

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

Rafał Wyszyński , Karol Struniawski \*

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

\*Corresponding author: Karol Struniawski ([karol.struniawski@sggw.edu.pl](mailto:karol.struniawski@sggw.edu.pl))

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

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

## 1. Introduction

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

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

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

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

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

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

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

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

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

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

## 2. Methods

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

### 2.1. Dataset information

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

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

### 2.2. Subimages retrieval

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

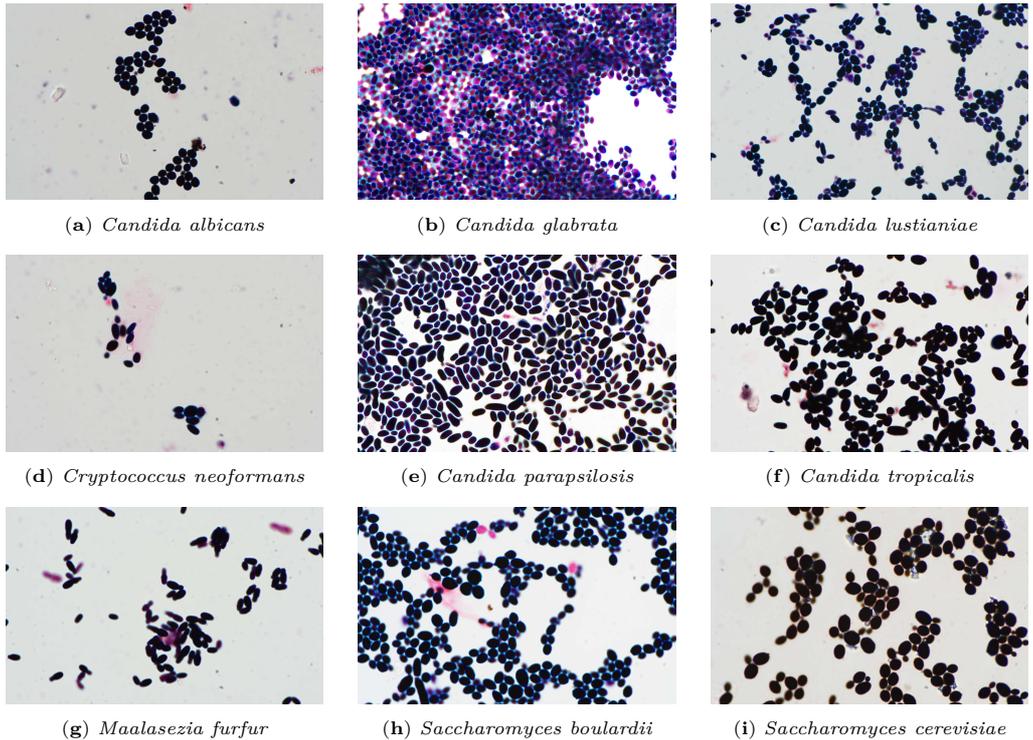


Fig. 1. Microscopic images of each fungus from the dataset.

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

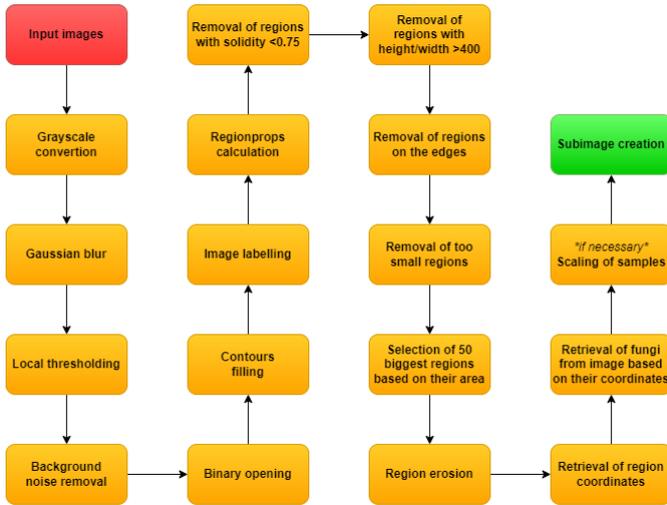


Fig. 2. The data flow diagram illustrating the proposed algorithm for subimages retrieval.

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

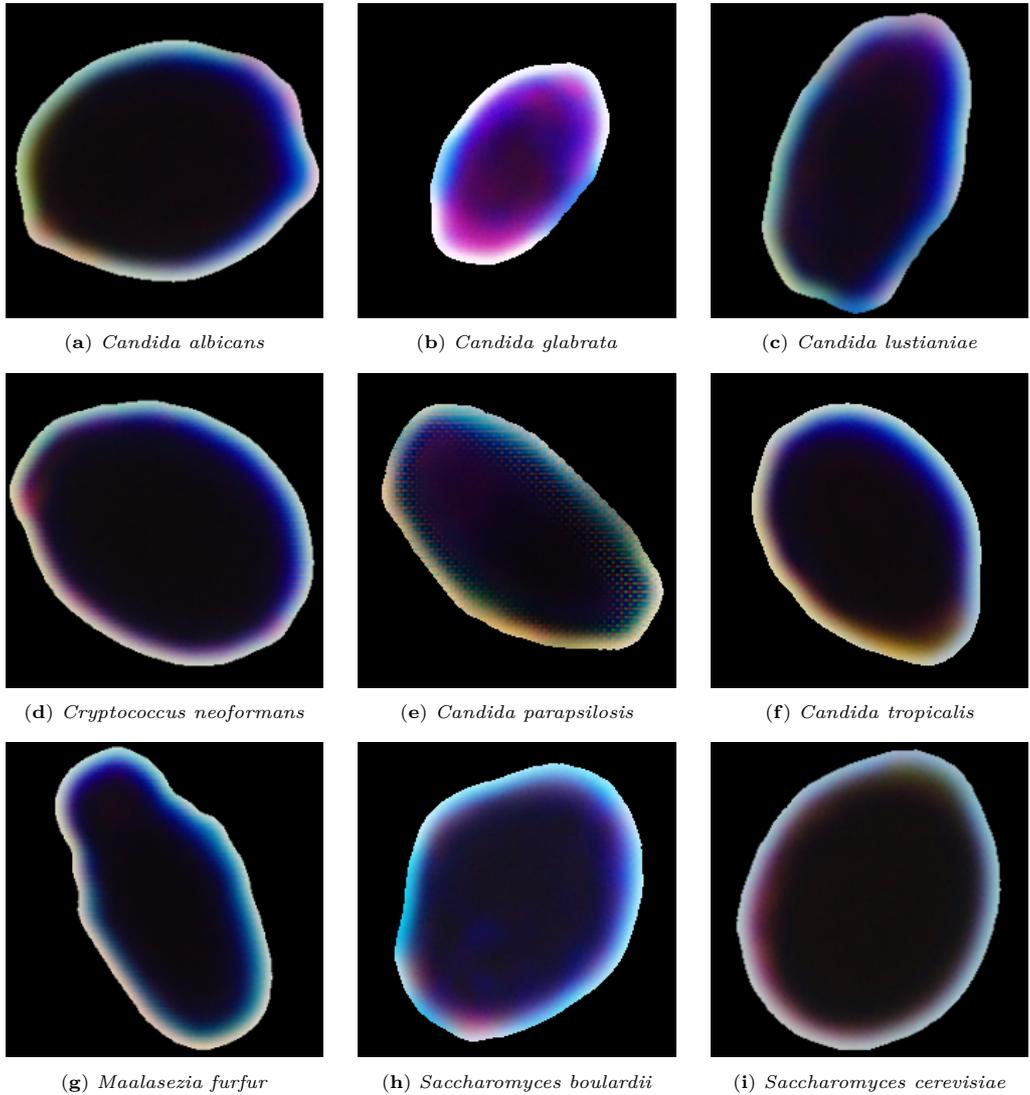


Fig. 3. Sample subimages retrieved from the original dataset.

### 2.3. Convolutional Neural Network

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

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

### 2.4. Residual Neural Network

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

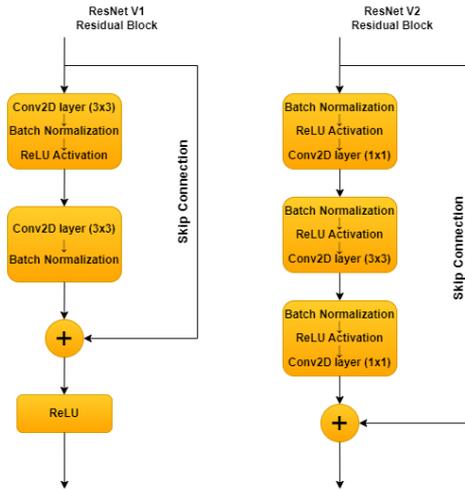


Fig. 4. Residual block architecture of ResNet V1 (left) and ResNet V2 (right).

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

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

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

## 2.5. Transfer learning

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

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

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

## 3. Experiments

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

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

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

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

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

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

**Model 5:** Model 1 with Trainable = TRUE

**Model 6:** Model 2 with Trainable = TRUE

**Model 7:** Model 3 with Trainable = TRUE

**Model 8:** Model 4 with Trainable = TRUE

**Model 9:** Model 1 with Batch Size = 64

**Model 10:** Model 2 with Batch Size = 64

**Model 11:** Model 3 with Batch Size = 64

**Model 12:** Model 4 with Batch Size = 64

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

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

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

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

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

**Model 18:** Model 17 with Learning Rate = 0.0001

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

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

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

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

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

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

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

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

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

**Model 28:** Model 27 with Learning Rate = 0.001

**Model 29:** Model 27 with Learning Rate = 0.0001

**Model 30:** Model 27 with Learning Rate = 0.00001

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

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

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

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

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

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

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

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

**Model 39:** Model 27 with Learning Rate = 0.000001

**Model 40:** Model 27 with Learning Rate = 0.0000001

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

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

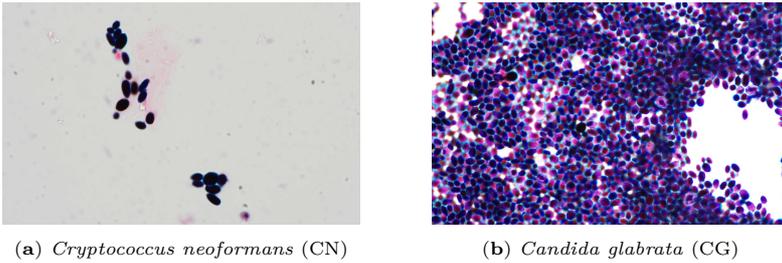


Fig. 5. CN class sample image structure compared to the CA class image.

#### 4. Results

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

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

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

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

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

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

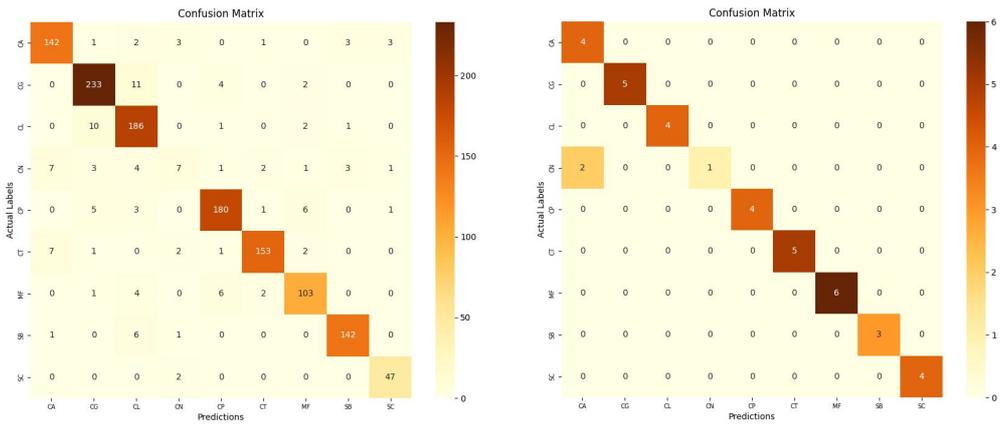


Fig. 6. Comparison of Confusion Matrices for the 30th model for each subimage (left) and after majority voting (right).

## 5. Conclusion

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

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

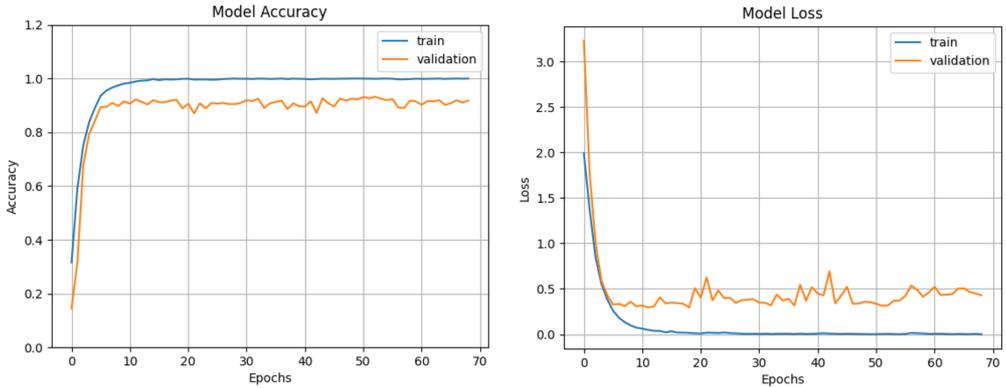


Fig. 7. Accuracy (left) and loss (right) functions during training of the 30th model on training and validation sets.

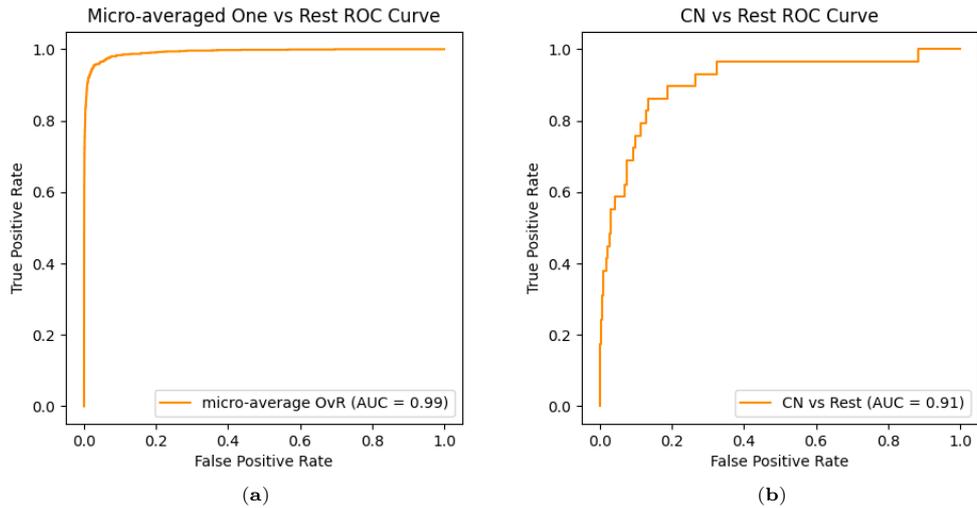


Fig. 8. AUC values and ROC curves of the 30th model on test set, micro-averaged: (a) One vs. Rest and (b) CN vs. Rest.

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

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

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

## References

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

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

- [24] Q. Li, W. Cai, X. Wang, et al. Medical image classification with convolutional neural network. In: *International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 844–848. Singapore, 10-12 Dec 2014. doi:10.1109/ICARCV.2014.7064414.
- [25] S. Liu, W. Cai, Y. Song, et al. Localized sparse code gradient in Alzheimer’s disease staging. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5398–5401. Osaka, Japan, 3-7 Jul 2013. doi:10.1109/EMBC.2013.6610769.
- [26] D. Misra. Mish: A self regularized non-monotonic activation function. *ArXiv*, 2020. ArXiv:1908.08681. doi:10.48550/arXiv.1908.08681.
- [27] O. M. Niall, C. Sean, and C. Anderson. In: *Computer Vision Conference (CVC)*, vol. 1, pp. 128–144. Las Vegas, USA, 25-26 Apr 2020. doi:10.1007/978-3-030-17795-9.
- [28] F. C. Odds. *Candida and candidosis*. Bailliere Tindall, 2nd edn., 1988.
- [29] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training Recurrent Neural Networks. *arXiv*, 2013. ArXiv:1211.5063. doi:10.48550/arXiv.1211.5063.
- [30] M. A. Pfaller and D. J. Diekema. Epidemiology of invasive candidiasis: a persistent public health problem. *Clin. Microbiol. Rev.*, 20(1):133–163, 2007. doi:10.1128/CMR.00029-06.
- [31] S. Rawat, B. Bisht, V. Bisht, et al. Mefunx: A novel meta-learning-based deep learning architecture to detect fungal infection directly from microscopic images. *Franklin Open*, 6:100069, 2024. doi:https://doi.org/10.1016/j.fraope.2023.100069.
- [32] C. F. Rodrigues, S. Silva, and M. Henriques. *Candida glabrata*: a review of its features and resistance. *Eur. J. Clin. Microbiol.*, 33(5):673–688, 2014. doi:10.1007/s10096-013-02009-3.
- [33] P. Sahoo, S. Soltani, and A. Wong. A survey of thresholding techniques. *Comput. Graph. Image Process.*, 41:233–260, 1988. doi:10.1016/0734-189X(88)90022-9.
- [34] M. Schaefer, S. Migge-Kleian, and S. Scheu. *The Role of Soil Fauna for Decomposition of Plant Residues*, pp. 207–230. Springer Berlin Heidelberg, 2009. doi:10.1007/b82392\_13.
- [35] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015. doi:10.1016/j.neunet.2014.09.003.
- [36] T. R. Singh, S. Roy, O. I. Singh, T. Sinam, and K. M. Singh. A new local adaptive thresholding technique in binarization. *ArXiv*, 2012. ArXiv:1201.5227. doi:10.48550/arXiv.1201.5227.
- [37] R. Srisha and A. Khan. Morphological operations for image processing : Understanding and its applications. pp. 17–19. Tamil Nadu, India, 26-27 Apr 2013. <http://www.annauniv.edu/pdf/NCSCV.pdf>.
- [38] K. Struniawski, A. Konopka, and R. Kozera. Identification of soil bacteria with machine learning and image processing techniques applying single cells’ region isolation. In: *Modelling and Simulation: The European Simulation and Modelling Conference (ESM)*, pp. 76–81. Porto, Portugal, 26-28 Oct 2022.
- [39] K. Struniawski, R. Kozera, P. Trzcinski, et al. Automated identification of soil fungi and chromista through convolutional neural networks. *Eng. Appl. Artif. Intell.*, 127, 2024. doi:10.1016/j.engappai.2023.107333.
- [40] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, et al. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. doi:10.7717/peerj.453.
- [41] G. Xie and W. Lu. Image edge detection based on OpenCv. *Int. J. Electr. Electron. Eng. Telecommun.*, 1(1):104–106, 2013. doi:10.12720/IJEEEE.1.2.104-106.
- [42] Z. Xie, J. Li, and H. Shi. A face recognition method based on cnn. In: *High Performance Computing and Computational Intelligence Conf.*, vol. 1395, p. 012006. Chengdu, China, 25–27 Oct 2019. doi:10.1088/1742-6596/1395/1/012006.

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