# Disease Detection in Coffee Plants Using Convolutional Neural Network


Manoj Kumar
*Computer Engineering Dept.*
*Delhi Technological University*
New Delhi, India
Email: mkumarg@dce.ac.in

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

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

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



**Abstract— Every year, the crop output of India is significantly affected due to delayed detection of diseases in crops. This research is a contribution to the farmers in their battle against coffee plants diseases. It will help the farmers in timely detection of coffee leaf diseases, resulting in increased coffee production output of India. In the last few years, the use of expert systems, machine learning techniques, genetic algorithms has been proposed for efficient and accurate detection of diseases. Such methods provide faster results but the accuracy may not always be sufficient for practical purposes. Many coffee plant diseases like Leaf Rust, Cercospora Spots have clear visual symptoms and thus can be extracted and their classification can be done. Convolutional Neural Networks (CNNs) has proved its efficiency and accuracy in the field of image classification, object detection and pattern recognition. Hence, we realized that it can act as a powerful tool in the diagnosis of coffee leaves diseases since these symptoms have clearly distinguishable patterns. Thus, we have proposed a Convolution Neural Network model which utilizes the technique of Transfer Learning, reducing the training time taken by the model significantly. Further, to achieve a higher success rate, Data Augmentation technique is applied to enlarge the dataset used to train the network. The proposed model has achieved a high accuracy of 97.61%.**



***Keywords— Convolutional Neural Network, Coffee Plant Disease, Data Augmentation, Transfer Learning.***


## I. INTRODUCTION

Coffee is one of the most popular beverages and an important commercial product which is cultivated in more than 50 countries across the globe. Every day an estimated 2.25 billion coffee cups are consumed throughout the world.[1] 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[2]. With an estimated 319,500 metric ton of coffee produced in the year 2018-19[3], of which almost 80% is exported throughout the globe. The amount of revenue generated from the coffee industry is projected to be at USD 987m in the year 2020 and its market is projected to increase annually at 7.4%. [4]

Coffee plants are highly vulnerable to various pests and diseases. Crop loss due to such diseases in coffee plants is estimated at 20-30% of the country's total production[5]. Chemical control’s success on any disease depends on the disease pathogenesis stage at which the fungicide is applied [1]. Inexperienced and careless usage of pesticide may result in increased resistance of the pathogens in the long run, severely affecting the capability of plants to fight. Thus, the detection of plant diseases in an accurate and timely manner is an essential aspect of precision agriculture [2].

Soft computing can play a crucial role in the accurate and timely detection of such biotic stresses that attack the leaf of a coffee plant. Various techniques such as genetic algorithms, machine learning, use of expert systems, computer vision and image processing have been an area of active research to detect such diseases [3][4]. These methods have not been widely adopted due to low efficiency and success rate. In [5], the authors have used both image processing and artificial neural network to achieve a success rate of 90%. Some plant diseases do not have any conspicuous visual symptoms, however most common diseases found in coffee leaves have clear visual symptoms as shown in the Fig. 1. For example, the symptoms of coffee leaf rust (CLR) disease are oily spots on the upper surface of coffee leaf which are minute and yellowish in color, and expands further into a large circular spot that with time, turn to bright orange to red and at last, to brown with a yellow border[6].

Deep Learning have completely dominated the field of image classification for the last few years, thus the proposed Convolution Neural Network is used to detect coffee leaf

[1] https://en.wikipedia.org/wiki/Economics_of_coffee
[2] https://en.wikipedia.org/wiki/Coffee_production_in_India
[3] https://www.indiacoffee.org/coffee-statistics.html
[4] https://www.statista.com/outlook/30010000/119/coffee/india#market-globalrevenue
[5] https://economictimes.indiatimes.com/news/economy/agriculture/india-to-get-594349-to-control-leaf-rust-in-coffee/articleshow/2466942.cms
[6] https://www.britannica.com/science/coffee-rust

diseases and categorize them into 5 classes – Healthy, Diseased Leaves with Cercospora spots, Phoma, Coffee Leaf Rust(CLR), and Leaf Miner with a high success rate.

The subsequent sections in the paper are organized in the following order: Section II describes the dataset used for training the model in detail. Section III explains the architectural components like Inception V3 model, Data augmentation technique and working of the proposed model. Section IV describes the various tests and experiments performed. In section V, the result is presented. Section VI contains a conclusion along with the future work we aim to explore in this domain.

## II. DATASET

The dataset [6] contains a wide variety of images of coffee leaves for disease detection. The images contain both disease-free and diseased leaves, affected by different types of biotic stresses. Dataset is divided in two parts – leaf dataset consisting of original images of the leaves and other one containing cropped images of symptoms. Fig. 1 shows the division of the dataset- disease categories with the corresponding number of leaf images.

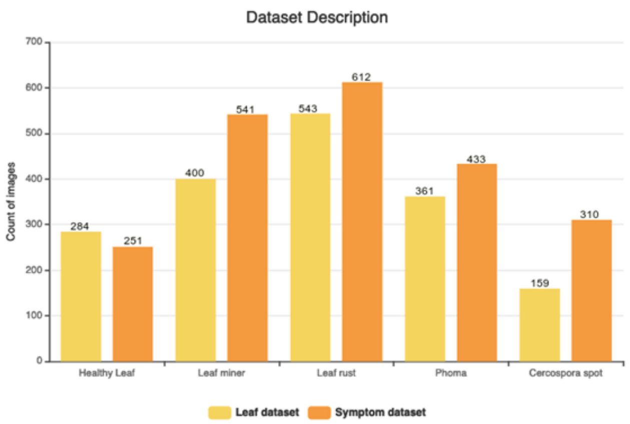


Fig. 1. Dataset Description

Details of each dataset is described below:

Leaf dataset: It contains images of coffee leaves and a CSV file with labels as different biotic stresses. All original images have the dimensions of 2048x1024 pixels.

Fig. 2. Sample of coffee leaves in Leaf Dataset

It consists of a total of 1747 images divided into 5 different classes - Healthy leaves, Infected leaves with Phoma, Leaf Miner, Coffee Leaf Rust and Cercospora Spots. Fig. 2 shows a sample of images found in this dataset.

Symptom dataset: A total of 2147 cropped images of the symptomatic parts of leaves are present. It is divided as per the named labelled classes - healthy leaves, leaf miner, leaf rust, phoma and cercospora. These are the 5 categories into which we have classified our coffee leaves. Fig. 3 shows a sample of images available in this dataset. It is further divided into training, validation and test folders.

(a) Healthy Leaf

(b) Leaf Minor

(c) Leaf Rust

(d) Cercospora Spot

(e) Phoma

Fig. 3. Sample of cropped images showing only symptomatic leaf parts from Symptom Dataset

## III. NETWORK ARCHITECTURE AND WORKING

The proposed network uses a Sequential model for classification. This Sequential model has been implemented using the Keras library in python language. The model has approximately 21 million parameters. Mini-Batch Gradient Descent (MBGD) is the optimization algorithm used to update the parameters. It performs much faster compared to conventional gradient descent algorithms. Fig. 4 represents the proposed neural network architecture. The images provided for training are input into the model in batches. The batch size value is kept as 32 with epoch value as 20. 1488 images belonging to 5 different classes is used for training the model and the image size has dimensions of 299x299 pixels.

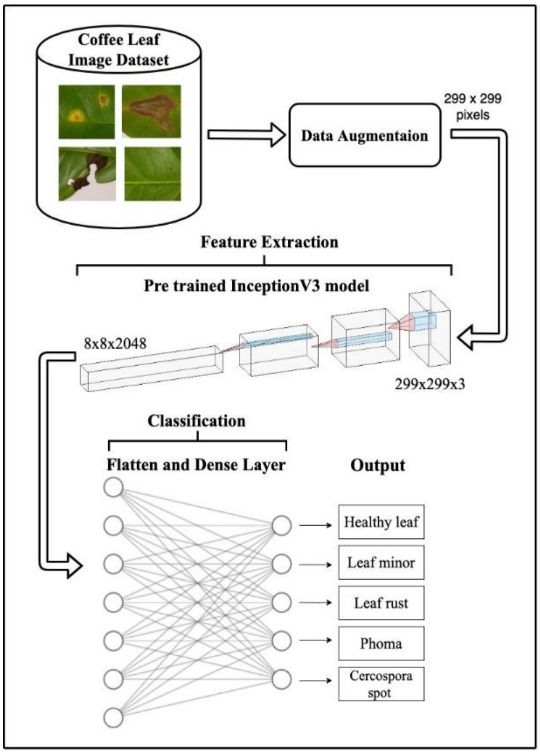


Fig. 4. Network Architecture

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

### A. Data Augmentation

Data Augmentation is used to enhance the generalizability of the model. It applies augmentation techniques to produce a set of new images that are modified versions of our input data by application of image transformations, contrast changes, blurring images without changing their labels. It creates a varied and enlarged dataset and so the model is able to learn more features in a robust manner [7] and also prevents the problem of overfitting.

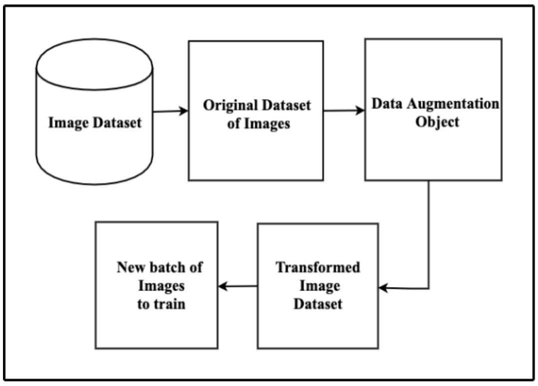


Fig. 5. Data Augmentation on the batch of images

In the proposed model, we have used the in-place data augmentation technique using the Keras library in python. Implementation is called in-place because the transformation is applied on runtime i.e. it ensures that at each and every epoch new transformed image is input to the network which ultimately ensures high accuracy.

### B. Transfer Learning

Transfer learning is a methodology 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 a large dataset and uses this knowledge to train on the new dataset. Image classification involves two processes, feature extraction and classification. With the help of transfer learning, one can reuse the part of the model involved in the feature extraction process and train the model for the classification part on a new dataset.

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 changes the dimensionality of data and prevents the problem of overfitting of data. It has been used in the classification part.

### C. Optimization Algorithm

Optimization algorithms help in the minimization (or maximization) of a given mathematical function F(x) which depends on the values of model's parameters and is used in carrying out computations for the output values. One such algorithm is Gradient Descent which falls under the umbrella of first-order optimization algorithms.

$$\theta = \theta - \alpha \nabla_{\theta} C(\theta; x_i, y_i) \quad (1)$$

where θ is optimization parameter, C (θ; xi, yi ) is the cost, (xi, yi ) represents the data-point from the training set.

Mini-Batch Gradient Descent algorithm is used and the dataset is divided into batches of 32 images each. It combines the advantages of both Stochastic Gradient Descent and Batch Gradient Descent algorithms. The disadvantage of BGD is that it runs over the whole dataset to take a single step in the direction of minima. In contrast, SGD computes gradient for each randomly chosen datapoint which is faster but loses the advantages of vectorization and is error-prone also.

### D. Categorical Cross-Entropy

Loss functions are used to evaluate the performance in the 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:

$$L(y, \hat{y}) = -\sum_{j=0}^{m} \sum_{i=0}^{n} (y_{ij} * log\ (\hat{y}_{ij})) \qquad (2)$$

where $\hat{y}$ is the predicted value and y representing the actual value, m and n represent the total number of data points and the total number of classes respectively.

## IV. EXPERIMENTS

To implement this model, we have divided the dataset for training, validation and testing. 70% of images are used for training the model, 15% each for validation and testing. To create a varied dataset, data augmentation is implemented. The new output images are of dimensions 299x299 pixels. To decrease the training time and increase the performance, we have used InceptionV3 structure which is a pre-trained network. The convolutional layers that are present in it have pre-trained weights and biases. After each epoch, a new set of images are produced that are passed to the inception model. It forms the feature extraction part of the network.

Along with it, a dense and flatten layer are added to our sequential model. These layers serve in the classification part of the model. The flattening layer is used to convert a multidimensional output from a convolution layer to a linear output such that it can be directly fed into the neural network layer or to regular dense layer. Then, the dense layer performs the calculations and applies the activation function. Fig. 6 shows the success rate and loss value obtained when different activation functions are applied. Other than ReLU, all the activation function gave a high success rate with Softmax function providing the maximum success rate.

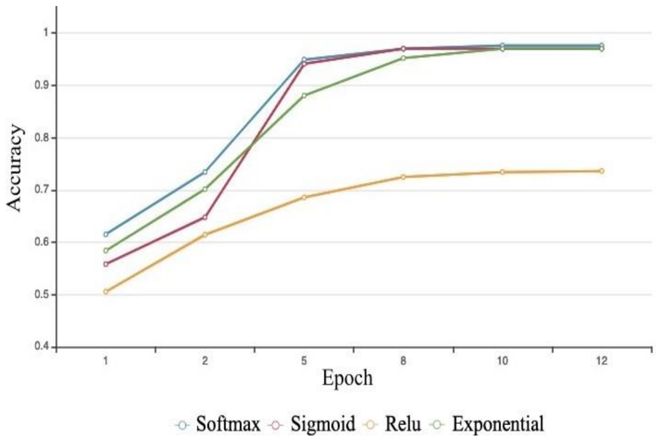


Fig. 6. Accuracy vs Epoch graph for the different activation function

Categorical cross-entropy is the loss function used. We also tested the model for different values of learning rate as presented in Table I.

TABLE I. ACCURACY AND LOSS VALUE VARIATION WITH DIFFERENT LEARNING RATES

| Learning Rate | Accuracy (%) | Loss Value |
|---|---|---|
| 0.0001 | 95.82 | 0.48 |
| 0.001 | 95.22 | 0.42 |
| 0.005 | 97.61 | 0.35 |
| 0.01 | 97.31 | 0.36 |
| 0.1 | 74.03 | 1.73 |
| 1 | 32.01 | 10.1 |

Accuracy of the model increased with increase in learning rate at first, reaching a maximum at 0.005 with an accuracy of 97.61%, and started to decrease further. Hence, final value of the learning rate in the model is 0.005. Mini-batch gradient descent is used as the optimizer algorithm, with a batch size of 32 for maximum accuracy as shown in Table II.

TABLE II. BATCH SIZE VS ACCURACY

| Batch Size | Accuracy (%) | Loss Value |
|---|---|---|
| 4 | 94.03 | 0.49 |
| 8 | 95.22 | 0.49 |
| 16 | 97.31 | 0.36 |
| 32 | 97.61 | 0.35 |
| 64 | 97.01 | 0.37 |

## V. RESULT

Using data-augmentation, transfer learning and MBGD as an optimizer, the proposed convolution neural network, has achieved an overall testing accuracy of 97.61% with a loss value of 0.35.

Table III represents how accuracy and loss improve with successive iterations. 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.

TABLE III. VARIATION IN ACCURACY AND LOSS VALUE WITH EPOCH

| Epoch Number | Accuracy (%) | Loss Value |
|---|---|---|
| 1 | 61.52 | 1.5752 |
| 2 | 73.43 | 0.7284 |
| 5 | 94.93 | 0.3632 |
| 8 | 97.01 | 0.3560 |
| 10 | 97.61 | 0.3535 |
| 12 | 97.61 | 0.3532 |

Further, Fig. 7 also clearly show that the value of loss and accuracy with varying epochs becomes nearly constant as more iterations are made.

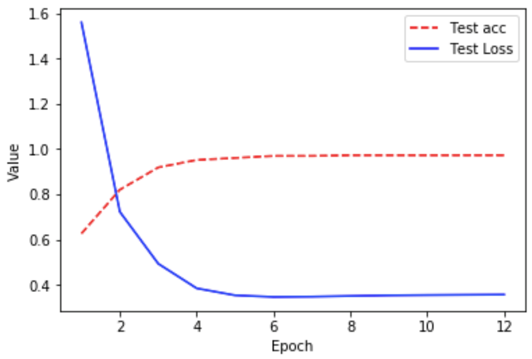


Fig. 7. Graph representing variation in accuracy and loss value with epoch

We have not directly used the coffee leaf images dataset but rather used Data Augmentation technique to enhance the variety of training data, resulting into a much more reliable and accurate model and also avoids the problem of overfitting. The Transfer Learning method is applied which makes the training process faster and efficient. Moreover, flatten and dense layers are added to the network which helps in the classification process. Table IV shows a comparison between various other techniques.

TABLE IV. COMPARISON OF DIFFERENT TECHNIQUES

| S. No. | Technique Used | Accuracy (%) | Loss Value |
|---|---|---|---|
| 1 | Fuzzy Logic Based Expert System (using Decision Tree) [5] | 85% | Not Available |
| 2 | Convolutional Neural Network (CNN) [8] | 90% | Not Available |
| 3 | CNN with Data augmentation [9] | 95% | 0.10 |
| 4 | CNN with data augmentation and transfer learning (using Inception V3 architecture) * | 97.61% | 0.35 |

*proposed model

The values of different hyper-parameters used in the proposed neural network are in Table V.

TABLE V. VALUES OF VARIOUS HYPER-PARAMETERS

| Hyper-parameter | Value |
|---|---|
| Activation function | Softmax |
| Loss function | Categorical cross-entropy |
| Momentum | 0.9 |
| Batch size | 32 |
| Learning rate | 0.005 |
| Optimization function | Mini-batch gradient descent |

## VI. 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.

In summary -

1. Identification and classification of coffee plant diseases by using the proposed neural network achieve 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 higher accuracy for a wide variety of diseases in coffee plants, additional variables such as temperature, air moisture, time of plantation and location weather are required.

## REFERENCES


[1] Ricardo Balardin. 2010. *Factors affecting fungicide efficacy in the tropics, fungicides*, odile carisse, intechopen, doi: 10.5772/13583.

[2] S. A. Miller, F. D. Beed, and C. L. Harmon, “Plant disease diagnostic capabilities and networks,” Annual Review of Phytopathology, vol. 47, pp. 15–38, 2009.

[3] 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.

[4] 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.

[5] Suhartono, Derwin & Aditya, Wahyu & Lestari, Miranty & Yasin, Muhammad. (2013). *Expert System in Detecting Coffee Plant Diseases*. International Journal of Electrical Energy. 156-162. 10.12720/ijoee.1.3.156-162

[6] Krohling, Renato A.; Esgario , Guilherme J. M.; Ventura, José A. (2019), “BRACOL - A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests”, Mendeley Data, v1. Available from: http://dx.doi.org/10.17632/yy2k5y8mxg.1

[7] . 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.

[8] 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 .

[9] 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.