International Journal of Computing and Digital Systems
2025, VOL. 17, NO.1, 1-15

http://dx.doi.org/10.12785/ijcds/1571064098

An intelligent leaf disease prediction for corn and maize using Convolutional Neural Network

Balasubramani M[1], Priyanka Nallusamy[2*], Adaline Suji R[3], Bright Anand D[4], Kunal Samnani[5] and Arumuga Arun R[6]

[1,2,3,6]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore – 632014, India,*
[4]*Computer Science and Engineering, Sreenivasa Institute of Technology and Management Studies (Sitams), Chittoor, Andhra Pradesh – 517127.*
[5]*Nextyn - Experts, Strategy Executive Search, Pinnacle Business Park, Mahakali Caves Rd, Gundavali, Andheri East, Mumbai, Maharashtra - 400093*

Received 16 August 2024, Revised 5 February 2025, Accepted 8 February 2025

Abstract: With around 40% of the population working directly in agriculture with an additional 20% active in related businesses, agriculture is an important contributor to India's economy. Corn, one of the most widely cultivated crops globally, is essential for food production, biofuels, and animal feed. It is quite subject to diseases, ranging though, which substantially reduce quality and farmer earnings, in particular adverse environmental circumstances. With advancements in digital technologies, modern farming methods increasingly rely on machine learning to monitor crop health and improve decision-making processes. Using a transfer learning methodological aspects, this study presents a deep learning-based framework for diagnosing diseases of corn leaves. A modified ResNet-18 Convolutional Neural Network is employed to classify leaf images into four categories—Common rust, Blight, Gray leaf spot, and Healthy. The model incorporates preprocessing techniques, data augmentation, and transfer learning to improve classification performance. Evaluation on a publicly available Kaggle dataset, consisting of 4,188 labeled images, demonstrated a high classification accuracy of 96%, outperforming conventional methods in precision, recall, and F1 scores. To enhance usability, the proposed framework integrates Gradio, a web-based interface, enabling farmers to upload leaf images and obtain real-time predictions. This interactive tool provides an accessible and reliable solution for disease detection, helping farmers mitigate crop losses and enhance productivity. The results indicate that the system can support large-scale agricultural applications and contribute to sustainable farming practices.

Keywords: ResNet-18, Convolutional Neural Network, Leaf diseases, Deep Learning, Corn.

1. Introduction

Leaf diseases represent the primary causes of crop failure, resulting in an approximate annual global output reduction of US$2000 billion [1], [2]. Plant health can be compromised by fungal, viral, or bacterial infections, leading to alterations in color, shape, or margins, and impacting leaves, fruits, or branches. Therefore, it is crucial to look at potential quick detection, control, and treatment solutions. The conventional method of the naked eye examination is the primary defense in disease identification, however, it is ineffective[3], [4], [5]. Furthermore, farmers might adopt the inappropriate chemical or treatment method[6]. TTypically, a plant pathologist is needed to evaluate and identify the particular kind of contamination, which is expensive, slower to act, or all of the above inaccessible[7], [8]. A crop that is widely grown in big amounts is a maze, which is also known as maize. In addition to being consumed directly as food, it also provides the essential raw material for an extensive variety of various products, which includes cooking oil, animal feed, flour, alcohol, starch, and biofuel. Alongside rice and wheat, corn stands out as one of the most crucial crops due to its substantial genetic variation and capacity for production, thriving across diverse environmental conditions[9]. In 2020, there were 1.15 billion tonnes of corn produced worldwide[10]. It is inherently vulnerable to a wide range of ailments that can affect the plant's leaves, trunk, and fruit at any stage of development. This directly affects corn harvest yield, which results in significant financial loss. Food shortages, famine, and possibly hunger may occur from the global production of essential crops like corn being reduced[11]. The deadliest of these ailments are those that affect the growth of maize leaves. In this work, Cercospora Leaf spot, Northern leaf blight and Common rust are three general leaf disease used[12].

The Cercospora leaf spot is caused by the fungi Cercospora zeina, and Cercospora zeae-maydis which alters the color and appearance of the leaves. These fungus live on the surface of the soil and produce necrotic lesions that range in color from black to grey and they demand a warm, moist

E-mail address: m.balasubramani@vit.ac.in, priyanka.n@vit.ac.in, adalinesuji.r@vit.ac.in, brightanand@gmail.com, kunal.samnani@nextyn.com, arumugaarun.r@vit.ac.in

environment. The administration of an appropriate fertilizer is a common treatment, but it has to be used before grain formation[13], [14]. Another fungus illness brought on by the Exserohilum turcicum fungus is Northern Leaf Blight. Crops that are in the process of growing and reproducing could suffer substantial losses if they strike during these times. Also, certain meteorological circumstances make it worse and deteriorate the growth of crops. The disease manifests itself on the leaves as angular, rectangular-shaped dark brown dots. The appropriate chemical agent is used to treat the disease[15], [16]. On the upper and lower surfaces of maize leaves, common rust manifests as dark, reddish-brown blisters and is brought on by the fungus Puccinia sorghi. Acceptable chemicals are used to treat it[17], [18], [19].

Numerous advances in agriculture have been made possible by modern technology. In particular, images of plants have an impact on machine learning as well as deep learning applications. Another kind of Deep Learning network is the Convolutional Neural Network (CNN) [20], which has a structure featuring rectified linear units, convolutional, and pooling layers that receive input into a fully corresponding layer. CNNs aim to detect image characteristics and associations. The various characteristics that the prior layers have uncovered are included in this layer. Despite multiple modifications that could affect the input images, CNNs are excellent at recognizing features[21]. When building systems based on CNNs, researchers have the option of starting from scratch or applying transfer learning. The process of reusing models and applying new approaches to innovative applications is called transfer learning. Following this technique, Deep transfer learning transforms contemporary models with partial or complete retraining in a way that suits the new application. It has the benefit of using earlier layers to identify generic properties (such as colors and borders) and tailoring subsequent levels for particular applications.

On the farmer's side, a CNN model is developed that can feed images directly from farmers' laptop computers. The model then performs disease detection and applies strategies for resizing and normalizing each dataset. Although data augmentation procedures are only used on the trained set to enrich the data with the intent that the model may generate more accurate findings. The model then displays the disease category as well as the confidence percentage and classification time it took to process the image. Farmers with minimal resources can now take photos of the infected plant leaves using a web app. On the user side, the web application runs on top of the CNN model. The application also shows the classification time required to process the image and the confidence percentage. In particular, a system is developed which is working locally on the PC of the user. We used a Kaggle open-source dataset that contained 4188 photos of healthy varieties and the three foremost prevalent diseases namely blight, a gray leaf spot, and common rust. Major contributions of the proposed work are,

- Employing transfer learning techniques to facilitate the creation of a web-based approach for immediate identification of leaf diseases.
- A technologically modified CNN model is implemented in which the farmers can directly feed images from their system.

In order to provide a disease detection tool that is both successful and easy to use, the Corn Leaf Disease Detection application with Gradio requires a number of critical user needs. The proposed structure has the following important features:

- *Easy-to-use interface:* Even for users who are unfamiliar with Machine Learning techniques or agricultural terminology, the Gradio interface for the disease detection tool is simple to use.
- *Quick and accurate results:* Users should be able to quickly and accurately obtain results for the disease detection tool. The tool should be able to process images quickly and provide accurate disease classifications within seconds.
- *Accurate disease classification:* The disease detection tool should be able to accurately classify corn leaf diseases, even when there are multiple diseases present in the same leaf or when the disease is in its early stages.
- *Clear and concise results:* The disease detection tool should provide clear and concise results that are easy for users to interpret. This may include visualizations or textual descriptions of the disease classification.
- *Reliable and secure:* Users should be able to trust that the disease detection tool is reliable and secure, with no risk of data breaches or other security issues.

Using a modified ResNet-18[22] CNN with transfer learning, this study introduces a distinctive deep learning framework for the precise categorization of corn leaf diseases. Unlike prior studies, which primarily focus on complex architectures requiring extensive computational resources, this approach achieves high accuracy while maintaining computational efficiency, making it accessible for real-world agricultural use. The key contributions includes,

- Advanced Model Design: Implementation of a pre-trained ResNet-18 model fine-tuned with transfer learning to classify corn leaf diseases with 96
- Comprehensive Dataset Utilization: Using Kaggle, an open-sourced dataset combined with 4,188 tagged photos improved through data enhancement and pre-processing techniques for better model generalization.
- Practical Application Tool: Integration of Gradio, a user-friendly web-based interface, enabling farmers

and researchers to upload images and receive real-time disease predictions without requiring technical expertise.

In particular, the Corn Leaf Disease Detection method with Gradio is focused on creating a user-friendly and effective disease detection tool that is accessible to an extensive range of users.

The purpose of this work is to create a deep learning-based framework for precisely identifying and categorizing diseases affecting corn leaf surfaces. The proposed solution leverages a modified ResNet-18 CNN with transfer learning techniques to address challenges like limited labeled datasets and high computational costs. To enhance accessibility, the system integrates Gradio, a user-friendly web-based interface, enabling farmers and researchers to upload leaf images and obtain real-time predictions, including confidence scores.

The section that follows summarizes the remainder of this study: The background, current situation, and recent developments for leaf disease detection are included in Section 2 . The requirements for deploying the proposed approach into practice are then briefly outlined. The proposed approach for addressing the challenge of detecting maize leaf disease is discussed in Section 4 along with some discussions. The findings of the simulation and testing are exhibited in Section 5, and the conclusion is specified in Section 6.

2. Related works

The contemporary schemes that are related to corn disease detection and the gaps found in the existing models are discussed in this section.

Rajeev et al.[23] used AlexNet a type of CNN model that has 5 convolution layers and 3 max pooling layers. In this technique, CNN was able to extract features directly by processing the raw images directly. However, it experienced lower accuracy for a lower number of epochs. Ali et al.[24] used CNN and image processing techniques for classifying the potato images into 5 classes namely Black Leg, Black Scurf, Pink Rot, Healthy, and Common Scab. This work utilized around 5000 images and achieved an accuracy of 99-100% in some of the classes. But, this technique failed to detect plant images with multiple diseases.

Monzurul et al.[25] used Multi-class support vector machines, imaging, and computer vision based on phenotyping were used to classify the potato images. This work paved the way for automated plant disease assessment on a broader scale, demonstrating a 95% accuracy rate on a dataset of around 300 images. But this technique used a very small dataset and the accuracy obtained is not precise. It was not able to perform with imbalanced image classes and also the noisy data provided uncertain results.

To start transfer learning for the categorization of corn illnesses, Wei et al.[26] introduced VGGNet [27], a form of CNN, together with the Adam optimizer. The evaluation score for this model is exceptionally high, with a 94.64% recognition rate. In contrast, this method was unable to identify plant images with numerous diseases.

A multi-layered deep learning model has been proposed by Javed et al.[28] for determining potato leaf diseases. First, the leaves were extracted from potato plant images using the YOLOv5 [29] image segmentation strategy. To differentiate between matured blight and premature blight on potato leaves, a separate Deep Learning method was created at the second level. Our technique was less parameter-intensive and easier to implement than state-of-the-art approaches. The impact of environmental factors on potato leaf diseases was also considered.

Using data augmentation techniques including image scaling, image transformation, image segregation, and and scaling down, Pan et al.[7] broadened their collection of 985 snapshots of healthy and unhealthy maize leaves to 31,005 images. The pytorch and kerras frameworks were used to implement numerous tested CNN models. With a 99.94% accuracy rate, the proposed methodology produced excellent results and an informative diagnosis of NCLB. Adversarial network-based data augmentation approaches can also improve the effectiveness of visual feature detection during the training of the DCNN models.

Divyanth et al.[30] developed a CNN model, to identify diseases in maize that combines depth with conventional artificial and feed-forward neural network methods. They introduced a new method for detecting and evaluating maize illnesses using a two-stage semantic segmentation procedure. During each step, semantic segmentation models were trained using various network architectures, including SegNet, DeepLabV3+, and UNet. The recognition rate and speed of this method are very high. It eliminates interference from the outside environment, quickly and accurately detects and identifies information on maize disease, and significantly increases detection accuracy. This method should take into account the various traits of diseases at various phases of disease development because incorrect decisions could have an impact on the recognition rate. Malusi et al.[31] trained CNN to recognize and categorize maize leaf diseases using Neuroph. The convolution and pooling feature extractions were included in the Neuroph library, which functioned as an IDE, to construct a more reliable CNN. Unfortunately, because the resolution setting of this method is limited to $10*20*3$ ($height*width*RGB$), the results of greyscale images are challenging.

Partial Least Squares (PLS) regression had been suggested by Farah et al.[32] as a method for determining attributes from a deep feature set acquired in an automated crop disease recognition system. Compared to the initial feature vectors, it used a fusion procedure that consumed longer to execute. The final vector also has a selection that failed to take into account some of the key attributes, which results in a very low accuracy.

Manavalan [33] examined around 109 articles that reported on early disease detection to upsurge production. The study's findings demonstrate that autonomous systems for diagnosing and classifying grain plant diseases are quite an infant stage. However, this research had trouble differentiating between diseases with related characteristics.

Rajeev et al.[23] experienced lower accuracy for a lower number of epochs. Ali et al.[24] were not able to detect plant images with multiple diseases. Monzurul et al.[25] used a very small dataset so the accuracy may not be very precise. It was not able to perform with imbalance image classes and also the noisy data was giving confusing results. Wei et al.[26] were not able to detect plant images with multiple diseases. Javed Rashid et al.[28] were not able to predict the severity of the diseases and were giving false results on leaves with multiple diseases. The study by Pan et al.[7] had a few shortcomings, like using an adversarial networks-based data augmentation approach, which could have improved the DCNN models' ability to detect features in images. A method that automatically changed hyperparameters might have been employed to improve performance.

The recognition rate may be impacted by incorrect estimation because Divyanth et al.[30] used an approach that ignores the various characteristics of diseases at different phases of the disease course. Malusi et al.[31] had some resolution setting constraints of $10*20*3$ ($height*width*RGB$) and also was not able to produce outputs for grey scale images. The accuracy was a little low at 90% since the Farah et al.[32] technique employed a fusion procedure that took longer to execute than the original feature vectors. The final vector also failed to take into account some of the major characteristics. Manavalan[33] faced issues in discriminating against diseases with similar properties.

Murugan et al.[34] used OpenMP to optimize performance on the Raspberry Pi hardware while constructing a CNN-based model for identifying corn leaf illnesses. For test samples, the model's maximum accuracy was 93%. However, the system faced limitations in handling complex real-world scenarios with overlapping disease features.

The systematic literature review by Mohamad et al. [35] looks at the utility of CNN in identifying illnesses in corn plants. By assessing different CNN models for the detection of maize leaf disease, the study seeks to improve precision agriculture and shed light on the efficacy of deep learning techniques in agricultural applications.

Olayiwola et al. [36] conducted a study that used a CNN-based deep learning system to detect three common maize leaf diseases: Leaf Blight, Common Rust, and Leaf Spot. With a 98.56% accuracy rate in accurately identifying the illnesses, the model showed promise for creating applications that would help plant pathologists and farmers determine and manage maize leaf diseases more effectively.

The motivation for this work stems from the global need to mitigate crop losses caused by leaf diseases, which significantly affect food security and farmer livelihoods. By providing a scalable and user-friendly detection tool, this research contributes to sustainable agriculture and supports efforts to address food security challenges. The system employs a modified ResNet-18 architecture, optimized with preprocessing and data augmentation for enhanced performance. Transfer learning enables efficient training with limited resources. Gradio facilitates a seamless user experience, offering real-time disease classification and confidence metrics in an intuitive interface.

3. Prerequisites

The system presumes that input images are of high quality and depict corn leaves. It also postulates that input images are correctly labeled according to their corresponding disease category. The system is constrained by the availability of labeled corn leaf images to train the ResNet-18 model.

A. System Requirements

The hardware requirements for the Corn Leaf Disease Detection project with Gradio would depend on the scale of the project and the size of the dataset. The application runs on a Central Processing Unit (CPU), but training and inference times may be slower compared to running on a Global Processing Unit (GPU). A GPU with CUDA support can significantly speed up training and inference times, especially for large datasets. The capacity of Random Access Memory (RAM) required depends on the dataset size, but at least 8GB of RAM is recommended. Storage requirements are determined by the dataset size and the model checkpoints number saved during training.

A very high network connection is required to install and download the necessary libraries and tools. Overall, the Corn Leaf Disease Detection application with Gradio executes on a standard computer with a CPU and at least 8GB of RAM. However, for optimal performance, a GPU with CUDA support and more RAM would be recommended, especially for larger datasets. Table I demonstrates the software requirements for the Corn Leaf Disease Detection application with Gradio. The software requirements for the Corn Leaf Disease Detection project with Gradio are mainly python libraries and tools, along with Flask for the web interface.

4. Proposed Work

A detailed explanation of CNN architecture, the proposed system, and the pre-trained model which is used are discussed in this section.

Residual Neural Network (ResNet), a CNN architecture, is utilized to create networks that outperform shallower networks by having up to hundreds of convolutional layers.

TABLE I. The approach for detecting corn leaf disease requires specific software

S.NO.	NAME OF THE SOFTWARE	DESCRIPTIONS
1.	Python	Python 3.7
2.	PyTorch	Deep learning library
3.	Gradio	Web interface
4.	NumPy	Numerical computing
5.	Matplotlib	Data visualization
6.	OpenCV	Image processing
7.	Pandas	Data manipulation
8.	Flask	Backend server for the Gradio web interface
9.	TorchVision	Image processing
10.	CUDA Toolkit	Optimal performance

This study makes use of ResNet-18, one of the variations that offer the benefit of being able to train on over a million photos in the ImageNet database. It consists of 18 layers of depth and is constructed with 72 layers. It classifies images into 1000 dissimilar object classifications, making it incredibly effective and useful in image classification. This enables a larger amount of CNN layers so that the classification is performed efficiently. However, having multiple deep layers leads to a vanishing gradient problem. ResNet's main goal is to employ jumping connections, frequently referred to by the terms shortcut connections or identify connections and the connections utilize the activation of previous layers. These hop from one layer to another creating a shortcut linkage between them. These identity mappings initially skip connections, using previous layer activations as a result. The skipping procedure compresses the network and henceforth learns earlier. After compression is completed, layer expansion occurs allowing the remaining parts of the network to train and explore feature space simultaneously. The network's input size is $224 * 224 * 3$, which has been predetermined. The network's intricate layered architecture essentially qualifies it as a Directed Acyclic Graph (DAG) network. Furthermore, it receives input from numerous layers and outputs to numerous layers.

This study aims to develop a user-friendly, high-accuracy framework for detecting corn leaf diseases using deep learning. The methodology consists of the following key stages namely Data Collection and Preprocessing, Model Architecture, Training Configuration, Real-Time Prediction Framework. Dataset utilized in this work is publicly available Kaggle dataset with 4,188 labeled images of corn leaves was used. The dataset consists of four classes: Blight, Gray Leaf Spot, Common Rust, and Healthy leaves. A pre-trained ResNet-18 architecture was fine-tuned using transfer learning. Initial layers of ResNet-18, trained on ImageNet, were used to extract general features like edges and textures. Final layers were retrained to classify specific corn leaf diseases. Stochastic Gradient Descent (SGD) is an optimizer included in the training configuration. It has a learning rate of 0.003, a batch size of 32, and 30 epochs. Training (70%), validation (20%), and testing (10%) sets were separated from the dataset. Real-Time Prediction Framework consists of Gradio, a Python-based library, was integrated to provide a web-based interface for uploading leaf images and obtaining instant disease predictions. The interface displays predictions, confidence scores, and processing times, making it accessible to non-technical users like farmers.

The proposed system architecture for the user-side laptop implementation of the maize leaf disease detector is shown in Figure 1. Layer 1 defines the Intermediate Representation (IR) model that runs on the device and undergoes training using the dataset, in addition to the deep learning model employed by the system (i.e., ResNet-18). The user interface based on Gradio that offers system users an interactive user interface is depicted in Layer 2 of the diagram. A detailed explanation of the architecture is given below,

- *User Interface:* Users can submit images of maize leaves to the Gradio interface's web-based user interface to receive predictions for the disease label associated with those images.
- *Application logic:* It includes the machine learning model that was accomplished to detect several forms of corn leaf diseases. This model is loaded into memory when the application starts up and is used to generate predictions for new images.
- *Gradio library:* It provides the backend for the web-based user interface. It handles tasks such as uploading images, displaying predictions, and managing user interactions.
- *Deep learning libraries:* Libraries such as PyTorch and Torchvision are used to construct and train the machine learning model that controls the application.
- *Deployment platform:* The application is deployed on a local machine.
- *Data sources:* The application employs a dataset of

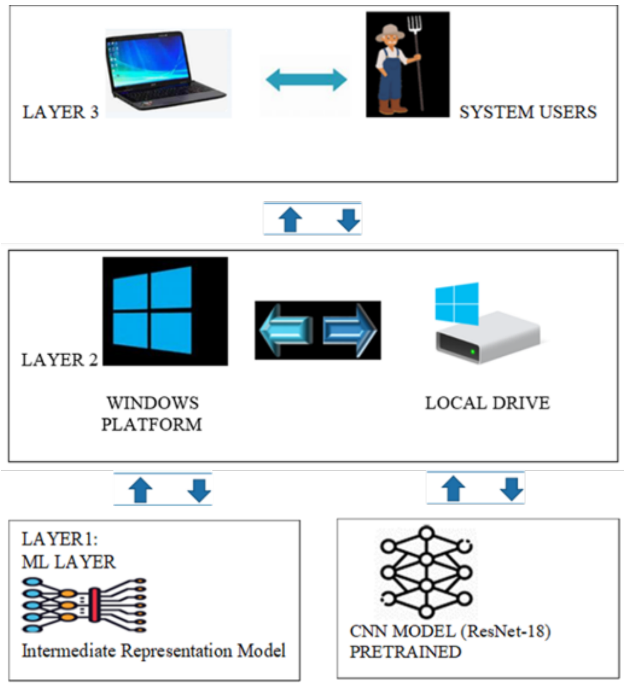

Figure 1. System architecture

labeled snapshots of leaves of maize to train the model. The application often retrieves this dataset when training through a database or file system.

The Corn Leaf Disease Detection application with Gradio has a client-server system architecture overall, with the deep learning libraries and application logic operating on the server side and the Gradio interface serving as the client. The deployment platform provides the necessary resources to run the application and manage user interactions.

The ResNet-18 model was used for categorizing diseases, followed by the proposed CNN that is based on data augmentation. When data augmentation improves the data, the transfer learning process begins, which increases the model's accuracy and generalization. Consequently, the ResNet-18 model is used for training which in turn accelerates the training process of CNN and uses test data feedback network training results. Using a pre-trained model, one can then adapt it to a new application by changing the output type and class count, for instance. Initial layers in this approach identify common low-level features like colors, edges, and blobs before subsequently learning the precise feature the customer needs. This is preferable to establishing random beginning weights since it speeds up learning, which is used in the current systems. Additionally, it facilitates learning from fewer images. The suggested method's process flow is shown in Figure 2.

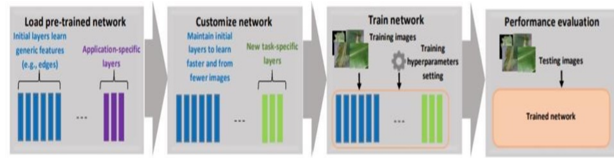

Figure 2. The process flow of the proposed disease detection framework

A. The proposed disease detection framework's process flow

By combining preprocessing, model training, and real-time prediction, the suggested framework ensures accuracy and efficiency in the detection and classification of maize leaf diseases. The steps are detailed below:

1. Data Acquisition and Preprocessing:
   - i Dataset Collection: The system makes use of a publicly accessible dataset from Kaggle that comprises 4,188 annotated snapshots divided into four groups: healthy leaves, common rust, blight, and gray leaf spot.
   - ii Image Preprocessing: To standardize the input dimensions used by the ResNet-18 model, images are scaled to 224x224 pixels.
   - iii Normalization: To enable more quickly and effectively training, pixel values are standardized between 0 and 1.
   - iv Data Augmentation: Rotation, flipping, and zooming are some of the techniques used to increase dataset variability, which improves the robustness and generalization of the model.

2. Model Architecture and Transfer Learning:
   - i Model Selection: ResNet-18, a deep CNN with residual connections, is chosen for its ability to handle vanishing gradients and extract high-level features effectively.
   - ii Transfer Learning: The model is initialized using pre-trained weights from the ImageNet dataset, which allows it to identify patterns with little computational effort and training data.
   - iii Fine-Tuning: The final layers of ResNet-18 are retrained to suit the specific task of classifying corn leaf diseases. The softmax activation function is applied to output probabilities for each class.

3. Model Training and Validation:
   - i Training Configuration:
     - Batch Size: 32
     - Learning Rate: 0.003
     - Optimizer: Stochastic Gradient Descent (SGD)
     - Epochs: 30
   - ii Data Splitting:
     - 70

- Cross-validation is performed to minimize overfitting.

iii Performance Metrics: Accuracy, precision, recall, F1-score, and ROC curves are computed to evaluate performance.

4. Deployment and Real-Time Prediction:
   i Web-Based Interface (Gradio): The model is integrated into Gradio, providing a user-friendly interface where farmers can upload leaf images directly from their devices.
   ii Real-Time Prediction: The framework processes input images instantly, displaying predictions along with confidence scores and processing times.
   iii Scalability and Accessibility: Designed to run on both CPU and GPU, making it adaptable to diverse hardware environments, including low-resource settings.

5. Output and Interpretation:
   i Classification Results: The model classifies the image into one of the four categories and displays the output, including: Predicted Disease Category, Confidence Score (%), Processing Time (Seconds).
   ii Performance Visualization: Confusion matrices and ROC curves are generated for post-analysis to highlight model accuracy and error patterns.

B. Role of the Gradio Library in the Framework

Gradio is a powerful Python library designed to create interactive web-based applications and machine learning model interfaces. In this study, Gradio is integrated to enable a seamless, user-friendly experience for deploying and testing the proposed deep learning model for corn leaf disease detection. Its role is elaborately listed below,

1. *User-Friendly Interface for Real-Time Predictions:* (i) Purpose-Gradio allows users, including farmers and agricultural researchers, to interact with the model without requiring programming knowledge. (ii) Functionality-Users can upload images of corn leaves directly through the interface. The uploaded image is processed in real-time, and the model predicts whether the leaf is healthy or affected by diseases such as blight, gray leaf spot, or common rust. Predictions are displayed with confidence scores (%) and classification results for transparency.

2. *Accessibility and Deployment:* (i) Cross-Platform Compatibility-The Gradio interface runs on web browsers, making it accessible on desktops, laptops, tablets, and mobile devices. (ii) No installation requirements for users-Users do not need to install specialized software or libraries, as the interface is hosted on a web server and accessible via a URL. (iii) Cloud and Local Deployment Support-The framework supports both local deployment for offline use and cloud deployment for scalability, allowing access even in remote agricultural areas.

3. *Features Enabling Practical Implementation:* (i) Drag-and-Drop Support-Farmers can easily upload leaf images using drag-and-drop functionality, simplifying the process for non-technical users. (ii) Live Testing and Feedback-Predictions appear instantly, and users can test multiple images without restarting the application. (iii) Error Handling and Visualization-Displays error messages if the input format is incorrect and supports visualizations, such as displaying uploaded images alongside predictions.

4. *Customization for Model Evaluation:* (i) Performance Visualization-ROC curves, confusion matrices, and performance metrics are integrated within the interface for model evaluation. (ii) Export Results-Users can export predictions and performance metrics for offline analysis and reporting.

5. *Practical Impact on Agricultural Applications:* (i) Real-World Use Case-Farmers can use their smartphones to capture leaf images in the field and upload them to the interface, enabling quick disease identification. (ii) Support for Agricultural Advisories-Agricultural extension officers can leverage the tool to diagnose diseases and provide treatment recommendations to farmers. (iii) Training and Demonstration Tool-The interface can also be used for educational purposes, helping researchers and students visualize model behavior and results.

Gradio significantly enhances the usability and accessibility of the proposed framework by providing an intuitive web interface for real-time testing. Its features make the tool practical for deployment in agricultural settings, empowering farmers and researchers with AI-based solutions for early disease detection and improved crop management.

C. Distinctive features of the proposed identification technique

Some of the main characteristics of the proposed Corn Leaf Disease Detection application are listed below:

- *Deep Learning model:* The application uses a Deep Learning model based on the ResNet-18 architecture to find maize leaves diseases. An extensive dataset of more than 14,000 images of maize leaves was used to train the model, which contributes to its excellent accuracy in disease identification.

- *Gradio interface:* The project uses Gradio to provide a user-friendly and interactive interface for disease detection. The Gradio interface allows users to upload images of corn leaves, adjust the threshold for disease detection, and get accurate predictions for the category of disease existing in the leaves.

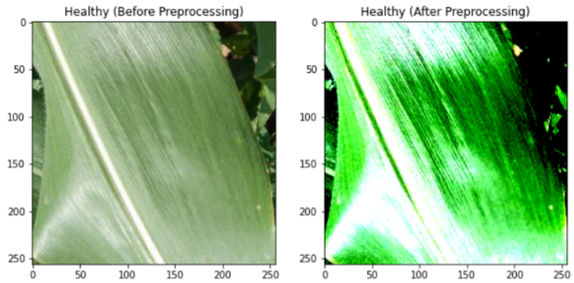

Figure 3. Process of data preprocessing

- *Multiple image upload:* The Gradio interface supports multiple image uploads, allowing users to upload and process multiple images of corn leaves simultaneously. This feature can help increase the efficiency of disease detection for large-scale agricultural operations.
- *Real-time prediction:* The Gradio interface provides a real-time prediction of disease presence in corn leaves as users adjust the threshold for disease detection. This feature allows users to see how the model responds to different thresholds and gain a better understanding of how the model works.
- *Cross-platform:* The Gradio interface is cross-platform, which means that it can be accessed from any device with a web browser. This feature allows users to access the disease-detection interface from a wide range of devices, including smartphones and tablets.

D. Non-functional Requirements

The system has an extreme accuracy rate in distinguishing corn leaf diseases, in addition, it classifies corn leaf images in real-time or near-real-time, with minimal delay. The system has a user-friendly interface that allows users to easily upload images and view the classification results. The system is scalable, allowing for additional disease categories to be added in the future.

- *Data Loading and Pre-processing:* It involves loading the images and applying some pre-processing steps such as resizing, normalization, and data augmentation. The time complexity of loading and pre-processing is determined by the integral value of images and the size of each image. For 'n' photos, the time complexity is $O(n)$ because each image is processed once, assuming that each image has an average size of $256 * 256$ pixels. The spatial complexity is proportional to the picture size and the number of images. The space complexity is $O(n * s)$ where 's' is the size of each image, assuming that the images are stored in memory. The background pre-processing procedure is shown in Figure 3.

5. Results and Discussions

The proposed framework offers several implications and insights beyond its technical contributions, highlighting its potential impact on agricultural practices and precision farming. It bridges the gap between technology and agriculture by reducing the dependency on manual expertise. It encourages the preventive measures and improves the scalability across diverse applications It adds economic and environmental benefits. However, it has certain limitations that includes the lack of real-world dataset variability and limited disease coverage.

This section discusses the evaluation matrix and performance metrics used in this work. Detailed requirement analysis for both functional and non-functional is performed. Based on the RAM capacity that is currently available, the starting batch size was set at 32 epochs. For the purpose of training the network, Stochastic Gradient Descent (SGD) was utilised, with a learning rate of 0.003. The dataset was partitioned into training, validation, and testing sets in order to accomplish data separation. Allocating 70% for training, 20% for validation, and 10% for testing was achieved using the split folder library. Data augmentation techniques were exclusively applied to the training set to enhance the dataset and improve the model's accuracy, while normalisation and resizing approaches were applied to all datasets. The following equations demonstrate some of the metrics that were used,

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precison(P) = \frac{TP}{TP + FP} \tag{2}$$

$$Recall(R) = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 * \frac{(P * R)}{(P + R)} \tag{4}$$

The True Positive (TP) in equations (1-4) denotes a correctly classified leaf image among disease states. A False Negative (FN) occurs when a leaf image is incorrectly classified as healthy when actually it belongs to a disease category. A False Positive (FP) indicates that a healthy leaf image was mistakenly identified as an infection. Additionally, TN stands for "True Negative," which is capable of correctly identifying a healthy leaf in an image. Out of all the positive images, the true positive rate (TPR) measures the extent to which the model can classify a leaf image as belonging to the proper disease class. Sensitivity is another name for TPR. A high sensitivity level may lead to a large number of FP, but it also means that leaf images are being recognized as disease representations quickly. Precision measure is defined as the percentage of FP relative to the total number of positives. Divide the total number of test images by the sum of TP and TN to calculate the accuracy. The F1 score is considered as an easier method to evaluate the model's performance in addressing class imbalances, especially when dealing with different categories that have uneven image amounts.

- *PyTorch:* Deep Learning model for disease detection in maize leaves is implemented by PyTorch, an open-source machine learning package.
- *NumPy:* It is a Python library for scientific computing that is used for numerical operations on corn leaf images.
- *Gradio:* A Python library for building and sharing custom machine-learning interfaces. It is used to create a user-friendly interface for disease detection in the Corn Leaf Disease Detection project.
- *Pillow:* It is a Python library for handling and processing image data. Whenever images of maize leaves are fed into the Deep Learning model, it is used to load and pre-process the images.
- *Matplotlib:* A Python library for creating visualizations. It is used to visualize the corn leaf images and their predicted disease labels.
- *Pandas:* It is a Python library for data manipulation and analysis. It is used to organize and manipulate the corn leaf image data.
- *Flask:* It is used to create a server for hosting the Gradio interface. It is python webframework that guarantees the efficacy and precision of the disease detection model. The Corn Leaf Disease Detection application using Gradio includes domain-specific criteria.

Some of the most important domain criteria for this approach are as follows,

- *Image quality:* The accuracy of the disease detection model depends on the quality of the input images. To ensure accurate results, the input images should be high-resolution and clear, with minimal noise and distortion.
- *Representative dataset:* When it comes to disease identification, the quality of the dataset means an excellent value. The training dataset should contain examples of all the many kinds of diseases that might harm maize crops so that the machine can identify them correctly in the leaves.
- The dataset used in this study was obtained from the publicly available Kaggle PlantVillage Dataset repository. It contains labeled images of corn leaves categorized based on different diseases and healthy samples[37].
- *Resolution and Preprocessing:* Original images in the dataset consists of varying resolutions. For consistency and compatibility with the ResNet-18 model, all images were resized to $224 * 224$ pixels during preprocessing. Additional preprocessing included normalization of pixel values to a range of [0, 1] to standardize inputs and facilitate stable gradient updates during training. It comprises 4188 images divided into 4 classes given in Table II.
- The dataset covers a diverse range of leaf appearances, ensuring representation of varying disease severities, lighting conditions, and angles.
- *Augmentation for Robustness:* (i) To further improve model generalization, data augmentation techniques such as random rotation (0–30 degrees), horizontal and vertical flipping, and zooming (0.8–1.2x) were applied. (ii) This effectively increased variability within the dataset, reducing overfitting and improving model performance under real-world conditions.
- Split Ratios:
  - i *Training Set:* 70% (2,931 images)
  - ii *Validation Set:* 20% (838 images)
  - iii *Testing Set:* 10% (419 images)

TABLE II. Classification of images with their count

S.NO.	NAME OF THE DISEASE	IMAGE COUNT
1	Blight	1146 images
2	Common rust	1307 images
3	Gray leaf spot	574 images
4	Healthy	1157 images

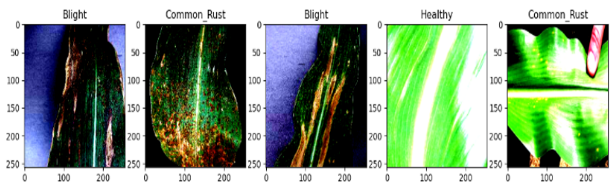

Figure 4. Sample images illustrating various disease

The images have been converted into .jpeg format and have $256 * 256$ pixel sizes and contains one leaf image in each file. Figure 4 displays samples of various diseases. The disease detection model must be able to accurately classify corn leaf diseases into different categories. To do this, the model must be trained on a taxonomy of corn leaf diseases that is accurate and up to date. The development and refinement of the disease detection model may require input from experts in the field of agriculture and plant pathology. Experts can provide insight into the most common and

```
In [19]: print(classification_report(test.targets, all_preds))

              precision    recall  f1-score   support

           0       0.95      0.90      0.92       115
           1       0.98      0.99      0.98       131
           2       0.84      0.92      0.88        59
           3       1.00      1.00      1.00       117

    accuracy                           0.96       422
   macro avg       0.94      0.95      0.95       422
weighted avg       0.96      0.96      0.96       422
```

Figure 5. Classification Performance Metrics Report

dangerous corn leaf diseases and help identify the key features that should be used for detection. As new corn leaf diseases are discovered and identified, the disease detection model must be updated to ensure that it can accurately detect these new diseases. Real-time updates to the model can help ensure that it remains effective and relevant over time. Overall, the Corn Leaf Disease Detection with Gradio has domain-specific requirements that are important for ensuring accurate and effective disease detection in corn crops.

Various Deep Learning models could produce different classification results, and need a variety of training and validation timeframes, otherwise, favor some classes over others. Additionally, while some models might be able to generalize to new data, others might not, depending on the training set. Moreover, the performance review is the selection of the images for each subset is random (e.g., training). The experiment should be repeated until positive findings are obtained, although this will not accurately reflect the performance. Machine Learning models might also be prone to overfitting and underfitting. Resizing and Normalizing methods were applied to all of the datasets but data augmentation procedures only pertained to the train set to enrich the data so that the model possibly will yield more accurate results. The testing accuracy of the model which we were able to achieve was around 96% after running the experiment 30 times.

The proposed application's categorizing report is illustrated in Figure 5. Here, *0* denotes the Blight class, *1* denotes common rust, *2* denotes gray leaf spot, and *3* denotes the Healthy class. From the observations, it is proved the proposed application achieved a 100% score in a healthy class for all the 3 parameters. The recall, accuracy and F1-score for the common rust class were all around 98%, 99%, and 98%, respectively. The class was able to acquire a 95% precision score, 90% recall score, and an 92% F1 score of blight. An 84% accuracy, 92% recall, and an 88% F1-score of gray leaf spot is achieved. Further deep diving into the results Confusion matrix is plotted for the model which is given in Figure 6. Figure 7 displays the accuracy, train, and validation losses that were plotted, and Table III lists the findings. Figure 8-11 represents the classification of images using confidence score in percentile for various leaf disease. Table IV discusses an extensive

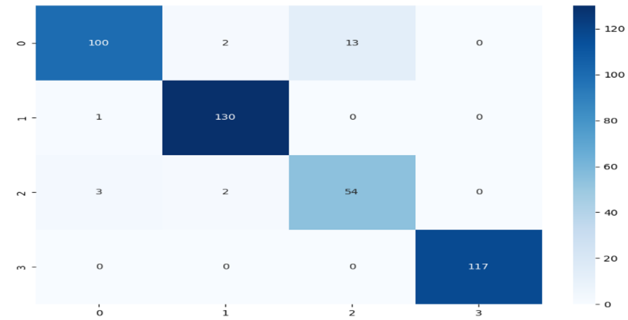

Figure 6. Confusion matrix

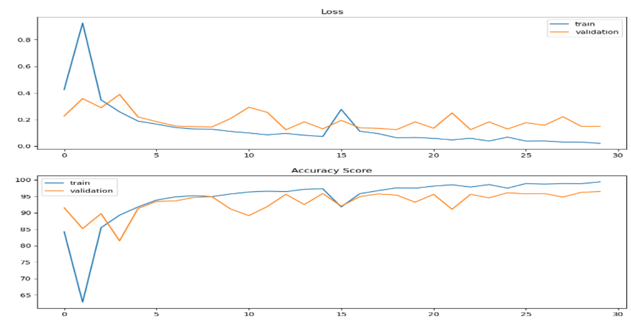

Figure 7. Train accuracy and Loss vs Validation accuracy and Loss

examination of the established investigation with other existing methodologies. Figure 14 describes the Receiver Operating Characteristic (ROC) curves used to analyze the model's predictive performance.

Padilla et al.[38] devised a system using Raspberry Pi as the hardware foundation. Demonstrating the feasibility of creating CNN-based applications with modest hardware, they employed OpenMP to enhance performance. Despite achieving an accuracy of up to 96%, their test accuracy reached only 93%.

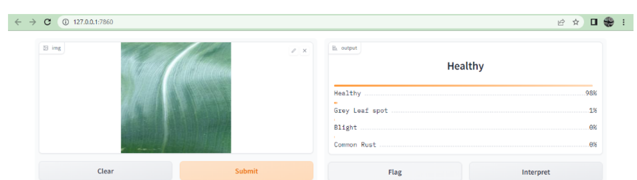

Figure 8. Healthy image classified with confidence score of 98 percentage

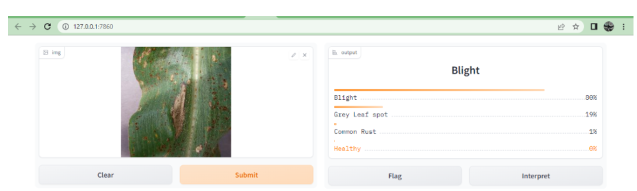

Figure 9. Blight classified with a confidence score of 80 percentile

TABLE III. Train Vs Validation Accuracy or Loss

S.NO.	NAME OF THE DISEASE	PARAMETERS	VALUES
1	Blight	Number of true positives Number of samples classified as Common rust Number of samples classified as grey leaf spot Number of samples classified as Healthy Number of false positives Accuracy	100 2 13 0 15 86%
2	Common Rust	Number of true positives Number of samples classified as Common rust Number of samples classified as grey leaf spot Number of samples classified as Healthy Number of false positives Accuracy	130 1 0 0 1 99.2%
3	Grey leaf spot	Number of true positives Number of samples classified as Common rust Number of samples classified as grey leaf spot Number of samples classified as Healthy Number of false positives Accuracy	54 2 3 0 5 91.52%
4	Healthy	Number of true positives Number of samples classified as Common rust Number of samples classified as grey leaf spot Number of samples classified as Healthy Number of false positives Accuracy	117 0 0 0 0 100%
5	Blight	True positives False positives Accuracy	401 21 95.02%

TABLE IV. The evaluation of the proposed model against existing methods

S.No.	EXISTING APPROACHES	OBJECTIVES	DATASET QUANTITY	TECHNIQUES	ACCURACY
1	Padilla et al.[38]	Classification of three different disease kinds, a healthy type, and an unidentified type	-	CNN employment using OpenMP on a Raspberry Pi	93%(Maximum)
2	Subramanian et al.[10]	Classification of three disease kinds, and a healthy type	18,888 images	VGG16, ResNet50, InceptionV3, and Xception	93.92 - 99.9%
3	Xu et al.[39]	Classification of 3 types of diseases and healthy type	17,600 augmented leaf images	VCGNET-16, Dense net, Resnet 50 and TCI ALEXNET(Modified Alexnet)	90% (mean)
4	Lu et al.[40]	Identify 10 common rice diseases	500 natural images	Multi-stage-CNN	95.48%
5	Proposed work	Classification of 3 types of diseases and healthy type	4188 augmented leaf images	Transfer learning on ResNet-18 framework in pytorch and kerras with gradio as frontend	96%

Figure 10. Common rust classified with a confidence score of 99 percentile

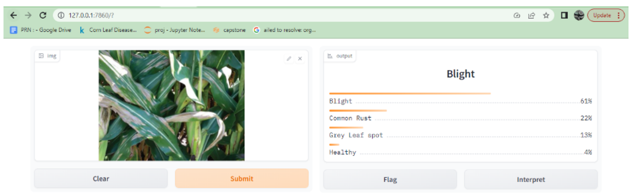

Figure 11. Blight classified with a confidence score of 61 percentile

Conversely, Subramanian et al.[10] explored multiple approaches to the interaction between two CNN models, EfficientNetB0 and DenseNet121. Their efforts yielded accuracies ranging from 93% to 99.993%. Lu et al.[40] implemented a method to augment the deep learning capability of CNNs, achieving significant advancements. Their CNN-based model successfully classifies 10 prevalent rice diseases through image recognition, boasting an accuracy of 95.48%. Meanwhile, Xu et al.[39] adapted AlexNet and developed a novel network to construct their CNNs, demonstrating innovative approaches in network architecture design.

The comparison of the proposed model with the existing techniques is given in Figure 13. Padilla-et al.[38] achieved recognition rates of 92% Leaf Blight, 89% Leaf Rust, and 89% Leaf Spot, respectively well our system achieved an accuracy of 86% for Blight, 99.2% for common rust, and 91.5% for grey leaf spot the mean accuracy is plotted in the above graph. Xu et al. [32] tried out 4 models i.e. VCGNET-16, Dense net, Resnet 50, and TCI ALEXNET, and respectively achieved a mean accuracy of 90%. Subramanian et al.[10] tried out the experiment in phases and the minimum accuracy achieved was 93% and the maximum achieved was 99% and an average accuracy of 97%. Lu et al.[40] implemented a technique to enhance the deep learning ability of CNNs and achieved an accuracy of 95.48% whereas our system achieved a mean accuracy

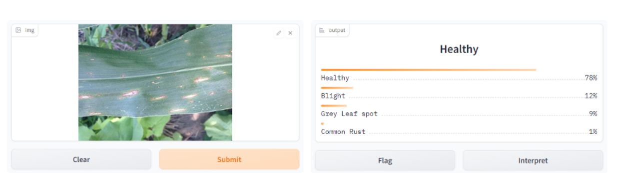

Figure 12. Blight classified as healthy

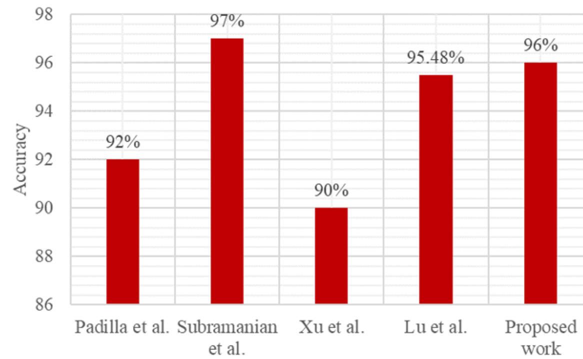

Figure 13. Comparison of Model Accuracy with Existing Approaches

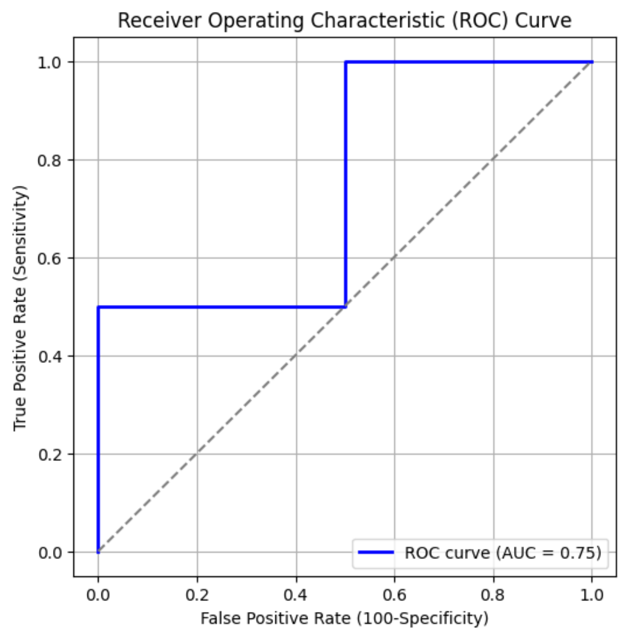

Figure 14. ROC Curves for Classification Performance Across Four Classes

of 96% which is better than these existing systems.

A. *Complexity Analysis*

- *Model Training:* Using the pre-processed data, the ResNet-18 model is trained. It takes a long time to train a model, according to the epoch count, batch size, and amount of the training dataset. The time complexity is $O((n*e)/b)$, where 'n' denotes Number of training samples, 'e' denotes number of epochs and 'b' denotes the batch size, if the model is trained for 'e' epochs with a batch size of 'b' and the training collection contains 'n' images. The magnitude of the model parameters and the batch size impact the space complexity. With the model kept in memory during training, the space complexity is $O(mb)$, where 'm' is the number of parameters.

- *Model Evaluation:* It involves evaluating the trained model on the validation set to measure its accuracy. Size of the validation set and processing time for each image determine the corresponding temporal complexity of the evaluation model. The time complexity is $O(m)$ if the validation set contains 'm' images and processing each image takes $O(1)$ time. Both the size of the validation set and the amount of memory required for keeping the model's predictions determine the space complexity. Assuming that the predictions are stored in memory, the space complexity is $O(m)$.
- *Inference:* It involves using the trained model to make predictions on new images. The time complexity of making predictions depends on the size of the input image and the time taken to process each image. Assuming that each image takes $O(1)$ time to process, the time complexity is $O(1)$ per image. The space complexity is based upon both the dimensions of the model parameters and the dimensions of the input image.The time complexity, assuming the model is kept in memory during inference, is $O(m + s)$, where 'm' is the number of parameters and s is the size of the input image.

The total time complexity of the proposed system is $O(e * n/b)$, where 'e' is the epoch count, 'n' is the picture count, and 'b' is the batch size. Since there are a finite number of parameters in the final model, their magnitude determines the space complexity, which is $O(m)$. Number of layers and size of each layer are the major factors that define the time complexity of the Deep Learning model used for disease diagnosis. For a single image, the temporal complexity of inference using the ResNet-18 model utilised in this study is $O(n^2)$, where 'n' is the number of layers. The time it takes to handle user queries and react with predictions, process and post-process images, and so on all contribute to the application's time complexity. Gradio can handle multiple user requests concurrently, but the overall response time may depend on the number of concurrent requests and the available resources. The space complexity of the application is mainly determined by the Deep Learning model size and the image number that is processed. The ResNet-18 model size is 44MB, which is relatively small compared to some other deep-learning models. The space required for image pre-processing and post-processing may also depend on the size and resolution of the input images and the number of images being processed concurrently.

6. Conclusions and Future Work

For both commercial and small-scale farming, corn is a key component of the diets of hundreds of millions of people worldwide. Furthermore, it creates the foundation for numerous industrial goods and biofuels. Yet, the effects of climate change on drought, severe weather, and unseasonably warm temperatures have severely impacted the world's agricultural output. Moreover, plant diseases can destroy maize yields and result in large financial losses. These reasons advance the requirement for incorporating technological advancements in various farming measures to preserve plants and provision farmers with the detection and restrain of diseases. In this paper, we intended to identify common maize illnesses from leaf images using Deep Learning algorithms. The performance evaluation's findings show that there is a significant amount of potential for creating and implementing commercial applications that meet the standards for accuracy and usability. Such measures could significantly assist farmers overcome maize diseases and preserve their standard of life. The investigation carried out in this work can be conceivably improved through an enhanced addition process with more images from the dataset's four classifications. Real-world input photos, for instance, could be of any type of background, which may not be identical to or consistent with the current dataset.

References

[1] G. Storey, Q. Meng, and B. Li, "Leaf disease segmentation and detection in apple orchards for precise smart spraying in sustainable agriculture," *Sustainability*, vol. 14, no. 3, p. 1458, 2022.

[2] A.-K. Mahlein, "Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping," *Plant disease*, vol. 100, no. 2, pp. 241–251, 2016.

[3] K. Kranthi Kumar, J. Goddu, P. Siva Prasad, A. Senthilrajan, and L. V. K. Rao, "An efficient approach for identification of multi-plant disease using image processing technique," in *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2021*. Springer, 2022, pp. 317–325.

[4] C. R. Rahman, P. S. Arko, M. E. Ali, M. A. I. Khan, S. H. Apon, F. Nowrin, and A. Wasif, "Identification and recognition of rice diseases and pests using convolutional neural networks," *Biosystems Engineering*, vol. 194, pp. 112–120, 2020.

[5] G. G. Manvi, K. Gayana, G. R. Sree, K. Divyanjali, and D. Patil, "Plant disease detection," *Int. J. Res. Appl. Sci. Eng. Technol*, vol. 10, pp. 4538–4542, 2022.

[6] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems engineering*, vol. 144, pp. 52–60, 2016.

[7] S.-Q. Pan, J.-F. Qiao, W. Rui, H.-L. Yu, W. Cheng, K. Taylor, and H.-Y. Pan, "Intelligent diagnosis of northern corn leaf blight with deep learning model," *Journal of integrative agriculture*, vol. 21, no. 4, pp. 1094–1105, 2022.

[8] I. Buja, E. Sabella, A. G. Monteduro, M. S. Chiriacò, L. De Bellis, A. Luvisi, and G. Maruccio, "Advances in plant disease detection and monitoring: From traditional assays to in-field diagnostics," *Sensors*, vol. 21, no. 6, p. 2129, 2021.

[9] H. Kaur, S. Kumar, K. Hooda, R. Gogoi, P. Bagaria, R. Singh, R. Mehra, and A. Kumar, "Leaf stripping: an alternative strategy to manage banded leaf and sheath blight of maize," *Indian Phytopathology*, vol. 73, pp. 203–211, 2020.

[10] M. Subramanian, K. Shanmugavadivel, and P. Nandhini, "On fine-tuning deep learning models using transfer learning and hyper-parameters optimization for disease identification in maize leaves," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13 951–13 968, 2022.

[11] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using efficientnet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2021.

[12] B. Shiferaw, B. M. Prasanna, J. Hellin, and M. Bänziger, "Crops that feed the world 6. past successes and future challenges to the role played by maize in global food security," *Food security*, vol. 3, pp. 307–327, 2011.

[13] J. E. Cairns, K. Sonder, P. Zaidi, N. Verhulst, G. Mahuku, R. Babu, S. Nair, B. Das, B. Govaerts, M. Vinayan *et al.*, "Maize production in a changing climate: impacts, adaptation, and mitigation strategies," *Advances in agronomy*, vol. 114, pp. 1–58, 2012.

[14] A. Tripathi, D. K. Tripathi, D. Chauhan, N. Kumar, and G. Singh, "Paradigms of climate change impacts on some major food sources of the world: A review on current knowledge and future prospects," *Agriculture, ecosystems & environment*, vol. 216, pp. 356–373, 2016.

[15] R. L. De Rossi, F. A. Guerra, M. C. Plazas, E. E. Vuletic, E. Brücher, G. Guerra, and E. Reis, "Crop damage, economic losses, and the economic damage threshold for northern corn leaf blight," *Crop Protection*, vol. 154, p. 105901, 2022.

[16] M. B. Zahra, B. Fayyaz, Z.-E.-H. Aftab, A. Akhter, T. Bahar, W. Anwar, and M. S. Haider, "Characterization and utilization of cow manure biochar as soil amendment for the management of northern corn leaf blight," *Journal of Soil Science and Plant Nutrition*, vol. 22, no. 3, pp. 3348–3363, 2022.

[17] U. Dey, S. Harlapur, D. Dhutraj, A. Suryawanshi, and R. Bhattacharjee, "Integrated disease management strategy of common rust of maize incited by puccinia sorghi schw," *African Journal of Microbiology Research*, vol. 9, no. 20, pp. 1345–1351, 2015.

[18] J. Sucher, R. Boni, P. Yang, P. Rogowsky, H. Büchner, C. Kastner, J. Kumlehn, S. G. Krattinger, and B. Keller, "The durable wheat disease resistance gene lr34 confers common rust and northern corn leaf blight resistance in maize," *Plant biotechnology journal*, vol. 15, no. 4, pp. 489–496, 2017.

[19] P. Wright, M. Parker, R. Van Tilburg, and D. Hedderley, "Effect of planting dates and azoxystrobin fungicide application regimes on common rust of maize," *New Zealand Journal of Crop and Horticultural Science*, vol. 42, no. 2, pp. 99–110, 2014.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[21] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia computer science*, vol. 132, pp. 377–384, 2018.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] R. K. Singh, A. Tiwari, and R. K. Gupta, "Deep transfer modeling for classification of maize plant leaf disease," *Multimedia tools and applications*, vol. 81, no. 5, pp. 6051–6067, 2022.

[24] A. Arshaghi, M. Ashourian, and L. Ghabeli, "Potato diseases detection and classification using deep learning methods," *Multimedia Tools and Applications*, vol. 82, no. 4, pp. 5725–5742, 2023.

[25] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," in *2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE)*. IEEE, 2017, pp. 1–4.

[26] W. Wenxuan, W. Qianshu, H. Chaofan, S. Xizhe, B. Ruiming, and T. T. Toe, "Leaf disease image classification method based on improved convolutional neural network," in *2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*. IEEE, 2022, pp. 210–216.

[27] K. Simonyan, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] J. Rashid, I. Khan, G. Ali, S. H. Almotiri, M. A. AlGhamdi, and K. Masood, "Multi-level deep learning model for potato leaf disease recognition," *Electronics*, vol. 10, no. 17, p. 2064, 2021.

[29] R. Khanam and M. Hussain, "What is yolov5: A deep look into the internal features of the popular object detector," *arXiv preprint arXiv:2407.20892*, 2024.

[30] L. Divyanth, A. Ahmad, and D. Saraswat, "A two-stage deep-learning based segmentation model for crop disease quantification based on corn field imagery," *Smart Agricultural Technology*, vol. 3, p. 100108, 2023.

[31] M. Sibiya and M. Sumbwanyambe, "A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks," *AgriEngineering*, vol. 1, no. 1, pp. 119–131, 2019.

[32] F. Saeed, M. A. Khan, M. Sharif, M. Mittal, L. M. Goyal, and S. Roy, "Deep neural network features fusion and selection based on pls regression with an application for crops diseases classification," *Applied Soft Computing*, vol. 103, p. 107164, 2021.

[33] R. Manavalan, "Automatic identification of diseases in grains crops through computational approaches: A review," *Computers and Electronics in Agriculture*, vol. 178, p. 105802, 2020.

[34] S. Santhi, M. Murugan, T. Srinivasan, and K. Shanthi, "Corn leaf disease detection using deep learning techniques," in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. IEEE, 2024, pp. 1536–1540.

[35] M. I. Abas, S. Syarif, I. Nurtanio, and Z. Tahir, "Detection of corn plant diseases using convolutional neural network: A review," in *AIP Conference Proceedings*, vol. 2952, no. 1. AIP Publishing, 2024.

[36] J. O. Olayiwola and J. A. Adejoju, "Maize (corn) leaf disease detection system using convolutional neural network (cnn)," in *International Conference on Computational Science and Its Applications*. Springer, 2023, pp. 309–321.

[37] https://www.kaggle.com/datasets/emmarex/plantdisease.

[38] D. A. Padilla, R. A. I. Pajes, and J. T. De Guzman, "Detection of corn leaf diseases using convolutional neural network with openmp implementation," in *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*. IEEE, 2020, pp. 1–6.

[39] Y. Xu, B. Zhao, Y. Zhai, Q. Chen, and Y. Zhou, "Maize diseases identification method based on multi-scale convolutional global

pooling neural network," *IEEE access*, vol. 9, pp. 27 959–27 970, 2021.

[40] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, "Identification of rice diseases using deep convolutional neural networks," *Neurocomputing*, vol. 267, pp. 378–384, 2017.