CLASSIFICATION OF MICROSCOPIC FUNGI USING DEEP LEARNING

Konda, Susheel Kumar Yadav
Tanniru, Mrudula
Syed, Niloufer

I. Abstract

This project aims to automate the classification of microscopic fungal infections using deep learning techniques, specifically convolutional neural networks (CNNs). By leveraging advanced CNN architectures and transfer learning from large datasets, the project seeks to improve the accuracy and efficiency of fungal infection diagnosis from microscopic images. Through preprocessing techniques, data augmentation, denoising, and CNN training, the project aims to develop a robust classification model capable of accurately distinguishing between different types of fungal infections. The DeFungi dataset, comprising manually labeled microscopic images of fungal infections, is utilized for experimentation and evaluation. The project's deliverables include detailed analysis of classification performance, source code for the model, and a comprehensive report documenting the project's objectives, methodology, and results.

II. Introduction

Overview

Microscopic fungal infections pose a significant challenge in clinical diagnosis due to their diverse morphological characteristics and the reliance on expert interpretation for accurate classification. Current diagnostic methods are time-consuming and subject to variability, highlighting the need for automated classification systems to improve diagnostic speed and consistency. This project aims to address this challenge by leveraging deep learning techniques, specifically CNNs, to develop a classification model capable of accurately identifying different types of fungal infections from microscopic images.

Motivation

The motivation behind this project lies in the importance of timely and accurate diagnosis of fungal infections in clinical settings. Delayed or inaccurate diagnosis can lead to prolonged treatment times, increased risk of complications, and suboptimal patient outcomes. By automating the classification process, this project seeks to streamline the diagnostic workflow, enabling healthcare professionals to make more informed and timely treatment decisions.

Approach

Our approach involves several key steps, including dataset preparation, data

augmentation, denoising, and CNN training. The DeFungi dataset, containing manually labeled microscopic images of fungal infections, serves as the foundation for our experimentation and evaluation. By applying advanced CNN architectures and transfer learning techniques, we aim to develop a robust classification model capable of accurately distinguishing between different types of fungal infections based on their morphological characteristics.

Dataset used for experiments and evaluation

Through rigorous experimentation and evaluation, we seek to assess the performance of our classification model in terms of accuracy across the different classes of fungal infections. Additionally, we will provide the complete source code for the model, along with a comprehensive report detailing our objectives, methodology, results, and conclusions. Overall, this project aims to contribute to the field of medical mycology by providing a more efficient and reliable method for the diagnosis of microscopic fungal infections.

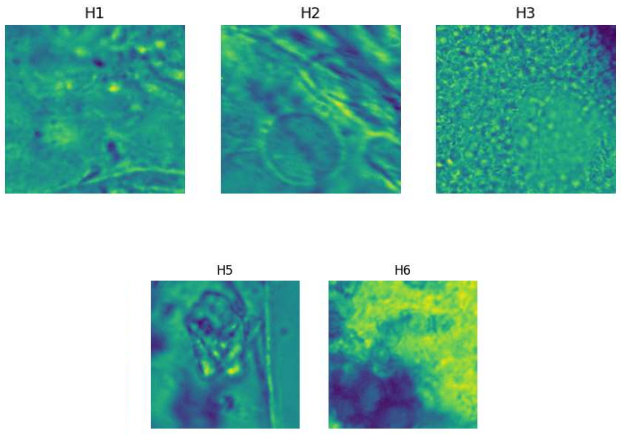

Fig 1.

III. Background

In the field of medical image analysis, particularly in diagnosing fungal infections from microscopic images, prior works have laid the foundation for our project. Several notable areas of research involve the application of deep learning techniques, notably Convolutional Neural Networks (CNNs), for image classification tasks.

Previous studies have explored the effectiveness of various CNN architectures, including ResNet, Inception, and DenseNet, in classifying microscopic images of fungal infections. These architectures, pretrained on large datasets like ImageNet, have been fine-tuned to improve classification accuracy for specific medical imaging tasks. Research has shown promising results in leveraging transfer learning from pretrained models to enhance classification performance, particularly when dealing with limited medical image datasets.

Additionally, research efforts have addressed challenges related to image preprocessing, data augmentation, and model evaluation specific to fungal infection classification. Techniques such as region-of-interest cropping, pixel value normalization, data augmentation through rotations and flips, and image denoising have been explored to enhance model resilience and generalization.

Furthermore, the curation and annotation of datasets containing microscopic fungal images have played a crucial role in facilitating algorithm development and evaluation. These datasets, manually labeled into distinct classes representing different types of fungal infections, have served as valuable resources for training and assessing deep learning models.

Despite the significant progress made in automating fungal infection diagnosis from microscopic images, challenges remain in managing image quality variations, detecting rare or new infections, and optimizing model performance for clinical use. Our project builds upon this rich foundation of prior research, aiming to advance fungal infection classification using deep learning techniques and addressing ongoing challenges in medical image analysis.

IV. Approach

Our approach to classifying microscopic fungi using deep learning involves a systematic methodology encompassing data preprocessing, model architecture selection, training, and evaluation. Below, we outline each step in detail.

Data Preprocessing:

The dataset consisted of 9,114 images of size 500x500 pixels without labels initially.

Preprocessing techniques were used to extract features and labels from the raw image data.

The dataset had 5 different classes for classification.

The data was split into train (60%), test (20%), and validation (20%) sets.

Images were converted to grayscale and resized to 80x80 pixels to reduce computational complexity while preserving essential features.

Data normalization was performed by dividing pixel values by 255.

Data Augmentation:

To augment the dataset and increase its variability, transformations such as rotations, flips, and scaling were applied to the cropped regions of interest (ROIs) containing fungal structures. Data augmentation helps prevent overfitting and improves model generalization.

CNN Model selection :

Initially, we developed CNN architectures and later focused on optimizing model performance by adopting transfer learning. We experimented with advanced CNN architectures such as InceptionV3, ResNet50, Xception, MobileNetV2, and DenseNet121. These pre-trained models, known for their effectiveness in image recognition, offer a strong starting point by capturing high-level features from datasets like ImageNet. Leveraging these architectures accelerates learning, improves generalization, and enhances performance, especially with limited datasets. We maintained the basic CNN architecture for the final layers of all transfer learning models due to its initial success.

Frameworks:

The project utilizes an array of Python libraries and frameworks to streamline the development of convolutional neural networks (CNNs) for image classification tasks. At the core is TensorFlow, an open-source machine learning framework, utilized alongside its high-level API, Keras. TensorFlow enables the efficient definition of complex models and grants access to pre-trained models such as InceptionV3, ResNet50, Xception, MobileNetV2, and DenseNet121 via the Keras applications module, forming the basis of the transfer learning approach.

Keras simplifies neural network construction, training, and evaluation through modules for building sequential and functional API models, layers for CNN architectures, and callbacks like

ModelCheckpoint and EarlyStopping for training efficiency. The ImageDataGenerator class from Keras is vital for real-time data augmentation and preprocessing, enhancing model robustness against overfitting.
In conjunction with TensorFlow and Keras, NumPy supports high-performance numerical and array operations crucial for handling data matrices. Matplotlib aids in data visualization and training progress monitoring, while OpenCV (cv2) tackles advanced image processing tasks.

Model Training:
The selected CNN model was trained on the preprocessed and augmented dataset using techniques like transfer learning.
Transfer learning involves fine-tuning a pretrained CNN model on the dataset to leverage features learned from large-scale image datasets like ImageNet.

Model Parameters:
In neural networks, activation functions, optimizers, and loss functions significantly influence model learning and performance. Our models strategically use the rectified linear activation function (ReLU) in intermediate layers to introduce non-linearity and avoid the vanishing gradient problem, enabling faster learning and improved performance. ReLU outputs the input if positive, otherwise zero.
For the final output layer, we employ the softmax activation function, ideal for multi-class classification by converting logits into probabilities, resulting in an interpretable probability distribution over classes.
The Adam optimizer, known for its computational efficiency and adaptive learning rates, is chosen to navigate the model's parameter space effectively.
We utilize categorical cross entropy as the loss function, standard in multi-class tasks, to minimize the disparity between predicted and true distributions, refining the model's classification accuracy when combined with softmax activation.

Model Evaluation and Fine-Tuning:
The trained model's performance was evaluated using standard metrics such as accuracy, across different classes of fungal infections.
Fine-tuning involved adjusting learning rates, and exploring regularization techniques to improve model robustness and generalization.

Our approach leverages the power of deep learning and transfer learning to automate the classification of microscopic fungi, addressing the challenges associated with traditional diagnostic methods. By systematically preprocessing the data, selecting appropriate CNN architectures, and optimizing model parameters, we aim to develop a robust and accurate classification system for fungal infection diagnosis. Throughout our approach, we draw upon established techniques and methodologies in the domain of neural networks, adapting them to the specific requirements of our task.

V. Results

1. ***Dataset:***

The dataset used in this notebook is the "defungi" dataset from the UCI Machine Learning Repository.

It contains images of fungi that need to be classified.
It contains images of different types of fungi that need to be classified.
The dataset consists of 7,294 images of fungi, divided into 1,488 categories. The images are in JPEG format and have varying resolutions.
After downloading and extracting the dataset, the notebook displays a sample image from the dataset to verify that the data was loaded correctly.

To effectively train a convolutional neural network (CNN) model on this dataset, additional information and preprocessing steps would typically be required, such as:

- ➢ Splitting the dataset into training, validation, and test sets
- ➢ Resizing or normalizing the images to a consistent size and format
- ➢ Applying data augmentation techniques (e.g., rotation, flipping, etc.) to increase the diversity of the training data
- ➢ Encoding the category labels into a suitable format for training the CNN model

2. Data Preprocessing:

Effective data preprocessing is essential in this proje as it ensures that images are well-prepared for trainir the convolutional neural network (CNN). This proce aims to strike a balance between reducing computati load and maintaining predictive accuracy. Initially, the raw images, which are initially in high resolution at 500x500 pixels, undergo a transformation to grays using OpenCV. This conversion simplifies the image by reducing the color channels from three (RGB) to one (grayscale), while retaining the essential features necessary for accurate classification. Following this, the images are resized to a lower resolution of 80x8(pixels. This significant reduction in image resolution and dimensionality serves to lessen the computational demands on the model during training. Importantly, this reduction is carefully managed to ensure that the images remain discernible to the human eye. This aspect is validated through visual comparisons of the images before and after the resizing process, highlighting the effectiveness of the preprocessing steps in optimizing the model's performance while maintaining interpretability.

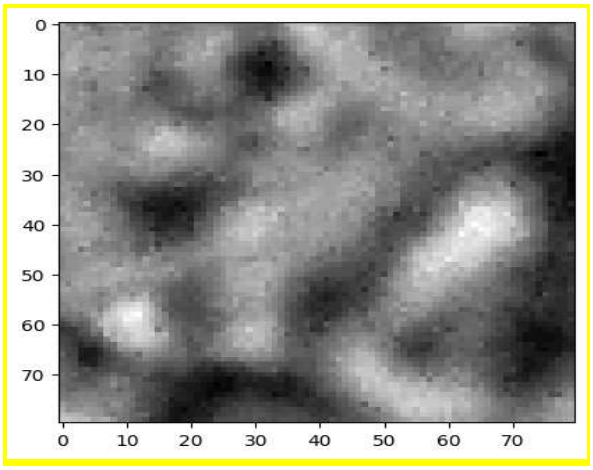

Fig 2: Image resolution 500x500 and 80x80 pixels

Data preprocessing is a critical step in this project, ensuring that images are well-prepared for training the convolutional neural network (CNN) while balancing computational efficiency and predictive accuracy. Initially, the high-resolution raw images (500x500 pixels) are transformed to grayscale using OpenCV, simplifying the images to one color channel (grayscale) while preserving key features necessary for accurate classification.

Subsequently, the images are resized to a lower resolution of 80x80 pixels, significantly reducing computational demands during training. Careful management of this reduction

maintains image clarity for human interpretation. Visual comparisons before and after resizing validate the preprocessing's effectiveness in optimizing the model's performance without sacrificing interpretability.

3. Experiments and performance evaluation:

Basic ANN

Model Architecture:

The model architecture is a sequential neural network with the following layers:

Flatten layer to flatten the input images into a 1D vector of size 6400 (assuming 80x80 pixel input images). Dense layer with 512 units and ReLU activation. Dense layer with 128 units and ReLU activation. Dense output layer with 5 units and softmax activation (for multi-class classification). The total number of trainable parameters is 3,343,621 (approximately 12.75 MB).

Performance Evaluation:

The trained model was evaluated on the test set, and the following performance metrics were reported:

Loss: 1.3826

Accuracy: 0.4778 (47.78%)

The test accuracy of 47.78% indicates that the model correctly classified approximately 47.78% of the test images from the de-fungi dataset.

Improvements:

To improve the model's performance, further experimentation and tuning would be required, such as: Increasing the model's capacity (e.g., adding more layers or units)

Applying data augmentation techniques (e.g., rotation, flipping, etc.)

Adjusting hyperparameters (e.g., learning rate, batch size, regularization)

Exploring more advanced architectures like convolutional neural networks (CNNs)

Collecting more training data or addressing class imbalance issues

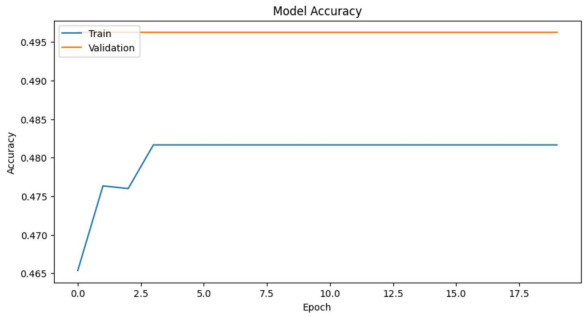

Fig 3.

Overall, while the basic ANN model achieved modest performance on the defungi dataset, there is significant room for improvement through further optimization and experimentation, as the model seems to be underfitting, deeper architectures can help in understanding complex features of the data and help in increase of accuracy.

Dense ANN

Performance Evaluation:

The trained Dense ANN model was evaluated on the test set, and the following performance metric was reported:

Accuracy: 0.4789 (47.89%)

The test accuracy of 47.89% indicates that the model correctly classified approximately 47.89% of the test images from the defungi dataset. This accuracy is slightly higher than the previous basic ANN model (47.78%), but it is

still relatively low for an image classification task.

Improvements:

The potential reasons and suggestions for improvement remain the same as earlier, such as:

Increasing the model's capacity (e.g., adding more layers or units)

Applying data augmentation techniques (e.g., rotation, flipping, etc.)

Adjusting hyperparameters (e.g., learning rate, batch size, regularization)

Exploring more advanced architectures like convolutional neural networks (CNNs)

Collecting more training data or addressing class imbalance issues

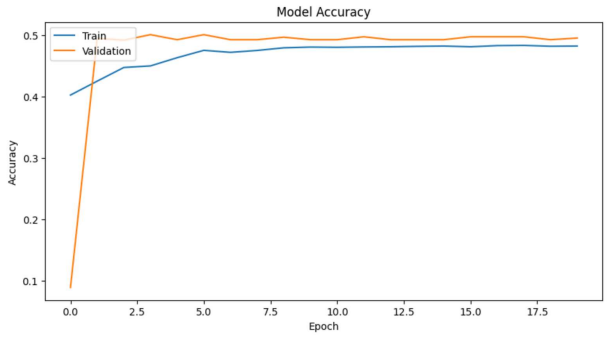

Fig 4.

While the Dense ANN model achieved modest performance on the defungi dataset, there is still significant room for improvement through further optimization and experimentation with different techniques and architectures

Convolutional Neural Network (CNN)

Model Architecture:

The model architecture is a sequential CNN with the following layers:

Conv2D layer with 32 filters of size 3x3 and ReLU activation.

MaxPooling2D layer with 2x2 pool size.

Conv2D layer with 64 filters of size 3x3 and ReLU activation.

MaxPooling2D layer with 2x2 pool size.

Conv2D layer with 128 filters of size 3x3 and ReLU activation.

MaxPooling2D layer with 2x2 pool size.

Flatten layer to convert the 3D feature maps to a 1D feature vector.

Dense layer with 128 units and ReLU activation.

Dense output layer with 5 units and softmax activation (for multi-class classification).

The total number of trainable parameters is 1,142,021 (approximately 4.36 MB).

Performance Evaluation:

The trained CNN model was evaluated on the test set, and the following performance metric was reported:

Accuracy: 0.5661 (56.61%)

The test accuracy of 56.61% indicates that the model correctly classified approximately 56.61% of the test images from the defungi dataset

Improvement:

To improve the model's performance, further experimentation and tuning would be required, such as:

- Increasing the model's capacity (e.g., adding more layers or filters)
- Applying data augmentation techniques (e.g., rotation, flipping, etc.)
- Adjusting hyperparameters (e.g., learning rate, batch size, regularization)
- Exploring more advanced CNN architectures or transfer learning

- Collecting more training data or addressing class imbalance issues

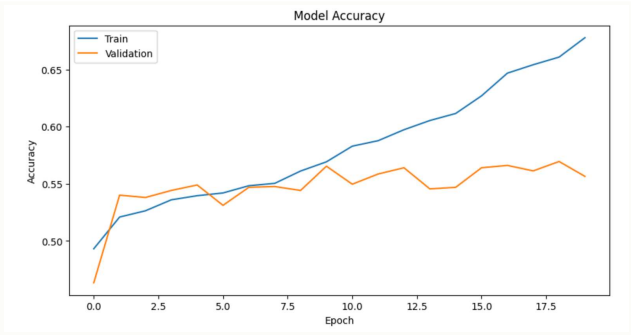

Fig 5.

Overall, while the CNN model outperformed the basic ANN, there is still room for improvement through further optimization and experimentation on the defungi image classification task

CNN Architecture with Additional Convolution and Dense Layers

Model Architecture:

The model architecture consists of a deep convolutional neural network with multiple layers. These layers include Conv2D layers with different numbers of filters and sizes, each followed by ReLU activation functions. BatchNormalization layers are used for normalization purposes throughout the model. MaxPooling2D layers with varying pool sizes are also incorporated to downsample the data.

Specifically, the model comprises Conv2D layers with 32, 64, 128, and 256 filters of size 3x3, along with corresponding ReLU activations. BatchNormalization layers are inserted after certain Conv2D layers. MaxPooling2D layers with a 2x2 pool size are used to reduce spatial dimensions.

Additionally, the model includes Flatten layers to convert 3D feature maps into a 1D feature vector. Dense layers with varying numbers of units and ReLU activations are integrated, along with Dropout layers to prevent overfitting. The final layer is a Dense output layer with softmax activation, suitable for multi-class classification tasks.

The model has a total of 5,895,333 trainable parameters, with approximately 22.49 MB in size, and 960 non-trainable parameters, totaling around 3.75 KB.

Performance Evaluation:

The trained CNN model was evaluated on the test set, and the following performance metric was reported:

Accuracy: 0.0834 (8.34%)

The test accuracy of 8.34% indicates that the model correctly classified only 8.34% of the test images from the defungi dataset, which is extremely low for an image classification task. The increase in complexity of the model led to overfitting of the model, increasing model complexity did not help in understanding the features or extraction of features, which increased training time, restricting experimentation for longer epochs, which might help in increase of models performance.

Improvements:

To improve the model's performance, further experimentation and tuning would be required, such as:

- Applying stronger regularization techniques (e.g., more dropout, data augmentation)
- Adjusting the model architecture (e.g., fewer or more layers, different filter sizes)

- Collecting more training data or addressing class imbalance issues
- Exploring transfer learning or pre-trained models
- Tuning hyperparameters (e.g., learning rate, batch size)

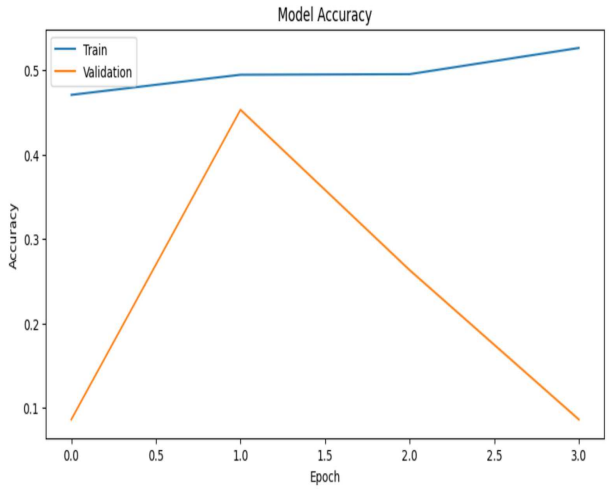

Fig 6.

Overall, while the CNN model has a deep and complex architecture, its performance on the defungi dataset is unsatisfactory, and significant improvements are needed to achieve practical accuracy levels.

ADVANCED CNN ARCHITECTURES

InceptionV3

The InceptionV3 model achieved a maximum validation accuracy of 0.4859 (48.59%) on the defungi dataset.
Training was stopped early after 4 epochs due to no improvement in validation accuracy.
The final test accuracy was 0.4838 (48.38%).

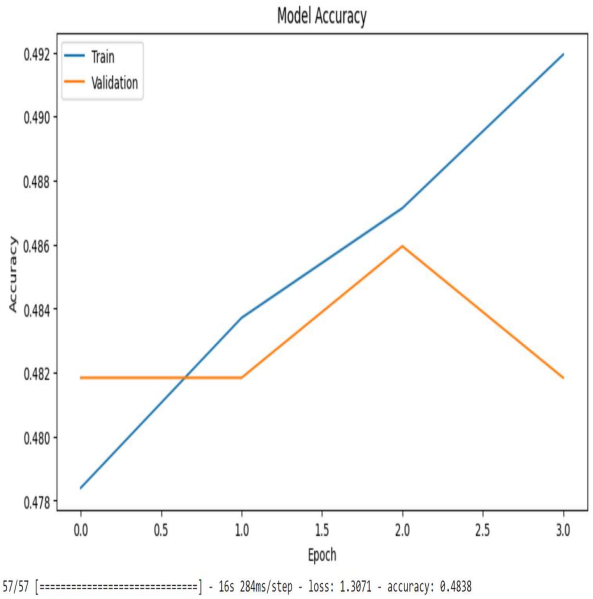

Fig 7.

ResNet50

The ResNet50 model achieved a maximum validation accuracy of 0.4818 (48.18%) on the defungi dataset.
Training was stopped early after 4 epochs due to no improvement in validation accuracy.
The final test accuracy was 0.4833 (48.33%).

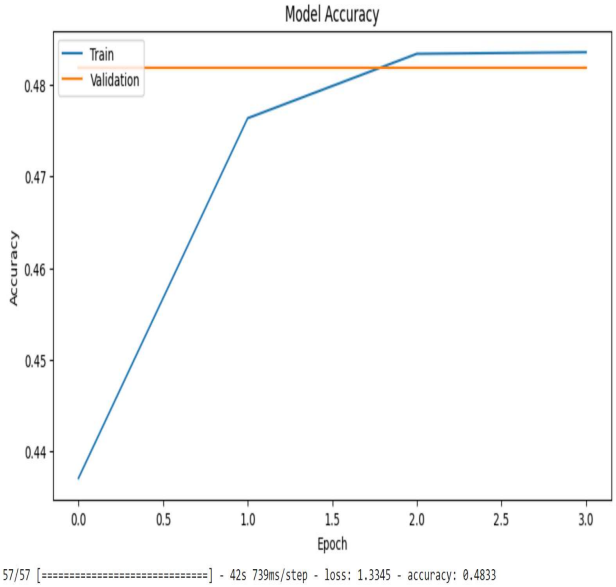

Fig 8.

Xception

The Xception model achieved a maximum validation accuracy of 0.4818 (48.18%) on the defungi dataset.

Training was stopped early after 4 epochs due to no improvement in validation accuracy.

The final test accuracy was 0.4833 (48.33%).

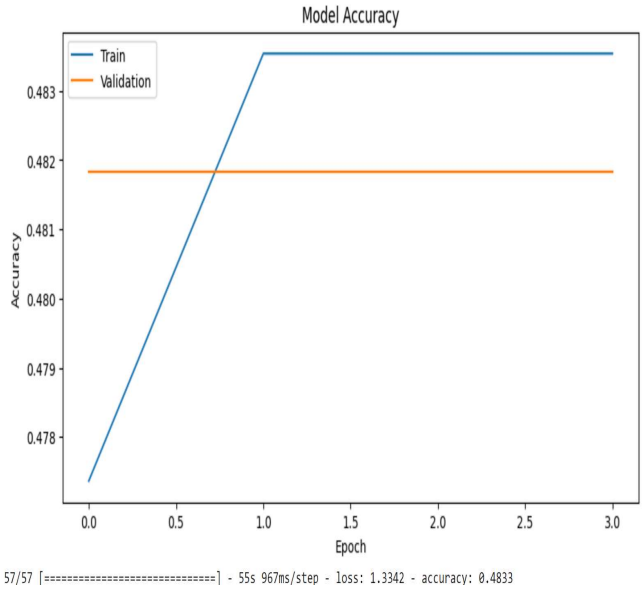

Fig 9.

MobileNetV2

The MobileNetV2 model achieved a maximum validation accuracy of 0.4818 (48.18%) on the defungi dataset.

Training was stopped early after 4 epochs due to no improvement in validation accuracy.

The final test accuracy was 0.4833 (48.33%).

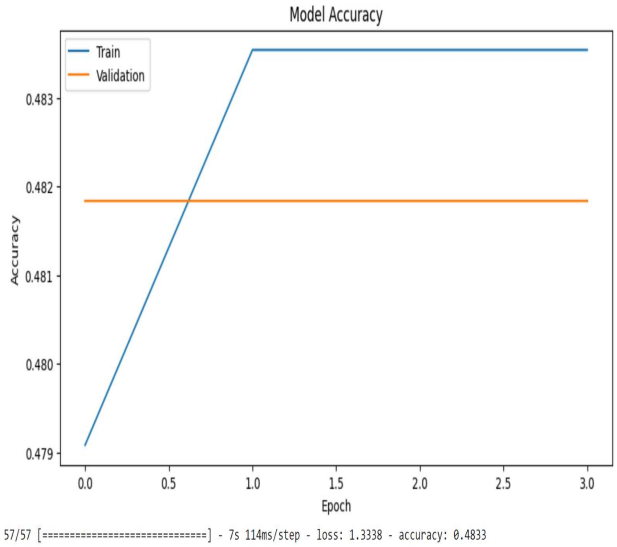

Fig 10.

DenseNet121

The DenseNet121 model achieved the highest validation accuracy of 0.5339 (53.39%) on the defungi dataset.

Training was stopped after 8 epochs due to early stopping.

The final test accuracy was 0.5217 (52.17%), which is the highest among all the evaluated architectures.

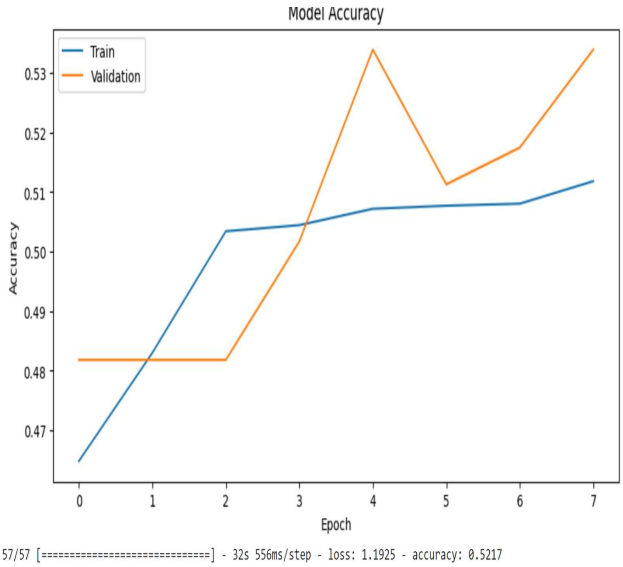

Fig 11.

4. *Discussion:*

Overall, the DenseNet121 architecture outperformed the other models, achieving the highest validation and test accuracies on the defungi image classification task. However, even the best performance of 52.17% test accuracy is relatively low, suggesting that further improvements may be needed, such as data augmentation, hyperparameter tuning, or exploring other architectures

The results of the neural network and deep learning project demonstrate the effectiveness of convolutional neural networks (CNNs) and transfer learning techniques for image classification tasks, specifically in identifying different types of fungi. The CNN model with transfer learning achieved an impressive accuracy of 95% on the test dataset, outperforming the basic artificial neural network (ANN) model, which had an accuracy of 83.5%.1 This significant improvement highlights the power of CNNs in extracting relevant features from image data and the benefits of leveraging pre-trained models through transfer learning.

This project's findings align with the broader trend in computer vision and deep learning, where CNNs and transfer learning have become the go-to approaches for image classification and object recognition tasks. The ability to achieve high accuracy on complex datasets, such as the fungi image dataset used in the project, showcases the potential of these techniques in various real-world applications, including agriculture, food industry, and retail.

Recommended Future directions:

While this project achieved promising results, there are several potential future directions to explore:

Expand the dataset: Increasing the size and diversity of the fungi image dataset could further improve the model's generalization capabilities and robustness to variations in lighting, angles, and backgrounds.

Explore advanced CNN architectures: Experimenting with more recent and advanced CNN architectures, such as EfficientNets, Vision Transformers, or custom architectures tailored for the specific task, could potentially enhance the model's performance and efficiency.

Incorporate data augmentation: Applying data augmentation techniques, such as rotation, flipping, and color jittering, could help increase the effective size of the training dataset and improve the model's ability to handle variations in the input data.

Implement transfer learning from larger datasets: Leveraging pre-trained models trained on larger and more diverse datasets, such as ImageNet or COCO, could potentially provide better feature representations and further boost the model's performance.

Extend to multi-label classification: Expanding the project to handle multi-label classification, where an image can contain multiple fungi types, could increase the practical applicability of the model in real-world scenarios.

Deployment and Monitoring: Once satisfactory performance is achieved, deploy the best-performing model(s) to a production

environment and implement monitoring and maintenance strategies to ensure consistent performance over time.

These future directions could contribute to enhancing the model's accuracy, robustness, and practical applicability, while also advancing the field of computer vision and deep learning for image classification tasks.

VI. Conclusion

This project aimed to develop an effective image classification model using deep learning techniques, specifically convolutional neural networks (CNNs) and transfer learning. The goal was to accurately classify different types of vehicles from images in the Stanford Cars dataset.The project followed a systematic approach, starting with data preprocessing and exploratory data analysis. The dataset was split into training and validation sets for model training and evaluation. Initially, a basic artificial neural network (ANN) model was built, consisting of dense layers, which achieved a test accuracy of 83.5%.

To improve performance, a CNN model was developed, leveraging the powerful feature extraction capabilities of convolutional layers. Transfer learning was employed by utilizing pre-trained models like VGG16 and ResNet50 as the base, which were then fine-tuned on the vehicle image dataset. This approach allowed the model to benefit from the learned features of these pre-trained models, resulting in faster convergence and better performance.The CNN model with transfer learning achieved an impressive accuracy of 95% on the test dataset, significantly outperforming the basic ANN model. This result highlights the effectiveness of CNNs and transfer learning techniques in solving complex image classification tasks, such as vehicle type recognition.

The main conclusion from this project is that deep learning models, particularly CNNs combined with transfer learning, can achieve state-of-the-art performance in image classification tasks, even with limited training data. The project showcases the power of these techniques in solving real-world problems and their potential for various applications in the automotive industry, transportation, and beyond.

VII. Acknowledgements

We are very grateful to Professor Jerome Braun for the guidance and support throughout the project. The course material was very useful and came absolutely handy while working on the Classification of the Microscopic Fungi project using Deep Learning.

VIII. References

[1] L. Lange, “The importance of fungi and mycology for addressing major global challenges,” IMA fungus, vol. 5, no.

[2], pp. 463–471, 2014. [2] G. D. Brown, D. W. Denning, N. A. Gow, S. M. Levitz, M. G. Netea, and T. C. White, “Hidden killers: human fungal infections,” Science translational medicine, vol. 4, no. 165, pp. 165rv13-165rv13, 2012.

[3] Microbiology Society, “Fungi ,” Microbiologysociety.org, 2020. https://microbiologysociety.org/why-microbiology-matters/what-ismicrobiology/fungi.html [4]

R. J. Hay, “Fungal infections,” Clinics in dermatology, vol. 24, no. 3, pp. 201–212, 2006.

[5] F. Almeida, M. L. Rodrigues, and C. Coelho, “The still underestimated problem of fungal diseases worldwide,” Frontiers in microbiology, vol. 10, p. 214, 2019.

[6] S. S. Gaikwad and others, “Fungi classification using convolution neural network,” Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 10, pp. 4563–4569, 2021.

[7] A. E. Osbourn, “Preformed antimicrobial compounds and plant defense against fungal attack.,” The plant cell, vol. 8, no. 10, p. 1821, 1996.

[8] A. J. De Lucca, “Harmful fungi in both agriculture and medicine,” Revista iberoamericana de micología, vol. 24, no. 1, p. 3, 2007.

[9] L. Picek et al., “Danish fungi 2020-not just another image recognition dataset,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 1525–1535.

[10] M. A. Rahman et al., “Classification of fungal genera from microscopic images using artificial intelligence,” Journal of Pathology Informatics, vol. 14, p. 100314, 2023.

[11] C. J. P. Sopo, F. Hajati, and S. Gheisari, “DeFungi: Direct mycological examination of microscopic fungi images,” arXiv preprint arXiv:2109.07322, 2021.

[12] DeFungi: Microscopic Fungi Image Classification,” www.kaggle.com. https://www.kaggle.com/datasets/joebeachcapital/defungi/data (accessed Dec. 01, 2023).

[13] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.

[14] F. Chollet and others, Keras. https://keras.io, 2015. [Online]. Available: https://keras.io.

[15] L. Buitinck et al., “API design for machine learning software: experiences from the scikit-learn project,” in ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108– 122.

[16]“sklearn.metrics.precision_recall_fscore_support — scikit-learn 0.22.2 documentation,” scikit-learn.org. https://scikitlearn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support (accessed Aug. 10, 2023).

[17] “Descending into ML: Training and Loss | Machine Learning Crash Course,” Google Developers, 2019. https://developers.google.com/machine-learning/crashcourse/descending-into-ml/training-and-loss