



A Deep Learning-Based Web Platform for Multi-Crop Disease Detection and Treatment Recommendation

Krupa Rani S¹, D K Soumya²

¹ Assistant Professor, Department of Computer Science ,GM University Davangere,India

² Student, Department of Computer Science ,GM University Davangere,India.

Article Info

Article History:

Published: 12 August 2026

Publication Issue:

*Volume 3, Issue 8
August-2026*

Page Number:

178-189

Corresponding Author:

D K Soumya

Abstract:

Both bacterial and viral pathogens cause plant diseases that are a serious threat to agricultural production and food security. The traditional method of diagnosing diseases is based on physical inspection, which takes a great deal of time and is complicated by lack of resources. Though various deep learning techniques have shown results, these innovations require large computational resources, low level of explainability, and operate almost exclusively offline. CropCare AI addresses the problems represented by traditional as well as deep learning technologies by implementing the use of EfficientNet-B0 for effective classification of plant diseases. The system uses the advanced localization algorithm based on color identification for detecting infected areas without sophisticated description requirements. The online web application is safe and it provides access to multiple users that can check their prediction histories and see where the disease has spread in the most effective way. Testing shows that the system has accuracy rates of 93.44%, which proves its efficiency in diagnosis making.

Keywords: Plant disease detection, EfficientNet-B0, Transfer learning, Deep learning, Multi-crop classification, Lesion annotation, Explainable artificial intelligence (XAI), Cloud computing, Precision agriculture, Convolutional neural networks (CNNs)

1.INTRODUCTION

Agriculture is the foundation of food security but plant diseases caused by fungi, bacteria and viruses still put crop production at risk especially when they are not caught early [3] [6]. Checking for these diseases by hand is not reliable takes a lot of work and is hard to do on a scale especially because there are not enough trained experts in rural areas [1] [3] [7] [8].

Deep learning has made it easier to detect plant Diseases automatically. Convolutional Neural Networks or CNNs can learn features directly from pictures of leaves and are better than old methods that rely on manually created features [1] [2]. Using -trained networks to learn from new data lowers the need for large amounts of data and powerful computers [2] [8]. EfficientNet-B0 [10] uses a way of scaling to balance how deep, wide and detailed the model is. It performs with fewer parameters than other larger CNNs. This makes it good for use in real-time situations where resourcesre limited which is why it is used in this work.

While the prediction accuracy of a model matters, being able to know how it can achieve certain predictions is problematic. Disease label information does not provide farmers information on how to locate the area affected by the disease or its severity. Existing methods for explaining the model workings are aimed mainly at AI researchers rather than actual users and many works deal with the model evaluation rather than working on real life systems that farmers could actually use [4], [6].

To overcome the mentioned drawbacks, the given paper builds the system for identifying plant diseases: an EfficientNet-B0 based classification will be complemented by an image-based method for lesion annotation. The following objectives are to be accomplished: (i) create a classification model based on EfficientNet-B0, (ii) design a rule-based lesion annotation method and thus enable the infection localization, (iii) combine the two components into a system.

2. LITERATURE REVIEW

A. Deep Learning for Plant Disease Detection

Early disease identification used image processing methods. These methods included color-space transforms, thresholding and handcrafted features. They were used with classifiers like SVM. All of these methods were sensitive to lighting and background variation [3]. CNNs replaced these methods. They used end-to-end feature learning. This approach consistently improved accuracy. It captured spatial patterns [1] [2] [5]. However deeper networks needed computation. They often did not generalize beyond their training crops [1]. Public datasets such as PlantVillage [11] were important, for this research direction.

B. Transfer Learning Approaches

Transfer learning using -trained models is now common to cut down on training time [2] [8]. Deep transfer learning methods give results when using multiple crops but have a lot of parameters and need a lot of computing power [2]. Methods that combine models improve reliability but make the process more complicated, during testing, which stops them from being used in real time [8].

C. Lightweight CNN Models

Lightweight architectures using depthwise separable convolutions and multi-scale feature aggregation achieve accuracy competitive with much larger networks while remaining efficient [4]; similar results hold for models fusing compact deep features with texture descriptors [5] and multi-scale residual designs [6]. EfficientNet-B0 [10] extends this direction through compound scaling, offering a favorable accuracy-efficiency trade-off relative to conventional and earlier lightweight CNNs.

D. Explainability and Disease Localization

Most disease classifiers remain black boxes, offering a predicted label without visual justification. Gradient- and activation-based visualization techniques improve transparency by highlighting influential regions [4], [6], but these remain internal to the model and are rarely translated into an independent, farmer-interpretable annotation of infected tissue. This is an underexplored direction, one that the present work addresses through rule-based image analysis.

E. Cloud-based Agricultural Platforms

Few studies extend beyond model evaluation into deployable systems. Field-oriented datasets and mobile applications have improved real-world applicability in select cases [9], but most works lack a user-facing interface, backend service layer, or persistent storage [8], [9], limiting practical, multi-user adoption.

Comparative Analysis

TABLE I. Comparative Analysis of Existing Studies

Ref.	Model	Acc.	Strength	Limitation
[1]	Patch CNN	94–97%	Cross-crop	Pigmentation errors
[2]	TL CNN	96–98%	Multi-crop	High cost

[3]	CNN+SVM	96–97%	Severity	Image sensitive
[4]	MS-CNN	97–99%	Real-time	Limited crops
[5]	LW CNN	97–99%	Texture fusion	Limited validation
[6]	Res-CNN	94–98%	Generalization	Similar diseases
[8]	Ensemble CNN	97–98%	Robust	Heavy model
[9]	Seg.+CNN	34%	Field support	Accuracy drop

As Table I shows, deep transfer learning and ensemble methods get the results but need a lot of computing power [2] [8]. Architectures that are light and use scales on the other hand give a better balance, between how well they work and how much resources they use [4] [5]. The ability to explain what the models do is still part of the model itself of being a separate visual note and systems that can be fully used in the cloud are still not common [4] [6] [8] [9]. These missing parts push the research direction that comes next.

3. RESEARCH GAP

The literature identifies a number of problems that can be linked to each other. In particular, training models with laboratory datasets results in poor performance in real-world circumstances. Moreover, deep transfer learning methods and ensemble techniques require high computational power, which affects the scalability and cloud usage. Most of the systems are designed to work offline; do not support multiple users; offer limited explanation for farmers; and do not provide secure features for the authentication or data management of the solution.

The proposed solution gives answers to those problems through the use of EfficientNet-B0, which allows performing successfully with low computational costs. Moreover, an algorithm for identification of diseases through colors is developed which certainly increases clarity of results for the farmers. The system also offers secure authentication, history of predictions, and cloud access.

4. PROPOSED METHODOLOGY

The proposed platform integrates dataset preparation, an EfficientNet-B0 classification model, a rule-based lesion annotation module, and a full-stack web platform with secure access and persistent storage.

A. Overall Workflow

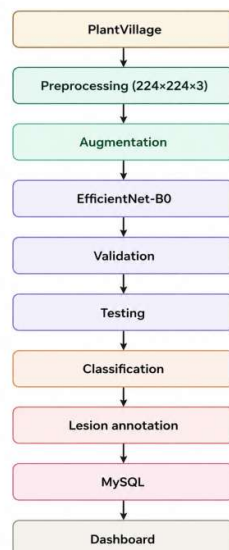


Figure1.Experimental Workflow of the Proposed Multi-Crop Disease Detection System

Figure 1 shows the process starting from getting an image all the way, to predicting a disease marking lesions and saving the prediction. Leaf images are. Sorted using EfficientNet-B0. This gives a Disease Label and a Confidence Score. These results are sent to the lesion annotation part to find the infection and create an Image. The results are sent through the Flask REST API to the React dashboard. Saved in MySQL.

B. Dataset Preparation

The system only uses the PlantVillage dataset. We make all the images the same size, which is 224 by 224 pixels. We also make sure the images are normalized using the settings as ImageNet so we can use transfer learning. To make the training data better we do things like rotate the images flip them and change how bright they are. We split the PlantVillage dataset into three parts: 80 percent is for training 10 percent is for validation and 10 percent is, for testing the PlantVillage dataset.

C. EfficientNet-B0 Classification Model

The classifier is based on EfficientNet-B0 [10] and utilizes compound scaling for the purpose of optimizing its depth, width and resolution so that a significantly less quantity of parameters is needed while still achieving a high accuracy in comparison with CNN's. The model itself has been initialized with the weights from ImageNet and then proceeded to the process of transfer learning. The last layer of the model is replaced with a dense one based on the number of classes. The implementation is made in PyTorch using AdamW [13] optimizer and Cross-Entropy loss as a loss function with 0.001 of learning rate, 32 of batch size and 20 epochs..

D. Color-Based Lesion Annotation

A custom rule-based lesion annotation algorithm has been created using Pillow (PIL) and NumPy. This algorithm relies on color tracking to identify the infection. This is done through calculating the amount of green for each pixel. Pixels with very little green are flagged as candidates. The algorithm is able to detect both very dark necrotic regions as well as very bright chlorotic/fungal segments and combines this information into one binary mask. Subsequently, the flagged pixels are analyzed based on their density and only those areas above a certain threshold are used in an output bounding box which generates the Annotated Image as well as an associated Lesion Percentage figure. Both of these outputs will come with the expected Disease Label attached.

E. Web Platform

The frontend part of the application is made using React.js with Vite and utilizes TypeScript for uploading images, displaying results, and reviewing history. The backend is built as a Flask REST API and runs the EfficientNet-B0 inference stream and the module for annotating lesions, which generates combined results. Authentication is done via Flask-JWT while Flask-SQLAlchemy is responsible for communicating with MySQL to save user accounts, images data, disease labels, confidence scores, percentage of lesions and references to annotated images. The system was designed to run in the cloud allowing users to access it from multiple devices without the need to install anything locally.

5. SYSTEM ARCHITECTURE

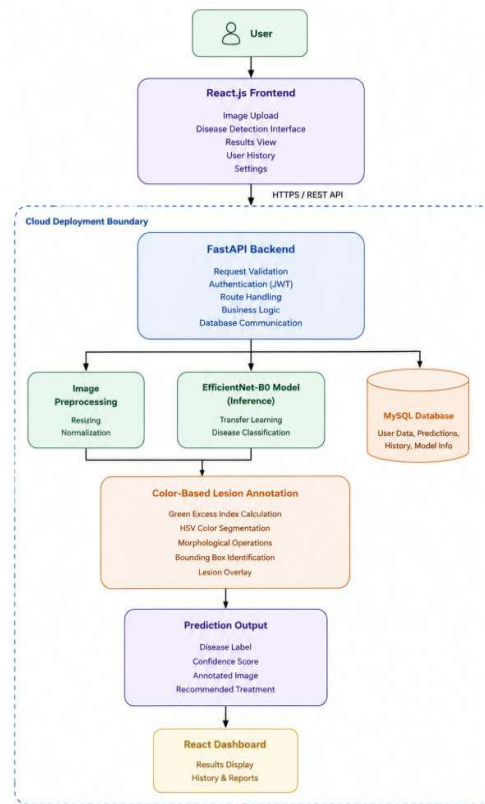


Figure 2. System Architecture of the Proposed Platform

In diagram 2 we can see how the general design looks: it consists of a client layer, a server application layer, a module for inference and lesion annotation, and a database layer that can be accessed via cloud technology.

A. Frontend

The combination of React.js with Vite and TypeScript is capable of uploads of images, secure authentication of users, and monitoring Disease Labels, Confidence Scores, and Annotated Images with the help of authenticated HTTP requests to the backend.

B. Backend

The authentication of requests on the side of the back end is achieved by Flask-JWT authentication, with routing of images done by the pipeline of inference and creation of predictions being made with the use of Flask-SQLAlchemy.

C. Inference Module

The EfficientNet-B0 model based on PyTorch [12] performs forwarding of the processed images for the creation of disease labels and percentages with the help of a service that can be independently updated.

D. Lesion Annotation Module

This module includes performing evaluations of images on the pixel level, the grid-based behavior, and creating of bounding-boxes (see Section IV-D) so that Annotated Image and Lesion Percentage could be produced

E. Database:

User accounts, image metadata, Disease Labels, Confidence Scores, Lesion Percentages, and Annotated Image references are kept by MySQL, which is connected through Flask-SQLAlchemy, while allowing many users to use it at the same time.

F. Cloud Deployment:

The whole platform is launched in a cloud, which enables users to access it remotely and maintain it in one place, as well as to carry out operations such as model updates and migration of databases.

G. Overall Data Flow:

An authenticated user uploads an image from the React front-end to the Flask REST API, which sends the uploaded image to EffectiveNet-B0 for classification and annotation. Then, the obtained Disease Label, Confidence Score, Lesion Percentage, and Annotated Image are saved in MySQL and sent back to the client's computer. The system has a modular design, allowing maintenance and scalability, and facilitating the inclusion of new crops or diagnostic modules without system modifications.

6.EXPERIMENTAL SETUP

The description of the dataset, software and hardware environment, training configuration, evaluation metrics, and overall workflow of the experiment as a part of the execution and assessment of the proposed system is given below.

A.Dataset

In the proposed system, the PlantVillage dataset is used, which is a publicly available dataset containing labeled leaf images of different crop species and diseases. All images in PlantVillage dataset were resized to 224×224 pixels in order to meet the requirements of EfficientNet-B0. Data augmentation techniques such as random rotation, flipping, and brightness adjustment were applied on the training set to enhance generalization. The dataset was divided into training, validation, and testing data sets using 80/10/10 ratio.

D. Training Configuration

The EfficientNet-B0 method was developed using transfer learning, with the backbone taken from a model trained on ImageNet and a new classification layer connected at the end to carry out this work. The techniques were executed using an optimizer called AdamW and a cross-entropy loss function with a learning rate of 0.001, a batch size of 32, and 20 epochs. The learning rate was adjusted by a learning rate schedule called ReduceLROnPlateau whenever performance stagnated because validation results did not change.

E. Performance Metrics

Estimating how well the model behaves was accomplished by using accuracy, precision, recall, and F1-measures determined based on the conventional definitions for multi-class classification. A confusion matrix has been used to display how correctly or incorrectly the model has classified a particular case in all classes.

F. Experimental Workflow

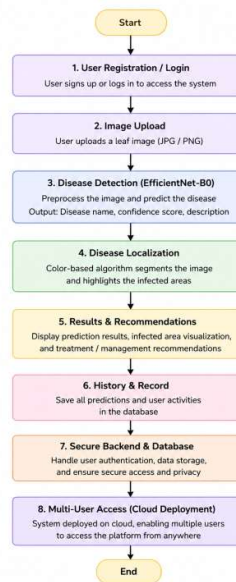


Figure 3. Experimental Workflow of the Proposed System

As shown in Figure 3, the procedures utilized in the experimental process include the preliminary step of using the PlantVillage dataset for preprocessing the information before using the data to train the model. Then the EfficientNet-B0 model in question is trained and validated, in order to be able to track the performance in terms of generalization and test the system on another set of data for the objective of obtaining accurate performance indication. Then for all the images used for testing the annotation tool processes the given image and computes the expected percentage, after which the information is recorded in the database and shown to the user via the React website.

VII. RESULTS AND DISCUSSION

The section explains the experimental findings of the proposed CropCare AI technology and talks about the results of EfficientNet-B0 classification model, rule-based lesion annotation system, and the entire cloud-implemented platform.

A. Classification Performance

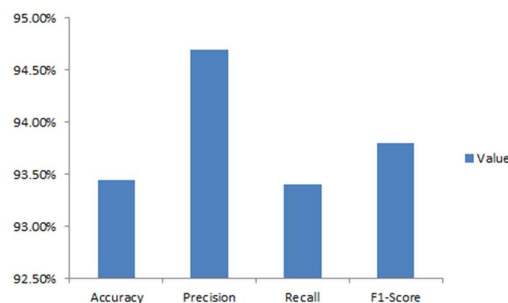


Figure 4. Overall Performance Metrics

The new EfficientNet-B0 model attained an overall classification accuracy of 93.44% on the PlantVillage dataset's test subset, which includes weighted precision of 94.70%, weighted recall of 93.40%, and weighted F1-score of 93.80%, as presented in Table 1 and illustrated in Figure 4. The comparable value of the precision and the recall shows

that the model does a fair prediction of the disease occurrence, thereby ruling out the possibility of any over-prediction or under-prediction of diseases. The results indicate transfer learning with EfficientNet-B0 is effective for multi-crop disease classification since it does not compromise on the efficiency of the compound-scaled architecture.

B. Class-wise Performance Analysis

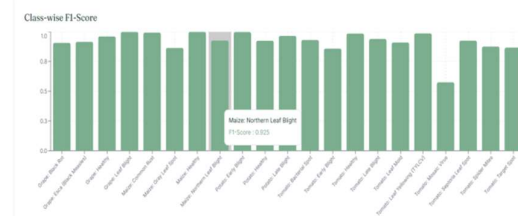


Figure 5. Class-wise F1-Score

In this figure, the F1-score distribution of different classes is shown with respect to the various diseases. The F1-score of many classes is very high which indicates that the disease has a good discriminative capability when there are pronounced lesions or changes in color. A small number of classes have a lower F1-score value which implies that the disease is visually confused with other diseases of the same crop, like early-stage fungal infection which could be mistaken for bacterial spots. The above results also indicate that the errors are only concentrated on certain pairs of diseases and this finding is consistent with the literature.

C. Precision vs Recall Analysis

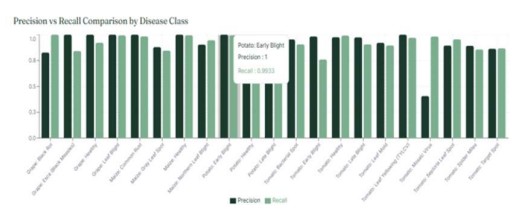


Figure 6. Precision vs Recall Comparison by Disease Class

In Figure 6, the relationship between precision and recall by individual classes is shown. The precision and recall figures remain fairly close for many of the classes indicating a decent amount of accuracy in identifying positive disease cases and a low occurrence of false positive cases. The closeness of precision and recall suggests that the model is able to generalize well. A high precision but low recall indicates that the model may either be overfitted on the dominant classes or be too careful for minor classes. Only a few classes show noticeable divergence that occurs because the recall is higher than precision, something that is likely experienced in diseases with highly similar symptoms to other diseases.

D. Confusion Matrix Analysis

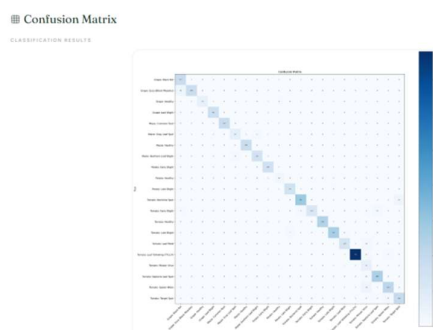


Figure 7. Confusion Matrix

Figure 7 shows the presence of a strong diagonal in the confusion matrix, implying that the model makes accurate predictions due to its high ability to differentiate among the various crop-disease pairs. The few predictions that were misclassified occurred between diseases that affected similar crops and showed similar symptoms as discussed earlier in the classification section. Because these errors are minimal and arise from identifiable visual vagueness rather than random misclassification, no major deficiencies in the product have been detected.

E.Prediction Dashboard

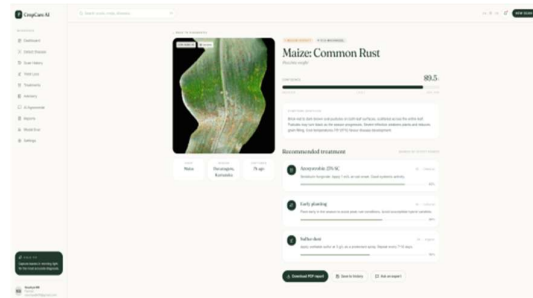


Figure 8. CropCare AI Prediction Dashboard

Figure 8 illustrates the prediction dashboard developed using React.js, which provides complete diagnostic information in one platform, such as disease prediction, confidence score, associated symptoms, guidance for treatment, and annotated image of lesion. By obtaining the outcomes of the diagnostic, right now, users can view the final classification result along with specific recommendation, which is particularly useful for non-expert population (e.g., farmers who need specific advice, not just the result).

F.Lesion Annotation Results



Figure 9:Annotaed Image

The lesion annotation algorithm based on rules has been created using Pillow and NumPy so that it can accurately find and display infected areas irrespective of the type of classification model. The combined use of green dominance analysis, the detection of dark/bright regions, and the act of creating grid-based bounding boxes makes it possible to obtain images with infected areas marked in red boxes and giving the percentage of lesions, as shown in Figure 9. Because of this, a user gets a complete explanation of the image he/she is interested in and can establish the correlation

between the infection extent and the class label predicted by the model. Besides, this approach is beneficial for the user since it does not require an understanding of the classification network logic behind predictions and yields comprehensible results.

G. Cloud Deployment

The entire platform, consisting of Flask REST API backend, React.js frontend, and MySQL database, has been launched in a cloud-based environment that allows remote multi-user access and does not require local resources. Personal accounts and prediction history are protected with JWT authentication, while Flask-SQLAlchemy-MYSQL integration allows safe and well-structured management of user data and predictions. The given solution supports multiple users at the same time while ensuring equal fast response times and minimizes the burden of model maintenance and updating for end-users. Overall, this deployment model deals with main obstacles of experimental and deployed systems for detecting diseases.

H. Comparison with Previous Studies

The suggested system utilizes EfficientNet-B0 to provide excellent balance between accuracy and efficiency along with complete end-to-end solution with secure

authentication and cloud storage. In contrast to previous studies that focus mainly on offline classification and model evaluation tasks, this system has been designed to allow real-time data prediction and implementation. Moreover, the proposed rule-based annotation module offers simple, image-level explanations which do not depend on the internal functioning of the model.

I. Limitations

Although these contributions are acknowledged, the current system has its share of drawbacks. The classification model has been trained and tested on PlantVillage dataset alone, which included the pictures of plants taken in a laboratory setting with uniform background and fixed lighting. Thus, model performance may be substantially different when applied to leaf images taken under real-life circumstances in the open air where clutter, different lighting conditions, and overlapping leaves are observed.

J. Future Work

The future development directions are listed as follows: the development of an app to enable farmers perform in-field diagnosis; validation and enhancement of the method with real photos taken at farmlands to strengthen the approach; introduction of new crops and the disease categories; improvement of the algorithm for lesions annotation for better segmentation; or integration of the system with the Internet of Things (IoT) sensor array to enable continuous crop health monitoring using images as well as IoT data.

8. CONCLUSION

The paper describes the CropCare AI software, an online system for identifying diseases in multiple crops and marking lesions. The implications of CropCare AI are twofold: it presents a model for classification and diagnosis based on transfer learning, as well as a lesion identification algorithm for marking infected areas on plant leaves, independent from the classification procedure. The proposed system thus makes it possible to overcome the issues related to the lack of interpretability typical of most existing models. Additionally, it demonstrates how the integration of the

solution into a complete and secure platform based on React.js, Flask REST API, and MySQL facilitates the availability of such models as fully functioning software tools.

The experimental results validate this contribution indeed, as the proposed model achieves an accuracy level of 93.44% in terms of overall accuracy, 94.70% in terms of weighted precision, 93.40% in terms of weighted recall, and 93.80% in terms of weighted F1 score. The analysis at the class level shows an efficient operation for the majority of diseases, with a relatively small number of errors in the classification being due to close similarities among pairs of diseases, rather than randomly distributed, while the presence of the clear diagonal at the confusion matrix proving this strong classification performance. The lesion identification module based on rules increased its diagnostic power, since it provides an illustrative and easy-to-understand explanation for users at the image level.

The practical importance of this work pertains to its conversion from a standalone classification model to a full-fledged accessible diagnostic device usable for agricultural purposes, closing the gap between deep learning experiments and practical solutions intended for farmers. However, this significance has its drawbacks due to the scope of evaluation: since the only source of information for the system was the PlantVillage dataset, real-world operational effectiveness still needs to be established. Further work will include extending the detection process to natural images, other types of crops, using mobile devices, improving lesion algorithms, and introducing IoT-based monitoring techniques.

References

- [1] I. Bouacida, B. Farou, L. Djakhadjakha, H. Seridi, and M. Kurulay, "Innovative deep learning approach for cross-crop plant disease detection: A generalized method for identifying unhealthy leaves," *Information Processing in Agriculture*, vol. 12, no. 1, pp. 54–67, 2025, doi: 10.1016/j.inpa.2024.03.002.
- [2] A. S. Paymode and V. B. Malode, "Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, 2022, doi: 10.1016/j.aiia.2021.12.002.
- [3] A. Shrotriya, A. Kumar Sharma, S. Prabhu, and A. Kumar Bairwa, "An approach toward classifying plant-leaf diseases and comparisons with the conventional classification," *IEEE Access*, vol. 12, pp. 117379–117398, 2024, doi: 10.1109/ACCESS.2024.3411013.
- [4] M. S. A. M. Al-Gaashani, R. Alkanhel, M. A. S. Ali, M. S. A. Muthanna, A. Aziz, and A. Muthanna, "MSCPNet: A multi-scale convolutional pooling network for maize disease classification," *IEEE Access*, vol. 13, pp. 11423–11446, 2025, doi: 10.1109/ACCESS.2024.3524729.
- [5] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-class classification of plant leaf diseases using feature fusion of deep convolutional neural network and local binary pattern," *IEEE Access*, vol. 11, pp. 62307–62317, 2023, doi: 10.1109/ACCESS.2023.3286730.
- [6] L. Liu, S. Qiao, J. Chang, W. Ding, C. Xu, J. Gu, T. Sun, and H. Qiao, "A multi-scale feature fusion neural network for multi-class disease classification on the maize leaf images," *Heliyon*, vol. 10, no. 7, art. no. e28264, 2024, doi: 10.1016/j.heliyon.2024.e28264.
- [7] M. Chilakalapudi and S. Jayachandran, "Multi-classification of disease induced in plant leaf using chronological Flamingo search optimization with transfer learning," *PeerJ Computer Science*, vol. 10, art. no. e1972, 2024, doi: 10.7717/peerj-cs.1972.

- [8] W. Shafik, A. Tufail, C. De Silva Liyanage, and R. A. Awg Haji Mohd Apong, "Using transfer learning-based plant disease classification and detection for sustainable agriculture," *BMC Plant Biology*, vol. 24, art. no. 136, 2024.
- [9] E. Moupojou, F. Retraint, H. Tapamo, M. Nkenlifack, C. Kacfah, and A. Tagne, "Segment anything model and fully convolutional data description for plant multi-disease detection on field images," *IEEE Access*, vol. 12, pp. 102592–102605, 2024, doi:10.1109/ACCESS.2024.3433495.
- [10] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, Long Beach, CA, USA, 2019, pp. 6105–6114.
- [11] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," *arXiv preprint*, arXiv:1511.08060, 2015.
- [12] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, Vancouver, BC, Canada, 2019, pp. 8024–8035.
- [13] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. 7th Int. Conf. Learning Representations (ICLR)*, New Orleans, LA, USA, 2019.