Deep learning approach to describe and classify fungi microscopic images

Bartosz Zieliński1,2*, Agnieszka Sroka-Oleksiak3,4, Dawid Rymarczyk1,2, Adam Piekarczyk1, Monika Brzychczy-Włoch4

1 Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland, 2 Ardigen, Kraków, Poland, 3 Department of Mycology, Chair of Microbiology, Faculty of Medicine, Jagiellonian University Medical College, Kraków, Poland, 4 Department of Molecular Medical Microbiology, Chair of Microbiology, Faculty of Medicine, Jagiellonian University Medical College, Kraków, Poland

* bartosz.zielinski@uj.edu.pl

Abstract

Preliminary diagnosis of fungal infections can rely on microscopic examination. However, in many cases, it does not allow unambiguous identification of the species due to their visual similarity. Therefore, it is usually necessary to use additional biochemical tests. That involves additional costs and extends the identification process up to 10 days. Such a delay in the implementation of targeted therapy may be grave in consequence as the mortality rate for immunosuppressed patients is high. In this paper, we apply a machine learning approach based on deep neural networks and bag-of-words to classify microscopic images of various fungi species. Our approach makes the last stage of biochemical identification redundant, shortening the identification process by 2-3 days, and reducing the cost of the diagnosis.

Introduction

Yeast and yeast-like fungi are a component of natural human microbiota [1]. However, as opportunistic pathogens, they can cause surface and systemic infections [2]. The leading causes of the fungal infections are impaired function of the immune system and imbalanced microbiota composition in the human body. Other factors of fungal infections include steroid treatment, invasive medical procedures, and long-term antibiotic treatment with a broad spectrum of antimicrobial agents [3–5].

The standard procedure in mycological diagnostics begins with collecting various types of test materials like swabs, scraps of skin lesions, urine, blood, or cerebrospinal fluid. Next, the clinical materials (marked as B in Fig 1) are directly cultured on special media, while the blood and cerebrospinal fluid samples (marked as A in Fig 1) require prior cultivation in automated closed systems for additional 2-3 days. Material incubates under specific temperature conditions (usually for 2-4 days in case of yeast-like fungi). The initial identification of fungi bases on the assessment of the cells' shapes observed under the microscope as well as the growth rate, type, shape, color, and the smell of the colonies. Such analysis allows the assignment to fungi type; however, identification of the species is usually impossible due to the significant
specific roles of these authors are articulated in the “author contributions” section.

**Competing interests:** Two of the authors, Bartosz Zieliński (BZ) and Dawid Rymarczyk (DR), are employed by a commercial company Ardigen. Ardigen provided support in the form of salaries for authors BZ and DR. This does not alter our adherence to PLOS ONE policies on sharing data and materials.

In this paper, we apply a machine learning approach based on deep neural networks and bag-of-words approaches to classify microscopic images of various fungus species. As a result, the last stage of biochemical identification is unnecessary, which shortens the identification process by 2-3 days and reduces the cost of diagnosis. It allows accelerating the decision about the introduction of an appropriate antifungal drug, which prevents the progression of the disease and shortens the time of patient recovery.

According to our best knowledge, there are no other methods for classifying fungi species based only on microscopic images. Existing methods involve techniques such as morphological identification of a type of fungi [6], fluorescence in situ hybridization (FISH) [7], biochemical techniques, molecular approaches, such as PCR [8], and sequencing [9]. However, all of them are costly. On the other hand, our method bases on basic microbiological staining (Gram staining) and a simple microscope equipped with a camera, and takes only a few minutes, which makes it easily applicable in many laboratories.

The paper is structured as follows. First, we introduce a fungus database and describe a classification method based on deep neural networks and bag-of-words methods. Then, we present experimental setup, results, and conclusion.

**Materials and methods**

**Materials**

One of the most common fungal infections is candidiasis [5], mainly caused by *Candida albicans* (50-70% of cases) [10]. Other species responsible for the diseases are *Candida glabrata* [2, 3], *Candida tropicalis* [4], *Candida krusei* [11], and *Candida parapsilosis* [3, 4]. In high-risk patients, severe infections can also be caused by *Cryptococcus neoformans* [12] and *Saccharomyces phylum* [13]. Taking those facts into consideration, we prepared database, which consists of five yeast-like fungal strains: *Candida albicans* ATCC 10231 (CA), *Candida glabrata* ATCC 15545 (CG), *Candida tropicalis* ATCC 1369 (CT), *Candida parapsilosis* ATCC 34136 (CP), and *Candida lustiana* ATCC 42720 (CL); two yeast strains: *Saccharomyces cerevisiae* ATCC 4098 (SC) and *Saccharomyces boulardii* ATCC 74012 (SB); and two strains belonging to the Basidiomycetes: *Maalasezia furfur* ATCC 14521 (MF) and *Cryptococcus neoformans* ATCC 204092 (CN). All strains are from the American Type Culture Collection. The species
in our database highly overlap with the most common fungal infections; however, they are not that identical due to the limitations of our repository.

The strains were cultured on Sabouraud agar at 37°C for 48h (together with olive oil in the case of Malassezia furfur). After this time, microscopic preparations were made (2 preparations for each fungal strain) and stained with Gram method. Images were taken using an Olympus BX43 microscope with 100 times a super-apochromatic objective under oil-immersion. The photographic documentation was then produced with an Olympus BP74 camera and CellSense software (Olympus).

Altogether, our Digital Images of Fungus Species database (DIFaS) contains 180 images (9 strains × 2 preparations × 10 images) of resolution 3600 × 5760 × 3 with 16-bits intensity range in every pixel. In Fig 2, we present three random thumbnails for each of the registered strains.

**Method**

Deep Neural Networks (DNN) have shown human-level performance in case of large amounts of training data; however, they are limited when it comes to the application on small datasets due to the large numbers of parameters. Therefore, in this work, we consider two types of domain adaptation, both based on DNN features initially pre-trained on a different task (i.e., instance classification [14]). As a baseline method, we fine-tune the classifier’s block of the well-known network architectures, i.e., AlexNet [15], DenseNet169 [16], InceptionV3 [17], and ResNet [18] (with frozen features’ block). As we present in results, such architectures are
not optimal due to the small training set (see S1 and S3 Tables). Hence, we propose to apply the deep bag-of-words multi-step algorithm shown in Fig 3. In contrast to baseline methods, which utilize “shallow” Neural Network to previously calculated features, our strategies aggregate those features using one of the bag-of-words approaches and then classify them with Support Vector Machine (SVM). Such a policy, previously applied to texture recognition [19] and bacteria colony classification [20], is more accurate than the baseline methods; however, it is not well known. Therefore, to make this paper self-contained, below, we describe its successive steps.

To generate robust image representation, AlexNet [15], InceptionV3 [17] or ResNet [18] pre-trained on ImageNet [14] database are used. Another option would be to use conventional handcrafted descriptors (like ORB [21] or DSIFT [22]); however, they are usually outperformed by deep features. Considered network architectures consist of two parts: convolutional layers, which are responsible for extracting image features (so-called features’ block), and fully connected layers, which are responsible for the classification (so-called classifier’s block). Classifier’s block cannot be directly used because it was trained for other types of images; however, features’ block encodes more general, reusable information. Therefore, removing the classifier block from the network and preserving convolutional layers allows us to generate robust image features. In the case of AlexNet, we obtain a set of points in 256-dimensional space, whose number depends on the input image’s resolution (e.g., in case of resolution 500 × 500 pixels, 169 points (13 × 13) are generated).

Since the classified patches are always of the same size, their features’ blocks could be used directly by the classifier. It, however, would lead to vast data dimensionality (i.e., the feature vector of size 43264), which according to our experiments, results in the lack of generalization, primarily due to the relatively small size of the training set (100 images). Therefore, to obtain a more reliable representation of patches, we pool the acquired set of points using Bag of Words, BoW [23, 24], or its more expressive modification called Fisher Vector, FV [25]. The idea behind both of them is to aggregate a set of points (representing the patch) with a so-called codebook. The codebook is usually generated from the subset of training data in an unsupervised manner using a clustering algorithm (e.g., k-Means or Expectation Maximization [26]). Given a codebook, the set of 256-dimensional points obtained with AlexNet for a particular image is encoded by assigning points to the nearest codeword. In traditional Bag of Words, this encoding leads to a codeword histogram, i.e., a histogram for which each
codeword contains points closest to this codeword. In the case of the Fisher Vector, the clusters are replaced with a Gaussian Mixture Model (GMM), and the representation encodes the log-likelihood gradients with respect to the parameters of this model. In this paper, we will use notations deep Bag of Words and deep Fisher Vector to refer to those two types of pooling methods. To make this article self-contained, we recall definitions of BoW and FV in S1 Appendix.

As a result of pooling, one fixed-size vector is obtained for each of the analyzed patches, which can be classified with any machine learning methods to distinguish between various fungus species. We decided to use Support Vector Machine and Random Forest classifiers for this step.

**Experimental setup and results**

For the experiments, we split our DIFaS database (9 strains × 2 preparations × 10 images) into two subsets, so that both of them contain images of all strains, but from different preparation. It is because each preparation has its characteristics, and according to our previous studies [20], using images from the same preparation both in training and test set can result in overstated accuracy. As an example, let us consider the background-size, which depends on the size of the colony moved by inoculation loop from Sabouraud agar to preparation. Because there are only two preparations for each species in the dataset, the classifier could end up learning clinically irrelevant background-size instead of relevant fungus features. Therefore, images from particular preparation should not be shared between training and test set. We decided to use 2-fold cross-validation (one fold with 90 images from the first preparations and the second fold with 90 images from the second preparations). Moreover, we decided to classify patches instead of the whole image (see Image preprocessing for details) and introduce additional class corresponding to the background (BG) to compensate for the preparation characteristic on the final result. Nevertheless, we report accuracy for both patch-level and scan-level classification (the latter with majority voting).

For each fold, we optimize the following parameters using internal 5-fold cross-validation: number of clusters in BoW ∈ [5, 10, 20, 50, 100, 200, 500]; number of clusters in FV ∈ [5, 10, 20, 50]; SVM kernel ∈ [linear, RBF]; SVM $C$ ∈ [1, 10, 100, 1000]; SVM $\gamma$ ∈ [0.001, 0.0001]. As the evaluation metric for grid search optimization, we use the accuracy classification score. Best results were obtained for FV with 10 clusters and SVM with RBF kernel, $C = 1$, and $\gamma = 0.0001$.

We performed all the experiments on a workstation with one 12 GB GPU and 256 GB RAM. On average, feature extraction, pooling, and classification take from 1 to 2 hours when training deep Fisher Vector. Such performance was possible thanks to the adaptation of the VLFeat library [27]. For comparison, the fine-tuning of the well-known architectures takes from 70 to 85 hours (see Table 1). Processing time in case of baseline methods was measured by multiplying the average time of an epoch by the number of epochs till the early stopping (i.e., the increase in validation loss). In the case of deep bag-of-words approaches, processing time was computed as a sum of all three steps of the algorithm (i.e., obtaining image representation, pooling, and classification).

The remaining part of this section is structured as follows. First, we describe image preprocessing, including contrast stretching and background removal. Then, we describe the results obtained for patch-based classification using deep bag-of-words approaches and compare them with the well-known network architectures. To explain the outcomes of deep BoW, we introduce an in-depth explanatory analysis of the obtained codebooks together with the microbiological feedback. We continue this investigation for a deep FV approach. Finally, we present
Table 1. Test accuracy of patch-based classification averaged over two runs (for two subsets described in Experimental setup).

<table>
<thead>
<tr>
<th>Method</th>
<th>CA</th>
<th>CG</th>
<th>CL</th>
<th>CN</th>
<th>CP</th>
<th>CT</th>
<th>MF</th>
<th>SB</th>
<th>SC</th>
<th>BG</th>
<th>Total</th>
<th>Training time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>78.6 ± 4.3</td>
<td>80.0 ± 1.4</td>
<td>55.8 ± 1.4</td>
<td>63.4 ± 14.7</td>
<td>75.0 ± 7.9</td>
<td>75.0 ± 6.4</td>
<td>71.6 ± 16.6</td>
<td>67.9 ± 5.0</td>
<td>72.1 ± 2.1</td>
<td>90.0 ± 0.1</td>
<td>71.6 ± 2.4</td>
<td>250600</td>
</tr>
<tr>
<td>DenseNet169</td>
<td>67.9 ± 17.9</td>
<td>72.1 ± 0.7</td>
<td>53.6 ± 0.7</td>
<td>56.3 ± 13.3</td>
<td>60.0 ± 12.7</td>
<td>81.4 ± 0.7</td>
<td>68.7 ± 5.6</td>
<td>85.0 ± 5.0</td>
<td>68.8 ± 8.6</td>
<td>81.2 ± 2.0</td>
<td>72.9 ± 0.6</td>
<td>271600</td>
</tr>
<tr>
<td>InceptionV3</td>
<td>67.3 ± 1.5</td>
<td>55.0 ± 2.2</td>
<td>59.3 ± 6.4</td>
<td>67.0 ± 18.2</td>
<td>64.3 ± 1.4</td>
<td>81.4 ± 2.9</td>
<td>61.1 ± 11.1</td>
<td>55.0 ± 0.7</td>
<td>89.3 ± 1.3</td>
<td>84.7 ± 3.1</td>
<td>69.9 ± 1.9</td>
<td>309400</td>
</tr>
<tr>
<td>ResNet18</td>
<td>91.4 ± 3.6</td>
<td>67.1 ± 5.7</td>
<td>67.1 ± 5.7</td>
<td>61.9 ± 7.5</td>
<td>65.0 ± 0.7</td>
<td>69.8 ± 2.8</td>
<td>54.8 ± 12.8</td>
<td>93.6 ± 3.2</td>
<td>93.5 ± 1.2</td>
<td>93.1 ± 0.5</td>
<td>75.9 ± 2.6</td>
<td>263900</td>
</tr>
<tr>
<td>ResNet50</td>
<td>86.4 ± 3.5</td>
<td>57.9 ± 2.1</td>
<td>89.3 ± 3.5</td>
<td>61.8 ± 11.5</td>
<td>60.0 ± 7.1</td>
<td>64.5 ± 6.0</td>
<td>66.4 ± 19.3</td>
<td>79.3 ± 0.7</td>
<td>64.3 ± 12.4</td>
<td>90.4 ± 2.7</td>
<td>73.9 ± 2.6</td>
<td>276500</td>
</tr>
<tr>
<td>AlexNet BoW RF</td>
<td>87.3 ± 9.7</td>
<td>39.0 ± 0.3</td>
<td>88.3 ± 7.0</td>
<td>64.0 ± 17.0</td>
<td>79.0 ± 11.0</td>
<td>79.3 ± 10.7</td>
<td>55.1 ± 3.3</td>
<td>93.3 ± 2.0</td>
<td>81.3 ± 6.0</td>
<td>92.6 ± 1.0</td>
<td>76.7 ± 1.0</td>
<td>156</td>
</tr>
<tr>
<td>InceptionV3 BoW RF</td>
<td>44.3 ± 7.7</td>
<td>51.0 ± 13.7</td>
<td>60.3 ± 0.3</td>
<td>34.2 ± 17.8</td>
<td>28.0 ± 0.7</td>
<td>40.7 ± 11.3</td>
<td>13.3 ± 1.4</td>
<td>39.7 ± 11.7</td>
<td>36.3 ± 8.3</td>
<td>83.5 ± 2.0</td>
<td>44.6 ± 0.3</td>
<td>155</td>
</tr>
<tr>
<td>ResNet18 BoW RF</td>
<td>54.3 ± 2.3</td>
<td>39.7 ± 8.3</td>
<td>80.7 ± 12.7</td>
<td>46.2 ± 20.1</td>
<td>72.0 ± 3.3</td>
<td>61.3 ± 11.3</td>
<td>42.1 ± 2.9</td>
<td>58.7 ± 8.7</td>
<td>68.0 ± 4.0</td>
<td>91.6 ± 2.0</td>
<td>62.7 ± 2.4</td>
<td>158</td>
</tr>
<tr>
<td>AlexNet BoW SVM</td>
<td>92.3 ± 3.0</td>
<td>44.3 ± 18.3</td>
<td>88.7 ± 9.3</td>
<td>62.2 ± 22.2</td>
<td>83.0 ± 5.7</td>
<td>71.0 ± 15.0</td>
<td>76.1 ± 11.8</td>
<td>85.0 ± 4.3</td>
<td>76.3 ± 7.0</td>
<td>89.5 ± 3.6</td>
<td>77.6 ± 1.2</td>
<td>5124</td>
</tr>
<tr>
<td>InceptionV3 BoW SVM</td>
<td>44.3 ± 10.3</td>
<td>32.3 ± 5.7</td>
<td>57.7 ± 2.3</td>
<td>33.3 ± 17.6</td>
<td>28.3 ± 1.0</td>
<td>39.7 ± 6.3</td>
<td>13.0 ± 1.7</td>
<td>34.7 ± 8.0</td>
<td>32.0 ± 2.0</td>
<td>78.8 ± 2.1</td>
<td>40.8 ± 1.3</td>
<td>5153</td>
</tr>
<tr>
<td>ResNet18 BoW SVM</td>
<td>59.0 ± 3.7</td>
<td>32.3 ± 5.7</td>
<td>84.0 ± 11.3</td>
<td>54.7 ± 22.0</td>
<td>66.0 ± 0.0</td>
<td>62.0 ± 4.0</td>
<td>39.7 ± 1.0</td>
<td>59.3 ± 2.0</td>
<td>73.0 ± 12.3</td>
<td>92.9 ± 1.8</td>
<td>63.4 ± 0.7</td>
<td>5195</td>
</tr>
<tr>
<td>AlexNet FV RF</td>
<td>83.3 ± 10.7</td>
<td>54.3 ± 31.7</td>
<td>81.7 ± 0.3</td>
<td>49.3 ± 32.0</td>
<td>78.0 ± 12.0</td>
<td>73.0 ± 15.7</td>
<td>76.5 ± 5.8</td>
<td>89.3 ± 2.0</td>
<td>74.7 ± 6.7</td>
<td>88.0 ± 0.9</td>
<td>75.8 ± 0.4</td>
<td>166</td>
</tr>
<tr>
<td>InceptionV3 FV RF</td>
<td>40.7 ± 9.3</td>
<td>53.3 ± 11.3</td>
<td>60.3 ± 5.0</td>
<td>37.3 ± 21.7</td>
<td>27.7 ± 6.3</td>
<td>47.7 ± 13.0</td>
<td>16.5 ± 0.2</td>
<td>35.3 ± 13.3</td>
<td>32.0 ± 10.7</td>
<td>81.4 ± 4.1</td>
<td>44.6 ± 0.7</td>
<td>168</td>
</tr>
<tr>
<td>ResNet18 FV RF</td>
<td>63.7 ± 2.3</td>
<td>37.0 ± 5.7</td>
<td>78.7 ± 12.7</td>
<td>52.5 ± 21.1</td>
<td>73.3 ± 4.0</td>
<td>64.3 ± 5.0</td>
<td>61.8 ± 6.8</td>
<td>61.0 ± 9.7</td>
<td>69.3 ± 7.3</td>
<td>92.7 ± 2.0</td>
<td>66.4 ± 2.1</td>
<td>167</td>
</tr>
<tr>
<td>AlexNet FV SVM</td>
<td>93.7 ± 2.3</td>
<td>53.7 ± 16.3</td>
<td>90.7 ± 4.0</td>
<td>59.6 ± 15.2</td>
<td>77.7 ± 14.3</td>
<td>87.7 ± 9.7</td>
<td>82.8 ± 6.9</td>
<td>97.3 ± 1.3</td>
<td>81.3 ± 10.0</td>
<td>91.1 ± 2.5</td>
<td>82.4 ± 0.2</td>
<td>1541</td>
</tr>
<tr>
<td>InceptionV3 FV SVM</td>
<td>46.0 ± 14.0</td>
<td>45.0 ± 20.3</td>
<td>58.3 ± 3.0</td>
<td>42.2 ± 20.4</td>
<td>24.0 ± 5.3</td>
<td>43.7 ± 3.7</td>
<td>13.0 ± 1.1</td>
<td>26.7 ± 5.0</td>
<td>76.7 ± 3.7</td>
<td>41.3 ± 1.9</td>
<td>41.3 ± 1.9</td>
<td>1511</td>
</tr>
<tr>
<td>ResNet18 FV SVM</td>
<td>71.3 ± 11.3</td>
<td>35.3 ± 1.3</td>
<td>59.3 ± 10.3</td>
<td>51.6 ± 31.1</td>
<td>77.0 ± 8.3</td>
<td>76.7 ± 4.7</td>
<td>57.5 ± 0.5</td>
<td>73.3 ± 4.7</td>
<td>77.7 ± 9.7</td>
<td>94.5 ± 1.3</td>
<td>71.3 ± 1.5</td>
<td>1535</td>
</tr>
</tbody>
</table>

https://doi.org/10.1371/journal.pone.0234806.t001

results obtained for scan-based classification, computed by aggregating patch-based scores. The code implemented in Python with PyTorch library is available at https://github.com/bziuij/fungus.

Image preprocessing

DIFaS database contains 180 images of relatively high resolution and intensity range (from 0 to 65535); however, the actual pixel values are usually between 0 and 1000 (see Fig 4a). Therefore, in the first step of preprocessing, we compute the lower and upper-intensity limits (separately for every image) and use them for contrast stretching (see Fig 4b). Moreover, images are scaled to the range [0, 1].

To overcome the issues with preparation characteristic (e.g., background-size), as the second step of preprocessing, we extract and classify only image patches with the reasonable foreground to background proportions (FBP), so patches with a rational number of foreground pixels. To obtain foreground-background segmentation on the pixel level, we apply thresholding (with threshold equal 0.5) to a grayscale and blurred version of the scanned image. Such a simple segmentation is sufficient and works for all the images from the dataset (see S1 and S2 Figs) because the background is always much brighter than the areas with fungi cells. We tested three possible options of FBP: 1: 2, 1: 1, and 2: 1 (see Fig 4d–4f). Based on empirical
studies, we decided to use FBP equal 2:1, which gains around 1.5% comparing to the other options. As a result, we obtain rough segmentations with approximated locations of foreground patches (those with FBP greater than 2:1) and background patches (those with FBP smaller than 1:100). Additionally, we experimented with two image scales: the original images and images scaled by factor 0.5 (with bicubic interpolation), concluding that the latter gains around 4% comparing to the former.

**Patch-based classification**

In this experiment, we use baseline models (well-known network architectures) as well as deep Bag of Words and deep Fisher Vector models to classify each patch of the image separately. As baseline models, we fine-tune the classifier’s block of the well-known network architectures, such as AlexNet [15], DenseNet169 [16], InceptionV3 [17], and ResNet [18] for 100 epochs (with frozen features' block). Every baseline model was previously pre-trained on the ImageNet database [14]. Before running all the experiments, we experimentally chose the optimal FBP (2:1), patch size (500 × 500 pixels), and image scale (0.5) using grid search optimization. The number of foreground patches overlapped by less than 50% oscillates between 2000 and 3000, depending on the fold. Moreover, the number of patches significantly varies depending on the strains (see S1 and S3 Tables). Therefore, we apply data augmentation (rotations, mirror reflection, and random noise) for better regularization.

The overall comparison of tested methods is presented in Table 1. One can observe that deep Fisher Vector works better than all the other techniques, including deep Bag of Words. However, its accuracy drops dramatically in the case of *Candida glabrata* (CG) and *Cryptococcus neoformans* (CN). In the case of CN it is most probably caused by a reduced number of samples, while in the case of CG due to its more substantial variance in the arrangement, appearance, and quantity (especially between two preparations, see S1 and S2 Figs). Moreover, CG images are hard to classify due to partial discoloration (pink color instead of purple) and vast overlapping of cells. As a result, CG is often classified as *Candida lustianiae* (CL).
belonging to the same genus (see confusion matrix in Fig 5b). However, the classification error should decrease if the biological material of microscopic preparation has the smallest possible density with separated cells, as overlapping is the leading cause of blurriness.

To further understand the reason for incorrect classification, we prepare a qualitative confusion matrix for deep Fisher Vector to show examples of correctly and incorrectly classified patches (see Fig 6). We observe a high morphological similarity between misclassified species belonging to genus Candida, Cryptococcus, and Saccharomyces, especially if the preparation

![Confusion Matrix](https://doi.org/10.1371/journal.pone.0234806.g005)

**Fig 5. Test confusion matrices.** Normalized test confusion matrices for the best baseline (ResNet18) and the best deep bag-of-words approach (deep Fisher Vector).

![Qualitative Confusion Matrix](https://doi.org/10.1371/journal.pone.0234806.g006)

**Fig 6. The qualitative confusion matrix for deep Fisher Vector (with AlexNet and SVM).** Each cell contains at most two patches of a "true" strain (represented by rows) that were classified as "predicted" strain (given by columns). Notice that the patches highly overlap, and they are often misclassified due to the artifacts in the background (such as purple trail).
with the biological material is discolored. Moreover, one can notice that deep Fisher Vector can return two different results for two highly overlapped patches from the same scan. It is usually caused by the artifacts in the background, such as purple trail in *Candida lusitaniae* (CL), predicted as *Cryptococcus neoformans* (CN), or *Malassezia furfur* (MF) in Fig 6. The other incorrect classifications appear due to the small number of incomplete (fragmented) cells (like *Candida glabrata* (CG) predicted as CN).

**Analysis of deep Bag of Words clusters**

In this section, we first analyze deep Bag of Words pooling step by visualizing clusters using the patches nearest to their centroids. Then, based on those patches, we introduce a description of the considered species using properties pre-defined by the microbiologists. Finally, we present the mean deep BoW for every species. To make our analysis clearer, in this section, we limit deep BoW to 10 clusters, although its optimal number obtained with grid search optimization is 50. Moreover, the presented properties are introduced only to explain the intrinsic rules of the method. They are not used in the automatic classification, which requires only a scan image as an input.

Ten nearest neighbors of ten deep BoW centroids obtained with k-Means algorithm are presented in Fig 7. One can observe that they share common features and, therefore, can be

<table>
<thead>
<tr>
<th>Cluster No.</th>
<th>Closest patches</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>1</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>3</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>7</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>8</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
<tr>
<td>9</td>
<td><img src="https://doi.org/10.1371/journal.pone.0234806.g007" alt="Image" /></td>
</tr>
</tbody>
</table>
used to determine which visual properties are essential for the classifier. We consider the following properties (see Table 2): brightness (dark or bright), size (small, medium or large), shape (circular, oval, longitudinal or variform), arrangement (regular or irregular), appearance (singular, grouped or fragmentary), color (pink, purple, blue or black), and quantity (low, medium or high). As a result, the standard set of parameters used to describe the species (size, shape, arrangement, and appearance) was significantly extended.

To investigate which visual properties are essential for the classifier, we calculate mean deep Bag of Words representation for every species (see Fig 8) and then examine how the visual information about their main clusters corresponds to the knowledge of a microbiologist. The main conclusions she drew are as follows:

• species of the genus *Candida* mainly belong to cluster 2 with black cells of medium or large size, and oval or longitudinal shape;

• *Maalasezia furfur* has been assigned to clusters 0, 2, 5 and 8, mostly representing the black and longitudinal shape of various size;

• *Saccharomyces boulardii* and *Saccharomyces cerevisiae* are mainly described by clusters 1, 2, 4 and 8, which are characterized by black color, medium or large size and longitudinal shape;

• *Candida tropicalis* and *Saccharomyces cerevisiae* have very similar mean Bag of Words, which confirms high morphological similarity described in [28], i.e., size 3.0-8.0 × 5.0-10 μm, oval shape, elongated, and occurring singly or in small groups).

### Analysis of deep Fisher Vector and SVM classifier

In this section, we first analyze the power of deep Fisher Vector representation using the t-SNE algorithm [29] by projecting it on a 2D surface. Then, we analyze classifier certainty based on the scores obtained for various patches.
Projection of high-dimensional deep Fisher Vector to 2D using the t-SNE algorithm is presented in Fig 9. One can observe that classes are generally well separated in the case of AlexNet and ResNet18 architectures. Nevertheless, species of the same genus are not more coherent than the other species, in contrast to what we expected. Moreover, one can observe that InceptionV3 fails domain adaptation in the case of microbiological images, which explains the results in Table 1. Direct explanation of this behavior is hard, due to the neural network’s black-box character. However, it is consistent with the results reported in [30–32], which suggest that ResNet representations are more robust than those obtained from InceptionV3.
The second task in this section was to analyze classifier certainty. For this purpose, we investigate the distance of patches’ representations from the classifier hyperplane, which roughly corresponds to how sure the classifier is of its decisions. Most left and right patches in Fig 10 are correctly classified with high probability, while the ones in the middle are ambiguous. The most representative fungal *Malassezia furfur* (MF) cells have oval, longitudinal shape, and often occur in the budding form, in which the daughter cells are as wide as the parent cells. While in the case of *Saccharomyces cerevisae* (SC), fungal cells characterize with round shapes, more significant in relation to *Candida albicans* (CA), which are arranged individually or in small groups.

![TSNE visualization of fungus representation from Fisher Vector with representation from different Ns](https://doi.org/10.1371/journal.pone.0234806.g009)

**Fig 9.** Projection of high-dimensional deep Fisher Vector to 2D using the t-SNE algorithm.

![Classifier certainty for deep Fisher Vector](https://doi.org/10.1371/journal.pone.0234806.g010)

**Fig 10.** Classifier certainty for deep Fisher Vector. From left to right, one can observe the most negative and most positive patches according to the classifier of the particular species.
Scan-based classification

To analyze classification score for the whole scan (instead of just patches, like in previous sections), we predict classification for all foreground patches of one scan and aggregate them to obtain the most frequently predicted species. As presented in Table 3, deep Fisher Vector performs better than the other methods, also in this case, obtaining 15.6% better accuracy than the best baseline method (ResNet18).

Conclusions and future work

In this paper, we apply deep neural networks and bag-of-words approaches to classify microscopic images of various fungi species. According to our experiments, the combination of features from deep neural networks with Fisher Vector works better than fine-tuning the classifier’s block of the well-known network architectures and has the potential to be successfully used by microbiologists in their daily practice.

A large part of this paper is dedicated to the explainability of deep bag-of-words approaches to increase the trust in deep neural networks. For this purpose, we introduce an in-depth visual description of the properties pre-defined by the microbiologists. We hope that it will help to understand similarities and differences between fungi species better.

In our experiment, we assumed that images are obtained from the same laboratory and with the same scanner (details are presented in the Materials). However, in our opinion, this method could be easily extended to more diversified datasets by using additional preprocessing steps, which unify the input data. Due to the lack of data for such experiments, we did not cover this issue in the current article; however, it is planned for future research. Moreover, we would like to extend the DIFaS database so that it contains more preparations for all species, also gathered from other laboratories and scanners. Finally, we plan to prepare scans containing more than one species, as the automatic classification of such images would help to exclude the culture phase from the microbiological pipeline.

Table 3. Test accuracy of scan-based classification obtained by aggregating patch-based classification and determining the most frequent prediction.

<table>
<thead>
<tr>
<th>Method</th>
<th>CA</th>
<th>CG</th>
<th>CL</th>
<th>CN</th>
<th>CP</th>
<th>CT</th>
<th>MF</th>
<th>SB</th>
<th>SC</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>95.5 ± 5.0</td>
<td>95.0 ± 5.0</td>
<td>70.0 ± 10.0</td>
<td>57.0 ± 6.9</td>
<td>95.0 ± 5.0</td>
<td>45.0 ± 10.0</td>
<td>70.0 ± 10.0</td>
<td>75.0 ± 5.0</td>
<td>85.0 ± 5.0</td>
<td>77.3 ± 4.2</td>
</tr>
<tr>
<td>DenseNet169</td>
<td>80.0 ± 10.0</td>
<td>85.0 ± 5.0</td>
<td>45.0 ± 5.0</td>
<td>57.0 ± 21.2</td>
<td>70.0 ± 10.0</td>
<td>100.0 ± 0.0</td>
<td>85.0 ± 15.0</td>
<td>95.0 ± 5.0</td>
<td>75.0 ± 10.0</td>
<td>77.6 ± 6.6</td>
</tr>
<tr>
<td>InceptionV3</td>
<td>50.0 ± 10.0</td>
<td>50.0 ± 10.0</td>
<td>80.0 ± 0.0</td>
<td>78.6 ± 14.3</td>
<td>55.0 ± 10.0</td>
<td>60.0 ± 10.0</td>
<td>50.0 ± 10.0</td>
<td>85.0 ± 5.0</td>
<td>85.0 ± 5.0</td>
<td>65.9 ± 4.9</td>
</tr>
<tr>
<td>ResNet18</td>
<td>100.0 ± 0.0</td>
<td>75.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>78.6 ± 6.9</td>
<td>50.0 ± 10.0</td>
<td>70.0 ± 0.0</td>
<td>70.0 ± 10.0</td>
<td>95.0 ± 5.0</td>
<td>80.0 ± 10.0</td>
<td>78.3 ± 5.4</td>
</tr>
<tr>
<td>ResNet50</td>
<td>100.0 ± 0.0</td>
<td>85.0 ± 15.0</td>
<td>100.0 ± 0.0</td>
<td>57.0 ± 6.9</td>
<td>50.0 ± 10.0</td>
<td>45.0 ± 15.0</td>
<td>80.0 ± 10.0</td>
<td>95.0 ± 5.0</td>
<td>75.0 ± 5.0</td>
<td>78.1 ± 8.3</td>
</tr>
<tr>
<td>AlexNet BoW RF</td>
<td>100.0 ± 0.0</td>
<td>75.0 ± 25.0</td>
<td>100.0 ± 0.0</td>
<td>75.0 ± 15.0</td>
<td>100.0 ± 0.0</td>
<td>90.0 ± 10.0</td>
<td>90.0 ± 10.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>92.2 ± 4.4</td>
</tr>
<tr>
<td>InceptionV3 BoW RF</td>
<td>80.0 ± 20.0</td>
<td>75.0 ± 25.0</td>
<td>100.0 ± 0.0</td>
<td>45.0 ± 15.0</td>
<td>40.0 ± 0.0</td>
<td>60.0 ± 30.0</td>
<td>0.0 ± 0.0</td>
<td>70.0 ± 10.0</td>
<td>45.0 ± 15.0</td>
<td>57.2 ± 2.8</td>
</tr>
<tr>
<td>ResNet18 BoW RF</td>
<td>100.0 ± 0.0</td>
<td>70.0 ± 10.0</td>
<td>100.0 ± 0.0</td>
<td>45.0 ± 15.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>80.0 ± 10.0</td>
<td>90.0 ± 10.0</td>
<td>100.0 ± 0.0</td>
<td>87.2 ± 1.7</td>
</tr>
<tr>
<td>AlexNet BoW SVM</td>
<td>100.0 ± 0.0</td>
<td>65.0 ± 25.0</td>
<td>100.0 ± 0.0</td>
<td>70.0 ± 10.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>85.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>91.1 ± 2.2</td>
</tr>
<tr>
<td>InceptionV3 BoW SVM</td>
<td>70.0 ± 10.0</td>
<td>55.0 ± 15.0</td>
<td>100.0 ± 0.0</td>
<td>40.0 ± 20.0</td>
<td>45.0 ± 5.0</td>
<td>55.0 ± 15.0</td>
<td>0.0 ± 0.0</td>
<td>55.0 ± 15.0</td>
<td>40.0 ± 20.0</td>
<td>51.1 ± 2.3</td>
</tr>
<tr>
<td>ResNet18 BoW SVM</td>
<td>100.0 ± 0.0</td>
<td>60.0 ± 20.0</td>
<td>100.0 ± 0.0</td>
<td>65.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>90.0 ± 10.0</td>
<td>60.0 ± 0.0</td>
<td>95.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>85.6 ± 2.2</td>
</tr>
<tr>
<td>AlexNet FV RF</td>
<td>100.0 ± 0.0</td>
<td>65.0 ± 35.0</td>
<td>100.0 ± 0.0</td>
<td>55.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>90.0 ± 10.0</td>
<td>95.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>89.4 ± 2.2</td>
</tr>
<tr>
<td>InceptionV3 FV RF</td>
<td>65.0 ± 5.0</td>
<td>95.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>50.0 ± 10.0</td>
<td>30.0 ± 10.0</td>
<td>75.0 ± 25.0</td>
<td>5.0 ± 5.0</td>
<td>45.0 ± 25.0</td>
<td>45.0 ± 35.0</td>
<td>56.7 ± 3.3</td>
</tr>
<tr>
<td>ResNet18 FV RF</td>
<td>95.0 ± 5.0</td>
<td>60.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>65.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>95.0 ± 5.0</td>
<td>95.0 ± 5.0</td>
<td>95.0 ± 5.0</td>
<td>89.4 ± 1.7</td>
</tr>
<tr>
<td>AlexNet FV SVM</td>
<td>100.0 ± 0.0</td>
<td>75.0 ± 25.0</td>
<td>100.0 ± 0.0</td>
<td>75.0 ± 15.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>95.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>93.9 ± 3.9</td>
</tr>
<tr>
<td>InceptionV3 FV SVM</td>
<td>75.0 ± 25.0</td>
<td>60.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>55.0 ± 5.0</td>
<td>45.0 ± 15.0</td>
<td>85.0 ± 5.0</td>
<td>5.0 ± 5.0</td>
<td>25.0 ± 15.0</td>
<td>45.0 ± 25.0</td>
<td>55.0 ± 5.6</td>
</tr>
<tr>
<td>ResNet18 FV SVM</td>
<td>100.0 ± 0.0</td>
<td>60.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>45.0 ± 15.0</td>
<td>95.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>95.0 ± 5.0</td>
<td>100.0 ± 0.0</td>
<td>100.0 ± 0.0</td>
<td>88.3 ± 2.7</td>
</tr>
</tbody>
</table>

https://doi.org/10.1371/journal.pone.0234806.t003
Supporting information

S1 Appendix. (PDF)

S2 Appendix. (PDF)

S1 Table. The number of foreground patches overlapped by less than 50% for the images from S1 Fig. (PDF)

S2 Table. The number of background patches overlapped by less than 50% for the images from S1 Fig. (PDF)

S3 Table. The number of foreground patches overlapped by less than 50% for the images from S2 Fig. (PDF)

S4 Table. The number of background patches overlapped by less than 50% for the images from S2 Fig. (PDF)

S1 Fig. Images of the first preparations from the DIFaS database and their foreground-background masks. (TIF)

S2 Fig. Images of the second preparations from the DIFaS database and their foreground-background masks. (TIF)

Author Contributions

Conceptualization: Bartosz Zieliński, Agnieszka Sroka-Oleksiak, Monika Brzychczy-Włoch.

Data curation: Agnieszka Sroka-Oleksiak.

Formal analysis: Bartosz Zieliński.

Funding acquisition: Bartosz Zieliński.

Investigation: Bartosz Zieliński, Agnieszka Sroka-Oleksiak, Dawid Rymarczyk.

Methodology: Bartosz Zieliński, Dawid Rymarczyk, Adam Piekarczyk, Monika Brzychczy-Włoch.

Project administration: Bartosz Zieliński.

Resources: Bartosz Zieliński, Dawid Rymarczyk.

Software: Dawid Rymarczyk, Adam Piekarczyk.

Supervision: Bartosz Zieliński, Monika Brzychczy-Włoch.

Validation: Bartosz Zieliński, Agnieszka Sroka-Oleksiak, Dawid Rymarczyk, Adam Piekarczyk.

Visualization: Dawid Rymarczyk, Adam Piekarczyk.
Writing – original draft: Bartosz Zieliński, Agnieszka Sroka-Oleksiak, Dawid Rymarczyk, Adam Piekarczyk, Monika Brzychczy-Włoch.

References

PLOS ONE | https://doi.org/10.1371/journal.pone.0234806 June 30, 2020 15 / 16


