Exploring Transfer Learning Approaches for Direct Mycological Examination of Microscopic Fungi Images

John Michael T. Elbambo
College of Computing and Information Technologies
National University
Manila, Philippines
jm.elbambo27@gmail.com

Abstract

Fungal infections such as superficial mycoses are widespread and pose a global threat. Current diagnostic methods such as direct microscopic examination have limitations in terms of time and reliability. Meanwhile, transfer learning has proven successful in computer vision tasks but faces challenges when applied to mycological examination due to the unique characteristics of microscopic fungi. This research investigates transfer learning approaches using a pre-trained EfficientNet V2 model to classify microscopic fungi images. The models created were able to generalize and adapt the learned features to the microscopic images used in this study. One of the model in the experiments achieved an accuracy score of 85.48% by just adding a few extra layers from the pre-trained model.

1 Introduction

Among the most common fungal infections are superficial mycoses of the skin, nails, and hair. Dermatophytes, non-dermatophyte moulds, yeasts, and yeast-like fungi cause these infections. Such fungal infections are common around the world, with tropical and subtropical regions being the most susceptible. Factors in the environment such as warmth, humidity, and poor sanitary conditions stimulate their growth and reproduction (Sharma and Nonzom, 2021). Over 150 million severe cases of fungal infections occur each year worldwide, resulting in nearly 1.7 million fatalities. These figures are rapidly increasing, due to a number of societal and medical advances in recent decades that have aided the spread of fungal diseases. Furthermore, the long-term therapeutic and preventative use of anti-fungal medications in high-risk individuals has aided in the creation of multi drug-resistant fungi. As a result, fungal infections are already a global threat that is becoming more prevalent (Kainz et al., 2020). Direct microscopic examination with potassium hydroxide is commonly used to diagnose superficial fungal infections. Although this kind of examination is faster than other diagnostic approaches, evaluating a whole sample can still take time. Additionally, it has the disadvantage of inconsistent reliability since the accuracy of the reading may vary depending on the examiner's experience (Koo et al., 2021).

In recent years, transfer learning has demonstrated remarkable success in various computer vision applications, including object recognition, image segmentation, and image classification. Transfer learning has gained significant attention in the field of computer vision due to its ability to leverage pre-trained models on large-scale datasets and adapt them to new domains with limited labeled data (Jayram et al., 2019). However, the direct application of transfer learning techniques to the domain of mycological examination of microscopic fungi images presents unique challenges. Fungal morphology and microscopic characteristics differ significantly from common objects and scenes found in general image datasets. To address these challenges, this research paper investigates various transfer learning approaches and methodologies for direct mycological examination of microscopic fungi images. Particularly, this study will explore a pre-trained model with an EfficientNet V2 architecture and assess its performance in classifying different types of microscopic fungi images.

2 Related Works

This paper from (Sopo et al., 2021), where the dataset in this study was obtained from, presents experimental results classifying five fungi types using three different convolutional neural network models, VGG16, Inception V3, and ResNet50. The study applied transfer learning using pre-trained models based on the ImageNet dataset. The study demonstrated that deep learning convolutional neural network (CNN) models using transfer learning to classify novel target class images can achieve state-of-the-art accuracy performance. The models achieved an average loss of 0.877 for ResNet50, 0.5947 for VGG16, and 0.6949 for Inception V3. For accuracy, the models achieved an average of 85.040%, 83.040%, and 82.800% for ResNet50, VGG16, and Inception V3 respectively.

(Tan and Le, 2020) explored model scaling and discovered that carefully balancing network depth, width, and resolution can lead to improved performance. Based on these findings, they proposed a new scaling method that employs a simple yet highly effective compound coefficient to uniformly scale all depth/width/resolution dimensions. The effectiveness of this method was proved by the researchers on MobileNets and ResNet. The researchers also employed neural architecture search to create a new baseline network and scale it up to create the EfficientNets family of models, which outperform prior ConvNets in accuracy and efficiency. The EfficientNet-B7, for instance, achieves state-of-the-art ImageNet accuracy of 84.4% top-1 / 97.1% top-5 while being 8.4x smaller and 6.1x faster on inference than the best existing ConvNet (Huang et al., 2018). With a large scale fewer parameters, the EfficientNet models transfer well and attain state-of-the-art accuracy on CIFAR-100 (91.7%), Flower (98.8%), and three other transfer learning datasets.

(Shoeibi et al., 2022) explored deep learning techniques for the diagnosis of myocarditis from cardiac magnetic resonance (CMR) images. Their paper presents a new method to detect myocarditis in CMR images by transfer learning using pre-trained models. The Z-Alizadeh Sani myocarditis dataset was used for the simulations, which included CMR images of normal subjects and myocardial infarction patients. To classify the input data, pre-trained models such as the EfficientNet B3, EfficientNet V2, HrNet, ResNetrs50, ResNest50d, and ResNet 50d were utilized. The EfficientNet V2 model achieved the highest performance with a 99.33% accuracy among all other pre-trained models used in their study.

3 Methodology

3.1 Dataset

The dataset used in this study is the DeFungi dataset (Sopo et al., 2021). DeFungi is a dataset for direct mycological examination of microscopic fungi images. The images are from superficial fungal infections caused by yeasts, moulds, or dermatophyte fungi. There are 9114 instances of images that have been manually labelled into five classes and curated with a subject matter expert assistance. The images have been cropped with automated algorithms to produce the final dataset which has been used in this study. A train, validation, and test split was performed on the dataset with the ratios 70%, 15%, and 15% accordingly. The resulting dataset contains 6377 instances for train, 1364 instances for validation, and 1373 instances for the test dataset. The images are also resized to 240x240 width by height which is the optimal arbitrary size since the pre-trained model to be used on the transfer learning approach was trained on this size.

3.2 Models

There are 3 models created for this study. These models take advantage of implementing a transfer learning approach using a pre-trained model with an EfficientNet V2 architecture which is shown on table 1.

Stage	Operator	Stride	Channels	Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	2	24	2
2	Fused-MBConv1, k3x3	1	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	2	160	9
6	MBConv6, k3x3, SE0.25	1	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

Table 1: EfficientNet V2 Architecture

The specific model used in this study is the `efficientnet_v2_imagenet21k_b1` obtained from TensorFlow Hub[1]. This pre-trained model contains feature vectors of images with input size 240x240, trained on imagenet-21k (Full ImageNet, Fall 2011 release). This pre-trained model will then be used later as a feature extractor layer.

All models created have a preprocessing rescaling layer as its first layer. In this layer, the image data are rescaled to be in the range of 0-1 for numerical stability during computations and to be optimized for the activation functions which will later be used. The second layer is the feature extractor layer previously created from the pre-trained model where its output will then be flattened before being sent to the next layer.

In the first model, the output has been immediately fed to a densely-connected neural network

[1] https://tfhub.dev/google/imagenet/efficientnet_v2_imagenet21k_b1/feature_vector/2

Layer	Output Shape	Params
Rescaling	(240, 240, 3)	0
EffNet V2 (Pre-Trained)	1280	6931124
Flatten	1280	0
Dense	5	6405

Table 2: Model 1 Architecture

layer which serves as the classification head. The details of each layer is illustrated on Table 2.

Layer	Output Shape	Params
Rescaling	(240, 240, 3)	0
EffNet V2 (Pre-Trained)	1280	6931124
Dropout	1280	0
Flatten	1280	0
Dense	128	163968
Dense	5	645

Table 3: Model 2 Architecture

In the second model, the output has been fed first to a dropout layer with a dropout rate of 20%, then sent to a densely-connected neural network layer with 128 units and a relu activation function, before finally sent to the classification head. The details of each layer is shown on Table 3.

Layer	Output Shape	Params
Rescaling	(240, 240, 3)	0
EffNet V2 (Pre-Trained)	1280	6931124
Dropout	1280	0
Flatten	1280	0
Dense	128	163968
Dense	128	16512
Dense	5	645

Table 4: Model 3 Architecture

For the third model, the output has also been fed first to a dropout layer with a dropout rate of 20%, then sent to 2 successive densely-connected neural network layer with 128 units and a relu activation function, before finally sent to the classification head. The details of each layer is shown on Table 4.

3.3 Algorithm

The created models have been trained with a batch size of 16 and a maximum number of epochs set to 50. The training phase applied an early stopping method to stop the training of the model and restore the best weights when the monitored metric has stopped improving from the last arbitrary number of epochs. In this study, we used the validation loss of the model as the monitored metric for early stopping and set the number of epochs to 5 to accommodate a somewhat noisy graph or improvement over time.

4 Results and Discussion

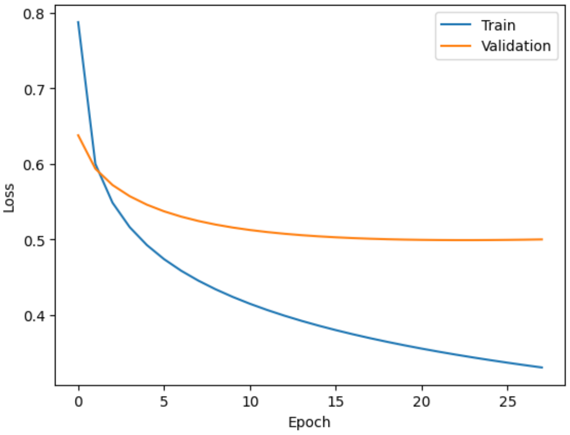

Figure 1: Model 1 Training and Validation Loss

The training phase of the model 1 stopped after 28 epochs with the best validation loss at 0.4993 while having a validation accuracy of 79.40%. Figure 1 illustrates the train and validation loss of model 1 throughout its training phase. We can see a steep decrease on both train and validation loss for the first 3 epochs followed by a gradual slow down in train loss while the validation loss seems linear and has very minimal improvements until the last epoch. The stopping was done even if the train loss is still improving because we do not want our model to be overfitted on the training data.

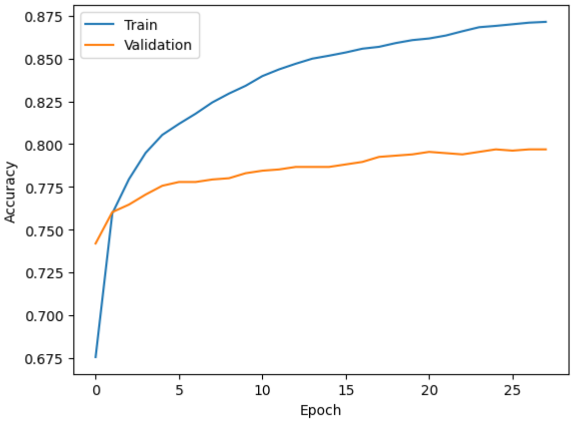

Figure 2: Model 1 Training and Validation Accuracy

Figure 2 illustrates the train and validation accuracy of model 1 throughout its training phase. The rates of improvement are similar to the model's loss where we can see a steep increase on both train and validation accuracy for the first 3 epochs followed by a gradual slow down in train accuracy while the validation accuracy seems linear and has very minimal improvements until the last epoch.

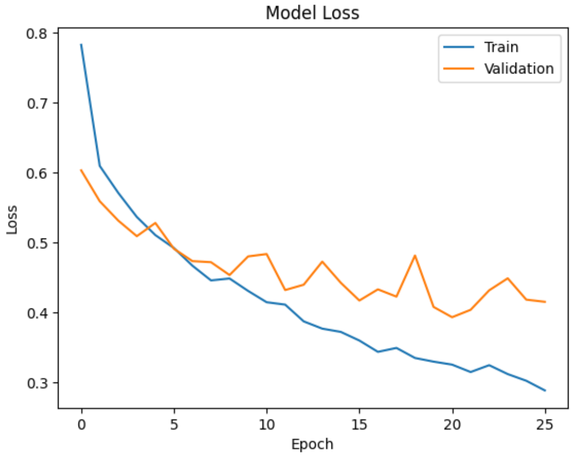

Figure 3: Model 2 Training and Validation Loss

The training phase of the model 2 stopped after 26 epochs with the best validation loss at 0.3933 while having a validation accuracy of 84.60%. Figure 3 illustrates the train and validation loss of model 2 throughout its training phase. We can see a quick decrease on both train and validation loss for around the first 5 epochs followed by a gradual slow down in train loss while the graph of the validation loss seems to be jagged with an overall very minimal improvement until the last epoch. The jaggedness or roughness of the line graph may have been caused by the introduction of a dropout layer which randomly drops learned features by setting the weight to 0.

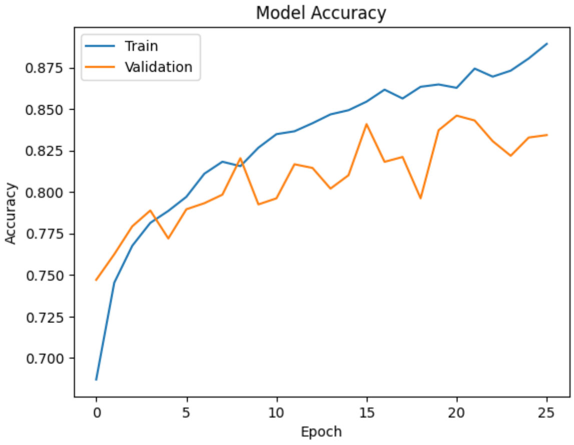

Figure 4: Model 2 Training and Validation Accuracy

Figure 4 illustrates the train and validation accuracy of model 2 throughout its training phase. The rates of improvement are similar to the model's loss where we can see a quick increase on both train and validation accuracy for around the first 5 epochs followed by a seemingly constant rate of improvement in train accuracy while the validation accuracy seems to have a noisy and minimal overall improvement.

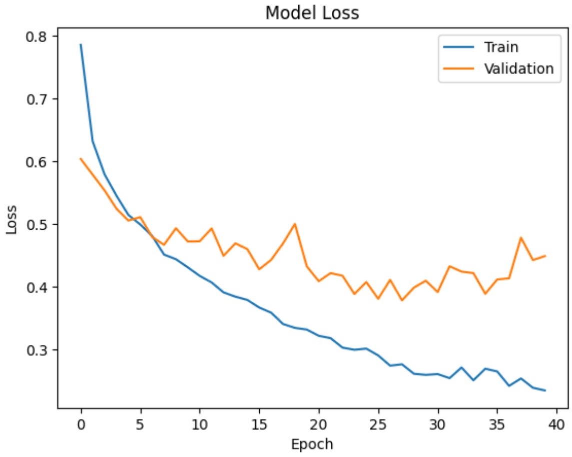

Figure 5: Model 3 Training and Validation Loss

The training phase of the model 3 stopped after 42 epochs with the best validation loss at 0.38891 while having a validation accuracy of 85.48%. Figure 5 illustrates the train and validation loss of model 3 throughout its training phase. We can see a steep drop on both train and validation loss for the first 8 epochs followed by a constant decrease in train loss until epoch 30 and a somewhat noisy graph to the end of the training, while the graph for validation loss has been noisy from the start but shows slow improvements over time.

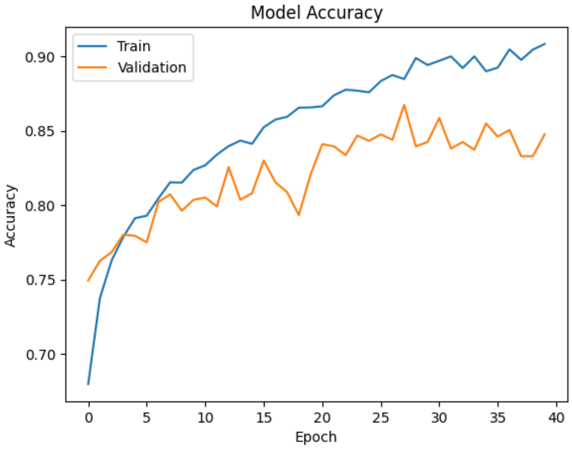

Figure 6: Model 3 Training and Validation Accuracy

Figure 6 illustrates the train and validation accuracy of model 3 throughout its training phase. The rates of improvement are similar to the model's loss where we can see a steep improvement on both train and validation accuracy for the first 8 epochs followed by a constant increase in train accuracy until epoch 30 and a somewhat noisy graph to the end of the training, while the graph for validation accuracy has been noisy from the start but shows slow improvements over time.

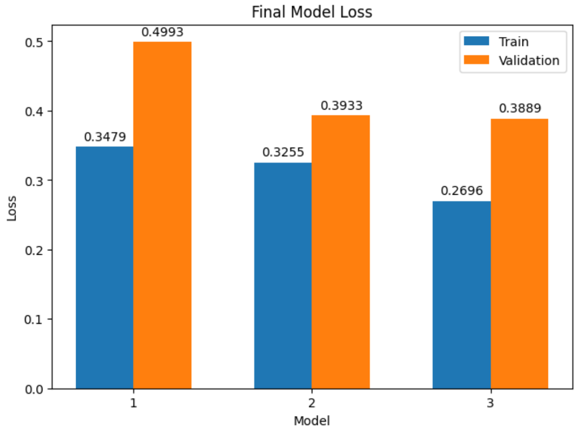

Figure 7: Final Models Training and Validation Loss

The final models' loss values are illustrated side by in Figure 7. Model 3 had the best or lowest validation loss with a value of 0.3889, closely followed by Model 2 with a value of 0.3933, while Model 1 trailed afar with a loss of 0.4993. In terms of training loss, Model 3 still performed the best with a training loss of 0.2696. Model 1 and 2 trailed behind with training losses of 0.3479 and 0.3255 respectively.

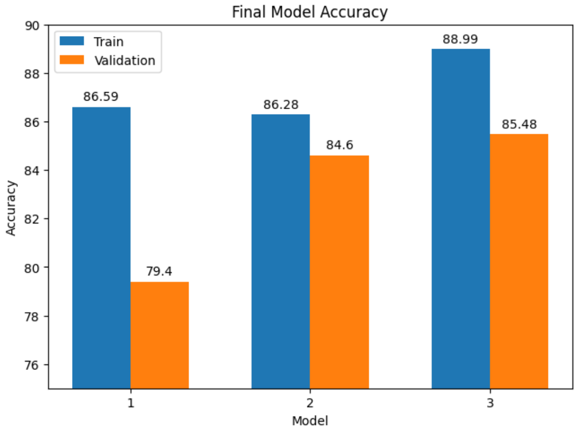

Figure 8: Final Models Training and Validation Accuracy

The final models' accuracy scores are illustrated side by in Figure 8. Model 3 had the best or highest validation accuracy with a score of 85.48%, closely followed by Model 2 with an accuracy of 84.6%, while Model 1 fell behind with an accuracy of 79.4%. In terms of training accuracy, Model 3 still performed the best with a training accuracy of 88.99%. Model 1 and 2 trailed behind with training accuracies of 86.59% and 86.28% respectively.

The evaluation metrics show in our validation set that Model 3, having 2 additional fully connected neural network layer, performed the best among the other models. Model 2, which has only one additional fully connected neural network layer, closely followed Model 3. While Model 1, which has no additional layers, fell behind on performance compared to the other two models.

5 Conclusion

This study demonstrated the effectiveness of using the EfficientNet V2 architecture in a transfer learning approach for the identification and classification of different superficial fungal infections obtained from the DeFungi dataset. The experiments show promising results where simply adding a classification head to the pre-trained EfficientNet V2 model yields a good accuracy of 79.40%, adding a dropout layer and a single dense layer greatly increased the accuracy to 84.60%, and adding another dense layer further increased the accuracy a bit to 85.48%. Further studies can improve the performance of the model by exploring the addition of other or more layers to the model. The conducted experiments also suggest that introducing a dropout layer to the model architectures adds noise to the graph of both loss and accuracy of the model.

Overall, the study found out that the use of transfer learning approaches, specifically using the EfficientNet V2 pre-trained model, generates promising results. The models were able to generalize and adapt the features learned from comnmon objects found in general image datasets to microscopic images used in this study.

References

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. 2018. Gpipe: Efficient training of giant neural networks using pipeline parallelism.

T. S. Jayram, Vincent Marois, Tomasz Kornuta, Vincent Albouy, Emre Sevgen, and Ahmet S. Ozcan. 2019. Transfer Learning in Visual and Relational Reasoning. *arXiv (Cornell University)*.

Katharina Kainz, Maria Bauer, Frank Madeo, and Didac Carmona-Gutierrez. 2020. Fungal infections in humans: the silent crisis. *Microbial Cell*, 7(6):143–145.

Taehan Koo, Moon-Hwan Kim, and Mihn-Sook Jue. 2021. Automated detection of superficial fungal infections from microscopic images through a regional convolutional neural network. *PLOS ONE*, 16(8):e0256290.

Bharti Sharma and Skarma Nonzom. 2021. Superficial mycoses, a matter of concern: Global and Indian scenario-an updated analysis. *Mycoses*, 64(8):890–908.

Afshin Shoeibi, Navid Ghassemi, Jónathan Heras, Mitra Rezaei, and Juan Manuel Górriz. 2022. *Automatic Diagnosis of Myocarditis in Cardiac Magnetic Images Using CycleGAN and Deep PreTrained Models*.

Camilo Javier Pineda Sopo, Farshid Hajati, and Soheila Gheisari. 2021. Defungi: Direct mycological examination of microscopic fungi images. *Medical Mycology*, 60.

Mingxing Tan and Quoc V. Le. 2020. Efficientnet: Rethinking model scaling for convolutional neural networks.