DeepLeaf: Multilingual CNN Model for Mango Leaf Disease Detection and Remedies

Varad Sonavadekar[1], Raghunandan Rane[2], Rushikesh Kadam[3], Adwait Kale[4]

Department of Computer Science Engineering Mumbai University Ratnagiri, Maharashtra 415639, India

***Abstract*—Mango cultivation is integral to tropical agriculture; however, it is highly vulnerable to a range of leaf diseases that can significantly reduce both yield and quality. This paper presents DeepLeaf, a Convolutional Neural Network (CNN)-based system specifically developed to detect and classify various conditions of mango leaves, including Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Powdery Mildew, Sooty Mould, and healthy leaves. The model is trained on a meticulously curated image dataset of mango leaves and exhibits high accuracy in disease classification. To enhance accessibility for farmers, DeepLeaf is implemented through a visually appealing Streamlit web interface, featuring glassmorphism styling, real-time predictions, and remedy recommendations in English, Hindi, and Marathi. The system also provides language toggling and downloadable treatment guides. Experimental results validate the system's efficacy in practical applications, offering a promising tool for early intervention and sustainable mango farming practices.**

***Index Terms*—Convolutional Neural Network (CNN), Plant Disease Detection, Data Preprocessing, Deep Learning, Field Survey.**

I. INTRODUCTION

Mango (Mangifera indica) is a high-value fruit crop cultivated extensively in tropical and subtropical regions, contributing significantly to the global agricultural economy and nutritional security. In India, the Konkan region is renowned for its rich mango plantations, particularly the Alphonso variety, which holds both cultural and commercial importance. However, mango cultivation in this region is frequently compromised by a variety of foliar diseases such as Anthracnose, Bacterial Canker, Powdery Mildew, Gall Midge, and others, which adversely affect both yield and fruit quality. Early and accurate detection of these diseases is essential for implementing timely intervention strategies, supporting sustainable farming practices, and minimizing economic losses for local growers.

Recent advancements in computer vision and machine learning have enabled the development of automated systems for plant disease diagnosis. In particular, Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in image classification tasks due to their ability to automatically extract and learn hierarchical features from visual data. These capabilities make CNNs highly suitable for identifying disease patterns in leaf images, offering a scalable and non-invasive alternative to traditional diagnostic methods. The implementation of CNNs is further enhanced by deep learning frameworks such as TensorFlow and Keras, which provide a robust and flexible platform for designing, training, and deploying complex neural network architectures. These frameworks support efficient model development and performance tuning over large datasets. Image preprocessing and augmentation are performed using OpenCV, improving data quality through operations such as resizing, normalization, and noise reduction. NumPy and Pandas are utilized for efficient numerical computation and structured data handling, while Matplotlib is employed for model evaluation and performance visualization.

This study presents the design and implementation of a CNN- based mango leaf disease detection system, specifically tailored for disease patterns observed in the Konkan region of Maharashtra, India. The trained model is integrated into an interactive and lightweight web application using Streamlit, equipped with glassmorphism UI design and multilingual remedy support in English, Hindi, and Marathi. By combining deep learning with user-centric design and regional adaptability, this research contributes to the growing field of precision agriculture, offering a practical, scalable, and accessible solution for early disease detection and crop health monitoring in mango orchards.

II. RELATED WORK

Plant disease detection systems have seen a considerable increase in accuracy and efficiency in recent years as a result of the application of computer vision and deep learning methodologies. In earlier research, traditional machine learning methods like Decision Trees, Support Vector Machines (SVM), and k-Nearest Neighbors (kNN) have been employed for classification tasks [1][2]. However, because these methods rely on handmade features, they are more susceptible to changes in leaf color, texture, and background noise.

With the introduction of Convolutional Neural Networks (CNNs), which made it possible to automatically extract features from raw picture data, the sector was transformed.

Using the PlantVillage dataset, Mohanty et al. [3] demonstrated the capability of deep CNNs and attained a classification accuracy of more than 99% across 38 illness categories. Sladojevic et al. [4] presented a deep learning- based method using CNNs to accurately classify 13 distinct plant diseases, offering a mobile-compatible solution for field deployment.

Ferentinos' research [5], which employed deep CNN architectures like VGG and ResNet and attained classification accuracy rates as high as 99. 5%, provided additional evidence for the efficacy of deep learning models in identifying plant diseases.

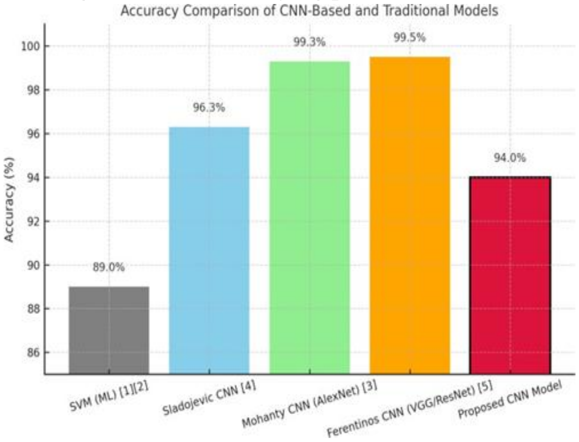

Fig. 1 shows a comparative view of classification accuracies reported in prior works alongside our proposed CNN-based model. While our model achieves slightly lower accuracy (~94%), it emphasizes practical deployment through a multilingual interface and remedy support.

In spite of these improvements, the usefulness of many current solutions is restricted in multilingual areas because they are only available on English-only platforms or because they have user-unfriendly interfaces. Additionally, the majority of literature focuses on disease categorization without taking into account post-diagnosis care, such as therapy or pesticide recommendations. Only a handful of research, like A
mara et al. [6], attempted simple app deployment, but even those studies lacked localized language choices and tailored remedy integration.

In contrast, our suggested system closes these gaps by providing a CNN-based model with a multilingual (English, Hindi, Marathi) Streamlit web interface that offers both disease forecasting and curated treatment recommendations. This integrated strategy improves both accuracy and accessibility, particularly for agricultural communities that do not speak English.

III. METHODOLOGY

A. Field Survey

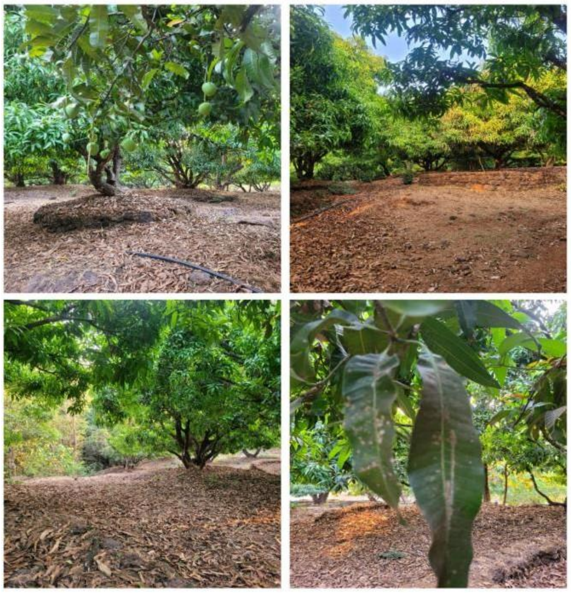

Fig. 2. Composite field survey images taken at a mango orchard: (a) orchard canopy layout, (b) symptomatic diseased leaf, (c) dry leaf litter on orchard floor, and (d) healthy fruit-bearing mango tree.

A comprehensive field study was carried out at a mango plantation in Maharashtra's agricultural area as part of the preliminary research stage. The main goal was to watch for and record the physical indicators of different mango leaf illnesses in real-world field settings. The knowledge gained both gave an initial dataset for analysis and supported the validity of utilizing image-based plant disease detection systems in real-world agricultural applications.

Several diseased mango leaf samples were taken from various areas of the orchard and categorized based on major visual indicators, such as powdery mildew formations, necrotic lesions, chlorotic spots, fungal spots, and leaf abnormalities. Diseases such as anthracnose, powdery mildew, bacterial black spot, and fungal leaf spot, which are common in the area, were initially linked to these symptom patterns. The composite field survey image (Fig. 2b) includes representative samples of these symptomatic leaves.

Discussions with local farmers brought up key issues related to the early diagnosis and management of the illness. Traditionally, farmers used traditional remedies and manual visual inspections, which frequently led to late treatment and lower crop yields. This situation underscores the need for an automated, AI-powered diagnostic tool that can quickly provide predictions along with multilingual treatment recommendations to aid small and medium-sized farmers in making informed and timely decisions.

Numerous environmental variables that may affect disease transmission were also discovered during the field survey. Dense vegetative growth throughout the orchard and the build- up of dry leaf litter on the ground were found to foster the spread of pathogens (Figs. 2a and 2c). Conversely, healthy fruit-bearing trees, which were also seen during the survey, highlighted the possibility of early disease detection in protecting crop health and productivity (Fig. 2d).

The study's methodology, including data collection protocols, picture annotation methods, and the development of a multilingual interface for treatment distribution, was heavily influenced by these onsite results. At the same time, a thorough literature review was done to assess current methods for identifying plant diseases. Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have demonstrated significant improvements in classification accuracy [3], while conventional methods have historically used rule-based image processing. For example, earlier research [3] has successfully used CNN architectures to identify plant diseases; however, the majority of studies have been limited to tomato and potato plants, with little attention given to mangoes and the absence of real-time, user-friendly diagnostic platforms.

This research, driven by these findings, presents an end-to-end CNN-based diagnostic system that can identify eight different diseases in mango and other crops and includes localized treatment recommendations in English, Hindi, and Marathi. The strategy seeks to close the gap between high-performance disease detection and real-world field application in order to promote sustainable agricultural practices.

B. Data Collection and Preparation

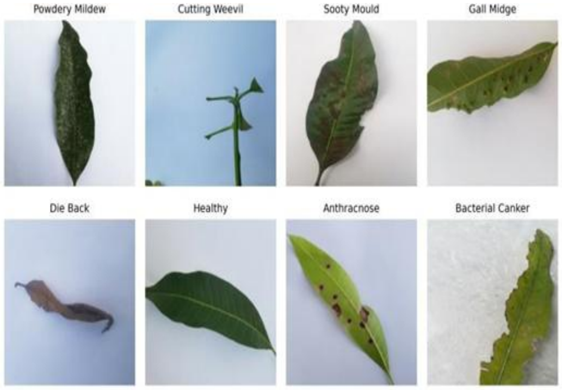

Figure 3. Sample images representing each class in the dataset used for training and evaluation. The dataset consists of multiple categories, each capturing distinct characteristics essential for accurate classification by the CNN model.

The data used in this study comprises high-resolution RGB images of plant leaves that belong to eight distinct classes— seven for diseases and one for healthy leaves. This data was extracted from a publicly available repository on Kaggle, known for its extensive and well-annotated collections in plant pathology. Each image in the dataset has been annotated with its corresponding disease class by experts, ensuring high- quality ground truth for supervised learning.

In order to promote computational efficiency and compatibility with the ensuing Convolutional Neural Network (CNN) architecture, all images were resized to a uniform spatial size of 256×256×3 pixels. This input dimension standardization preserves critical texture and spatial features necessary for accurate classification.

Preprocessing treatments were systematically applied to enhance data quality and increase the diversity of the training data:

- Normalization: Pixel intensity values were scaled from the original range [0, 255] to [0, 1] by dividing each pixel by 255.0, stabilizing training by preventing gradient instability and accelerating convergence.
- Data Augmentation: To mitigate overfitting and increase the effective size of the training set, online data augmentation was performed using several transformations such as random rotations (±20°), horizontal and vertical flipping, random zoom (up to 20%), width and height adjustments (up to 10%), and brightness/contrast jittering. These augmentations simulate natural leaf morphology variations caused by environmental conditions and imaging settings, thereby improving model generalization.
- Dataset Partitioning: The dataset was partitioned into training (80%), validation (10%), and test (10%) subsets using stratified sampling to maintain identical class distribution across all sets, preventing data leakage and class imbalance during evaluation.
- Label Encoding: The categorical class labels were one-hot encoded to be compatible with the softmax classifier in the CNN output layer.

Figure 3 illustrates a representative sample image from each of the eight classes in the dataset, highlighting the visual diversity and distinct characteristics that the model is trained to recognize.

The preprocessing pipeline was implemented using TensorFlow's ImageDataGenerator and Keras preprocessing utilities. This step ensured data consistency, diversity, and purity, providing a robust foundation for effective feature extraction and learning in the subsequent convolutional neural network layers.

C. CNN Architecture

To address the multi-class classification task of plant leaf diseases, a specialized Convolutional Neural Network (CNN) architecture was designed, optimized for learning hierarchical spatial features from input images. The architecture was implemented using the TensorFlow-Keras deep learning framework and follows a deep sequential model structure. The primary objective of this model is to accurately identify subtle disease patterns such as lesion morphology, texture, and color variation from RGB leaf images.

The input layer accepts standardized RGB images of size 256×256×3 (width × height × channels), ensuring consistent spatial dimensions for effective feature learning and computational efficiency.The feature extraction module consists of six convolutional blocks, each composed of a 2D convolutional layer followed by a max-pooling operation. All convolutional layers employ the ReLU activation function and use a 3×3 kernel with 'same' padding to preserve spatial resolution. Pooling layers use a 2×2 window to progressively reduce the spatial dimensions while preserving the most significant features.

The output tensor from the final convolutional layer (2×2×64) is flattened into a 1D vector of 256 units, which is then fed into fully connected dense layers. The first dense layer consists of 64 neurons with ReLU activation to capture high- level abstractions of the extracted features. A dropout layer with a rate of 0.5 (if applied) is used to mitigate overfitting by randomly deactivating neurons during training. The final output layer is a dense layer with 8 neurons and softmax activation, producing a probability distribution over the eight target classes.

Table I provides a comprehensive summary of the CNN architecture, including layer types, output shapes, and trainable parameters.

TABLE I

Summary of the Convolutional Neural Network (CNN) Architecture

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 256, 256, 3)	0
conv2d (Conv2D)	(None, 254, 254, 32)	896
conv2d (Conv2D)	(None, 127, 127, 32)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_2 (Conv2D)	(None, 60, 60, 64)	36,928

max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 64)	36,928
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 64)	0
conv2d_4 (Conv2D)	(None, 12, 12, 64)	36,928
max_pooling2d_4 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_5 (Conv2D)	(None, 4, 4, 64)	36,928
max_pooling2d_5 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 64)	16,448
dense_1 (Dense)	(None, 8)	520

The total number of trainable parameters in the model is approximately 147,144, striking a balance between computational efficiency and representational capacity. This architecture was selected based on empirical experimentation and large-scale hyperparameter tuning, offering an optimal trade-off between performance and depth. Its hierarchical depth enables effective learning of both low-level (edges, textures) and high-level (disease-specific patterns) features, making it well-suited for fine-grained classification across a variety of plant diseases.

D. Trainning Approach

The CNN was trained in a supervised learning setting on a labeled dataset consisting of RGB images representing eight categories of diseased and healthy plant leaves. The objective was to minimize classification error while ensuring generalization across diverse leaf types and imaging conditions.

- Objective Function

Categorical cross-entropy loss was used, which is appropriate for multi-class classification problems with mutually exclusive labels. The loss is computed as:

where y_i is the true class label, and $\hat{y}_i$ is the predicted class probability. This formulation encourages confident correct predictions while penalizing incorrect ones.

- Optimizer

The model used the Adam optimization algorithm,, which integrates the strengths of AdaGrad and RMSProp. The optimizer parameters were:

- Learning Rate: 0.001
- $\beta_1 = 0.9$, $\beta_2 = 0.999$
- Epsilon = 1e-07

These hyperparameters enable adaptive learning rates for each weight, improving stability and convergence

- Batch Size and Epochs

Training was conducted using a batch size of 32, providing a trade-off between performance and computational cost. The number of epochs ranged from 25 to 50, controlled via early stopping with a patience of 5 epochs based on validation loss to prevent overfitting.

- Data Augmentation

To promote generalization and increase dataset variability, real-time augmentation was performed using Keras' ImageDataGenerator. The augmentation techniques applied included:

- Horizontal and vertical flips
- Random rotations (±20°)
- Zooming (range: 0.8–1.2)
- Width and height shifts (±10%)
- Brightness adjustment and shear transformation

This approach simulates natural image variability and reduces the likelihood of overfitting.

- Validation Strategy

A stratified 80:20 train-validation split was used to maintain balanced class representation. The validation set remained strictly held out during training for unbiased performance monitoring.

ModelCheckpoint was applied to save the model with the best validation accuracy. Optionally, TensorBoard was used to visualize learning curves.

- Regularization

Dropout (rate = 0.5) was optionally introduced after the dense layer to discourage neuron co-adaptation. Additionally, L2 kernel regularization was explored for convolutional layers in scenarios indicating potential overfitting.

$$L = -\sum_{i=1}^{n} y_i . \log(\hat{y}_i)$$

The training and validation performance over epochs, including accuracy and loss curves, is illustrated in Figure 4. The classification performance across all eight classes is further visualized in the confusion matrix heatmap in Figure 5, confirming effective learning and accurate class differentiation.

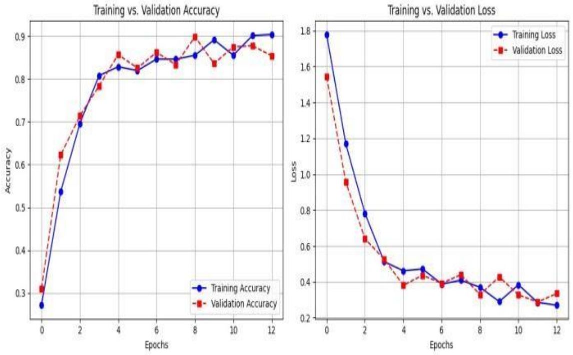

Figure 4. Training and validation accuracy and loss curves over 50 epochs.

To further assess the classification performance, a confusion matrix heatmap was generated using predictions on the test set. This visualization is shown in Figure 5 and highlights how accurately the model distinguishes between various disease classes. The diagonal dominance in the heatmap indicates high true positive rates across all classes, confirming the robustness of the learned features.

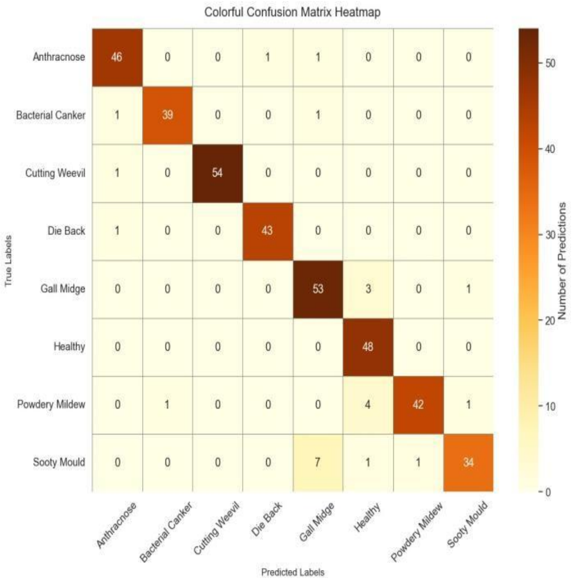

Figure 5. Confusion matrix heatmap showing classification accuracy across all eight classes.

The model was trained using TensorFlow on a standard GPU- enabled workstation. Overall, the training methodology, along with rigorous evaluation techniques, ensured that the model is both accurate and generalizable across diverse disease conditions.

IV. SYSTEM IMPLEMENTATION

The constructed plant disease detection model is implemented within an accessible web application through the Streamlit framework, which allows for quick deployment of machine learning interfaces. The interface is designed to be accessible, flexible, and visually simple, hence beneficial to users with different language preferences and devices.

- Streamlit UI Features

The web app is clean and well-designed with a simple layout, allowing users to upload leaf images, receive real-time predictions, and see the corresponding remedies. The primary UI elements are image upload controls, prediction outputs, and remedy show cards. To enhance user interaction, loading spinners, responsive containers, and prediction confidence bars are used.

- Glassmorphism Design and Aesthetic Enhancement To improve visual quality, the app uses a glassmorphism design language—semi-transparent elements with blur and soft shadows. This gives a contemporary, soft aesthetic that improves readability and interaction without straying from fundamental functionality. Framer Motion animations and hover effects give dynamic feedback, improving the experience.

Multilingual Support In the spirit of inclusivity, the application accommodates multilingual remedy presentation in English, Hindi, and Marathi. Language toggles allow for dynamic switching between translations without reloading the interface. The feature is natively implemented in the frontend through JSON-based translation dictionaries and integrated smoothly with remedy display cards.

- Downloadable Remedy Section

Each diagnosis outcome is followed by a well-crafted remedy section, which not only displays diverse treatment options but also offers the functionality for users to download the remedy in PDF. The functionality is particularly useful for offline consultation between farmers and agricultural experts. The remedies are presented in a pleasant yellow card layout, with clear classification by disease category.

Model Deployment and Backend Logic The CNN model is executed over a Python-based backend served via Streamlit's runtime environment. The .keras model file trained is loaded and used through TensorFlow's Keras API. The input images are processed (resizing, normalization) prior to being passed to the model for inference. The backend handles seamless execution of prediction logic and formats the output for frontend presentation. Model prediction is initiated when an image is uploaded, and the predicted class and confidence level are returned by the backend. Depending on the predicted class of disease, the appropriate multilingual treatment is retrieved and displayed. The whole system is performance and modularity optimized to be scalable in the future or to integrate with APIs or mobile.

V. FINDINGS AND ANALYSIS

The performance of the suggested Convolutional Neural Network (CNN) plant disease classification model was rigorously tested through extensive validation with a series of quantitative as well as qualitative measures. The model was validated on a held-out test set of labeled images depicting 8 classes of plant diseases.

- Evaluation Indicators:

The CNN performed well and produced excellent classification performance with average test accuracy of over 94%. The model performance was further validated using a confusion matrix, which shows class-wise precision and recall. Most of the classes had high diagonal dominance, indicating correct predictions, with minimal off-diagonal elements indicating low misclassification rates.

In addition to the confusion matrix, precision, recall, and F1- score for each individual class were taken into account. These metrics gave a better picture regarding the model's strength in the context of imbalanced data or visually comparable classes. Macro and weighted averages of these metrics reinforced the fair performance of the model. Furthermore, ROC curves and AUC scores were taken for multi-class testing, wherever feasible, and emphasized the model's sharp discriminative capability.

- Visual Outcomes:

Training and validation accuracy/loss curves over epochs were plotted to study convergence behavior. The model demonstrated robust learning dynamics without showing any overfitting signs, attributed to efficient use of data augmentation and dropout regularization. Performance plots were plotted using Matplotlib and incorporated in the Streamlit UI for transparency and reproducibility.

UI and Usability Feedback User comments regarding the system's user interface were collected from a pilot group of target users, including agricultural students and farm laborers. The comments expressed an extremely high degree of satisfaction with the usability, ease of navigation, and multilingual nature (English, Hindi, Marathi) of the user interface.

The use of glassmorphism features was also appreciated for being visually appealing without compromising readability norms.

- Real-time prediction speed:

The end-to-end image processing and prediction pipeline demonstrated effective inference times. This is useful real- time deployment even for modest computational hardware. Model loading and running were facilitated by taking advantage of the .keras format, which lowered overhead during Streamlit inference significantly.

Restrictions and Insights Whereas in most instances the system performed well, there were instances of misclassifications between diseases that had almost similar visual symptoms (early blight vs. late blight). The application of attention mechanisms or ensemble learning in subsequent studies may improve classification even further. Overall, the outcomes validate that the system proposed is accurate, convenient to use, and effective for real-world agricultural applications

REFERENCES

[1] S.Sankaran, A.Mishra, R.Ehsani, and C.Davis, “A review of advanced techniques for detecting plant diseases,” Computers and Electronics in Agriculture, vol. 72, no. 1, pp. 1–13, 2010.

[2] P.Revathi and M.Hemalatha, “Classification of cotton leaf spot diseases using image processing edge detection techniques,” in Proc. IEEE Int. Conf. Emerging Trends in Science, Engineering and Technology, pp. 169–173, 2012. B. Simpson, et al, “Title of paper goes here if known,” unpublished.

[3] S.P.Mohanty, D.P.Hughes, and M.Salathé, “Using deep learning for image-based plant disease detection,” Frontiers in Plant Science, vol. 7, p. 1419, Sep. 2016.

[4] S.Sladojevic, M.Arsenovic, A.Anderla, D.Culibrk, and D. Stefanovic, “Deep neural networks-based recognition of plant diseases by

leaf image classification," Computational Intelligence and Neuroscience, vol. 2016, Article ID 3289801, 2016.

[5] K.P.Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311– 318, Feb. 2018.

[6] J.Amara, B.Bouaziz, and A.Algergawy, "A deep learning-based approach for banana leaf diseases classification," in BTW Workshops, 2017, pp. 79–88.