A Comprehensive Survey of Methods and Applications for Detecting Infected Plant Leaves

Kedar Sawant
Dept. Computer Engineering
Agnel Institute of Technology and Design
Mapusa, India
ksa@aitdgoa.edu.in

Sagar Naik
Dept. Computer Engineering
Agnel Institute of Technology and Design
Mapusa, India
skn@aitdgoa.edu.in

Anuya Naik
Dept. Computer Engineering
Agnel Institute of Technology and Design
Mapusa, India
21co08@aitdgoa.edu.in

Tanaya Sangodkar
Dept. Computer Engineering
Agnel Institute of Technology and Design
Mapusa, India
21co66@aitdgoa.edu.in

Savari Karpe
Dept. Computer Engineering
Agnel Institute of Technology and Design
Mapusa, India
21co76@aitdgoa.edu.in

Abstract— **Agriculture is the backbone of the Indian economy. Now that crop production is decreasing more quickly, farmers find it difficult to sell their goods when their output is decreased due to diseases that affect trees, particularly the leaves. To improve quality and productivity, it is imperative to treat any serious illnesses as soon as possible. This issue led to the creation of expert systems for disease prevention and innovative technologies for the detection and diagnosis of plant diseases. A detailed literature survey highlighted the applications of different architectures of deep learning models for identifying infected and healthy leaves. Most of the existing literature commonly emphasized on the use of machine learning techniques for activities like disease detection, fruit identification which are related to agriculture. This paper uses five deep learning models YOLOv8, Faster R-CNN, Swin Transformer, EfficientNet, and Custom CNN to classify whether the leaf is healthy or diseased. A dataset containing two classes of images of healthy and infected leaves has been used, and EfficientNet achieves the highest accuracy of 98.75% when compared to Faster R-CNN (87.98%), YOLOv8 (96.56%), Swin Transformer (97.00%) and Custom CNN (85.64%). However, more optimization is required, taking into account variables like dataset variations, computational performance, real-world applicability, and combining with IoT.**

Keywords— Custom CNN, deep learning, EfficientNet, Faster RCNN, Swin Transformer, YOLOv8

I. INTRODUCTION

Agriculture is the backbone of the Indian economy[1][2]. Crop production is growing rapidly at a rate that is going down at an even faster rate. This happens mainly because of pests and diseases, which impact the quality as well as yield of the crop produced [3]. Some diseases target the leaves only, and their impact adds up to make things worse, as they further retard the agricultural production process [4]. Due to this reason, traditionally farmers have been unable to diagnose this disease with any degree of certainty by visual inspection [5][6]. To secure accurate disease detection, experts were involved, which would inevitably increase the costs of the whole process [5].

However, it has been resolved by various techniques of image preprocessing and deep learning models [7]. Through computer vision, artificial intelligence, deep learning, and machine learning approaches, solutions have been made for effortless detection of diseases rather than expensive ones as other traditional methods; useful for crop quality detection, identification of plant disease, destruction of weeds, three-dimensional analysis, and many more [8]. This article offers a comparative review of deep learning models, YOLOv8, Faster R-CNN, Custom CNN, EfficientNet, and Swin Transformer for the detection of healthy and diseased leaves. The paper is segmented into the following sections: Section 2 gives a literature review of the work done previously, Section 3 is the proposed methodology, Section 4 is the results obtained, and Section 5 is the conclusion which also discusses the future scope.

II. LITERATURE REVIEW

The subsection analyzes the current literature around leaf diseases detection and classification techniques, with a specific emphasis on CNN models. There have been many approaches and solutions proposed and deployed in this space.

K. S. Chethan et al. [9] designed a mobile application for recognizing diseases in plants and suggesting treatment. Intermediate processing of images, segmentation, and classification using neural networks, are used to target eight common diseases including apple scab and rust. The app, built on top of Python and the Kivy library, enables farmers to send images of their plants to a server communicating via TCP, and be instantly diagnosed without the need to take samples and send them for analysis. They found that CNNs (Convolutional Neural Networks) were superior to SVMs (Support Vector Machines) giving them 99% accuracy. Pankaj Kumar et al. [10] proposed to use VGG-16 based CNN to classify mango leaves as healthy, infected and other types of leaves. Their model, which achieved 96.16% accuracy on a dataset pre-processed by cropping sprites, resizing the image, and applying histogram equalization to all images, outperformed previous travaux by some margin. Sharma Rohan et al. [1] using data augmentation and preprocessing created a CNN model in which the output can either be a healthy or diseased mango leaf. Thaseentaj Shaik et al. [2] the model achieved accuracy of 99.27%, classified 13 diseases and healthy leaves using the CNN models GoogLeNet, ResNet-50 and EfficientNet on the dataset of 1,344 leaf images which were captured from Chittoor, Shunlong Chen et al. [11] in order for faster recognition of strawberry disease, used GhostConv, Involution layers, CBAM and CARAFE to propose a new enhanced model of YOLOv5. Evaluated with 2,246 images of seven disease categories, the model achieved precision, recall and mean average precision, mAP@0.5, values of 93.3, 90.3 and 94.7, respectively. Diksha Tandekar et al. [4] better than the baseline YOLOv5 employed VGG-19 along with CNN networks for identification of plant diseases with an accuracy of 85.4% and 83% respectively. Using image processing and treatment suggestions, and a Tkinter based GUI, to analyze and diagnose, Sukruth S. Puranik et al. [12] constructed a mobile application model using MobileNetV3 to detect the disease in mango leaves in real-time. Trained on the MangoLeafBD dataset consisting of 4,000 images, it achieved 98% training and 96% testing accuracy, predicting most of the diseases correctly, Lawrence C. Ngugi et al. [13], classify lesions using GoogLeNet and segment both leaf and lesions using KijaniNet, achieving high accuracy 94.99% and strong segmentaion results (Leaf mIoU : 0.8448, Lesion mIoU : 0.6257) on dataset of 601 leaf images and 22,835 lesion images, Nimisha Manoharan et al. [14] in this study a dataset of 450 images of mango leaves were used with pre-processing and augmentation to improve accuracy. The accuracy was 98% for custom CNN, Venkatesh et al. [8] utilize V2IncepNet which is the variation VGGNet Architecture, VGGNet features combined in the Inception module to distinguish Anthracnose disease. Using Batch Normalization, Swish activation, and feature extraction measures the model achieved 92% accuracy; more than the traditional VGGNet by using a dataset of both self-captured and PlantVillage images, Vijayakumar Ponnusamy et al. [15] also proposed real-time tomato leaf disease detection based on YOLOv3 and CNN used to classify leaves as healthy or unhealthy. It was trained on 304 images (264 for training and 40 for testing), obtaining 82.38% accuracy is integrated with the economic Smart Glass for real time detection, Nosin Ibna Mahbub et al. [16] designed an LCNN model to classify eight mango leaf diseases with 98% testing accuracy, outperforming VGG16 and ResNet. They also designed the "Save Mango" Android app for disease diagnosis, Madhumini Mohapatra et al. [17] designed an RNN-based model for plant disease detection with 93.8% accuracy, outperforming SVM, NN, and CNN. Their method involves median filtering, Otsu thresholding, feature extraction, and a new AOCDO optimization method for enhanced RNN performance, N. Srikanth et al. [18] designed PDDC-Net, a ResNet-CNN-based model for rice plant disease detection and pesticide recommendation with 99.485% accuracy, outperforming SVM, Decision Tree, and Naïve Bayes. The model improves image quality using contrast enhancement and histogram equalization prior to classification and pesticide recommendation using a hybrid optimization approach, Shiva Mehta et al. [19] employed federated learning for decentralized training in plant disease detection to ensure data privacy for agricultural stakeholders. Their model, for five disease classes, achieved 97-98% accuracy, with high precision, recall, and F1-scores, Munshi Omar Faruque Rabbi et al. [20] designed a hybrid CNN model (InceptionV3 + DenseNet121) for mango leaf disease detection with 97.92% accuracy, outperforming DenseNet121, VGG16, and ResNet50. Employing image augmentation for robustness, their model performs well in reducing false positives and enhancing prediction reliability, Husnul Ajra et al. [21] employed ResNet-50 on a Kaggle dataset of potato and tomato leaves with 97% accuracy for healthy vs. unhealthy classification and 96.1% for disease detection, outperforming AlexNet. Their work also features a graphical user interface to recommend preventive measures for farmers.

Ankita Suryavanshi et al. [22] proposed to use of Federated Learning combined with CNNs to detect and classify 5 types of Brinjal leaf diseases. The dataset was collected from 5 different clients. The methodology included data augmentation, CNN based feature extraction and federated model training using the Federated Averaging algorithm. Client io_3 achieved an accuracy of 97.20% and Client io_5 achieved 81.51%. The research gap identified was privacy-preserving plant detection and highlighted Federation Learning potential in agricultural diagnostics. Nilay Jadav et al. [23] used an AI-powered quad-wheeled robot equipped with NVIDIA JETSON Xavier NX and YOLOv8. The dataset contained 11,498 annotated images for multiple classes of plant disease. The robot integrated IoT sensors for real-time environmental monitoring and autonomous navigation using GPS and IMU sensors. YOLOv8 achieved high precision, recall, and mAP also showed high detection accuracy. The gap that was captured in their research was automated greenhouse disease detection and the potential of AI and robotics for precision agriculture. Bin Yang et al. [24] introduced a triple-branch Swin Transformer (TSTC) with deep supervision and severity classification of plant disease. The AI Challenger 2018 dataset (30,215 images) was used, the model integrates multitask feature extraction, feature fusion, and deep supervision. Disease classification and Severity classification showed an accuracy of 99.00% and 88.73% accuracy respectively. The study enhances generalisability in plant disease detection and addresses the problem of joint disease severity classification. RVS Praveen et al. [25] developed a CNN-based plant disease classification system is developed using the PlantVillage dataset with 97.27% accuracy. It also calls for improved feature extraction, pre-processing, and dataset expansion for practical application. Given the need for privacy and scaling, Hari Kishan Kondaveeti et al. [26] discusses Federated Learning (FL) for Smart Agriculture on disease identification, yield prediction, etc. Although models such as CNNs and EfficientNet reached accuracies of up to 99.95%, there remain challenges related to data heterogeneity, as well as privacy and compatibility issues with existing systems.

Despite attaining high test accuracy for plant disease detection in the reviewed studies, they share common challenges. A lot of the models use computationally expensive architectures that preclude their deployment on resource constrained. Furthermore, most datasets lack diversity, consisting of only a few images taken in controlled settings, which can undermine robustness in real-world situations. Additionally on small datasets, manual labelling, low segmentation performance in challenging conditions, and computational inefficiency [13]. Because the features are weak or poorly defined, and sensitive to ambient lighting and background changes, the above models suffer from low-disease early detection. The YOLOv3 model suffers from low data availability, 82.38% moderate accuracy and hardware overheating while used in real-time applications [15].

III. Methodology

A. Dataset Creation

The dataset selection process retrieved only mango leaf images from the dataset "A Database of Leaf Images" found on Kaggle; there were 170 healthy leaves and 265 diseased leaves, respectively. Data augmentation techniques were used to increase the size of the original dataset to 4000 images to enhance the dataset's variability and improve in generalization performance of the model. It split the dataset into 2798 images for training, 802 images for validation, and 400 images for testing.

B. YOLO v8 (You Only Look Once)

YOLO is an advanced convolutional neural network, designed for real-time object detection and localization. Instead of using a region proposal network, YOLO integrates object classification and detection into a single regression task [15]. This innovative design transforms raw image data into bounding box coordinates and class probabilities directly through regression. Although the model sacrifices some accuracy for speed, its speed makes it highly effective. YOLO divides the image into an M×M grid and YOLO computes objects whose center lies inside one of these boundary cells. For every one of these boundary cells, for each cell on the grid, YOLO assigns a bounding box and its confidence measure, which tells how strong this presence likelihood is. The model uses five anchor boxes for each cell. It contains information like confidence scores, coordinates (center, width, height), and class identifiers. Unlike most models, YOLO applies a single neural network to evaluate the entire image at once. This allows it to make parallel predictions of probability and bounding boxes at various regions in the image. The overall design reduces detection efficiency and time complexity into a single forward pass of the network. The one recently developed version in YOLOs; YOLOv8 maintains a highly superior accuracy level concerning former versions while with increased efficiency [27]. Its lean structure focuses more on simplicity with only a one-stage approach used in unified feature extraction, object localization, and probability prediction. With its enhanced backbone consisting of convolutional layers and residual connections, YOLOv8 enhanced feature representation [28]. The "Focus" module also makes feature fusion optimal in detection. Rebuilding the head with an improved anchor box selection mechanism, besides a multi-scale prediction strategy, is used to perform object detection for objects of all scales and sizes. In this respect, YOLOv8 has performed excellently well, considering high accuracy in detection, higher speed of inference, and better memory usage efficiency [27]. For training, standard hyperparameter values including a learning rate of 0.01, 16 epochs, batch size of 16, and image size of 640px imgsz=640 have been selected to find an optimal balance between performance and computational efficiency. The values achieve good training time along with accuracy. There are several pre-trained models in varying sizes that come into focus regarding versatility and make it possible to choose the best one to suit specific user needs and available hardware capacities. Improvements put YOLOv8 in a highly robust and flexible framework for object detection tasks.

The YOLOv8 architecture is a state-of-the-art object detection technology, combining the most recent innovations. The backbone network is optimized to extract the most critical features from input images and maximize information capture. The intermediate neck layers enhance feature extraction and transformation, further improving the accuracy of object recognition. YOLOv8, the new detection head being anchor-free is the main advantage for faster predictions in bounding boxes and class probabilities over the older versions. It adapts the loss function at the time of training of the network for optimal performance across a wide variety of hardware platforms. Beyond the excellent hardware adaptability, YOLOv8 works fluently on a range of devices and is therefore very applicable for specialized use cases and versatile applications. These innovations together enhance the speed, accuracy, and performance of the framework, making YOLOv8 a leading solution for modern object detection challenges in computer vision and intelligent systems [27].

C. Faster R-CNN

Faster R-CNN, is used when detection should be accurate and yet in a reasonable time, is the most widely accepted method of object detection throughout the world. Faster R-CNN takes the base R-CNN structure to the next level by composing region proposal generation and object detection into a single unified architecture; hence redundancy is minimized, and inference is accelerated [13].

A deep convolutional neural network like VGG16, ResNet, or something between those architectures can be used as a feature extractor, which outputs a feature map of the input image. The feature map has very rich spatial and semantic information, which is very important for both object identification and precise location. It's the pre-training of very large datasets like ImageNet to be robust from a feature extractor standpoint, effectively maximizing the representation of the processing. Thus it slides over a proposed set of features, as proposed by some region proposal networks, and upon feeding such features, the RPN passes through the image of a particular region in a feature map where the bounding boxes are most likely to be contained. An anchor-based approach was applied, where anchors are reference points, defined here as boxes of various scales and aspect ratios. The RPN predicts an objectness score (whether the anchor contains an object or not) and refines the anchor coordinates for each anchor. Redundant overlapping proposals are eliminated with the help of non-maximum suppression (NMS) to keep only the most informative findings. The final stage in region-based CNN approaches is fully connected, and it not only classifies each region proposal into an object difference but also refines the bounding box predictions. This network uses a Region of Interest pooling layer to crop and resize features in the feature map, respectively, corresponding to each proposed region. These feature vectors of fixed size are forwarded through fully connected layers to obtain class scores and bounding box regression parameters. During training, hyperparameter values such as a learning rate of 0.001, 8 epochs, and a batch size of 2 were used to optimize the model's performance. Trained on the region proposals provided by the RPN, Faster R-CNN uses shared convolutional computations between the RPN and the fully connected layer to achieve high efficiency compared with earlier approaches, such as Fast R-CNN and R-CNN. Faster R-CNN, due to its coding-in-unified design, has been the basis framework in object detection with a wide range of applications, including autonomous driving, medical imaging, and surveillance. Its trade-off of efficiency and accuracy has influenced follow-ups such as models expanding its capabilities to tasks relevant for real-time object detection and complex situations such as multiscale object detection [28].

D. EfficientNet

EfficientNet model architecture is developed to strike a balance between optimal performance and efficiency within a specified layer configuration. The stem starts with a 3×3 convolutional layer with batch normalization as well as ReLU6 activation for the production of early low-level features. Mobile Inverted Bottleneck Convolution (MBConv) blocks constitute the foundational blocks of the network and are replaced with conventional convolutional layers in pursuit of the highest efficiency. These MBConv blocks also consist of depth-wise separable convolutions that efficiently diminish computational complexity without sacrificing representational power. MBConv blocks apply squeeze-and-excitation (SE) layers for selective scaling of important channels to scale the feature maps. Inspired by MobileNetV2, inverted residual connections retain low-dimensional representations before feature expansion within a block so that efficient gradients and improved learning are possible. EfficientNet-B2 enhances performance through the compound scaling method, in which network depth (layers), width (channels), and resolution (input image dimensions) are proportionally scaled in relation to a constant coefficient. B2 has a larger input size (260×260) compared to EfficientNet-B0 with depth and width adjusted for precision. The final head of classification includes a 1×1 convolution, global average pooling, and a fully connected SoftMax prediction head. With accurate convolutions, attention, and a balanced scaling approach, EfficientNet attains higher accuracy making it suitable for computationally intensive deep-learning applications to image classification, object detection, and segmentation.

E. Swin Transformer

Swin Transformer is a powerful vision model with hierarchical architecture for processing high-resolution images with shifted window-based self-attention. In contrast to traditional Transformers, in which global self-attention is employed, Swin Transformer divides an image into patches and computes self-attention on local windows with much lower computational cost. This hierarchical approach enables the model to scale economically and preserve fine-grained information while capturing long-range dependencies, thus being well-suited to applications such as image classification, object detection, and segmentation. The model architecture is a multi-stage processing pipeline. An image is initially divided into fixed-sized patches, and they are represented as feature vectors. Self-attention is computed inside non-overlapping windows to acquire local features efficiently. To allow cross-window interaction, the shifted window mechanism imposes overlap between layers such that information diffuses globally. As the model proceeds through its hierarchical steps, it increasingly pools features, refining both local and global representations. This cooperation between localized attention, window shifting, and hierarchical feature extraction enables the Swin Transformer to achieve state-of-the-art performance with the advantage of retaining computational efficiency.

F. Custom CNN

A custom Convolutional Neural Network (CNN) model was designed to classify healthy and infected leaves, using a dataset gathered at Goa College of Agriculture. Preprocessing techniques include histogram equalization, normalization, image resize to 320×320, and Gaussian blur for better feature extraction and model robustness. Data augmentation was executed to reach 3000 images in the dataset by applying horizontal and vertical flips, random 90-degree rotation, rotations within ±45 degrees, shift-scale-rotate transformations, and brightness-contrast adjustments. The dataset was further divided into training (70%), testing (20%), and validation (10%) for a structured evaluation of the model performance.

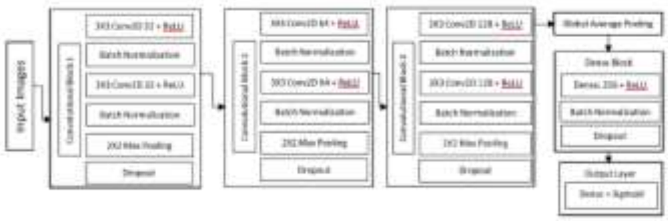

Fig. 1. Workflow Framework

The above Figure 1 shows that CNN was made up of three Conv2D blocks with two layers, ReLU activation, L2 regularization, batch normalization, max pooling, and dropout layers for overfitting regularization. The model concluded with a global average pooling layer, a dense layer and a final sigmoid output layer for binary classification. The model was trained using binary cross-entropy loss, optimized with Adam (learning rate =

0.0001). Initially, the model achieved an accuracy of 84.63% on the test set. To enhance performance, fine-tuning was conducted by freezing the first two convolutional blocks while keeping the last block and dense layers trainable, allowing the model to retain previously learned low-level features while refining high-level representations. Additionally, the learning rate was reduced to 0.00001 to ensure more stable weight updates and prevent drastic changes that could lead to overfitting. Early stopping and a ReduceLROnPlateau scheduler were also implemented to optimize training efficiency. After fine-tuning, the model's test accuracy improved to 85.64%, demonstrating the effectiveness of these refinements in enhancing classification performance for real-world mango leaf disease detection.

G. *Workflow Framework*

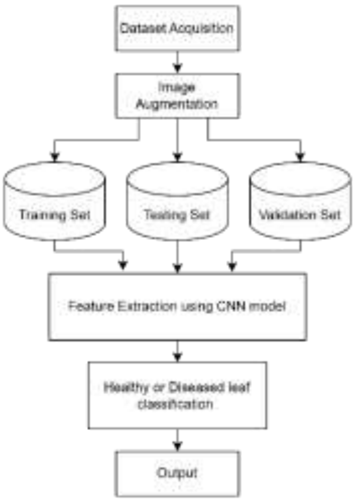

Fig. 2. Workflow Framework

Figure 2 shows the classification process of healthy and diseased leaves using YOLO v8, Faster R-CNN, Custom CNN, EfficientNet, and Swin Transformer. The process starts with getting the dataset that we downloaded from Kaggle to explore and analyze it. Dataset augmentation includes rotating, flipping, brightness adjustment, saturation, translating, and resizing images to 640 X 640 pixels. More diversity, and therefore better model. The dataset was split into training, validation and testing sets. The models are trained on the augmented dataset. That is followed by testing the model's performance on fresh data used for the testing set to evaluate how well they can generalize to the data they were not trained on and whether they can provide accurate predictions. During the feature extraction step, deep learning frameworks extracted the most important features from the images. In this manner they were able to find the defining features of the leaves. Here the trained models are ready to be used for classification based on the extracted features. The task of assessing their performance was based on using precision, recall, and F1-score as the metrics to gauge the accuracy and efficiency of YOLO v8, Faster R-CNN, Custom CNN, EfficientNet, and Swin Transformer.

IV. RESULTS AND DISCUSSION

A. *Performance Metrics*

The network performance of the fruit model is assessed using a range of performance metrics such as speed, and accuracy recognition, to ensure real-time use. Models are evaluated with respect to the following:

1) Precision: The level to which a model's positive prediction comes true. It shows the accuracy rate at which a model predicts a positive outcome,, as defined in equation (1).

$$Precision(P) = \frac{TP}{TP+FP} \quad [29] \quad (1)$$

2) Recall (True Positive Rate or Sensitivity):): It is the percentage of true positive result in the dataset that a model is correctly identifies out of all positive results. It measures the model's accuracy in identifying all relevant instances, regardless of the number of false positives, as defined in equation (2)

$$Recall(R) = \frac{TP}{TP+FN} \quad [29] \quad (2)$$

3) F1-Score: A model's accuracy is gauged by its F1-score, which strikes a compromise between recall and precision. It is very helpful when working with imbalanced datasets, where certain classes may have more samples than others. It is the harmonic mean of precision and recall, as defined in equation (3)

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad [30] \quad (3)$$

B. *Results*

Implementation showcases how the different models excel in specific precision and accuracy aspects, empowering researchers to choose the ideal model for the classification.

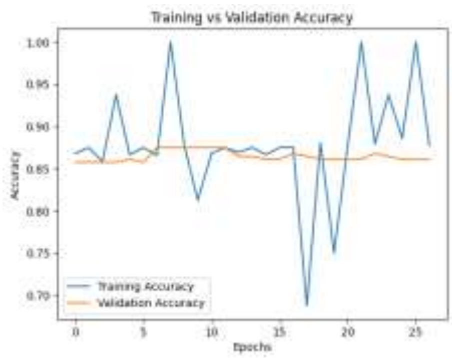

Fig. 3. Training vs Validation Accuracy of Custom CNN

The above Figure 3 indicates the training and validation accuracy across several epochs of a Custom CNN model. The training accuracy is highly variable, with steep peaks and troughs, which reflects the volatility of the learning process. In contrast, the validation accuracy is fairly stable around 0.85.

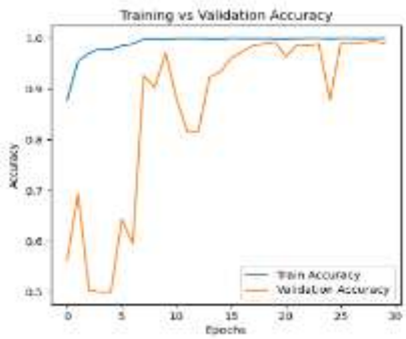

Fig. 4. Training vs Validation Accuracy of EfficientNet

The above Figure 4 demonstrates that the EfficientNet reaches really high training accuracy after 30 epochs, being close to 100%, which indicates proper learning on the training input. Validation accuracy also varies widely in the first half of the epochs but stabilizes around accuracy of 95-100% in the latter half. Training accuracy and validation accuracy converges over time indicating the model generalizes well to implement on unseen data.

TABLE I. COMPARISON TABLE BETWEEN YOLOv8, FASTER R-CNN, CUSTOM CNN, EFFICIENT NET AND SWIN TRANSFORMER

Criteria	YOLO v8	Faster R-CNN	Custom CNN	Efficient Net	Swin Transformer
Precision	0.9637	0.7594	0.7533	0.9899	0.9875
Recall	0.9638	0.8715	0.9456	0.98	0.9875
F1-score	0.9639	0.8115	0.8382	0.9851	0.9875

EfficientNet shows an accuracy of 98.75% when compared to Swin Transformer (97.00%), YOLOv8 (96.56%), Faster R-CNN (87.98%), and Custom CNN (85.64%) in classifying whether the given leaf to the model is a healthy or diseased leaf. As shown in TABLE I the models used are compared to each other based on the parameters such as precision, recall, and F1-score. Based on this one can conclude that the precision of EfficientNet is highest when compared to all other models. Recall and F1-score of the Swin Transformer is the highest when compared to all the other models.

Despite increasing the number of blocks in the Custom CNN model, using SGD optimizer, and increasing the batch size there was no improvement in the accuracy and in some scenarios, it even decreased. The above results demonstrate that EfficientNet has a good scope in accurate and reliable detection that makes it very suitable for classifying healthy and diseased leaves.

CONCLUSION

Agriculture is the main source of economic growth in India. Manual identification of diseased leaves is time-consuming and expensive. This research paper shows how deep learning models can help us to overcome this issue and improve the field of agriculture. The method focuses on the comparison between the deep learning models such as YOLOv8, Faster R-CNN, EfficientNet, Swin Transformer, and Custom CNN for classifying healthy and diseased leaves. According to the results, EfficientNet outperforms all the other models.

Deep learning models such as YOLOv8, Faster R-CNN, Swin Transformer, and EfficientNet, coupled with IoT-based surveillance and drone-assisted crop surveillance, facilitate leaf disease diagnosis in smart agriculture. YOLOv8 facilitates real-time on-ground identification, while Faster R-CNN facilitates high-resolution identification in cloud computing. Swin Transformer facilitates multi-scale features, and EfficientNet boosts accuracy through scaling optimization. These models, coupled with drones and IoT, enable disease detection at an early stage and precise spraying of pesticides, making the process more efficient and reducing losses. However, computation cost, energy consumption, and environmental footprint are some of the issues to be resolved with model optimization techniques such as quantization, pruning, and hardware acceleration for real-time processing.

REFERENCES

[1] R. Sharma, Kartik Suvarna, Shreyas Sudarsan, and G. P. Revathi, “Detecting Diseases in Mango Leaves Using Convolutional Neural Networks,” *2023 15th International Conference on Computer and Automation Engineering (ICCAE)*, vol. 1, Aug., pp. 183–192, 2021.

[2] T. Shaik and Sudhakar Ilango Swamykan, “Identification of Diseases Affecting Mango Leaves Using Deep Learning Models,” *Communications in computer and information science*, Jan., vol. 1907, pp. 132–144,. 2023.

[3] M. A. Jasim and J. M. AL-Tuwaijari, “Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques,” *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020.

[4] Diksha Tandekar and Snehlata Dongre, “Identification of Various Diseases in Plant Leaves Using Image Processing and CNN Approach,” *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023.

[5] P. Sharma, P. Hans, and S. C. Gupta, “Classification Of Plant Leaf Diseases Using Machine Learning And Image Preprocessing Techniques,” *IEEE Xplore*, 2020.

[6] A. Kaur, V. Kukreja, N. C. S, N. Garg, and R. Sharma, “Towards Sustainable Mango Cultivation: Automated Severity Classification of Mango Rust Disease using CNN-SVM,” *2024 Fourth International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, 2024.

[7] Yohannes Agegnehu Bezabh, A. M. Ayalew, Biniyam Mulugeta Abuhayi, Tensay Nigussie Demlie, Eshete Ayenew Awoke, and T.E. Mengistu, “Classification of Mango Disease Using Ensemble Convolutional Neural Network,” *Smart Agricultural Technology*, vol. 8, May, pp. 100476–100476, 2024.

[8] Venkatesh, N. Y, S. T. S, S. S, and S. U. Hegde, “Transfer Learning based Convolutional Neural Network Model for Classification of Mango Leaves Infected by Anthracnose,” *2020 IEEE International Conference for Innovation in Technology (INOCON)*, 2020.

[9] K. S. Chethan, Sumanth Donepudi, H. V. Supreeth, and V. D. Maani, “Mobile Application for Classification of Plant Leaf Diseases Using Image Processing and Neural Networks,” *Algorithms for intelligent systems*, Jan., pp. 287–306, 2021.

[10] P. Kumar, S. Ashtekar, S. S. Jayakrishna, K. P. Bharath, P. T. Vanathi, and M. Rajesh Kumar, “Classification of Mango Leaves Infected by Fungal Disease Anthracnose Using Deep Learning,” *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021.

[11] S. Chen, Y. Liao, F. Lin, and B. Huang, “An improved lightweight YOLOV5 algorithm for detecting strawberry diseases,” *IEEE Access*, vol. 11, Jan., pp. 54080–54092, 2023.

[12] S. S. Puranik, Siddharth R Hanamakkanavar, Anupama P Bidargaddi, Vighnesh V Ballur, Pratham T Joshi, Meena S M, and Uday Kulkarni, “MobileNetV3 for Mango Leaf Disease Detection:An efficient Deep Learning Approach for Precision Agriculture,” *2024 5th International Conference for Emerging Technology (INCET)*, vol. V1, May, pp. 1–7, 2024.

[13] L. C. Ngugi, M. Abdelwahab, and M. Abo-Zahhad, "A new approach to learning and recognizing leaf diseases from individual lesions using convolutional neural networks," *Information Processing in Agriculture*, vol. 10, no. 1, Oct. pp. 11–27, 2021.

[14] N. Manoharan, V. J. Thomas, and D. Anto Sahaya Dhas, "Identification of Mango Leaf Disease Using Deep Learning," *Asian Conference on Innovation in Technology (ASIANCON)*, Aug., pp. 132-144, 2021.

[15] V. Ponnusamy, A. Coumaran, A. S. Shunmugam, K. Rajaram, and S. Senthilvelavan, "Smart Glass: Real-Time Leaf Disease Detection using YOLO Transfer Learning," *International Conference on Communication and Signal Processing*, Jul., pp. 1150–1154, 2020.

[16] Nosin Ibna Mahbub, Feroza Naznin, M. I. Hasan, Syed, M. A. Hossain, and M. Z. Islam, "Detect Bangladeshi Mango Leaf Diseases Using Lightweight Convolutional Neural Network," *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Feb., pp. 1–6, 2023.

[17] M. Mohapatra, Ami Kumar Parida, Pradeep Kumar Mallick, and Neelamadhab Padhy, "Mango Leaf Disease Detection Based on Deep Learning Approach," *2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)*, 2022.

[18] N. Srikanth, Bolla Tirupathi Rao, Gutla Sri Lakshmi Bhargavi, and L. Sai, "Deep Learning Model for Plant Disease Detection and Classification with Pesticide Suggestion," *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)* , 2023.

[19] S. Mehta, V. Kukreja, and S. Vats, "Advancing Agricultural Practices: Federated Learning-based CNN for Mango Leaf Disease Detection," *IEEE Xplore*, 2023.

[20] O. Faruque, I. Jahan, Lamia Rukhsara, and Tapasy Rabeya, "Deep Learning for Detection of Mango Leaf Disease: A Comparative Study Using Convolutional Neural Networks Models," *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, vol. 10, May, pp. 675–680, 2024.

[21] H. Ajra, Mst. K. Nahar, L. Sarkar, and Md. S. Islam, "Disease Detection of Plant Leaf using Image Processing and CNN with Preventive Measures," *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, 2020.

[22] Ankita Suryavanshi, V. Kukreja, P. Srivastava, S. Mehta, and K. Joshi, "Smart Farming: Integrated Federated Learning CNNs for Brinjal Leaf Disease Detection," *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Mar., pp. 1–6, 2024.

[23] N. Jadav, Harsh Chhajed, U. Patel, D. Jani, and Aum Barai, "AI-Enhanced Quad-Wheeled Robot for Targeted Plant Disease Surveillance in Greenhouses," *2023 Global Conference on Information Technologies and Communications (GCITC)*, Dec., pp. 1–7, 2023.

[24] B. Yang et al., "Identifying plant disease and severity from leaves: A deep multitask learning framework using triple-branch Swin Transformer and deep supervision," *Computers and Electronics in Agriculture*, vol. 209, Apr., pp. 107809–107809, 2023.

[25] RVS Praveen, U Hemavathi, R. Sathya, A. A. Siddiq, M. Gokul Sanjay, and S. Gowdish, "AI Powered Plant Identification and Plant Disease Classification System," *2024 4th International Conference on Sustainable Expert Systems (ICSES)*, Oct., pp. 1610–1616, 2024.

[26] Hari Kishan Kondaveeti, G. B. Sai, S. A. Athar, Valli Kumari Vatsavayi, A. Mitra, and Preethi Ananthachari, "Federated Learning for Smart Agriculture: Challenges and Opportunities," *2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, Apr., pp. 1–7, 2024.

[27] X. Yao, F. Yang, and J. Yao, "YOLO-Wheat: A wheat disease detection algorithm improved by YOLOv8s," *IEEE Access*, vol. 12, Jan., pp. 1–1, 2024.

[28] K. Sawant, Shyam Chandrakant, R. D. Shirwaikar, Bhupesh Nilkant Tirodkar, Raj Ravindra Ugavekar, Pramesh Gawas and Ashwek Padolkar, "Fruit Identification using YOLO v8 and Faster R- CNN -A Commparative Study," *International Conference on Distributed Computing and Optimization Techniques (ICDCOT)*, 2024.

[29] K. Goyal, P. Kumar, and K. Verma, "AI-based fruit identification and quality detection system", *Multimedia Tools and Applications*, vol. 82, Nov., pp. 24573–24604, 2022.

[30] S. Jain and P. Jaidka, "Mango Leaf disease Classification using deep learning Hybrid Model," *IEEE Xplore*, 2023.