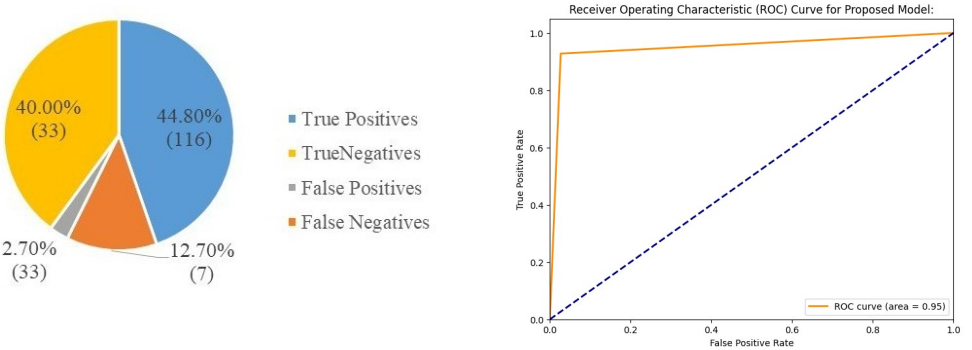

Fig. 9. Data derived from the confusion matrix and ROC for the proposed MetaMag Efficiency model.

accuracy. Both the training curve and validation curve correlate with each other projecting that the train dataset and test dataset are most probably similar to each other. Further, the data derived from the confusion matrix are drawn for the proposed method to analyze effectiveness. The model loss indicates how the model's prediction was on the input samples. If the loss is minimal, then the efficacy of the proposed approach will be enhanced. In Figure 8, the x-axis denotes the loss and the y-axis signifies the number of model training epochs. It is noted that the validation accuracy is higher than the training accuracy for some epochs. Both the training and validation curve follows a uniformity as the number of epochs increases. This denotes that the loss decreased with an increase in accuracy. The data derived from the confusion matrix and the ROC of the proposed model are represented in Figure 9.

The confusion matrix, also known as the error matrix, represents the counts from predicted and actual values. The True Positives value represents the number of positive samples that are accurately classified, while True Negatives denotes the number of negative samples categorized correctly. False Positives value signifies the number of actual negative samples classified as positive, and False Negatives is the number of actual positive samples classified as negative. From Figure 9 it is inferred that 144 samples were correctly classified as normal images, and 104 abnormal samples were classified accurately. Only 4 normal samples were misclassified as abnormal, and 8 abnormal samples as normal. With minimum error, the precision of the proposed approach is improved. Additionally, the area under the ROC curve of the proposed model is found to be 0.93, indicating improved performance. Further, the performance of the conventional EfficientNet is also analyzed. The data derived from the confusion matrix and ROC of the traditional EfficientNet model are shown in Figure 10.

The data derived from the confusion matrix of the traditional EfficientNet model is analyzed, and it is found that 116 normal samples are correctly classified, and 104

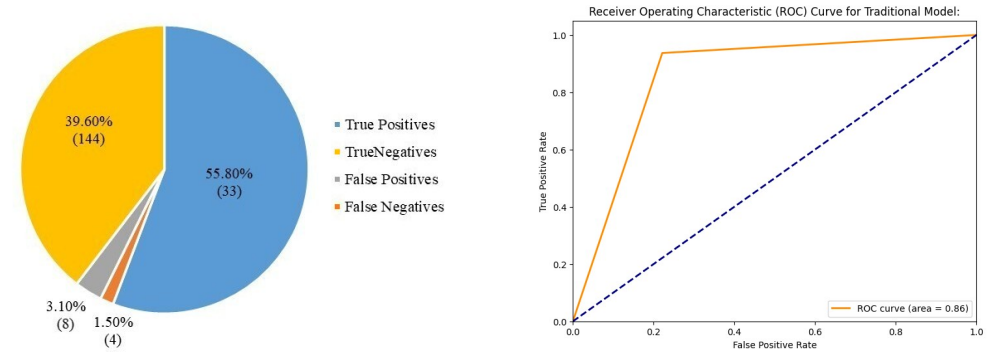

Fig. 10. Data derived from the confusion matrix and ROC for the traditional EfficientNet model.

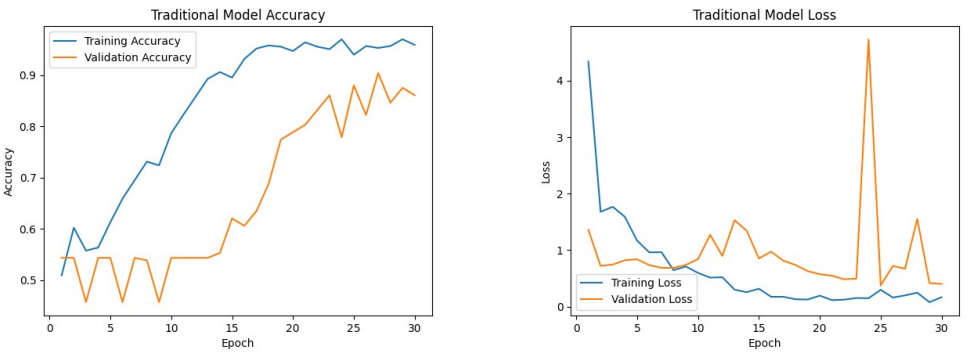

Fig. 11. Accuracy and loss prediction of the traditional EfficientNet Model.

abnormal images are classified accurately, while 33 normal images are wrongly classified as abnormal and 7 abnormal images are classified incorrectly as normal samples. The count of correctly classified samples is less than the count classified by the proposed model. Further, the area under the ROC curve of the traditional model is 0.86, denoting decreased accuracy and performance. Then, the model accuracy and loss prediction for the conventional method is shown in Figure 11.

The training and testing curves are partially correlated in the model accuracy plot, denoting decreased accuracy. Further, the loss plot denotes that both the loss curve interlinks with each other, representing increased model loss. This denotes that increased model loss leads to reduced performance of the model.

Tab. 3. Comparison of the performance of the proposed model and the existing model from [18].

Model	Recall	Specificity	Accuracy
DCNN Triquetral fracture (n=50) – 2-stage	0.96	0.88	0.92
DCNN Triquetral fracture (n=50) – 1-stage	0.96	0.64	0.80
Second fracture (n=24) – 2-stage	0.917	0.917	0.917
Second fracture (n=24) – 1-stage	0.917	0.917	0.917
Proposed	**0.97**	**0.93**	**0.95**

4.5. Comparative analysis

The comparison of the proposed method with other existing methods enumerates the efficiency of the proposed system. Here, the study compares the existing DenseNet-161 model in terms of lower extremities' accuracy, sensitivity, and specificity. The outcome of the conventional model and the proposed system is exemplified in Table 3.

Table 3 indicates that the proposed model attained better values than the existing models. It attained 95% of sensitivity, 97% of specificity and 95% of accuracy which shows the value of the proposed efficient model. Table 4 depicts the comparative analysis of the proposed and another existing model.

From Table 4 and Figure 12 it can be inferred that the existing DenseNet-161 model produced an accuracy of 79%, precision of 97%, and recall of 66%. Whereas the proposed MetaMag Efficiency model produced importantly improved overall accuracy of 95%, precision of 95% – slightly worse, and recall of 97% – improved.

Only a few studies have focused on detecting bone fractures in DM patients. So, only a limited comparison is provided to analyze the model's working. Thus, from analyzing using different evaluation indicators, it is identified that the proposed model has achieved improved performance compared to other existing models. The basic EfficientNet model tends to provide limited performance, whereas the proposed MetaMag efficiency model provides improved performance due to the implementation of the MetaMag scaling and Unify normalisation. The MetaMag scaling supports the model in enlarging the radiographic image and finely detecting significant patterns of abnormalities in the bone. Further, the unified normalization reduces the losses produced by the input samples and thus increases the model's efficiency.

Tab. 4. Comparison of the performance of the proposed model and the DenseNet model [30].

Model	Precision	Recall	Accuracy
DenseNet-161 [30]	0.97	0.66	0.79
Proposed	0.95	**0.97**	**0.95**

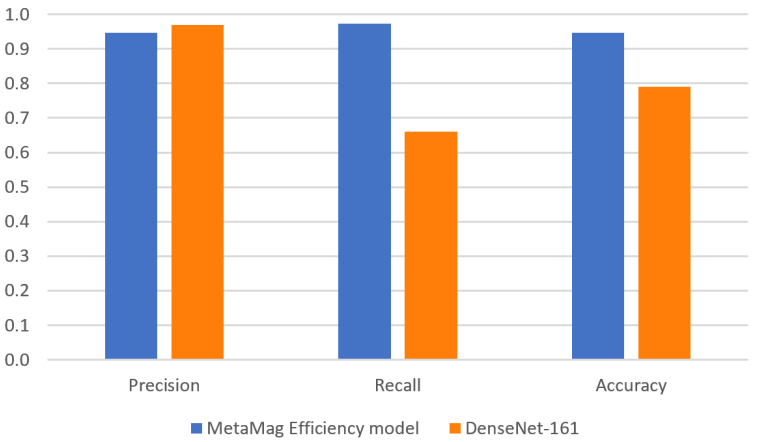

Fig. 12. Graphical representation of the comparison of the proposed model and the DenseNet-161 model [30].

5. Conclusion

Various deep learning methods are involved in diagnosing various diseases and have produced efficient outcomes. In that case, the previous phase concentrated on detecting foot ulcers in diabetes mellitus (DM) patients by using the Deep Convolutional Neural Network (DCNN) based Xception model. This approach produced improved outcomes and aided in efficiently classifying healthy and diabetic foot ulcer (DFU) images. On the other hand, the present phase focused on identifying fractures in diabetes patients. DM is associated with several other factors, and delay in treatment may lead to complex patient risks. Once a fracture occurs in diabetes patients, it is difficult to cure, and abnormalities exploit the routine lifestyle of patients. So, early detection of fractures can help physiologists efficiently cure the complications. Hence, the proposed approach implemented a MetaMag efficiency model to detect and classify normal and abnormal images from the given input radiograph images. Along with the classifier, MetaMag scaling and Unify normalization approaches were used to effectively obtain the fine details of input samples and reduce the loss that occurred in the proposed system. The outcomes of the proposed method produced an accuracy of 95%, compared with the traditional EfficientNet model, which produced an accuracy of 85%. This denoted the improved performance of the proposed MetaMag efficiency model. The study can be further improved by using different approaches of deep learning algorithms to produce higher classification accuracy.

References

[1] R. Ali, J. H. Chuah, M. S. A. Talip, N. Mokhtar, and M. A. Shoaib. Structural crack detection using deep convolutional neural networks. *Automation in Construction*, 133:103989, 2022. doi:10.1016/j.autcon.2021.103989.

[2] M. J. Awan, M. S. M. Rahim, N. Salim, M. A. Mohammed, B. Garcia-Zapirain, et al. Efficient detection of knee anterior cruciate ligament from magnetic resonance imaging using deep learning approach. *Diagnostics*, 11(1):105, 2021. doi:10.3390/diagnostics11010105.

[3] R. Bagaria, S. Wadhwani, and A. K. Wadhwani. Bone fracture detection in X-ray images using convolutional neural network. In: *SCRS Conference Proceedings on Intelligent Systems*, pp. 459–466. SCRS, India, 2022. doi:10.52458/978-93-91842-08-6-43.

[4] Z. Cao, L. Xu, D. Z. Chen, H. Gao, and J. Wu. A robust shape-aware rib fracture detection and segmentation framework with contrastive learning. *IEEE Transactions on Multimedia*, 25:1584–1591, 2023. doi:10.1109/TMM.2023.3263074.

[5] W. Chen, D. HolcDorf, M. W. McCusker, F. Gaillard, and P. D. Howe. Perceptual training to improve hip fracture identification in conventional radiographs. *PloS One*, 12(12):e0189192, 2017. doi:10.1371/journal.pone.0189192.

[6] P. Chłąd and M. R. Ogiela. Deep learning and cloud-based computation for cervical spine fracture detection system. *Electronics*, 12(9):2056, 2023. doi:10.3390/electronics12092056.

[7] M. Davenport and M. P. Oczypok. Knee and leg injuries. *Emergency Medicine Clinics*, 38(1):143–165, 2020. doi:10.1016/j.emc.2019.09.012.

[8] M. R. Delavar. Hybrid machine learning approaches for classification and detection of fractures in carbonate reservoir. *Journal of Petroleum Science and Engineering*, 208:109327, 2022. doi:10.1016/j.petrol.2021.109327.

[9] C. Germann, A. N. Meyer, M. Staib, R. Sutter, and B. Fritz. Performance of a deep convolutional neural network for MRI-based vertebral body measurements and insufficiency fracture detection. *European Radiology*, 33(5):3188–3199, 2023. doi:10.1007/s00330-022-09354-6.

[10] N. Gougoulias, H. Oshba, A. Dimitroulias, A. Sakellariou, and A. Wee. Ankle fractures in diabetic patients. *EFORT Open Reviews*, 5(8):457–463, 2020. doi:10.1302/2058-5241.5.200025.

[11] O. Q. Groot, M. E. R. Bongers, P. T. Ogink, J. T. Senders, A. V. Karhade, et al. Does artificial intelligence outperform natural intelligence in interpreting musculoskeletal radiological studies? A systematic review. *Clinical Orthopaedics and Related Research*, 478(12):2751, 2020. doi:10.1097/CORR.0000000000001360.

[12] B. Guan, J. Yao, G. Zhang, and X. Wang. Thigh fracture detection using deep learning method based on new dilated convolutional feature pyramid network. *Pattern Recognition Letters*, 125:521–526, 2019. doi:10.1016/j.patrec.2019.06.015.

[13] K. Karthik and S. S. Kamath. MSDNet: A deep neural ensemble model for abnormality detection and classification of plain radiographs. *Journal of Ambient Intelligence and Humanized Computing*, 14:16099–16113, 2023. doi:10.1007/s12652-022-03835-8.

[14] C. Kokkotis, S. Moustakidis, G. Giakas, and D. Tsaopoulos. Identification of risk factors and machine learning-based prediction models for knee osteoarthritis patients. *Applied Sciences*, 10(19):6797, 2020. doi:10.3390/app10196797.

[15] S. Kumar, P. Goswami, and S. Batra. Fuzzy rank-based ensemble model for accurate diagnosis of osteoporosis in knee radiographs. *International Journal of Advanced Computer Science and Applications*, 14(4):262–270, 2023. doi:10.14569/IJACSA.2023.0140430.

[16] Y. Ma and Y. Luo. Bone fracture detection through the two-stage system of crack-sensitive convolutional neural network. *Informatics in Medicine Unlocked*, 22:100452, 2021. doi:10.1016/j.imu.2020.100452.

[17] Stanford University School of Medicine. LERA- Lower Extremity RAdiographs, 2024. https://aimi.stanford.edu/lera-lower-extremity-radiographs, [Accessed: September 19, 2024].

[18] M. Ren and P. H. J. S. R. Yi. Deep learning detection of subtle fractures using staged algorithms to mimic radiologist search pattern. *Skeletal Radiology*, 51(2):345–353, 2022. doi:10.1007/s00256-021-03739-2.

[19] A. Sasidhar, M. Thanabal, and P. Ramya. Efficient transfer learning model for humerus bone fracture detection. *Annals of the Romanian Society for Cell Biology*, 25(2):3932–3942, 2021. http://annalsofrscb.ro/index.php/journal/article/view/1398.

[20] T. Schmidt, N. M. Simske, M. A. Audet, A. Benedick, C.-Y. Kim, et al. Effects of diabetes mellitus on functional outcomes and complications after torsional ankle fracture. *Journal of the American Academy of Orthopaedic Surgeons*, 28(16):661–670, 2020. doi:10.5435/JAAOS-D-19-00545.

[21] H. Sun, X. Wang, Z. Li, A. Liu, S. Xu, et al. Automated rib fracture detection on chest X-ray using contrastive learning. *Journal of Digital Imaging*, 36(5):2138–2147, 2023. doi:10.1007/s10278-023-00868-z.

[22] Y. L. Thian, Y. Li, P. Jagmohan, D. Sia, V. E. Y. Chan, et al. Convolutional neural networks for automated fracture detection and localization on wrist radiographs. *Radiology: Artificial Intelligence*, 1(1):e180001, 2019. doi:10.1148/ryai.2019180001.

[23] K. A. Thomas, Ł. Kidziński, E. Halilaj, S. L. Fleming, G. R. Venkataraman, et al. Automated classification of radiographic knee osteoarthritis severity using deep neural networks. *Radiology: Artificial Intelligence*, 2(2):e190065, 2020. doi:10.1148/ryai.2020190065.

[24] M. Tian, B. Li, H. Xu, D. Yan, Y. Gao, et al. Deep learning assisted well log inversion for fracture identification. *Geophysical Prospecting*, 69(2):419–433, 2021. doi:10.1111/1365-2478.13054.

[25] M. Varma, M. Lu, R. Gardner, J. Dunnmon, N. Khandwala, et al. Automated abnormality detection in lower extremity radiographs using deep learning. *Nature Machine Intelligence*, 1(12):578–583, 2019. doi:10.1038/s42256-019-0126-0.

[26] S. Verma, S. Kulshrestha, C. Rajput, and S. Patel. Detecting bone fracture using transfer learning. In: O. P. Verma, S. Roy, S. C. Pandey, and M. Mittal, eds., *Advancement of Machine Intelligence in Interactive Medical Image Analysis*, pp. 215–228, Algorithms for Intelligent Systems. Springer Singapore, 2020. doi:10.1007/978-981-15-1100-4_10.

[27] M. Wu, Z. Chai, G. Qian, H. Lin, Q. Wang, et al. Development and evaluation of a deep learning algorithm for rib segmentation and fracture detection from multicenter chest CT images. *Radiology: Artificial Intelligence*, 3(5):e200248, 2021. doi:10.1148/ryai.2021200248.

[28] L. Xue, W. Yan, P. Luo, X. Zhang, T. Chaikovska, et al. Detection and localization of hand fractures based on GA_Faster R-CNN. *Alexandria Engineering Journal*, 60(5):4555–4562, 2021. doi:10.1016/j.aej.2021.03.005.

[29] D. P. Yadav, A. Sharma, S. Athithan, A. Bhola, B. Sharma, et al. Hybrid SFNet model for bone fracture detection and classification using ML/DL. *Sensors*, 22(15):5823, 2022. doi:10.3390/s22155823.

[30] J. Zhang, Z. Li, S. Yan, H. Cao, J. Liu, et al. An algorithm for automatic rib fracture recognition combined with nnU-Net and DenseNet. *Evidence-Based Complementary and Alternative Medicine*, 2022(1):5841451, 2022. doi:10.1155/2022/5841451.

[31] K. Üreten, H. F. Sevinç, U. İğdeli, A. Onay, and Y. Maraş. Use of deep learning methods for hand fracture detection from plain hand radiographs. *Turkish Journal of Trauma and Emergency Surgery*, 28(2):196, 2022. doi:10.14744/tjtes.2020.06944.

S. Rajeashwari is a Full Time Research Scholar in the Department of Computer Science, Sri S. Ramasamy Naidu Memorial College, Affiliated to Madurai Kamaraj University, Madurai, India. Her major research areas include Data Mining, Knowledge Engineering, Image Processing, Medical Image Analysis, and Machine Learning.

Dr. K. Arunesh is an Associate Professor in the Department of Computer science working in S. Ramasamy Naidu Memorial College, Affiliated to Madurai Kamaraj University, Madurai, India. He has 34 years of teaching experience. His major research areas include Machine Learning, Web Usage Mining, Recommender System, Data Mining, and Computational Intelligence. He has over 40 publications in refereed journals and serves as a reviewer for several esteemed journals. He has also been an advisory committe member for various conferences.