Disease Detection in Coffee Plants Using Convolutional Neural Network

Pranav Gupta
Department of Computer Engineering
Delhi Technological University
New Delhi, India
Email: 7pranavgupta@gmail.com

Puneet Madhav
Department of Computer Engineering
Delhi Technological University
New Delhi, India
Email: puneet.m98@gmail.com

Sachin
Department of Computer Engineering
Delhi Technological University
New Delhi, India
Email: sachinrathore454@gmail.com

Abstract—In the last few years, convolutional neural networks (CNNs) has proved its efficiency and accuracy in the field of image classification. As Deep Convolutional Neural Networks have proved their usefulness in research areas like object detection and pattern recognition, hence we realized that it can act as a powerful tool in generation and classification of patterns. This research is a contribution to our farmers in their battle against coffee plants diseases. Coffee plant diseases like - Leaf Rust, Cercospora Spots, Phoma, Leaf Miner have clear visual symptoms which can be extracted and categorized, thus enabling us to increase the coffee crop production output. Since these symptoms are clearly distinguishable patterns thus we have proposed Convolution Neural Network which utilizes the technique of Transfer Learning, thus reducing the training time of the model significantly. Moreover, we have also applied Data Augmentation technique to generate a large number of transformed images for training our model such that it prevent the problem of overfitting of the model. This will help the farmers in timely detection of coffee leaf diseases, thus increasing the overall coffee production output of India. We have achieved a record breaking accuracy of 97.61 % with our trained deep convolutional neural network model.

I. INTRODUCTION

Coffee is one of the most popular beverage and an important commercial product, which is cultivated in more than 50 countries across the world. Everyday around 2.25 billion cups of coffee are consumed in the world.

India is among the top producers of coffee globally and has the finest quality of coffee grown in the shade rather than direct sunlight, anywhere in the world. With an estimated 319,500 metric ton of coffee produced in the year 2018-19, of which almost 80% is exported throughout the globe[1]**,** the amount of revenue generated from the coffee industry is projected to be at USD987m in 2020 and the market is expected to grow at 7.4% annually [2].

What these estimates do not show is that the coffee plants are highly vulnerable to various pests and diseases. Crop loss due to such diseases in coffee plants is estimated at 20-30 per cent of the country's total production. Chemical control's success on any disease depends on the disease pathogenesis stage at which the fungicide is applied [3]. Inexperienced pesticide usage may result in the development of long-term resistance of the pathogens, extremely reducing the ability to fight back. Timely and accurate detection of plant diseases is one of the pillars of precision agriculture[4]. In this sense, soft computing can play a vital role in the timely detection of such biotic stresses having visible symptoms that attack the leaf of a coffee plant. Some plant diseases do not have any conspicuous visual symptoms, however most common diseases found in coffee leaves like - Leaf Rust, Cercospora Spots, Phoma, Leaf Miner have clear visual symptoms as shown in the figure. For example, the symptoms of leaf rusts include small, yellowish, oily spots on the upper leaf surface that expand into larger round spots that turn bright orange to red and finally brown with a yellow border[5].

Healthy Leaf　　Leaf Minor　　Leaf Rust　　Phoma　　Cercospora Spot

Figure 1: images of Coffee leaves affected by different diseases

Various techniques such as genetic algorithms, computer vision and image processing have been an area of active research to detect such diseases[6][7]. In [8], the authors have used both image processing and machine learning techniques and have achieved a success rate of 94%.

Deep Learning have completely dominated in the field of image classification and since the symptoms of infected leaf are clearly distinguishable patterns thus we have proposed Convolution Neural Network which can be used to detect coffee leaf diseases and categorize them into 5 classes – Healthy, Diseased Leaves with Cercospora spots, Phoma, Coffee Leaf Rust(CLR), and Leaf Miner with a very high accuracy.

The remaining sections of this paper is organized as follows: Section II explains the various materials (data source) and methodologies that has been used like the SoftMax loss function and optimization algorithm in the model. Section III explains the architectural components like Inception V3 model used, Data Augmentation and Transfer Learning technique, Section IV describe the results we have achieved using the proposed model. In the section V, we have our conclusion and the work we aim to explore in the future.

II. MATERIAL AND METHODOLOGY

A. DATASET

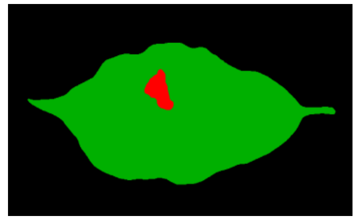
The dataset used by us contains various types of images of coffee leaves for disease detection. The leaf images within our dataset contains both disease-free and diseased leaves, affected by different types of biotic stresses, especially coffee leaf rust (CLR). We have used a publicly available dataset from the internet since collecting such large scale coffee leaf data through fieldwork requires a lot of time and effort. The different biotic stresses that affect coffee plants are: leaf miner, leaf rust, Cercospora leaf spot and Phoma. Each image has the dimensions of 2048x1024 pixels.

Our dataset is divided into 3 directories described below:

- Leaf dataset: It contains images of coffee leaves and a csv file with labels as different biotic stresses.
- Segmentation dataset: It contains images that are annotated for better performance and accuracy.
- Symptom dataset: It contains folders that are named as labelled classes - healthy leaves, leaf miner, leaf rust, phoma and Cercospora. These are the 5 categories into which we divide our leaves. All leaves that suffer from some form of biotic stress are unhealthy leaves, and are detected as diseased.

Each of our datasets is divided into training, testing and validation images.

Sample of Coffee leaf



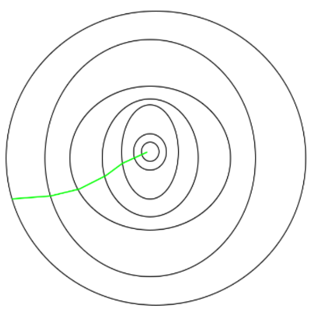
Annotation image indicating presence of Leaf rust

Figure: Images present in Segmentation dataset representing data annotation

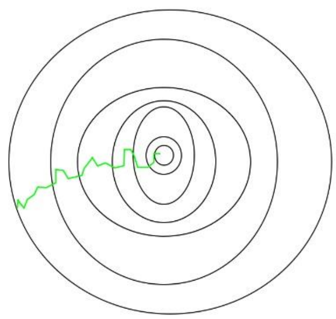
B. OPTIMIZATION ALGORITHM

Optimization algorithms helps us to minimize (or maximize) an Objective function O(x) which is a mathematical function dependent on the Model's internal learnable parameters and used in computing the output values(Y) . Gradient Descent falls under the umbrella of first order optimization algorithms

There are various type of gradient descent to optimize our parameters like Batch Gradient Descent, Stochastic Gradient Descent and Mini Batch Gradient. Computational time required in Stochastic Gradient Descent is least in contrast to BGD which is very expensive computationally. In the figure below we have tried to represent the path taken by SGD with BGD.



Path taken by BGD optimization



Path taken by SGD optimization

It can be seen in the figure that the path taken by SGD has some fluctuation in comparison to the path taken by BGD. In the proposed model Stochastic Gradient Descent(SGD) is used as the optimization algorithm, the term Stochastic is linked with random probability distribution, since in this algorithm we pick a random datapoint to find the gradient w.r.t parameter formula for SGD which is :-

$$\theta = \theta - \alpha \nabla_{\theta} J\,(\theta\,;\, x_{(i)}\,, y_{(i)}\,)$$

where θ is the parameter, J (θ ; x(i),y(i)) is the cost, (x(i),y(i)) are the datapoint from the training set

SGD is slightly noisier than Batch Gradient Descent, and generally takes higher number of iteration to reach the minima, since only a single random data point is used to update the parameters, but as we are using a single data point to find the gradient in comparison to Batch gradient, so the overall time taken by more number of iterations is compensated for, hence making SGD is computationally much less expensive than conventional Gradient Descent.

C. CATEGORICAL CROSS- ENTROPY

Loss functions are used to evaluate the performance in model. If our model predicts data that is highly deviated from the actual results, then loss function scales up to a large value. Thus better predictions mean minimum value of loss function. There are many loss functions like Mean squared Error, Cosine Similarity, Binary Cross-entropy etc. One such loss function is Categorical Cross- Entropy. It is used for multi-class classification. It compares the distribution of our predictions with the actual labels. The probability of the true class is kept as 1 and for all other classes, the probability is kept 0. The equation is given as:

$$\mathrm{L}(y, \hat{y}) = -\sum_{j=0}^{m} \sum_{i=0}^{n} (y_{ij} * log(\widehat{y_{ij}}))$$

where y is the actual value and $\hat{y}$ is the predicted value. m and n represent the total number of data points and total number of classes respectively. We use this function only when one of the classes is correct.

III. NEURAL NETWORK ARCHITECTURE AND WORKING

The proposed network uses a Sequential model for classification. This Sequential model has been implemented using the Keras library. To complement the performance of our network, we have used Inceptionv3 structure that is a pre-trained network. This pre-trained network is the feature extraction part of our model. Along with this network, we have added dense and flatten layer to our sequential model. The flattening layer is used when we want to convert a multidimensional output from a convolution layer to a linear output such that it can be directly fed into our neural network layer or to regular dense layer. The dense layer then performs the calculations and then applies the activation function. The output of the dense layer is specified by the number of units that are given input in the argument. These layers serve in the classification part of our model. We have around 21 million parameters. The image size to be used in the pre-trained model has dimensions of 299x299 pixels.

The optimization algorithm we have used is Stochastic Gradient Descent(SGD). It updates the parameters while training our network. It performs much faster because it makes a single update at a time. Optimization Algorithm has been briefly explained in the previous section already, and SGD is compared to Batch Gradient Descent algorithm to prove the point that latter is much more computationally expensive to perform.

The value of the learning rate value is kept at 0.005. To calculate the performance of our model, we have used Categorical Cross-entropy as our loss function. This function is also commonly referred to as SoftMax loss function. A brief explanation of loss function, particularly categorical cross-entropy loss function is provided in above section.

The images provided for training are input in batches. There are around 1539 images belonging to 5 different classes. The batch size value is kept as 24. The epoch value is kept as 20. While training, first we perform Data Augmentation that transforms our data and enlarges our dataset. The class ImageDataGenerator is used to perform this augmentation. The new images output from this class is of dimensions 299x299 pixels. These images are then passed into our Sequential model. The inception v3 model performs transfer learning. The CNN layers that are present in this model have pre-trained weights and biases. After each epoch, a new set of images are produced that are passed to the inception model. The images are then passed through a flatten and a dense layer. In this way, our model trains on the dataset. Hence after each and every epoch, we evaluate the accuracy and cross-entropy of our model.

In the following sub-sections, we explain the techniques of data augmentation and transfer learning and how it has enabled our model to provide such highly accurate testing results.

A. DATA AUGMENTATION

The main purpose of applying data augmentation is to enhance the generalizability of our model [9] . This reduces the overfitting in our network. In data augmentation, we transform our images to produce a set of new images that are modified versions of our input data. Since our network constantly sees new modified images, it is able to learn more features in a much more robust manner. This increases our accuracy and enlarges our dataset. In such augmentations, we mainly apply geometric transformations like random translation, rotation, flips, shearing etc to our images that transform our images but does not actually change the labels and enhances the learning rate of the model.

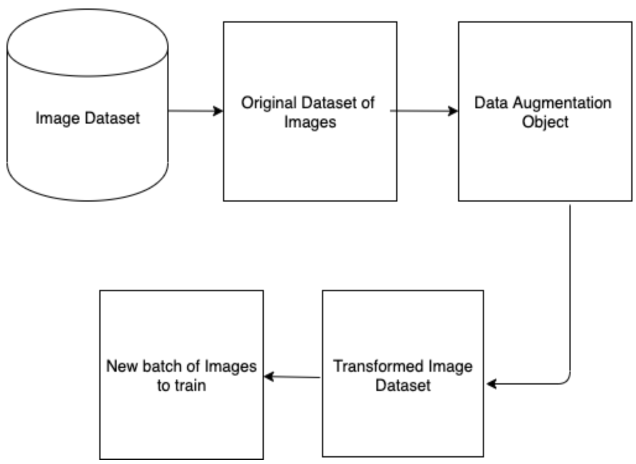


Figure: Data Augmentation on batch of images

In our model, we have used the in-place data augmentation technique using the Keras library. The image data generator class provides this functionality. This class returns the newly transformed images. It is called in-place because the transformation is done at the runtime. This class ensures that at each and every epoch we see new transformations in our data which ultimately ensures high training accuracy.

B. TRANSFER LEARNING

Transfer learning is a methodology in machine learning that uses a pre-trained neural network. It makes use of knowledge gained while training on one dataset to be used on other datasets. It uses parametric values like weights and biases obtained while working on some dataset and use these values to train on our new dataset. In transfer learning, we reuse the feature extraction part of some model and train again on the classification part using our own dataset. In this way, we achieve high accuracy with significantly less computation complexity and lesser training time.

The model that we have used here for transfer learning is Inception v3 model. It is a 48 layer deep neural network. This model has been trained over a million images from ImageNet Database belonging to around 1000 different classes. We have used TensorFlow hub library to perform transfer learning in our model. We further added a flatten and a dense layer to our network. It further reduces the computation burden and avoids overfitting of data.

C. PERFORMED TESTS

We have used the validation set to measure the performance of our model. The proposed pixel size for our validation images is 299 X 299 pixels. The metric used to measure the performance of our model is accuracy and cross-entropy loss function. Lesser the value of our loss function and higher the accuracy, better is our model. To train our network we have used the augmented images produced by our model so that our model trains on variations of data. For testing, we use the original images from our dataset to be used for measuring performance. We have also measured the training accuracy of the model. Since data augmentation provides a unique set of data at each epoch, so our model trains on high variations of data. This is the reason why our model gives a high training accuracy.

IV. RESULT

Using our proposed convolution neural network, we are able to achieve an overall testing accuracy of 97.61% with a loss value of 0.35. The results obtained are achieved by training our model on a much larger dataset generated using data augmentation technique on publicly existing coffee leaf images dataset, as convolution neural networks perform best when trained upon larger and varied datasets.

We further experimented and observed that using a learning rate value of 0.005, and an epoch value of 10, most accurate results were obtained. Initially, during successive iterations, the accuracy increased significantly and a drop in the loss value is observed and afterwards, no significant change in accuracy was observed. The table below represents how accuracy and loss improve with each iteration. With every epoch, the accuracy increases and loss decrease significantly. After iteration 10, there is no significant improvement in the accuracy of our model although loss value continues to decrease slightly.

No. of Epochs	Time (in sec)	Accuracy	Loss Value
1	925	35.52	1.3248
2	801	73.43	1.3109
5	731	94.93	0.3632
8	726	97.01	0.3560
10	715	97.61	0.3535
12	714	97.61	0.3530

Figure: Table representing varying accuracy and loss values with different epoch values

Further, the graphs of loss and accuracy with varying epochs also clearly show that the loss, accuracy values becomes nearly constant as more iterations are made.

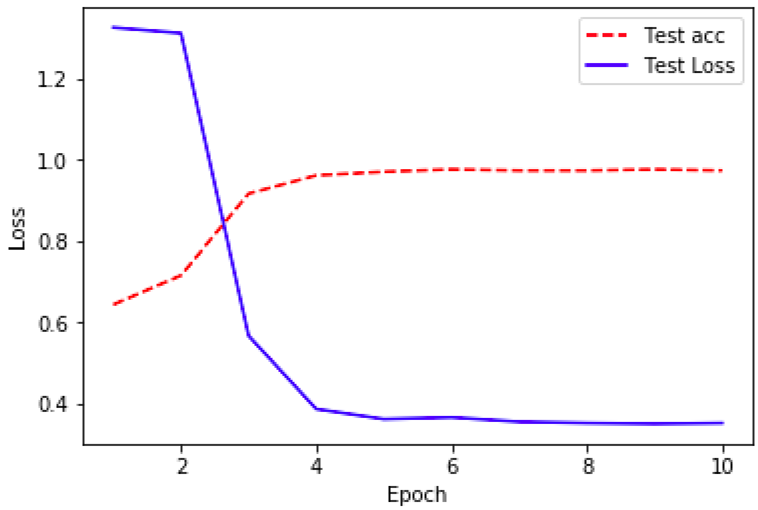


Figure: Graph representing how accuracy and loss vary with epoch value

According to good practice principles, comparison of results with other similar research papers is a must. Considering the fact that we have not directly used the publicly available coffee leaf images dataset but rather used Data Augmentation technique to enhance the variety and quantity of our training data, our model is much more reliable and accurate. We have used the Transfer Learning method and the model we have used has been trained over a million images from ImageNet Database belonging to around 1000 different classes. Moreover, we have also added flatten and dense layers to the network which also help in reducing overfitting. Finally, after comparing our result with various recently published research papers in similar areas, it can be seen that our methodology has provided with better results than [8][11][12].

V. CONCLUSION AND FUTURE WORK

This research paper presents the use of deep convolutional neural networks in the detection and classification of healthy and unhealthy coffee plants. The unhealthy plant leaves are classified into 4 classes – Coffee Leaf Rust, Cercospora Spots, Leaf Miner, Phoma which are commonly occurring biotic stresses in coffee plants. One of the main contributions in our work is to propose a neural network for coffee plant disease detection with a very high level of accuracy. Encouraging results were achieved using our methodology.

In summary -

1. Identification and classification of coffee plant diseases by using the above proposed neural network achieves a record breaking accuracy of 97.61% .

2. With the help of this methodology, coffee plants farmer could be helped in timely identification of the diseases which will help in increasing the coffee production output of India.

As future work, we aim to explore the requirements and necessary changes to our neural network to be able to accommodate various other new diseases in coffee plants and also account for variable environmental factors.

To make an even better prediction and achieve a higher accuracy for wide variety of diseases in coffee plants, additional variables such as temperature, air moisture, time of plantation and location weather will be needed. In this research, those variables have not been accounted for due to the limitation of a reliable dataset.

VI. References

[1] "Statistics on Coffee" 2018-2019. [Online]. Available: https://www.indiacoffee.org/coffee-statistics.html

[2] "Coffee Stats Highlights" . [Online]. Available: https://www.statista.com/outlook/30010000/119/coffee/india#market-globalRevenue

[3] Ricardo Balardin (December 14th 2010). Factors Affecting Fungicide Efficacy in the Tropics, Fungicides, Odile Carisse, IntechOpen, DOI: 10.5772/13583. Available from: https://www.intechopen.com/books/fungicides/factors-affecting-fungicide-efficacy-in-the-tropics

[4] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Computational Intelligence and Neuroscience, 2016, 1–11. doi:10.1155/2016/3289801 . Available from: https://www.hindawi.com/journals/cin/2016/3289801/

[5] "Coffee Rust " by editors of encyclopaedia britannica. 2018. Available from : https://www.britannica.com/science/coffee-rust

[6] P. Marcos, N. L. Silva Rodovalho and A. R. Backes, "Coffee Leaf Rust Detection Using Genetic Algorithm," 2019 XV Workshop de Visão Computacional (WVC), São Bernardo do Campo, Brazil, 2019, pp. 16-20.

[7] L. Han, M. S. Haleem and M. Taylor, "A novel computer vision-based approach to automatic detection and severity assessment of crop diseases," 2015 Science and Information Conference (SAI), London, 2015, pp. 638-644.

[8] U. B. Korkut, Ö. B. Göktürk and O. Yildiz, "Detection of plant diseases by machine learning," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4.

[9] J. Shijie, W. Ping, J. Peiyi and H. Siping, "Research on data augmentation for image classification based on convolution neural networks," 2017 Chinese Automation Congress (CAC), Jinan, 2017, pp. 4165-4170.

[10] Yosuke Toda and Fumio Okura, "How Convolutional Neural Networks Diagnose Plant Disease," Plant Phenomics, vol. 2019, Article ID 9237136, 14 pages, 2019.

[11] Marcos, A. P., Rodovalho, N. L. S., & Backes, A. R. (2019). Coffee Leaf Rust Detection Using Convolutional Neural Network. 2019. XV Workshop de Visão Computacional (WVC). doi:10.1109/wvc.2019.8876931.

[12] S. Ramesh and D. Vydeki, "Rice Blast Disease Detection and Classification Using Machine Learning Algorithm," 2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), Ghaziabad, India, 2018, pp. 255-259.