COFFEE LEAF DISEASES CLASSIFICATION AND THE EFFECT OF FINE-TUNING ON DEEP CONVOLUTIONAL NEURAL NETWORKS

Enoch Binney[1], Dongxiao Ren[2]

[1,2]Zhejiang University of Science and Technology,
Hangzhou 310023, China

Abstract:
This research proposes a method for the automatic diagnosis and classification of leaf diseases in Kenyan Arabica coffee leaves. We trained Deep Convolution Neural Learning models on the JMuBEN2 obtained from Mendeley data public access to determine whether a particular leaf image contains Phoma, Cercospora, or Rust. The proposed models for this work were the ResNet50, Densenet-121, and VGG19 architectures, all are well-known classification models. They were trained using transfer learning and fine-tuning and their respective outputs were compared based on these methods of training. After training the dataset using the aforementioned models, the Densenet-121 model was superior to the others gaining an accuracy of 95.44% after transfer learning and 99.36% after fine-tuning the model.

Keywords: Deep learning, Convolutional Neural Network, Leaf Rust, Phoma, Cerscospora, Transfer Learning, Fine-Tuning.

INTRODUCTION

To meet projected demand, global crop production must increase by at least 50% by 2050 [1]. Coffee is one of the world's most traded agricultural commodities, supporting the livelihoods of 100 million people worldwide [2]. In 2014, an estimated 26 million farmers in 52 countries cultivated more than 8.5 million tons of coffee, worth $39 billion in those countries [3]. Due to a combination of factors, world coffee production is expected to be down 8.5 million bags from the previous year to 167.5 million, whereas global consumption is expected to rise by 1.5 million bags to 164.9 million, with the European Union, the United States, and Brazil saw the greatest increases [4]. Coffee has a much higher retail value, with sales in the United States reaching as much as $87 billion in 2019 [5]. Even though agriculture is facing resource constraints due to climate change and decreasing water and land availability, pests and diseases are also reducing crop yields.

Coffee Leaf Rust is one of the most serious diseases afflicting Arabica coffee (CLR). It is regarded as the most important Arabica coffee disease, having ravaged Arabica coffee farms in Ceylon at the end of the nineteenth century, causing them to be replaced by tea plantations. Despite the development of effective fungicides and resistant strains to combat rust, the pathogen continues to cause production declines of 20% or more in a variety of regions [6]. The rapid development of machine learning in recent years has aided in identifying some of the leading causes of plant and crop diseases at an earlier stage than usual. Traditional machine learning and the recent development of deep learning are all making significant strides in the field of crop disease detection, classification, and segmentation to aid in the resolution of such issues faster than traditional methods expect.

Suhartono [7] employs a fuzzy logic and decision tree-based expert system. A human expert is used to collect symptoms and disease information. Following that, rules are established based on the information gathered from the experts, and a decision tree is used for classification. Pujari [8] describes a method for identifying leaves infected with fungal diseases. The radon-transform and support vector machines are used in this method (SVM). It first determines whether a crop is normal or fungi-infected. To segment the fungal region, K-means segmentation is used. After the fungal area has been identified, SVM is used to classify it. Gutte [9] performs diseased region segmentation and classification. This is accomplished through the use of K-means clustering and segmentation. A support vector machine is used for classification. Color, shape, and texture information are extracted from the segmented region.

Arivazhago demonstrated a CCM-based system for extracting features from native plants [10]. Green pixels are removed from the input image before thresholding. After that, the segmented image is used to extract features. The color and texture aspects are chosen by CCM. SVM is also used in the classification of plant leaves. As a result, such procedures are required to provide proper guidance for qualitative products for diagnoses, ensure consistency in the outcomes during the observation process, and promote objectivity. The complexity of traditional machine learning algorithms' operations, such as preprocessing, segmentation, feature extraction, and so on, reduces the system's efficiency and accuracy.

If not taken on time, proper steps may result in a significant loss of both production and time. As a result, it is critical to identify coffee leaf infections as soon as possible. In this case, machine learning can be extremely beneficial. Deep learning (DL) was developed to address the shortcomings of traditional machine learning techniques to extract relevant information from raw images and use it efficiently for categorization [11], [12]. Several deep learning-based algorithms for detecting coffee leaf disease have already been presented [13].

DL can be used to detect anomalies in both humans and plants. To classify diseases based on their impact on plants, pixel-wise procedures are used to evaluate leaves collected from sick plants. The visible patterns in these leaves are used to determine which diseases are harmful to the plants and how to deal with them to keep them from spreading. According to the research, the use of DL technology can be as accurate as 98.59 percent [14]. The need to create a cutting-edge convolutional system that supports image detection technology and plant disease classification has resulted in a plethora of research programs to provide scientists with the necessary information [15]. Image detection could be used to distinguish between healthy and unhealthy leaves. Convolutional neural networks (CNNs) provide differences between plant images that aid in the detection of anomalies in plants in their natural habitat [16]. Divya V Naik et al. [17] used an open dataset of 200 or more leaf images, divided into four distinct classes (three types of leaf parasites and a healthy leaf), to train CNN to detect disease.

Using a Convolutional Neural Network, Xiao et al. [18] have achieved 100 percent accuracy in the identification of leaf blight cases in strawberries. Esgario et al. [19] presented a ResNet50 architecture coupled with data augmentation approaches to improve the system's robustness. The resulting biotic stress classification accuracy was 95.24%. De Oliveira Aparecido et al. [20] offered an alternative method for addressing the problem of infections and pests by combining the link between infection rates and weather variables with machine learning techniques. Eduardo Lisboa et al [21] present a method for automatically classifying leaf diseases and pests in Brazilian Arabica Coffee leaves. With an accuracy of 98.04 %, a

Machine Learning model was developed and trained using the BRACOL public image dataset to determine if a particular leaf image has a disease or pest affected.
This paper's remaining sections are organized as follows: In Section II, the Research Design and Dataset used in the research. Section III discusses the Models and Theories of the individual deep CNN utilized in this research. Results are reported in Section IV, followed by a conclusion in Section V.

RESEARCH DESIGN AND DATASET

This research suggests using Deep Convolutional Neural Networks to develop a practical and reasonably accurate method for recognizing and categorizing coffee leaf diseases and tends to identify the effect of fine-tuning on the dataset as compared to transfer learning. If this study is successful, it will be able to give a complete and non-invasive method for detecting and classifying coffee leaf diseases. The experimentation will be carried out using broad methods of image classification algorithms.

Research Design

A deep learning model will be needed to construct the method that will be used to solve the problem of detecting, and classifying three sets of coffee diseases. This design approach enables the suggested model to be trained, validated, and tested on data. This will allow us to gain measurable results that can be compared to prior studies that used deep learning to tackle the same problem. In this work, transfer learning and fine-tuning models have been realized using Keras and Tensorflow and were deployed using python for image detection. The selected models have been trained using data obtained online from Mendeley Data Public Access. The results are analyzed concerning prediction.

The Dataset

The Arabica dataset is composed of two sets, namely JMuBEN and JMuBEN2 [30]. In the Mutira coffee plantation in Kirinyaga County, Kenya, images were captured under real-world conditions using a digital camera and a pathologist. The JMuBEN2 dataset includes three zipped folders containing images. The first file has 7682 Cercospora images, the second file contains 8337 Rust images, and the third file contains 6572 Phoma images. The collection contains 22,591 leaf images distributed among three classes (Phoma, Cescospora, and Rust) and annotated with the leaf condition and disease names. The Arabica datasets contain images that enable training and validation when using deep learning algorithms for leaf disease recognition and classification on coffee plants. This information can be utilized to improve the accuracy of arabica coffee leaf disease detection and classification because the model does not need to learn additional background characteristics. The distribution of the data can be found in Table 1 below.

Table 1. Distribution of the data.

Condition of leaf	Number of leaves
Cercospora	7682
Rust	8337
Phoma	6572

Splitting Data

The dataset contains 22,591 images of 3 classes as aforementioned. In the split, 85, 14, and 10 percent were allocated for training, validation, and testing. It is common practice to divide the data so that the training data makes up more than two-thirds of the total data. Training data finally contains 18,973 cell images after the split (both infected and uninfected). While the validation and test set each contain 3,387 and 228 images of coffee leave diseases respectively.

MODELS AND THEORIES

Deep Learning

Before one can appreciate the concepts and operations underlying convolutional neural networks, one must understand their architecture. A neural network is composed of various layers. Convolutional neural networks include at least one convolutional layer. The greater the number of convolutional layers, the greater the network depth. In addition to convolutional layers, there are pooling and fully connected layers. In addition, each convolutional layer concludes with an activation function.

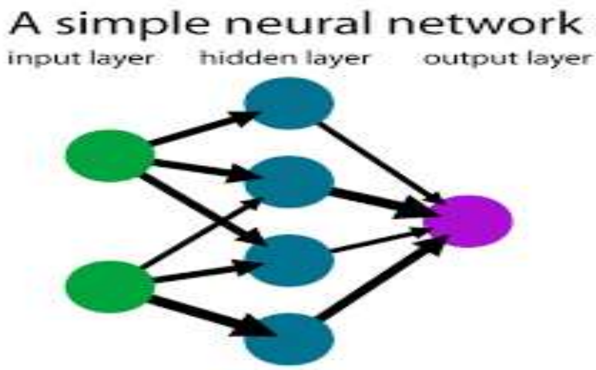

Fig. 1. Image of a simple neural network.

The convolutional layer (CL) is the primary building component of the architecture, where the network's intensive operations occur. The first CL applies a set of filters to the input image [height, width, depth]. Each filter has a small size, yet works to the full depth dimension of the image by processing each dimension individually.

Transfer Learning and Fine Tuning

In transfer learning, a base network is initially trained on a base dataset, and then the learned features from the first task are transferred to a second network for training on a second dataset and task. This procedure will be successful if the features are adequate for both the base and target objectives [31]. Using pre-trained models on similar data has proven effective for image classification applications [32, 33].
Because fine-tuning requires a great deal of time, which is dependent on computer resources, and an enormous amount of data, which is not always available, the majority of researchers no longer train the entire CNN with random weights initialization. It is typical to take a CNN that has already been trained on a large dataset in order to have initialized weights.
The first consists of removing the final fully-connected layer and replacing it with a new layer containing neurons associated with a number of classes or objects to be classified. Before classifying images, the next step is to train only this FC layer with our data, extracting features (whose size depends on the architecture) from the final layer.
In addition to replacing a classifier, another strategy involves fine-tuning the weights in the previous layers. In this manner, we "freeze" the weights so that they do not change, with the exception of those weights

contained within layers that we have fine-tuned. It can possibly fine-tune the whole network, but it increases the risk of overfitting. Therefore, it must find a balance between fine-tuning and our data volume.
Few groups have developed models such as the Oxford VGG Model [35], the Facebook Densenet Model [36], and the Microsoft ResNet Model [37], which require weeks to train on contemporary technology. These models can be downloaded and incorporated with new models that use images as input to produce more accurate results.

Oxford VGG Model
At the 2014 Image Net Large Scale Visual Recognition Challenge (ILSVRC), the VGG neural network depicted in Figure 4 placed first in the image localization test and second in the image classification task. This model was built by a team of Oxford researchers, who made the structure and weights available online. As shown in Figure 4, the structure was constructed utilizing only 3*3 convolutional layers, 2*2 max-pooling layers, and completely linked layers. The supplied image should be 224 by 224 by 3 pixels (RGB image). The only disadvantage of the VGG model is that it has 160 million parameters, the majority of which are utilized by the fully connected layers.

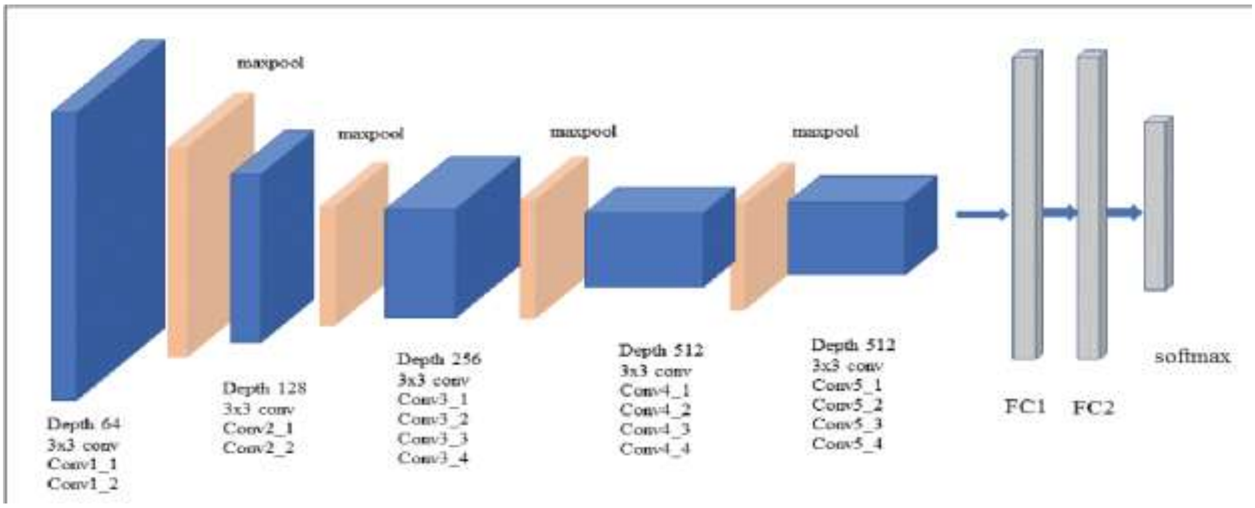

Fig. 2. Image detailing VGG19 architecture.

ResNet
In recent years, there have been several advancements in the field of computer vision. With the advent of deep convolutional neural networks, we are achieving state-of-the-art results in image classification and image recognition problems. Therefore, researchers have tended to create deeper neural networks (adding more layers) over the years to solve such complex tasks and improve classification and recognition accuracy. However, it has been observed that as more layers are added to a neural network, it becomes more difficult to train them and their accuracy begins to saturate and then decline. These are a result of problems such as the vanishing gradient problem [28] and the degradation problem [29]. ResNet or residual networks are composed of residual blocks.
Each layer of a neural network acquires low- or high-level features while being trained for the current task. Instead of attempting to learn features, residual learning attempts to learn residual. As seen in Figure 8, the activation is performed by adding the input 'x' as a residue to the output of the weight layers. The ResNet model incorporates Relu activations. ResNet50 is a 50-layer Residual network with variations including ResNet101 and ResNet152. Utilizing ResNet as a trained model for medical image categorization has yielded positive results [34].

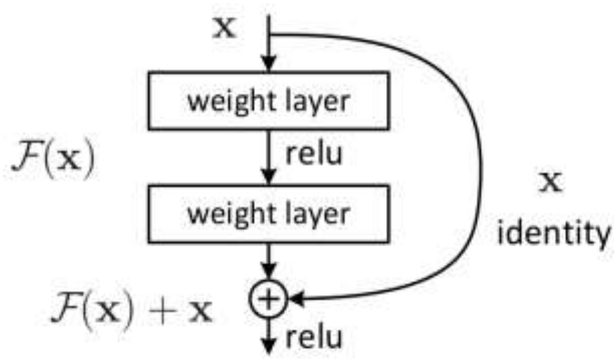

Fig. 3. Residual Learning; a building block

The proposed model has two layers, Pre-trained ResNet and dense. The input data will be trained with pre-learned weights, and only the dense layer will learn by backpropagation. Few layers, such as Batch Normalization (BN) layers, should not be frozen because the dataset's mean and variance are unlikely to match those of the pre-trained weights.

The rationale to utilize softmax activation instead of sigmoid is because the current problem involves multiple class classifications of images, and sigmoid performs well with binary classification.

DenseNet

Densenet121 has recently demonstrated remarkable precision. DenseNet is remarkably similar to ResNet, with significant distinctions. ResNet utilizes an additive approach (+) that combines the previous layer (identity) with the subsequent layer, whereas DenseNet concatenates (.) the previous layer's output with the subsequent layer's output.

DenseNet was developed primarily to improve the accuracy loss in high-level neural networks caused by the gradient's vanishing. Due to the larger distance between the input layer and the output layer, the information is lost before it reaches its destination.

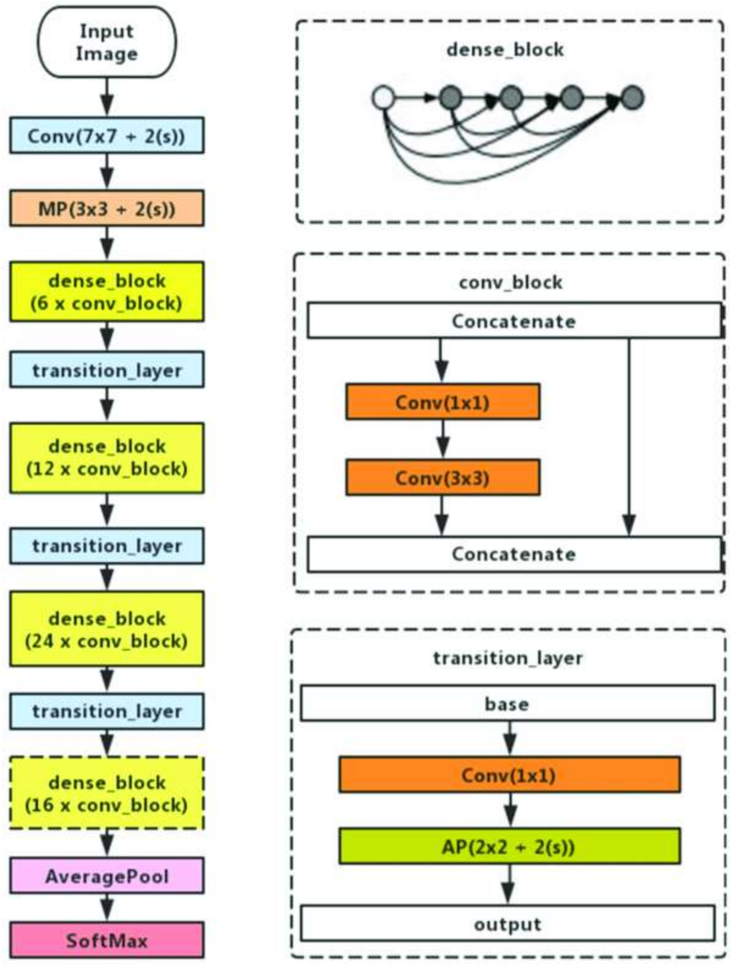

Fig. 4. Image detailing DenseNet-121 architecture.

After applying a combination of operations, conventional feed-forward neural networks connect the output of one layer to the next. As previously mentioned, this composite typically consists of convolution or pooling layers, batch normalization, and an activation function.

DenseNet-121 has 120 Convolutions and 4 AvgPool.ie. 1 7x7 Convolution, 58 3x3 Convolution, 61 1x1 Convolution, 4 AvgPool, and 1 Fully Connected Layer as shown in the above figure.
The rationale to utilize softmax activation instead of sigmoid is because the current problem involves multiple class classifications of images, and sigmoid performs well with binary classification.

DATA PREPROCESSING AND TRAINING OF THE MODEL

The image patch size, the batch size, and the number of epochs were the parameters utilized to form the neural network. All experiments utilized the identical optimizer, learning rate, and loss function, namely Sparse Categorical Cross-Entropy. Training neural networks was based on minimizing loss as opposed to optimizing precision.
The input image size employed for this study was 224x224. Therefore, all of the images were resized to the desired 224x224 size. The batch size has been set at 100. It results in 190 steps for each training cycle. The number of total training objects divided by the batch size yields the training steps per epoch. The model needs to be compiled following preprocessing and before the network is trained. A few parameters

that must be declared during training are the optimizer, loss function, and metrics. The fundamental components that provide the network with the ability to deal with the data are the optimizer and loss function. Simply put, an optimizer controls how quickly a neural network learns [23]. The aforementioned models were optimized using the Adam optimizer, Stochastic Gradient Descend, and RMSprop. The Adam optimizer has demonstrated superior performance to several other optimizers when it comes to ResNet. Selecting a loss function might be challenging as well [22]. Categorical-Cross Entropy is the loss function that was utilized to calculate the loss for the models. We all employed softmax activation because the current problem involves multi-class classifications of images.

Environment

We decrease the training time for neural networks by reducing the original images to 224 by 224 pixels. Experiments were conducted on a computer equipped with an AMD A8 PRO-7150B R5, @ 1.9GHz processor with 10 Compute cores 4c+6G, 4GB DDR3L PC3L-12800 RAM, and the Windows 10 Pro operating system.

Python 3.8.7 was used to implement the algorithms, with the libraries TensorFlow 2.4.1 [24], Keras 2.4.3 [25] contained in the TensorFlow library for the building and training of the neural network, sci-kit-learn 0.24.1[26] for the accuracy analysis, and matplotlib 3.3.4 [27] for the graph modeling.

RESULTS:

Accuracy and Loss were the metrics measured during the dataset's training. Both training and validation data were used to measure these measures. As autotuning was being done and no GPUs were being used, training the model took a while. Tables I, II, and III lists the accuracy and loss for the training and validation sets of data. Figures 5, 6, and 7 depict the accuracy and loss during transfer learning and fine-tuning with the green line indication beginning of fine-tuning. It can be realized from each of the loss graphs that each of our models was not overfitting and hence trained well.

It can be observed from the information given and illustrated below that fine-tuning improves model accuracy as compared to transfer learning. VGG19 after transfer learning produced a validation accuracy of 85.27% and a training accuracy of 80.90%. Upon model evaluation, it was realized that VGG19’s transfer learning model predicted about 83.77% of the test data indicating that the model is doing well. However, the fine-tuning model outperformed the transfer learning model by producing a validation accuracy of 90.40% with a training accuracy of 89.59%. Also, it predicted about 88.60% of the test dataset supporting our argument that fine-tuning outperforms transfer learning. Below are the results of the VGG19 model as highlighted:

Table I. Accuracy and Loss indicators of VGG19.

State	Metrics	Value
Transfer Learning	Training Accuracy	0.8090
	Training Loss	0.5973
	Validation Accuracy	0.8527
	Validation Loss	0.5673
Fine Tuning	Training Accuracy	0.8959
	Training Loss	0.346
	Validation Accuracy	0.9040
	Validation Loss	0.3088

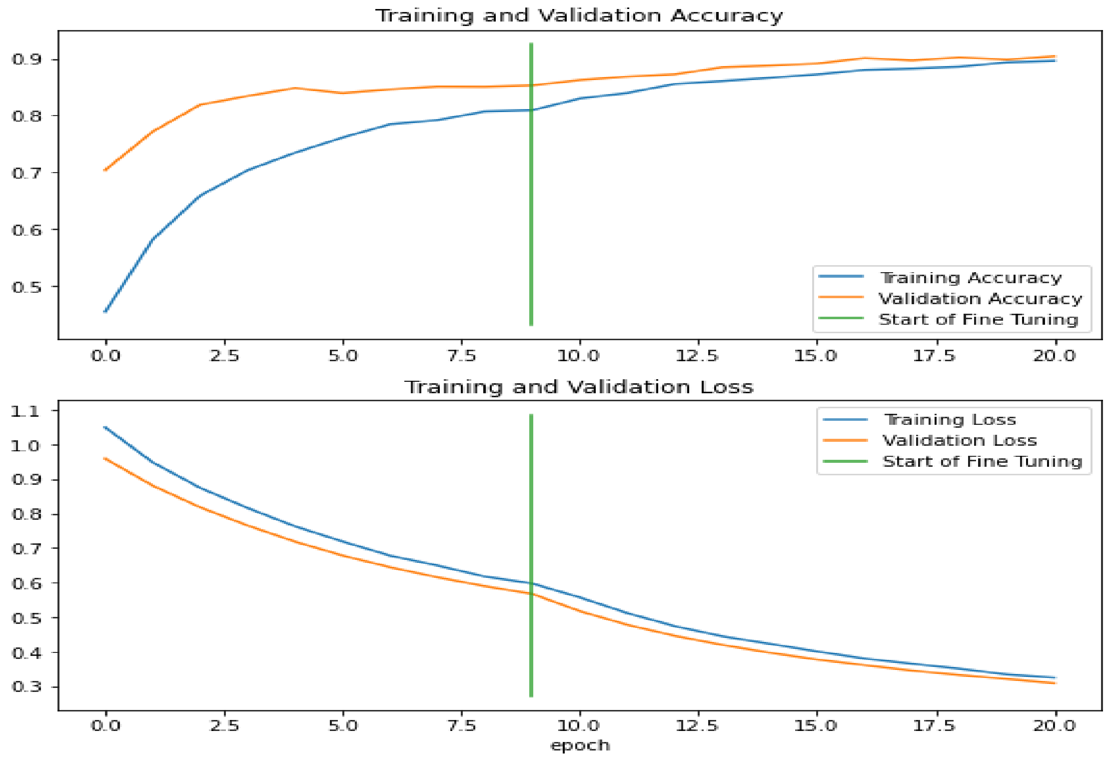

Fig. 5. Graph of Loss during training of VGG19

It can be observed from the information given and illustrated below that fine-tuning improves model accuracy as compared to transfer learning. ResNet50 after transfer learning produced a validation accuracy of 78.00% and a training accuracy of 74.63%. Upon model evaluation, it was realized that ResNet50's transfer learning model predicted about 77.19% of the test data indicating that the model is doing well. However, the fine-tuning model outperformed the transfer learning model by producing a validation accuracy of 83.35% with a training accuracy of 76.71%. Also, it predicted about 82.46% of the test dataset supporting our argument that fine-tuning outperforms transfer learning. Below are the results of the ResNet50 model as highlighted:

Table II. Accuracy and Loss indicators of ResNet50.

State	Metrics	Value
Transfer Learning	Training Accuracy	0.7463
	Training Loss	0.6303
	Validation Accuracy	0.7800
	Validation Loss	0.6052
Fine Tuning	Training Accuracy	0.7671
	Training Loss	0.5134
	Validation Accuracy	0.8335
	Validation Loss	0.4204

International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

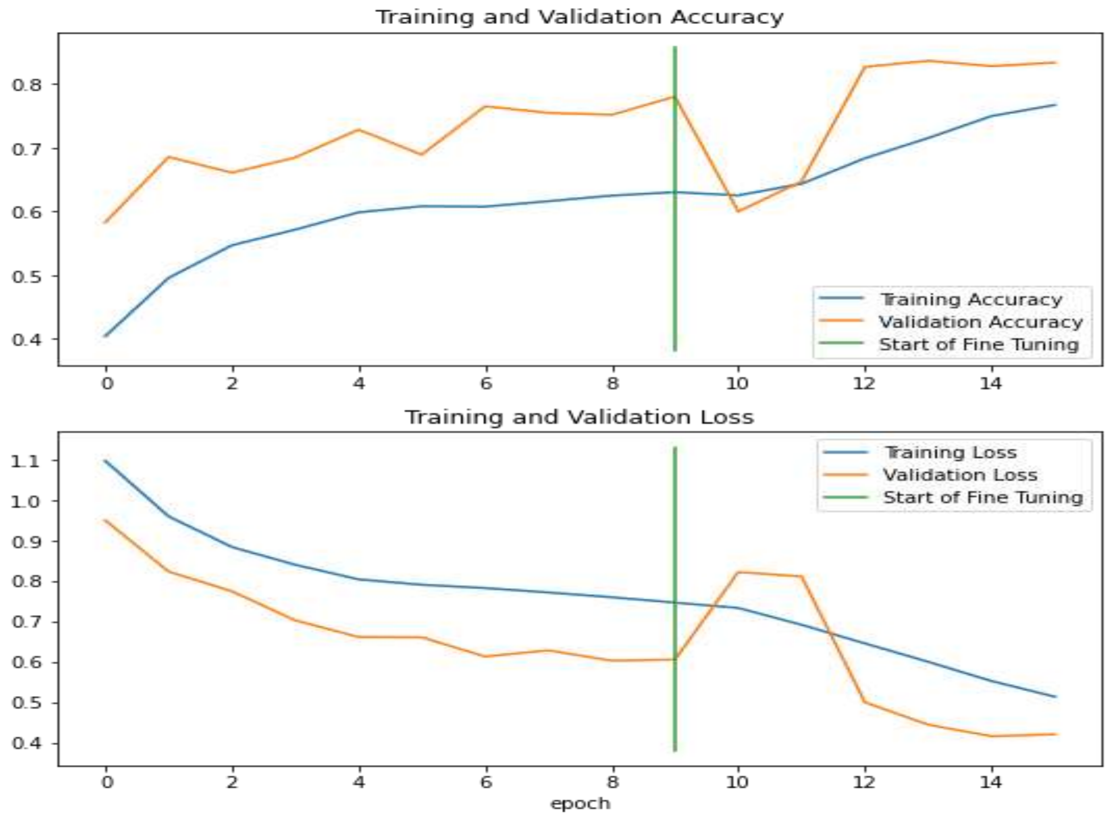

Fig. 6. Graph of Loss during training of ResNet50.

It can be observed from the information given and illustrated below that fine-tuning improves model accuracy as compared to transfer learning. DenseNet-121 after transfer learning produced a validation accuracy of 97.38% and a training accuracy of 95.44%. Upon model evaluation, it was realized that DenseNet-121's transfer learning model predicted about 96.05% of the test data indicating that the model is doing well. However, the fine-tuning model outperformed the transfer learning model by producing a validation accuracy of 99.36% with a training accuracy of 98.03%. Also, it predicted about 100% of the test dataset supporting our argument that fine-tuning outperforms transfer learning. Below are the results of the DenseNet-121 model as highlighted:

Table III. Accuracy and Loss indicators of DenseNet121.

State	Metrics	Value
Transfer Learning	Training Accuracy	0.9544
	Training Loss	0.8240
	Validation Accuracy	0.9738
	Validation Loss	0.7651
Fine Tuning	Training Accuracy	0.9803
	Training Loss	0.4912
	Validation Accuracy	0.9936
	Validation Loss	0.4517

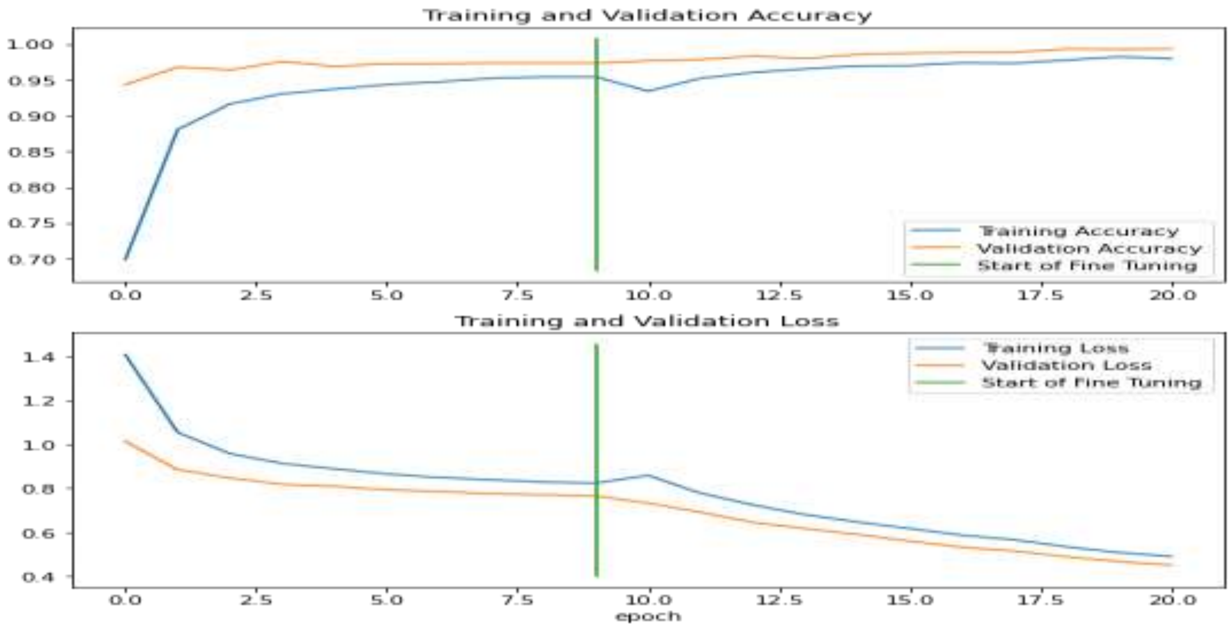

Fig. 7. Graph of Loss during training of DenseNet121

DISCUSSIONS

The classification of coffee leave disease images using Transfer Learning and fine-tuning has yielded positive results. The models upon training in 10 (transfer learning) + 10 (fine tuning) epochs with 190 steps per epoch tend to be a good fit as their output accuracy indicates. Overall, the model that best performed on the coffee leaf disease dataset, after transfer learning and fine-tuning was DenseNet-121, followed by VGG19 and then ResNet50. Other models, such as Google Inception and other variants of the proposed models like ResNet101, and ResNet152 should also be applied to the coffee leaf disease dataset, in addition to the proposed Deep CNN used in this study. Perhaps utilizing the Inception or Mobilenet model can improve accuracy. In addition, using proposed model variants with larger layers may increase the accuracy and reduce the loss.

Additionally, it can be concluded that fine-tuning had a positive effect on the models as compared to transfer learning although this method also trained well. However, fine-tuning a model to better perform a specific task can result in better results. On the other hand, when not tuned well can have a devastating effect on the pre-trained model.

REFERENCES

[1] M. Hunter, R. Smith, M. Schipanski, L. Atwood and D. Mortensen, "Agriculture in 2050: Recalibrating Targets for Sustainable Intensification", BioScience, vol. 67, no. 4, pp. 386-391, 2017. Available: 10.1093/biosci/bix010

[2] M. Pendergrast, Uncommon Grounds: The History of Coffee and How It Transformed our World (Basic Books, 2010).

[3] M. Hirons et al., Pursuing climate resilient coffee in Ethiopia—A critical review. Geoforum 91, 108–116 (2018).

[4] United States Department of Agriculture. "Coffee: World Markets and Trade, https://apps.fas.usda.gov/psdonline/circulars/coffee.pdf. Accessed April 10, 2022.

[5] Specialty Coffee Association, Price crisis response initiative: Summary of work (Specialty Coffee Association,2019).
https://static1.squarespace.com/static/584f6bbef5e23149e5522201/t/5ebd4d5f1e9467498632e0b8/1589464434242/AW_SCA_PCR_Report2020+-+December+2019+-+Update+May+2020.pdf. Accessed 10 April 2022.
[6] J.M. Waller, Coffee rust epidemiology and control Crop Prot, 1 (4) (1982), pp. 385-404
[7] Suhartono, Derwin, Wahyu Aditya, Miranty Lestari, and Muhamad Yasin. (2013). "Expert System in Detecting Coffee Plant Diseases." International Journal of Electrical Energy 1 (3): 156–162.
[8] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "Detection and classification of fungal disease with radon transform and support vector machine affected on cereals," Int. J. of Computational Vision and Robotics, vol. 4, pp. 261–280, 2014.
[9] Gutte, Vitthal S., Maharudra A.Gitte. (2018). "A Survey on Recognition of Plant Disease with Help of Algorithm". International Journal of Engineering Science and Computing, 6(6), 7100-7102.
[10] Selvaraj, Arivazhagan & Shebiah, Newlin & Ananthi, S. & Varthini, S. (2013), "Detection of the unhealthy region of plant leaves and classification of plant leaf diseases using texture features", Agricultural Engineering International: CIGR Journal. 15. 211-217.
[11] Y. LeCun, Y. Bengio, G. Hinton. Deep learning Nature, 521 (7553) (2015), p.436
[12] Y. Bengio, A. Courville, P. Vincent. Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell., 35 (8) (2013), pp. 1798-1828
[13] A. P. Marcos, N. L. Silva Rodovalho and A. R. Backes, "Coffee Leaf Rust Detection Using Convolutional Neural Network," 2019 XV Workshop de Visão Computational (WVC), São Bernardo do Campo, Brazil, 2019, pp. 38-42, doi:10.1109/WVC.2019.8876931.
[14] A. Dhakal and S. Shakya, "Image-based plant disease detection with deep learning," International Journal of Computer Trends and Technology, vol. 61, no. 1, pp. 26–29, 2018. View at: Publisher Site | Google Scholar
[15] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," Biosystems Engineering, vol. 172, pp. 84–91, 2018. View at: Publisher Site | Google Scholar
[16] A. Ramcharan, P. McCloskey, K. Baranowski, et al., "A mobile-based deep learning model for cassava disease diagnosis," Frontiers in Plant Science, vol. 10, p. 272, 2019. View at: Publisher Site | Google Scholar.
[17] Divya V Naik;Pallavi Shetty. Leaf disease detection using Mask-RCNN. 2020.
[18] J.-R. Xiao, P.-C. Chung, H.-Y. Wu, Q.-H. Phan, J.-L. A. Yeh, and M. T.- K. Hou, "Detection of strawberry diseases using a convolutional neural network," Plants, vol. 10, no. 1, 2021.
[19] J. G. Esgario, R. A. Krohling, and J. A. Ventura, "Deep learning for classification and severity estimation of coffee leaf biotic stress," Computers and Electronics in Agriculture, vol. 169, p. 105162, 2020.
[20] L. E. de Oliveira Aparecido, G. de Souza Rolim, J. R. da Silva Cabral De Moraes, C. T. S. Costa, and P. S. de Souza, "Machine learning algorithms for forecasting the incidence of Coffea arabica pests and diseases," International Journal of Biometeorology, vol. 64, no. 4, pp.
671–688, Apr 2020.
[21] LISBOA, Eduardo; LIMA, Givanildo; QUEIROZ, Fabiane. Coffee Leaf Diseases Identification and Severity Classification using Deep Learning. *In*: WORKSHOP DE TRABALHOS DA GRADUAÇÃO - CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES (SIBGRAPI), 34. , 2021, Online. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021 . p. 201-205.
DOI: https://doi.org/10.5753/sibgrapi.est.2021.20039.
[22] Rosasco L, Vito ED, Caponnetto A, Piana M, Verri A. Are loss functions all the same? Neural Computation. 2004 May 1;16(5):1063-76.
[23] Reyes AK, Caicedo JC, Camargo JE. Fine-tuning Deep Convolutional Networks for Plant Recognition. CLEF (Working Notes). 2015 Sep 8;1391.

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Man´e, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vi´egas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," https://www.tensorflow.org/, 2015.
[25] F. Chollet et al., "Keras," https://keras.io, 2015.
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion,O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
[27] J. D. Hunter, "Matplotlib: A 2d graphics environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90–95, 2007.
[28] Habibzadeh M, Jannesari M, Rezaei Z, Baharvand H, Totonchi M. Automatic white blood cell classification using pre-trained deep learning models: Resnet and inception. InTenth International Conference on Machine Vision (ICMV 2017) 2018 Apr 13 (Vol. 10696, p. 1069612). International Society for Optics and Photonics.
[29] Wichrowska O, Maheswaranathan N, Hoffman MW, Colmenarejo SG, Denil M, de Freitas N, Sohl-Dickstein J. Learned optimizers that scale and generalize. InProceedings of the 34th International Conference on Machine Learning-Volume 70 2017 Aug 6 (pp. 3751- 3760). JMLR.org.
[30] Arabica coffee leaf images dataset for coffee leaf disease detection and classification, https://doi.org/10.1016/j.dib.2021.107142, Jennifer Jepkoecha, David Muchangi Mugoa, Benson K.Kenduiywob, Edna Chebet Tooc, June 2021.
[31] Yosinski J, Clune J, Bengio Y, and Lipson H. How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27 (NIPS '14), NIPS Foundation, 2014.
[32] Quattoni A, Collins M, Darrell T. Transfer learning for image classification with sparse prototype representations. In2008 IEEE Conference on Computer Vision and Pattern Recognition 2008 Jun 23 (pp. 1-8). IEEE.
[33] Zhu Y, Chen Y, Lu Z, Pan SJ, Xue GR, Yu Y, Yang Q. Heterogenous transfer learning for image classificaion. InTwenty- Fifth AAAI Conference on Artificial Intelligence 2011 Aug 4.
[34] Keskar NS, Socher R. Improving generalization performance by switching from adam to sgd. arXiv preprint arXiv:1712.07628. 2017 Dec 20.
[35] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. http://arxiv.org/abs/1409.1556. 4 Sept. 2014
[36] G. Huang, Z. Liu and L. van der Maaten, "Densely Connected Convolutional Networks," 2018.
[37] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).