Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 233 (2024) 22–34

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024)

Integrated Leaf Disease Recognition Across Diverse Crops through Transfer Learning

Utkarsh Yashwant Tambe[a,*], Dr. A. Shanthini[b], Pao-Ann Hsiung[c]

[a]*Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur (Chennai) 603203, India*
ty7171@srmist.edu.in
[b]*Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur (Chennai) 603203, India*
shanthia@srmist.edu.in
[c]*Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621301, Taiwan (R.O.C.)*
pahsiung@cs.ccu.edu.tw

Abstract

Global food security is seriously threatened by plant diseases and can cause severe economic losses to farmers. Automated detection of plant diseases using computer vision and machine learning techniques has become a popular research area due to its potential to provide faster and more accurate results than traditional methods. In this research, we propose a plant disease detection model based on transfer learning using the MobileNetV2. We evaluate the suggested method on a dataset consisting of 38 various kinds of diseases across 14 different plants. Our experimental findings indicate that the proposed method has a typical accuracy of 91.98% and outperforms additional cutting-edge CNN models for plant disease detection. The experimental findings support the suggested approach's validity and show that it effectively detects plant diseases. We also have put forth the evaluation metrics to investigate the model. The suggested method has potential applications in real-world scenarios and can help farmers detect diseases in their crops at an early stage, allowing them to take timely action to minimize crop losses.

Keywords: Plant Disease Detection; Crop Protection; Image Processing; Crop Health Monitoring; Computers in Agriculture.

1. Introduction to Study

Plant diseases can severely affect crop yields and lead to significant economic loss for farmers. It's vital to identify plant diseases quickly and accurately for timely intervention and treatment to minimize crop losses [1]. With the

10.1016/j.procs.2024.03.192

advancements in computer vision and machine learning, automated detection of plant diseases has become possible, providing faster and more accurate results [2].

Convolutional Neural Networks (CNNs) have emerged as a popular technique for plant disease detection due to their ability to learn complex patterns and features from images. Transfer learning, a technique that involves fine-tuning pre-trained CNN models on new datasets, has further enhanced the accuracy of plant disease detection [3].

In this study, we suggest plant disease detection based on transfer learning using the MobileNetV2 model [4]. Utilizing a dataset with 38 different disease kinds spread over 14 different plants, we assess the performance of the proposed approach. The data was sourced from one source, which is open dataset. This study's main objective is to create an accurate and efficient automated plant disease detection model that can help farmers detect diseases in their crops at an early stage, allowing them to take timely action to minimize crop losses. In order to accomplish this, we focused on the following scientific advancements:

- Building of a precise deep learning model for the detection of numerous crop diseases in their early stages using photographs available in the dataset.
- Convey a comprehensive performance analysis of the MobileNetV2 CNN model for this disease classification task.
- Providing a comparison of MobileNetV2 and other cutting-edge models.
- Developing an explainable AI approach as it will provide aid to farmers for recognizing diseases of the plants without any confusion.

In this study, we collected a dataset consisting of images of healthy leaves from around 14 different plant species. The images in the dataset are captured using high-resolution camera under controlled lighting conditions to ensure consistency across the dataset.

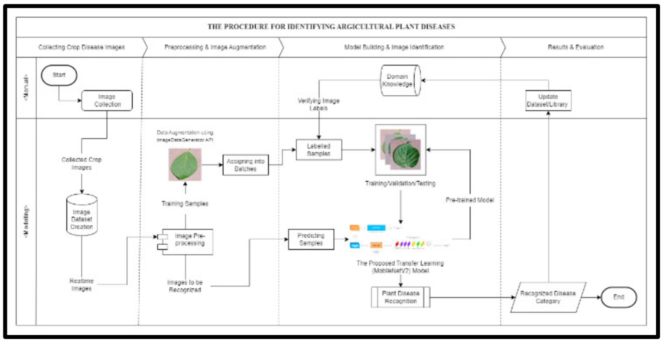

Figure 1: Experiment Overview (An Overall Process of Crop Disease Recognition)

Here is a breakdown of the remaining sections of this study: A synopsis of related research in this field of plant disease detection using deep learning approaches is provided in Section 2. The dataset utilized in the present research is described in Section 3, including exploratory data analysis, pre-processing & augmentation, and data splitting. Section 4 presents the proposed plant disease detection system, including details on the MobileNetV2 model, transfer learning, and training procedures. Section 5 provides experimental results and analysis, including performance evaluation, comparison with state-of-the-art models, and sensitivity analysis. Finally, the work is concluded and future research directions are covered in Section 6. Workflow of the experiment can be understood clearly from fig. 1.

2. Previous Research in this Field

Deep learning has shown great potential for revolutionizing agriculture by providing accurate, efficient, and automated solutions to a wide range of challenges faced by farmers and agricultural researchers. Sabrol and Satish [5] stated the use of image processing for plant disease recognition and classification, specifically for tomato plants. The authors conducted a study where they classified five types of tomato diseases and healthy tomato plants based on characteristics of colour, form, and texture taken from segmented images. The classification was done using a classification tree and yielded an overall accuracy of 97.3%.

In the next year, P. Kulinavar & V. Hadimani [6] discussed the use of image processing techniques for the detection and classification of leaf diseases in plants. The paper proposes a system that uses K-means clustering for segmentation, color and texture features for extraction, and Multiclass SVM classification for disease detection and classification. Then there was a survey conducted that tells us the different approaches used in machine vision for plant disease detection. The complexity of the visual scene determines the type of machine vision used. The paper also presents a review of various disease sorting methods used in plant leaf disease detection and a segmentation technique for identifying and categorizing plant leaf diseases was provided by Polke et al. [7].

Sadiku et al. [8] briefly introduce the idea of artificial intelligence and its application in the agriculture sector. It discusses the need for modern agriculture to conserve resources and adapt to climate change. The paper highlights the potential of machine learning in accurately diagnosing diseases and predicting crop diseases. However, it does not provide a detailed literature survey of previous research in this area. J. Chen et al. [9] had conducted a literature survey and reviewed various methods proposed for solving this task. They discovered that because of deep learning's excellent performance, it is increasingly the preferred method. The Inception module and ImageNet were already covered in the VGGNet's pre-training and are chosen by the authors for transfer learning, as well as the recently built neural networks were trained using their unique datasets. Comparing the proposed methodology to other cutting-edge techniques, it shows a significant performance improvement. The proposed approach's validity is supported by experimental findings, and it effectively detects plant diseases.

Also, the recent study by Paymode and Malode [10] discussed the agricultural application of artificial intelligence (AI), specifically for detecting and classifying diseases in tomato and grape leaves. The authors use a Convolutional Neural Network (CNN) method, improving performance metrics, in particular using the Visual Geometry Group (VGG) model. The paper also briefly mentions the utility of hyperspectral and multispectral data acquisition techniques in enhancing agricultural output and weed detection with deep learning technologies in vegetable plantations.

3. Data Overview for Model Preparation

The success of transfer learning is heavily reliant on the caliber and volume of the data used for fine-tuning. Insufficient data can lead to overfitting or poor generalization, while a well-curated and diverse dataset can improve the model's ability to capture and generalize features. Therefore, obtaining and preparing high-quality data is a critical step in the transfer learning process.

3.1. Data Acquisition

The collection of images that includes multiple plant species affected by various diseases was acquired from '**Kaggle**' platform named '**New Plant Diseases Dataset**' [11]. The dataset consists of **87,867** images. The dataset's images are of high resolution and captured under different lighting conditions, angles, and distances, making it a challenging dataset for image recognition tasks. The dataset contains **38** different classes of plant diseases, including Potato Early Blight, Tomato Mosaic Virus, Tomato Yellow Leaf Curl Virus, and Apple Scab, among others. Additionally, the dataset includes images of healthy plants to serve as a baseline for comparison. It offers a large and varied collection of images that may be used to test and refine image recognition algorithms. Images have been rescaled to a size of **224 x 224** pixels. The images present in the dataset are in JPEG/JPG (Joint Photographic Experts Group) format, renowned for its efficient compression and compatibility with various platforms.

3.2. Exploratory Data Analysis

The dataset contains **14 unique** plant species, making it a diverse collection that spans a variety of crops. This variety of plants contains a total of **38 different** classes of plant diseases, representing a diverse range of crops, regions and conditions. These include diseases such as early blight and late blight in tomatoes, bacterial spot in peppers and potatoes, black rot in apples and grapes, powdery mildew in cherries and squash, and leaf scorch in strawberries. In addition to the diseased samples, the dataset also includes healthy samples of several plants, such as soybeans, blueberries, and grapes, which can be used as a reference for comparison and among them, **26 are unique** diseases.

We observed that some classes, such as 'Orange Haunglongbing' and 'Soybean Healthy' have a relatively large number of images, while others, such as 'Corn (Maize) Cercospora (gray) Leaf Spot' and 'Cherry (including sour) Powdery Mildew' have fewer images. These class imbalances can pose a challenge for learning algorithms, which may struggle to accurately classify less-represented classes and also is depicted in fig. 2. As such, it is recommended that researchers consider balancing the class distributions or applying techniques such as data augmentation to improve model performance on underrepresented classes.

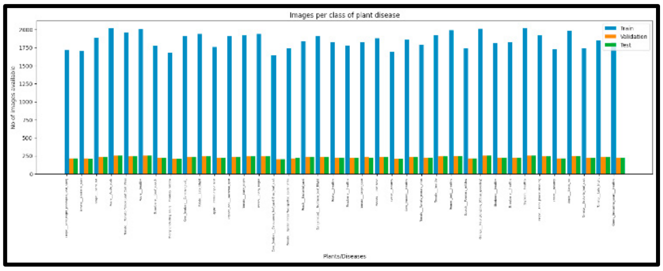

Figure 2: Distribution of Images per Disease Class

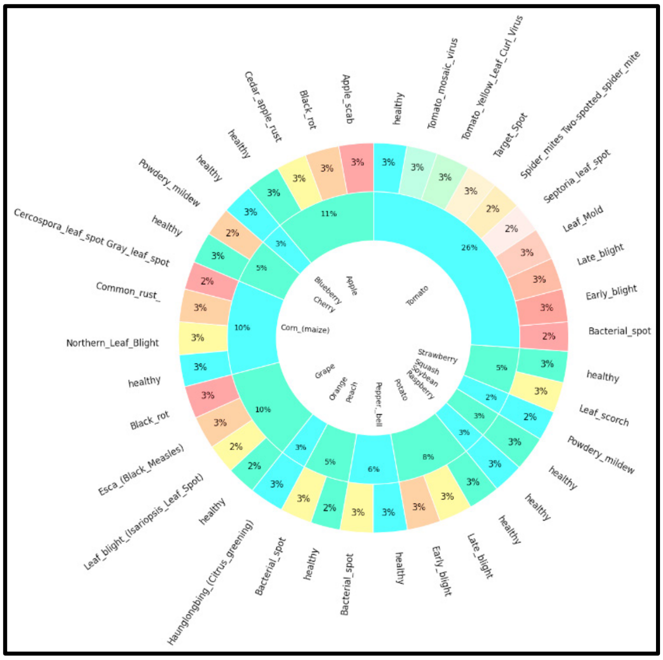

Figure 3: Pie-chart Distribution of Plant Species with respect to Number of Images

To gain a better understanding of the distribution of disease classes, we created a pie chart as seen in fig. 3 visualizing the percentage contribution of every disease class among all leaf images in the dataset. The resulting chart shows that majorly all the disease classes make up to **3%** contribution to the total leaf images in the dataset and some restrict till **2%**. In addition, we also sectioned the same pie chart visualizing the contribution of each plant species to the entire dataset. This chart shows that Tomato and Apple are the most well-represented species in the dataset, with each making up roughly **26%** and **11%** respectively of the total images. These pie charts provide valuable insights into the composition of the dataset and can help guide future research and analysis.

3.3. Data Preprocessing and Augmentation

To prepare the dataset for use in **training** a machine learning model, several image pre-processing techniques were applied to the dataset. The dataset was augmented by randomly zooming, shifting, and shearing the photos using the operations with a zoom range of 0.2, width and height shift range of 0.2, and a shear range of 0.2. These methods assist broaden the range of photographs in the dataset and lessen the likelihood that the model would overfit the training set. The horizontal flip = true technique was used to flip images horizontally, further increasing the variety of images in the dataset. Rotation range = 20 allows for random rotation of input images within a range of 20 degrees. It increases the training set's variability and strengthens the generalizability of the model and recognize objects under different orientations. The preprocessing_function = preprocess_input applies a standardized set of pre-processing operations to input images. This typically includes mean subtraction, normalization, or other transformations ensuring that the images are in a suitable format for efficient training and accurate predictions. These image pre-processing methods work together to increase the model's ability to generalize to new, unexplored data [12]. All the mentioned pre-processing techniques were applied to the dataset using Keras' ImageDataGenerator function which allows for data augmentation and pre-processing on-the-fly during model training. To expand the training dataset and decrease overfitting, data augmentation techniques are frequently used during training. However, using these strategies on the **validation** and **test** datasets would artificially inflate the model's accuracy on those datasets, thus they are not used there.

Instead, for the validation and test datasets, only the preprocessing_function = preprocess_input function is applied. It is not necessary to utilize additional data augmentation techniques on the validation and test datasets because they are not used to train the model. All of the images from the train, validation, and test datasets have been separated into 32, 16, 16 batches respectively with a target size of (224, 224).

3.4. Data Splitting

Table 1: Dataset Split

Category	**No. of Images**
Training	70295
Validation	8790
Testing	8782
Realtime Images	33
Total	**87900**

The dataset, which included **87,867** images, was arbitrarily divided in the ratios of **80:10:10** for training, validation, and test sets and the 33 real-time images which are present already in dataset are available separately. Table 1 provides a summary of the number of images in each collection for better understanding.

4. Proposing System for Detecting Plant Disease

4.1. Convolutional Neural Networks

Outstanding results have been attained by Convolutional Neural Networks, a subtype of neural networks created for image classification and recognition [13-15]. Instead of manually extracting the precise features as is the case with older methods, high-level robust properties can be instantly learned by CNNs from the source image. It has been demonstrated that CNNs can outperform traditional feature extraction techniques in identifying plant types and illnesses [16-17]. The majority of the layers in a standard CNN design are convolution layers, full connection layers, and pooling layers [18], which are outlined below:

- *Convolutional layers*: A CNN's convolutional layer, which uses various convolution kernel sizes to extract distinctive elements from images, is an essential part of the system. It is the backbone of the network, capturing elements like as edges, textures, and shapes at different scales and orientations via learnable filters or kernels. By layering these layers one after another, more sophisticated information is gradually abstracted and encoded, allowing the network to differentiate between various objects and visual characteristics. The layer extracts hierarchical visual representations as convolutional operations are conducted repeatedly. This enables the network to recognize complex patterns and subtleties, an essential step in applications like object detection and image categorization. Let H_i be a representation of the CNN feature map's i-th layer, then this is how the H_i can be produced:

$$H_i = \varphi\,(H_{i-1} * W_i + b_i) \ (1)$$

  where, H_i stands for the present network layer's feature map, H_{i-1} for the preceding layer's convolution feature (H_0 is the initial picture/photo), W_i for the weight of a layer, b_i for its offset vector, and $\varphi(\cdot)$ for the function of the ReLU (rectified linear unit).

- *Pooling layers*: Pooling layer's objective is to decrease spatial dimension; this can reduce processing costs and successfully handle the potential for over-fitting. One can calculate an output feature using eq. 2 on the l-th pooling layer's j-th local receptive field.

$$x_j^l = down\,(x_j^{l-1}, s) \ (2)$$

  where, 's' is the pooling size, x_j^{l-1} is the layer's feature vector before it, and down($\cdot$) is the down-sampling algorithm.

- *Fully-connected layers*: One or more fully connected (FC) layers are placed after the convolutional and pooling layers, and their function is to use the retrieved features for picture categorization. With the characteristics taken out of the preceding layers, the class prediction is frequently carried out using the **softmax** function. In mathematics, the softmax function is represented by:

$$softmax(z)_j = e^{z_j} / \sum_{k=1}^{K} e^{z_k} \ (for\ j = 1,2,\ldots,K) \ (3)$$

  where, 'K' stands for the z-vector's dimension.

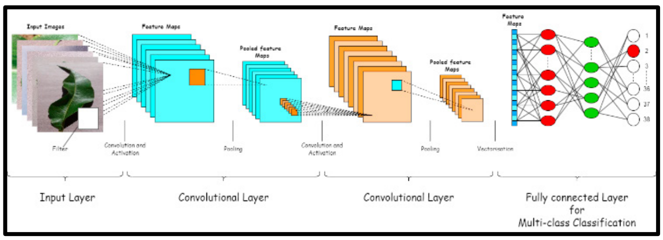

Figure 4: Building Blocks of Traditional CNN

These layers serve as the fundamental building elements of a CNN and are in charge of quantifying the final classes, as shown in fig. 4.

4.2. MobileNet V2 Model

MobileNetV2 [4] is a Google-developed convolutional neural network architecture which is built on the ideas of MobileNetV1 [19]. MobileNetV2 splits a traditional convolution into two sections using depthwise separable convolutions: both a pointwise and a depthwise convolution. As a result, there are fewer parameters in the network and allows for faster computation as compared with CNNs which can be computationally expensive and memory-intensive, making them less suitable for resource-constrained devices. In addition, MobileNetV2 uses linear bottlenecks with shortcut connections to increase the network's representational power while keeping the number of parameters low where, blue blocks represent composite convolutional building blocks.

In MobileNetV2, a more efficient module with an inverted residual structure is added. This time, thin layer non-linearities are removed. The use of MobileNetV2 as the base for feature extraction allows for the achievement of contemporary performances for object detection and semantic segmentation. In MobileNetV2, blocks come in two different varieties. There is a stride of one block. A block that can be reduced by two strides is an additional. Each of the two types of blocks has three levels. This time, 1x1 convolution with ReLU6 makes up the first layer. The depthwise convolution is the second layer. The third layer makes use of a further 1x1 convolution, although this one is linear. This claim states that only when ReLU is applied a second time do deep networks match the performance of a linear classifier on the non-zero volume region of the output domain [4].

4.3. Transfer Learning

Using CNNs that have been trained for one task as the basis for a model for another is a machine learning technique known as transfer learning [20]. We can initialize an already-trained network using massive labelled datasets, such as public picture datasets, etc., rather than starting from scratch and randomly initializing the weights. This study uses the viability of utilizing the pre-trained models found in the vast common dataset ImageNet, and then transferring them to the particular job discovered from the objective dataset. The following is a description of the transfer learning approach's main steps:

- *Determine the base Network*: Using the pre-trained CNN model, identify the transfer learning base networks and assign the network weights (W_1, W_2,…, W_n). One can download the weights of the lower layers from a well-trained CNN (https://www.tensorflow.org/api_docs/python/tf/keras/applications/mobilenet_v2).
- *Establish a Neural Network*: The network structure can be changed based on the bottom layers by, for example, replacing layers, adding layers, and removing layers from networks. A fresh network structure can be created in this manner.
- *Fine-tune the Neural Network*: As mentioned, utilizing our own dataset X and its corresponding labels Y, the recently built networks can be altered to reduce the loss function 'E' as in eq. 4

$$E(W) = -\frac{1}{n}\sum_{x_i=1}^{n}\sum_{k=1}^{K}[y_{ik}\, log\, P\,(x_i = k) + (1 - y_{ik})\, log\,(1 - P(x_i = k))]\ (4)$$

where, 'n' is the quantity of practice samples, the training sample index is 'I', and the class index is 'k', and 'W' stands for the fully-connected and convolutional layers' weighting matrices. The chance that input x_i belongs to the anticipated k-th class is given by $P(x_i = k)$.

It is common practice to determine the ideal 'W' through the target dataset's loss function 'E' being minimized using the stochastic gradient descent (SGD) [21] method, which is given in eq. 5.

$$W_k = W_{k-1} - a(\partial E\,(W)\,/\,\partial W)\ (5)$$

where, 'k' is class index & 'a' is rate of learning.

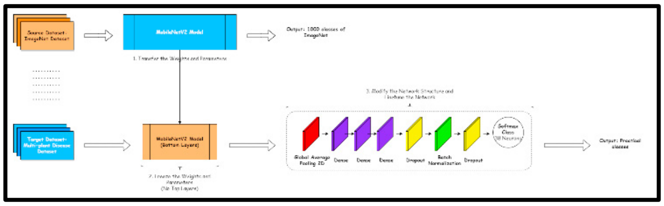

Figure 5: Transfer Learning Process of MobileNetV2

Since MobileNetV2 was already trained on ImageNet, we used it for transfer learning in this study. Using our own datasets, we then trained the newly constructed neural networks. The major operations of the method, which mixes weights from our own dataset and MobileNetV2, are shown in fig. 5.

After extracting picture characteristics using the pre-trained model, and by establishing layers before the classifier was trained, the top levels are shortened. The additional layers featured batch normalization, dropout, global average pooling, completely connected layers with ReLU activation, layers that are fully coupled/connected and activated by softmax, and a dense output layer. The model was developed using categorical cross-entropy loss and the Adam optimizer. We had excellent results with our strategy. Also, for a number of reasons, the suggested approach must use MobileNetV2 transfer learning. First off, MobileNetV2 is a tried-and-true architecture with a high degree of efficiency that is renowned for its capacity to extract complex information from photos [4]. Using this pre-trained model helps projects with limited resources by speeding up training and lowering the requirement for large amounts of labelled data. Furthermore, transfer learning makes it possible to apply knowledge from a sizable dataset to our particular plant disease classification problem, which improves the model's overall classification accuracy and its capacity to identify pertinent patterns [20].

5. Experimental Results and Analysis

5.1. Experiments on the Training Dataset

It should be noted that each photograph was shot with a straightforward background and that the illumination intensity was fairly constant. The same leaf is photographed in some instances from various angles. All of the photos have the same size, 224 x 224 pixels. This dataset's sample distributions are undoubtedly unbalanced. Squash leaf photos are few in quantity and much fewer than those of other classes in the tomato dataset. As a result, the sample photos of the squash classes are enhanced using data augmentation techniques to assure the balance of the sample data as mentioned above.

Particularly, to test the model's efficacy, a certain number of raw/unprocessed photos are kept in order to be cognizant of how the recommended technique might function given new, untested data. Using the technique suggested in section 4.3, we used the New Plant Diseases dataset for model training and validation. The train and validation, loss as well as accuracy of the approach is obtained in fig. 6.

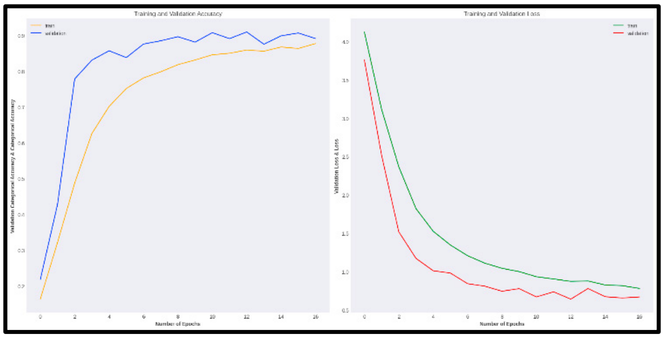

Figure 6: MobileNetV2, Right is Model's Loss while the left is Accuracy

Using the indicators, such as the Precision, Sensitivity, and F1-score, as given in eqs. (6), (7), and (8) we may evaluate how well the models work, taking into account the statistics of true positives, false positives, true negatives, and false negatives.

$$Precision = \frac{TP}{(TP + FP)} \quad (6)$$

$$Sensitivity/Recall = \frac{TP}{(TP + FN)} \quad (7)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2TP}{(2TP + FP + FN)} \quad (8)$$

FN (false negatives), on the other hand, the number of cases that should have been classified as class C but were instead misclassified, TP (true positives) being the number of occasions when the classifier correctly identified the examples as belonging to class C. The number of occurrences that were wrongly classified as class C but do not actually belong there are referred to FP (false positives). The number of instances that are correctly detected but actually do not belong in class C is referred as TN (true negatives). The suggested technique's testing accuracy reaches **0.9198** and the loss is **0.6258**; the evaluation metrics of the same are as shown in table 2.

Table 2: Evaluation Metrics for Identifying Results

Metrics	Score
Accuracy	0.919834
Precision	0.928369
Recall	0.916106
F1-Score	0.916839
Cohen's Kappa	0.913145
ROC-AUC	0.997361

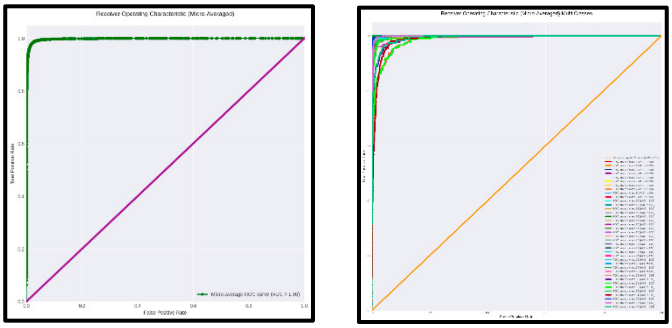

Figure 7: ROC-AUC curve - left is the ROC (micro-averaged) curve while, right is the ROC (micro-averaged) curve for every class

As previously mentioned, an experiment was run on fresh, previously undiscovered data to verify the success of the suggested approach. Table 3 shows that in the suggested experiment, the average prediction accuracy reached **91.98%**, demonstrating that the MobileNetV2 technique has a strong capacity to identify the various diseases compared to previous studies. In particular, the suggested method discriminated between many categories of plant illnesses in addition to identifying healthy and diseased plants. Thus, it can be inferred from the empirical analysis that the suggested strategy is successful in identifying a variety of disorders.

The ROC-AUC (Receiver Operating Characteristic - Area Under Curve) score is a widely used metric for classification models. It evaluates the model's ability to distinguish between negative and positive classes at various levels. The ROC curve plots TPR against FPR, and the AUC represents the model's overall performance. A high ROC-AUC score indicates accurate predictions. It helps evaluate deep learning models by providing insights into performance across different thresholds and identifying the optimal threshold for predictions [22]. This model’s ROC-AUC score is **99.73** and curve of the same is seen in fig. 7.

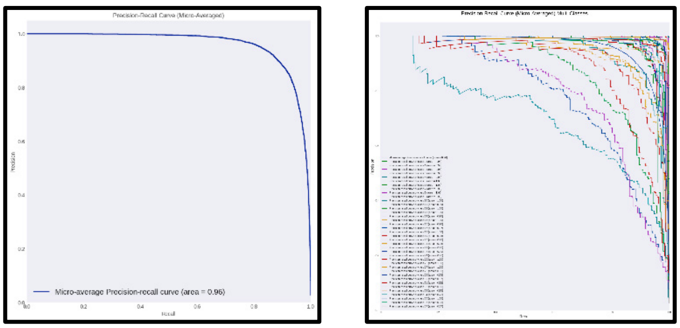

Figure 8: AP curve - left is the AP (micro-averaged) curve while, right is the AP (micro-averaged) curve for every class

The AP (Average Precision) score evaluates a binary or multiclass classification model's performance by measuring the area under the precision-recall curve at different thresholds. It provides a single value that represents overall performance, with higher scores indicating accurate positive predictions while minimizing false positives. This metric is useful for deep learning models, particularly when class distribution is unbalanced or when focusing on positive cases. It can also help to identify the optimal prediction threshold [23]. The AP score for this model is **95.93** and curve of the same as shown in fig. 8.

5.2. Experiments on the Test and Custom Dataset

The various metrics and their curves prove us that, the developed model is effective at identifying a variety of agricultural diseases available in the dataset and the same can be confirmed once again. Fig. 9 depicts all the disease classes and their predicted labels.

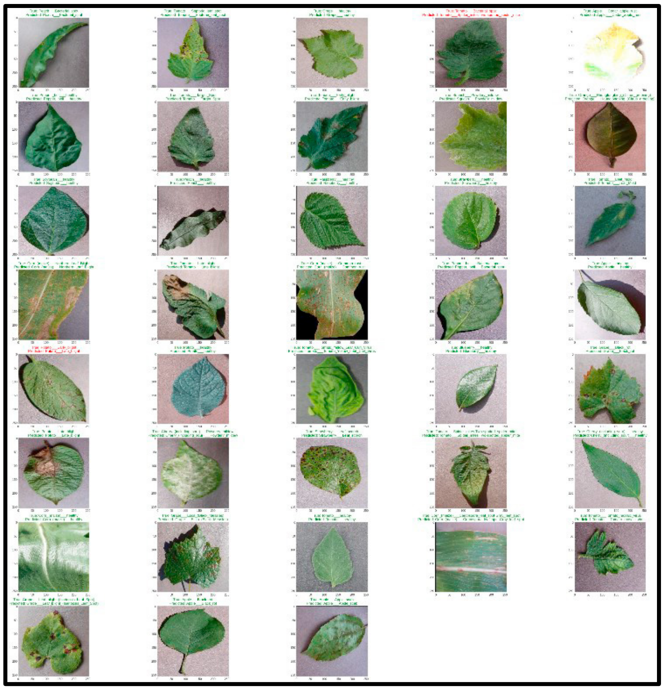

Figure 9: Plant Disease Images with their Predicted Labels

The model was also tested on real-time images available in the other set which are randomly sourced from Internet and to surprise, the model also performed very well here too. The model was also modified little so that it is able to tell the probability at which it is predicting a specific disease of the different plant classes.

Finally, let’s have comprehensive insights of the hyperparameters that governed our experimental setup. Specifically, we used a learning rate of 0.001 and a batch size of 32 during training. Although the original plan was to train for 50 epochs, we employed early stopping, which terminated training at the 18th epoch based on validation accuracy monitoring criterion. The optimization algorithm used was Adam, and L2 regularization with a parameter of 0.001 was applied to mitigate overfitting.

Table 3: Comparison between Proposed Study & Related Works

Study	**Method**	**Accuracy (%)**
Edna Chebet Too et al. [24]	VGG16 with Transfer Learning	81.92

Mohanty S. P., Hughes D. P., & Salathé M [15]	AlexNet with Training from Scratch	85.53
L. Poole and D. Brown [25]	VGG16 with Transfer Learning	87.34
Proposed Study	**MobileNetV2 with Transfer Learning**	**91.98**

6. Conclusion and Future Research Work

Plant infections are the primary impediment to the evolution of agriculture worldwide because they have a detrimental effect on how safely food is produced. Plant diseases might entirely halt a harvest in some cases. Therefore, disease detection in plants automatically in agricultural information is highly desired [26]. Rather than utilizing models that require a lot of computational resources, we proposed a novel methodology for the diagnosis of plant disease imaging by using MobileNetV2 with transfer learning. This method optimized both accuracy and efficiency and is well-suited for web and mobile applications. According to experimental results, the model excelled on both, the dataset and the customized photos. It has a testing accuracy of **91.98%** on the publicly available dataset outperforming previous models in the comparative analysis, see table 3. In order to automatically monitor and identify a greater variety of plant disease information, we plan to use it on mobile platforms in the future. For the interim, we want to use it for more useful things, including computer-aided diagnosis (CAD), among other things.

However, it's crucial to understand that the model have limitations in certain situations, such as when there are multiple leaves present in an image. In certain circumstances, the model might not be capable of reliably identify the crop-specific disease class. With the above proposed study, it started to predict false crop diseases somewhere. This shows the limitations of the model and needs to be overcome. This limitation could be addressed in future research by exploring ways to improve the model's capacity to differentiate leaves affected with different diseases and multiple leaves in an image. This study presents approach to using deep learning for plant disease identification, with potential for further development and improvement.

References

[1] Y. Fang and R. Ramasamy, "Current and Prospective Methods for Plant Disease Detection," *Biosensors*, vol. 5, no. 3, pp. 537–561, Aug. 2015, doi: https://doi.org/10.3390/bios5030537

[2] C. Jackulin and S. Murugavalli, "A comprehensive review on detection of plant disease using machine learning and deep learning approaches," *Measurement: Sensors*, vol. 24, p. 100441, Dec. 2022, doi: https://doi.org/10.1016/j.measen.2022.100441.

[3] W. Koehrsen, "Transfer Learning with Convolutional Neural Networks in PyTorch," *Medium*, Nov. 26, 2018. Accessed: Apr. 16, 2023. [Online]. Available: https://towardsdatascience.com/transfer-learning-with-convolutional-neural-networks-in-pytorch-dd09190245ce

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, doi: https://doi.org/10.1109/cvpr.2018.00474

[5] H. Sabrol and K. Satish, "Tomato plant disease classification in digital images using classification tree," *International Conference on Communication and Signal Processing*, Apr. 2016, 1242-1246. https://doi.org/10.1109/ICCSP.2016.7754351

[6] Pooja Kulinavar, Vidya I. Hadimani. Classification of Leaf Disease Based On Multiclass SVM Classifier, *International Journal of Advance Research, Ideas and Innovations in Technology, 2017*, pp. 321-327. https://www.ijariit.com/manuscripts/v3i4/V3I4-1205.pdf

[7] A. Polke, K. Joshi, and P. Gouda, "Leaf Disease Detection Based on Machine Learning," *Lecture Notes in Computational Vision and Biomechanics*, pp. 1875–1884, Jan. 2019, doi: https://doi.org/10.1007/978-3-030-00665-5_172

[8] M. N. O. Sadiku, C. M. M. Kotteti, and S. M. Musa, "Machine Learning in Agriculture," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, no. 6, p. 26, Jun. 2018, doi: https://doi.org/10.23956/ijarcsse.v8i6.713

[9] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, Jun. 2020, doi: https://doi.org/10.1016/j.compag.2020.105393

[10] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural networks VGG," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23-33, Jan. 2022, doi: https://doi.org/10.1016/j.aiia.2021.12.002.

[11] S. Bhattarai. "New Plant Diseases Dataset", Version 2. *Kaggle*, Apr. 2019. Accessed: Sept. 23, 2022. Available: < https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset>

[12] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, Jul. 2019, doi: https://doi.org/10.1186/s40537-019-0197-0

[13] C. Szegedy *et al.*, "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015, doi: https://doi.org/10.1109/cvpr.2015.7298594

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, Jul. 2017, doi: https://doi.org/10.1109/cvpr.2017.243

[15] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning Convolutional Neural Networks for fine art classification," *Expert Systems with Applications*, vol. 114, pp. 107–118, Dec. 2018, doi: https://doi.org/10.1016/j.eswa.2018.07.026.

[16] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, Sep. 2016, doi: https://doi.org/10.3389/fpls.2016.01419

[17] M. Dyrmann, H. Karstoft, and H. S. Midtiby, "Plant species classification using deep convolutional neural network," *Biosystems Engineering*, vol. 151, pp. 72–80, Nov. 2016, doi: https://doi.org/10.1016/j.biosystemseng.2016.08.024

[18] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, Dec. 2017, doi: https://doi.org/10.1016/j.neucom.2017.06.023

[19] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv (Cornell University)*, Apr. 2017, doi: https://doi.org/10.48550/arxiv.1704.04861

[20] A. Lumini and L. Nanni, "Deep learning and transfer learning features for plankton classification," *Ecological Informatics*, vol. 51, pp. 33–43, May 2019, doi: https://doi.org/10.1016/j.ecoinf.2019.02.007

[21] M. Mehdipour Ghazi, B. Yanikoglu, and E. Aptoula, "Plant identification using deep neural networks via optimization of transfer learning parameters," *Neurocomputing*, vol. 235, pp. 228–235, Apr. 2017, doi: https://doi.org/10.1016/j.neucom.2017.01.018

[22] Google, "Classification: ROC Curve and AUC | Machine Learning Crash Course," *Google Developers*, 2019. Accessed: Apr. 26, 2023. Online. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

[23] Scikit-learn, "Precision-Recall," *Scikit-learn*, 2019. Accessed: Apr. 26, 2023. Online. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

[24] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, Mar. 2018, doi: https://doi.org/10.1016/j.compag.2018.03.032

[25] G. Poole and D. Brown, "Investigating Popular CNN Architectures for Plant Disease Detection," *2021 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD),* pp. 1-5, Aug. 2021, doi: https://doi.org/10.1109/icabcd51485.2021.9519341

[26] R. N. Strange and P. R. Scott, "Plant disease: a threat to global food security," *Annual Review of Phytopathology*, vol. 43, pp. 83–116, 2005, doi: https://doi.org/10.1146/annurev.phyto.43.113004.133839