Deep Learning-Based Detection of Mango Leaf Diseases Using Convolutional Neural Networks

Ms. Tanishta
Manipal Institute of Technology
Bengaluru, India

Dr. Megha P Arakeri
Manipal Institute of Technology
Bengaluru,India

Mr. Vishesh Goyal
Manipal Institute of Technology
Bengaluru India

***Abstract* - Examining whether any leaf is diseased or not by visual inspection of the naked human eye is often unreliable and incorrect. Technological developments in the form of techniques like deep learning have proved to be extremely useful in agricultural threat detection by the detection of leaf diseases. It is a well-known fact that whenever a plant is affected by any disease, its first symptoms can be identified in the leaves of the tree. In this paper, we have provided research work done on mango plant leaf diseases using the EfficientNet Model and Fast-AI framework. The Fast-AI framework provides a better ensemble for mango leaf disease recognition (MLDR).**

I. Introduction

India, being an agriculture-based nation, generates a huge portion of national income (20%) from the agricultural sector and offers a wide range of jobs. The market size of mangoes in this industry can be estimated by the fact that there are 1500 different varieties of mangoes each with its own distinct flavour. Furthermore, India is a leading producer of mangoes producing 50% of the mangoes being produced worldwide. Mango tree conservation is necessary for the preservation of biodiversity not only in the Indian Subcontinent but worldwide as well [1]. Mango plant diseases, however, make it extremely difficult to produce enough fruits to meet the demand. If ailments of mango trees could be precisely recognized, this industry may grow more. But it's difficult to diagnose these illnesses with the unaided eye. Mango trees suffer from various infections such as anthracnose, bacterial canker, cutting weevil, die back, gall midge, sooty mould, powdery mildew, and rust leaf disease. Like any other plant, the early symptoms can be first observed on the leaves. Mango Bacterial Canker can result in a 10-100% loss in mango yield. Powdery mildew alone has damaged up to 23% of unsprayed mango trees globally, while Anthracnose has caused losses of up to 39%. These diseases lead to significant economic and agricultural losses. Thus, it is imperative that the computer vision is employed to identify unhealthy plants in their early stages for quick remedies and to prevent the spread of infection to the whole plantation. Given the above discussion, there is a strong need for the automatic detection, identification, and classification of mango plant diseases. This article addresses data augmentation to increase the dataset, tracking the colour, physical appearance and spots on the leaves all the while handling background variation, properly segmenting the unhealthy leaves and detecting which disease has affected the plant. This model has shown considerably increased accuracy increasing reliability and efficiency.

II. Related Work

For citrus plants, various researches have been done for image classification and disease detection [2]. Utilizing leaf images, complex Deep Convolutional Neural Nets (DCNN) have been constructed to classify disease using neural networking [3]. A VGG convolutional neural network was developed and used by Ferentinos et. al [4]. to compare images of healthy and diseased leaves. Scientists used a plant village dataset that comprised 14 healthy and 38 diseased leaf classifications and investigated four deep convolutional network designs: ResNet, Inception V4, VGG 16, and DenseNets[5]. DenseNets reduced calculation time while still producing higher classification results. Using images from the Digipathos repository as validation, research on deep learning algorithms in plant pathology explores key characteristics and issues affecting network effectiveness [6]. Transfer learning via deep learning techniques is being used to identify ailments in mangoes and grapes for developing precision farming. A pre-trained CNN was used for automatic feature extraction which resulted in a 99% accuracy for grape leaves and 89% accuracy for mango leaves. A multilayer CNN was used on the J&K dataset comprising 1070 images which diagnosed the anthracnose disease in mango leaves [7].

A high level of accuracy of 87.55% was achieved using deep learning techniques using pre-trained models and classifiers. Using real-time data on tomato early blight disease, the study illustrates the usefulness of the framework made by merging the Fast-AI framework with pre-trained models, providing a useful tool for early disease diagnosis and prevention [8]. Techniques like CLAHE, image sharpening, and GAN-based resampling have been heavily used to improve class imbalance in plant disease datasets that help boost accuracy. The method obtains an average accuracy of 97.69% using ResNet-50[9].

Iqbal et al. [10] examined segmentation, recognition, and identification methods and discovered that the majority of them are still in their infancy. They talked about the models, advantages, restrictions, difficulties, and current approaches of machine learning (ML) for the detection and recognition of diseases.

Shin et al. [11] combined an artificial neural network (ANN) with speeded-up robust features (SURF) for strawberry leaf powdery mildew detection, achieving 94.34% accuracy. Bhatia et al. [12] used the Friedman test to rank classifiers and post hoc analysis with the Nemenyi test, finding MGSVM to be the superior classifier with 94.74% accuracy. While the study found MGSVM to be superior with 94.74% accuracy, the use of the Friedman test and Nemenyi post hoc analysis might not fully account for the complexities of real-world data. Additionally, the performance metrics beyond accuracy, such as precision, recall, and F1-score, are not mentioned, which are critical for understanding the classifier's performance comprehensively.

Shah and Jain [13] detected cotton leaf diseases by extracting colour features. However, the focus on extracting colour features for detecting cotton leaf diseases may limit the model's effectiveness, as it might not capture the full range of relevant information, such as texture or shape features. Also, the accuracy of the method is not provided. There might also be a lack of comparative analysis with other feature extraction methods. Sharif et al. [14] recommended a computerised system for segmenting and classifying citrus plant diseases, achieving 90% accuracy using a hybrid feature selection technique. Although a 90% accuracy was achieved using a hybrid feature selection technique, this study might have limitations in the diversity of the dataset used. The term "hybrid feature selection" is broad, and without details, it is unclear how features were chosen and whether this selection process was optimal.

III. Material and Methods

Details of the proposed models' training, testing and validation architecture has been mentioned below. It discusses about the basic algorithm, preprocessing of images, tools and frameworks and training architecture that has been used.

A. Basic Algorithm

The training and making of the model were a tedious task as depicted in Fig. 1. Procurement of diseased mango leaves was one of the initial challenges in the project. The 240x240 pixel images were used from various datasets, websites and government data that was available online. All the images were pre-processed using Fast-AI module's 'aug_transformations' function.

The pre-trained EfficientNet model and the Fast-AI framework were used in the construction of the suggested mango leaf disease recognition model which was retrained on the augmented dataset. The EfficientNet pre-trained model with Fast-AI enables the distinction between diseased and healthy leaves. The proposed model follows five key steps: acquiring images, pre-processing them, separating the dataset into training and testing datasets, determining the optimal learning rate for the model based on the training data, and finally training the model. To ensure reliability and accurate results, the new images must also undergo similar image preprocessing.

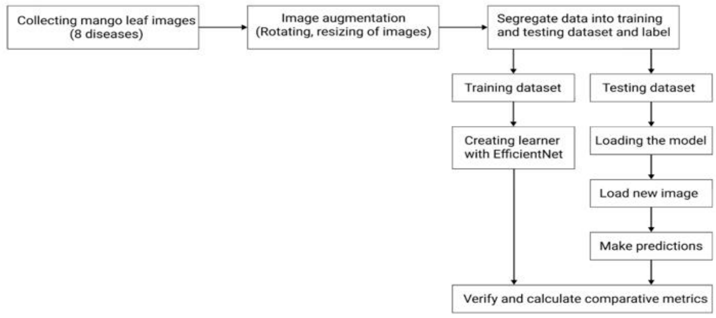

Fig. 1. Flowchart for disease detection model

B. Preprocessing of images

The concerned dataset was pre-processed using the Fast-AI framework. The dataset that was already labelled and classified according to different diseases was loaded. All the images of this dataset were ensured to be strictly 240x240 pixels. Two images were randomly picked from each of the categories for being used as test images as an unfamiliar input. Training and validation subsets were then created by splicing the rest of the dataset in a 4:1 ratio. As a next step, the image augmentation techniques were applied to standardised 448x448 pixel images. Random flipping, resizing, rotation, enhancement, zooming and contrast enhancement were done using the 'aug_transformations' method of Fast-AI library. However, no distortions were allowed so as to keep the image patterns and spots clear thereby increasing the accuracy and reliability of the model. Finally, all image sizes were once again downsized to 224x224 pixels so that the training process happens quickly without any hindrances.

C. Tools and Frameworks

PyTorch-based Fast-AI model was used to create the models for image identification, segmentation, and classification. The built-in data cleansing functionality and widget help in faster computing. Another of its primary merits is its user-friendly approach, which makes troubleshooting much more effortless. Fast-AI has a visual component that includes all the operations needed to construct a database and train vision-based models. The module named 'vision.data' has a distinctive utility function called 'ImageDataLoaders', which accepts input from image directories, image lists, .csv files, and other sources. The input is then divided into training and validation. Further sub-modules such as 'vision.transform', and 'vision.learn', provide methods for transforming/augmenting and training data accordingly. The 'DataBunch' approach facilitates the transformation from data blocks to training system models by allowing quick operations. After receiving data, 'vision.learner' provides all the functions needed to train the model. Fast-AI data pipeline architecture is represented in Fig. 2.

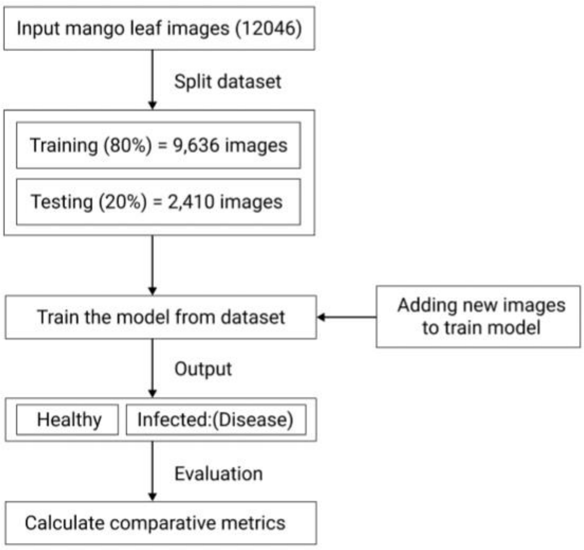

Fig. 2. Fast-AI data pipeline Architecture

The model is majorly based on the Convolution Neural Network (CNN) which is a subpart of the Fast-AI library. Fast-AI is a deep learning library that provides high-level components that can quickly provide results in standard deep learning domains. Fast-AI is organized around two main design goals: to be approachable and rapidly productive, while also being deeply hackable and configurable. This is a carefully layered architecture, which expresses common underlying patterns of many deep learning and data processing techniques. Fast-AI includes a new type dispatch system for Python along with a semantic type hierarchy for tensors, a GPU-optimized computer vision library that can be extended in pure Python and much more.

D. Training Architecture

The training architecture for classifying images of leaves involves a carefully orchestrated process. A test image is taken as input and then processed on the user's machine to determine if it depicts a healthy or infected leaf. The entire flow of this training model is visually depicted in Fig. 3, providing a clear overview of the process. A pre-processing step is initially undertaken to ensure uniformity in data, standardising all images to possess identical dimensions and contrast levels.

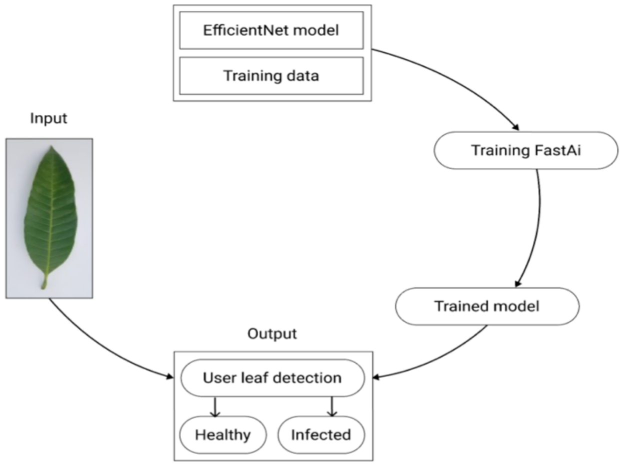

Fig. 3. Training architecture for MLDR model

A random splitter divides the dataset into training and validation sets, allocating 20% of the data for validation to assess the model's performance. Labels are extracted from the file paths using a function named parent label, and the images are organised in folders by class. Additionally, the images are resized to a uniform size of 224x224 pixels using the Resize transformation. Next, the data loaders are created, which facilitate the loading of data in batches during training.

The model was trained using pre trained EfficientNet-B0 model. It is a part of the EfficientNet family. The family introduces a novel scaling method that balances the trade-off between model accuracy and computational efficiency. In traditional CNN models, to increase the training accuracy, either the number of layers, number of channels in layers or image resolution was increased. However, the EfficientNet model improves all three simultaneously in a systematic way while being more computationally efficient.

The scaling factors are determined by a formula based on a fixed coefficient Φ, which controls the overall scaling factor, and three parameters α, β, and ɣ that determine how to proportionally scale the depth, width, and resolution :

- Depth $\propto \alpha^{\Phi}$
- Width $\propto \beta^{\Phi}$
- Resolution $\propto \gamma^{\Phi}$

The values are chosen such that

$$\alpha \times \beta^2 \times \gamma^2 \approx 2 \qquad (1)$$

EfficientNet-B0 is based on MobileNetV2 architecture which uses inverted residual blocks and depth wise separable convolutions. This helps to reduce the computational resources and parameters required. The MobileNetV2 architecture uses a bottleneck approach, implying middle layers are larger in size as compared to the input and output layers. This model uses the Swish Activation function:

$$swish(x) = x \times sigmoid(x) = \frac{x}{1+e^{-x}} \qquad (2)$$

This is a smooth monotonic function that improves model accuracy. Fig. 4 depicts how the EfficientNet-b0 model takes in the input image, analyses it and gives the output.

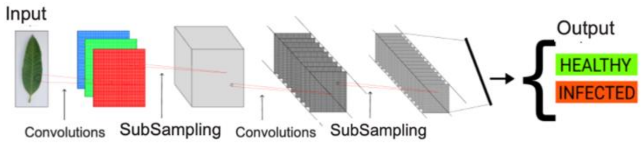

Fig. 4. EfficientNet-b0 Architecture

Each of the models was run for 6 epoch cycles to ensure greater accuracy and training time required was taken into consideration when judging the models.

IV. Results and Discussions

The results achieved following the above process and discussions that concluded are given below.

A. Dataset

A collaborative dataset from datasets available on Kaggle comprising a total of 12046 images was used to train this model. Both healthy and infected images from our dataset are shown in Fig.5. All images are labelled to their corresponding classes based on the category. Fig. 6 shows how some of the images looked after applying the augmentery transformations.


Fig. 5(a) Healthy


Fig. 5(b) Sooty Mould


Fig. 5(c) Rust Leaf


Fig. 5(d) Powdery Mildew


Fig. 5(e) Gall Midge


Fig. 5(f) Cutting weevil


Fig. 5(g) Anthracnose

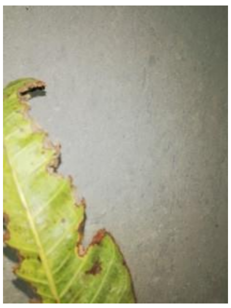
Fig. 5(h) Bacterial Canker

Fig. 5. Sample Healthy and Infected Images from the dataset


Fig. 6(a) Real


Fig. 6(b) Transformed (Rotation, Contrast)

Fig. 6. Sample Real and Augmented Images from the dataset

B. Results

All the models showed remarkable and improving results with each epoch. Their increased accuracy is depicted in the graph of Fig. 7.

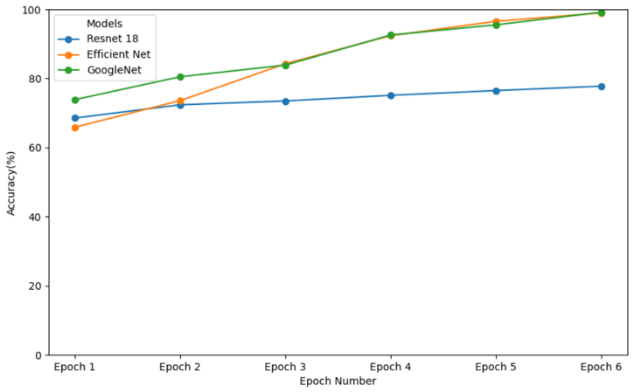

Fig. 7. Accuracy vs Epoch Cycles

These results made it quite evident that the GoogleNet and EfficientNet-B0 architecture outperformed the ResNet-18 model. Also, while the EfficientNet model took an average of 128 minutes per epoch for training, GoogleNet took an average of 463 minutes per epoch. Taking the time efficiency-accuracy trade off into consideration for industrial deployment, GoogleNet should also be rejected.

After that, the model's performance must be analysed using various metrics, including accuracy, precision, recall, and confusion matrix. The confusion matrix was used to know about the number of healthy and infected leaves. These numbers were then used to calculate the metrics, the F1 score (primary because it balances precision and recall), accuracy, precision and recall.

1. Confusion Matrix: A detailed table illustrating the model's performance across different classes, showing true positives, false positives, true negatives, and false negatives. Fig. 8 represents the confusion matrix achieved.

	PREDICTED NEGATIVE	PREDICTED POSITIVE
ACTUAL NEGATIVE (HEALTHY)	True Negative (TN) 1,251	False Positive (FP) 102
ACTUAL POSITIVE (INFECTED)	False Negative (FN) 91	True Positive (TP) 10,601

Fig. 8. Confusion Matrix

- Total healthy leaves (Negative class): 1,354
- Total infected leaves (Positive class): 10,692
- Total leaves: 12,046

TP is a true positive, FN is false negatives and FP is a false positive

2. Accuracy: The proportion of correctly classified images out of the total images.

$$Accuracy = \frac{Correct\ Output\ Data}{Total\ Input\ Data} \times 100 \qquad (3)$$

3. Precision: The ratio of true positive predictions to the sum of true positive and false positive predictions.

$$Precision = \frac{TP}{TP+FP} \times 100 \qquad (4)$$

4. Recall: The ratio of true positive predictions to the sum of true positive and false negative predictions.

$$Recall = \frac{TP}{TP+FN} \times 100 \qquad (5)$$

5. F1 Score: The harmonic mean of precision and recall, providing a single metric for performance evaluation.

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision+Recall} \qquad (6)$$

Following these calculations, the following results were achieved:

- Accuracy: 98.97% = 0.9897
- Precision: 99.05% = 0.9905
- F1 Score: 99.10% = 0.9910

The comparative analysis of the proposed approach with the available reference algorithms is made and given in Table1.

Table 1. Performance Comparison of New and Existing Models

Study	Accuracy	Precision	F1 Score
Proposed Work	98.97%	99.05%	99.10%
Kumar et al. [15]	91.20%	81.22%	88.13%
Simhadri et al. [16]	92.64%	84.66%	92.52%
Kshitij Dhawan et al. [17]	99.98%	98.65%	97.53%

This visual representation provides a clear and informative comparison of the models, offering valuable insights into their respective strengths and areas for potential improvement.

V. Conclusion and Future Scope

A cutting-edge computer vision framework was used to import and train Images quickly and deliver the results without any latency. Further, a model was proposed to detect the Mangifera Indica Leaf Disease and implemented using the EfficientNet transfer learning model, thus providing a high accuracy rate of 99.05%. The proposed model was also compared with the existing techniques mentioned by the researchers with performance metrics accuracy, precision and F1 Score.

The successful implementation of this model can significantly aid farmers and agricultural stakeholders in early disease detection and management, ultimately improving mango production and quality. Future work will focus on expanding the dataset, incorporating additional disease classes, and exploring the real-time deployment of the model through mobile applications and edge devices for on-site disease detection.

References

[1] National Horticulture Board (India). Available: https://nhb.gov.in/report_files/mango/mango.htm

[2] Iqbal, Zahid, et al." An automated detection and classification of citrus plant diseases using image processing techniques: A review." Computers and electronics in agriculture, 153 (2018): 12-32. Available: An automated detection and classification of citrus plant diseases using image processing techniques: A review - ScienceDirect.

[3] Golhani, Kamlesh, et al." A review of neural networks in plant disease detection using hyperspectral data." Information Processing in Agriculture, 5.3 (2018): 354-371. Available: A review of neural networks in plant disease detection using hyperspectral data - ScienceDirect.

[4] Ferentinos, Konstantinos P." Deep learning models for plant disease detection and diagnosis." Computers and electronics in agriculture, 145 (2018): 311-318. Available: Deep learning models for plant disease detection and diagnosis - ScienceDirect

[5] Too, Edna Chebet, et al." A comparative study of fine-tuning deep learning models for plant disease identification." Computers and Electronics in Agriculture, 161 (2019): 272-279. Available: A comparative study of fine-tuning deep learning models for plant disease identification - ScienceDirect

[6] Barbedo, Jayme Garcia Arnal. " A review on the main challenges in automatic plant disease identification based on visible range images." Biosystems Engineering, 144 (2016): 52-60. Available: A review on the main challenges in automatic plant disease identification based on visible range images - ScienceDirect

[7] Singh, Uday Pratap, et al." Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease." IEEE access 7 (2019):43721-43729. Available: Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease

[8] Chug, Anuradha, et al." A novel framework for image-based plant disease detection using hybrid deep learning approach." Soft Computing 27.18 (2023): 13613-13638. Available: A novel framework for image-based plant disease detection using hybrid deep learning approach | Soft Computing

[9] Ojo, Mike O., and Azlan Zahid. " Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline." Agronomy 13.3 (2023): 887. Available: Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline

[10] Iqbal, Z.; Khan, M.A.; Sharif, M.; Shah, J.H.; ur Rehman, M.H.; Javed, K. An automated detection and classification of citrus plant diseases using image processing techniques: A review. Comput. Electr. Agric. 2018, 153, 12–32. Available: Classification of Citrus Plant Diseases Using Deep Transfer Learning

[11] Shin, J.; Chang, Y.K.; Heung, B.; Nguyen-Quang, T.; Price, G.W.; Al-Mallahi, A. Effect of directional augmentation using supervised machine learning technologies: A case study of strawberry powdery mildew detection. Biosyst. Eng. 2020, 194, 49–60. Available: (PDF) Comparison of Image Texture Based Supervised Learning Classifiers for Strawberry Powdery Mildew Detection | TRI NGUYEN-QUANG - Academia.edu

[12] Bhatia, A.; Chug, A.; Singh, A.P. Statistical analysis of machine learning techniques for predicting powdery mildew disease in tomato plants. Int. J. Intell. Eng. Inform. 2021, 9, 24–58. Available: A machine learning-based spray prediction model for tomato powdery mildew disease | Indian Phytopathology

[13] Shah, N. Jain, S. Detection of disease in cotton leaf using artificial neural network. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 473–476. Available: Detection of Disease in Cotton Leaf using Artificial Neural Network | IEEE Conference Publication

[14] Sharif, M.; Khan, M.A.; Iqbal, Z.; Azam, M.F.; Lali, M.I.U.; Javed, M.Y. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. Comput. Electr. Agric. 2018, 150, 220–234. Available: Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection - ScienceDirect

[15] Kumar et. al: Automated fruit identification using modified AlexNet Feature Extraction. Available: Automated Fruit Identification using Modified AlexNet Feature Extraction

[16] Simhadiri et. al: Deep learning model for rice leaf disease detection. Available: Deep learning for rice leaf disease detection

[17] Kshitij Dhawan et. al: Deep learning model using CNN. Available: https://www.researchgate.net/publication/369012949