



AI-Based Crop Monitoring System: Cotton Disease Detection **Using CNN**

Aarya Sonawane

Dept. of Artificial Intelligence and Machine Learning

MPSTME, SVKM's NMIMS

Shirpur, India [aarya.sonawane038@nmims.in]

Anmol Patil

Dept. of Artificial Intelligence and Machine Learning

MPSTME, SVKM's NMIMS

Shirpur, India [anmol.patil052@nmims.in]

Dr. Suyog Pande

Dept. of Artificial Intelligence and Machine Learning

MPSTME, SVKM's NMIMS

Shirpur, India [Suyog.Pande@nmims.edu]

Abstract- Agricultural production depends on the diseases that affect plants and pests; early recognition is essential to crop management. Traditional visual analysis is labor-intensive and subjective because it might depend on the expertise of an analyst; this paper presents an AI-driven cotton disease identification system using Convolutional Neural Networks (CNNs). This model is supported by a web dashboard and a mobile app. The model was trained using 9,495 images of cotton leaves from nine categories: Alternaria Leaf Spot, Aphids, Army Worm, Bacterial Blight, Fusarium Wilt, Healthy, Powdery Mildew, Target Spot, and Verticillium Wilt. All images were, in other words, scaled to 224×224 pixels and normalized to zero-to-one in the real-life scenario. Augmentation such as rotation, shifting, zooming, horizontal flipping, and shearing is implemented during the training process. What I mean to say is that the dataset was split into 7,594 training images, 947 validation images, and 954 test images. Custom CNN reached the accuracy of 95.18%, with the macro F1-score of 93.49% and the weighted F1-score of 95.15%. Flask-based backend supports the application, while the web dashboard provides the information and the model's analytics. Further evaluation using images collected under the mixed real-field conditions is needed to establish concrete reliability and generalization.

Index Terms- *Artificial Intelligence, Cotton Disease Detection, Convolutional Neural Network, Deep Learning, Image Processing, Precision Agriculture, on the whole.*



I. Introduction

Cotton is mainly one of the most significant commercial crops and weighs heavily in the agricultural economy, yet cotton plants are typically vulnerable to different diseases and pest-related conditions that can negatively alter plant health and crop productivity, in many respects; indeed, early identification of these conditions is vital for supporting timely monitoring and suitable agricultural management, in actual application. More often than not, plant diseases are, more accurately, spotted through the visual inspection by farmers or agricultural experts, and while, this method is widely employed, it can be time-consuming and may rest heavily on the observer's experience, particularly when multiple disease symptoms appear comparable, so an automated and consistent strategy to plant disease identification is highly desirable.

Recent advances in AI technology, especially Deep Learning, have provided new possibilities for automated plant disease diagnosis, to an extent. Convolutional neural networks (CNNs) have demonstrated the potential of classifying images through automatic learning of relevant features including edges, textures, patterns and colour variations from input images. In particular, unlike classical image processing techniques that generally involve hand-crafted features, CNN learns layered representation from images automatically. Deep learning algorithms have shown their potential in detecting and classifying plant diseases based on leaf images, as illustrated in some earlier studies [2], [3], [12], [13]. The advancement in the development of deep CNN models has also improved recognition accuracy on image data [11].

Nevertheless, creating such a disease detection system entails more than reaching high classification accuracy; on the contrary, the correct approach requires the development of an easily accessible interface where people can upload images of leaves and get predictions and information about potential diseases in general. There has been quite some research conducted in agricultural monitoring which is concerned with using computer vision, mobile applications, and drones in order to help recognize the state of crops [5], [6].

To deal with the above problem, This paper discusses an Artificial Intelligence-based Cotton Disease Detection System Using Convolutional Neural Network (CNN). In this connection, the proposed system makes use of a tailored Sequential CNN model that has classified the cotton leaf images into nine different classes including Alternaria Leaf Spot, Aphids, Army Worm, Bacterial Blight, Fusarium Wilt, Healthy, Powdery Mildew, Target Spot, and Verticillium Wilt, etc.; the dataset has 9,495 images and contains training, validation, and testing portions. Image preprocessing, normalization, and data augmentation are used to preprocess the dataset for the training process; the trained model achieves a test accuracy of **95.18%**.

In addition, for the classification model, the project incorporates a backend built using Flask, and the web dashboard designed using React, besides a mobile application interface; all of this setup is



aimed at the moment towards making the process of disease classification easier through integration of predictions based on images with an application interface.

The primary insight of this work is the development of a cotton leaf disease classification approach employing a custom CNN architecture, combined with an application-oriented monitoring framework, broadly speaking; the suggested method, system architecture, experimental results, and conclusion are being outlined by the following sections, to a significant degree.

II. Related Work

A. CNN-Based Plant Disease Detection

Tejaswini et al. [1] studied CNN-based plant disease detection and reported an accuracy of 86.21%, on the whole; what their study showed was the potential of convolutional neural networks to spot disease-related visual patterns from plant images, broadly speaking. Sutaji and Rosyid [2] evaluated pretrained CNN architectures for crop disease classification, reporting 98.97% accuracy using DenseNet121. Their findings pointed out the significance of taking into consideration the trade-off between accuracy and computational needs.

B. Deep Learning Approaches for Crop Disease Classification

Shoaib et al. [3] have proposed novel deep learning methodologies for plant diseases as well as problems related to availability of labelled data, computational performance and generalization capability to some extent. Similarly, Salka et al. [4] have reviewed the classification of plant leaf diseases using CNNs and highlighted the benefits of feature extraction, as well as the problems encountered in the field environment; this research has been helpful in understanding the merits and demerits of deep learning methodologies for plant disease detection to some extent.

C. Drone-Assisted Agricultural Monitoring

The methodology that incorporated drone-based agricultural surveillance with CNN-based image recognition techniques along with application interfaces was put forward by Shah et al. [5]. 98.80% accuracy level was achieved with their EfficientNet-B3 model in the experimental setup described; yang et al. [6] came up with an intelligent crop health surveillance mechanism which made use of drone remote sensing along with CNN-based image recognition system. The modified MobileNetV2 model of theirs achieved 97.14% accuracy with an estimated size of 2.01 MB; these examples practically demonstrate the scope of integration between aerial imaging and AI-based crop monitoring systems.

D. Advanced Grouping and Object Detection Techniques

When it comes to the practical application in disease detection in tomatoes, Sowmya and Guruprasad [7] have employed Inception-v4 for category detection while utilizing YOLOv8 for object detection. They obtained an accuracy rate of 96% when it came to categorizing disease symptoms, whereas the mAP@0.5 value was 86%. On the other hand, Manoj et al. [8] have studied the implementation of drones for disease detection in crops.

E. Explainable AI and Attention-Based Disease Detection

Chihaoui et al. [9] provided a review on the use of AI for crop disease detection via leaf images and explainable AI. Some of the drawbacks identified by the authors include biased datasets, comments, cross-domain validation, and computational constraints. Gani et al. [10] performed research on the use of attention-based CNN with explainable AI for papaya leaf diseases.

F. Foundational Research on CNN and Plant Disease Classification

The efficiency of using deep convolutional neural networks in large scale image classification was proved by Krizhevsky et al. [11]. The application of deep learning in image based plant disease detection was studied by Mohanty et al. [12]. In reality, deep learning models for detecting and diagnosing plant diseases were further experimented with by ferentinos [13].

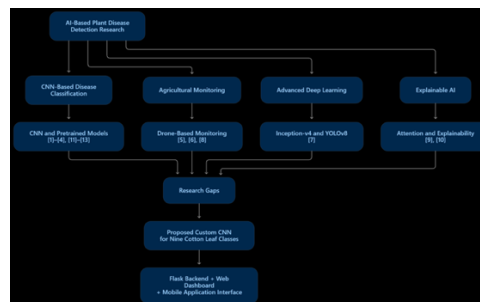


Fig.1. Category of Existing Research Areas

G. Research Gap and Motivation for the Proposed System

In this sense, it can be seen from the studies discussed above that CNNs and pre-trained models have the ability to reach very good classification results on different plant-disease datasets; however, important problems still exist in terms of field testing, computational efficiency, and disease-classification models' integration with monitoring systems. Still, performance depends on the dataset used, crops, and classes as well as evaluation methods.



Fig.2. Image Preprocessing Pipeline

Motivated by these problems, the present work is constructing a custom CNN model for classifying nine cotton leaf categories and connects it with a Flask-based backend, a React web dashboard, and a mobile application interface, in actual application, and the model hit 95.18% accuracy on its held-out cotton test dataset. The proposed framework focuses on linking disease classification using images with the digital interface. Going a step further validation employing images collected under the real agricultural conditions is needed to establish its field-level reliability, in actual application.

III. Materials and Methods

A. Dataset Collection and Description

The proposed system is employing a labelled cotton leaf image dataset collected from publicly available sources, including Kaggle and Google Dataset Search, to a significant degree; the dataset is comprising 9,495 images belonging to nine cotton disease and health categories, broadly speaking. Each picture, however, is linked to its corresponding class label and is employed for supervised learning in the CNN; the pictures are in JPG, JPEG, and PNG format.

The nine categories included in this research are generally Alternaria Leaf Spot, Aphid, Army Worm, Bacterial Blight, Fusarium Wilt, Healthy, Powdery Mildew, Target Spot, and Verticillium Wilt, in general; indeed, in practical cases, the data set is usually categorized into three sets including training, validation, and testing for learning, validating, and evaluating the performance of the model.

Table III. *Dataset Distribution*

Dataset subset	Number of images
Training	7,594
Validation	947
Testing	954
Total	9,495

B. Image Preprocessing and Augmentation

Preprocessing is done on the image in order to ensure the consistency of the input data. It is extremely crucial so that the input images can be prepared for classification using CNNs. In practice, normalization of the pixel intensity values from 0 to 255 to 0 to 1 is achieved by dividing all the pixel intensity values by 255.

Input images are actually resized to 224×224 with 3 RGB channels. Data augmentation can be used in order to increase diversity in the dataset and prevent overfitting of the model. Validation

and testing datasets will not undergo any random augmentation. Image rotation, width and height shifting, zooming, horizontal flipping, and shearing are being encompassed by the augmentation operations; these transformations introduce changes in image orientation and appearance while keeping the original disease category.

The pre-processed images obtained as such are fed into the CNN with an input size of $224 \times 224 \times 3$, overall.

C. Proposed CNN Architecture

The neural network architecture comprises four convolutional blocks containing 32, 64, 128 and 256 number of filters each; the convolutional layers extract visual features like edges, textures, colour changes, spots and disease symptoms from the input image data in general; therefore, a customized Sequential Convolutional Neural Network is developed for classification of cotton leaf images in practice.

Batch Normalization is included to stabilize the learning process; so, dropout layers are employed to decrease overfitting during the training; these days, max Pooling lowers the spatial dimensions of feature maps, broadly speaking; Global Average Pooling converts the feature maps into a compact feature representation after feature extraction, broadly speaking. A fully connected Thick layer covering 256 neurons is then applied, followed by a final Softmax output layer comprising nine neurons.

What's more, the class with the highest output score is picked as the forecasted category, to a significant degree; in fact, output scores for the nine cotton leaf categories are being created by the Softmax layer, to a significant degree.

Table IV. *CNN Architecture Components*

Component	Configuration
Input layer	$224 \times 224 \times 3$ RGB
Convolutional blocks	4
Convolution filters	32, 64, 128, 256
Normalization	Batch Normalization
Pooling	Max Pooling
Regularization	Dropout
Feature aggregation	Global Average Pooling
Dense layer	256 neurons
Output layer	9 neurons with Softmax

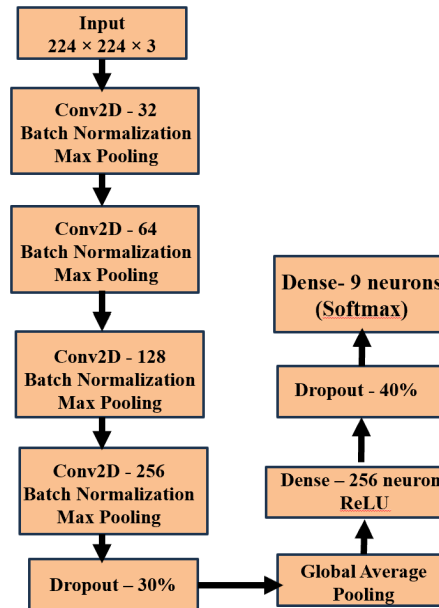


Fig.3. CNN Architecture

D. Model Training Configuration

The training process is using labelled cotton leaf images to learn the relationship between visual attributes and their associated disease categories, in many respects; in fact, the CNN model is trained employing TensorFlow with the Keras framework, so the Adam optimizer is used to update model parameters by minimizing the classification loss, in many respects.

Categorical Cross-Entropy is, in practice, chosen as the loss function for the nine-class classification task. The model is trained for 30 epochs with a batch size of 32 and a learning rate of 0.0001, on the whole. The validation dataset is employed to monitor the training process and assess model performance during the training, in real-world scenarios.

Table V. Model Training Parameters

Parameter	Configuration
Framework	TensorFlow with Keras
Architecture	Custom Sequential CNN
Optimizer	Adam
Learning rate	0.0001
Loss function	Categorical Cross-Entropy
Batch size	32
Epochs	30
Input dimensions	224 × 224 × 3

Number of classes	9
Training hardware	T4 GPU

E. System Process and Application Integration

The proposed system is following an image-based classification workflow that's connecting image input, preprocessing, CNN inference, and prediction presentation, to a significant degree, and a cotton leaf image is picked by the user and supplied to the prediction workflow, in actual application. In reality, the picture is processed and validated based on the needs of the trained CNN, and from that, scores are generated from the nine classes that are supported, and also, the class having the highest score is selected as the prediction.

Flask backend is developed with the aim of providing a way for interaction between the trained machine learning model and application interfaces, which is why the web dashboard has been developed using React framework for purposes of allowing the user to upload images, make predictions, visualize results, and monitor data, amongst others.

The whole process can be summed up as follows:

1. Obtaining the image of cotton leaf.
2. Validation and pre-processing of the image.
3. Extracting features and classification through CNN.
4. Prediction of one of the nine classes of cotton leaves.
5. Sending of prediction result through backend.
6. Display of results through the web and mobile application interfaces.

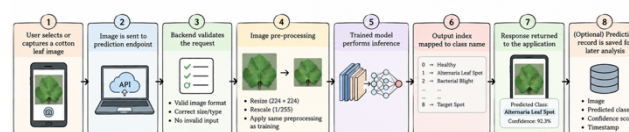


Fig.4. Unified Crop Monitoring Workflow

F. Performance Evaluation Metrics

The performance of the trained CNN is tested by use of the common classification measures such as the accuracy, precision, recall, and F1 score, which measure the performance of the classification process and how well the model can identify the different types of cotton leaves.

Precision measures the proportion of correctly classified images among all tested images; so, precision is suggesting the proportion of correctly predicted positive samples, to a significant degree; at the same time, Recall is the ratio of true positives correctly identified by the machine learning algorithm in real-life situations. F1-score is the harmonic mean of precision and recall.

The evaluation measures are, more accurately, employed along with the confusion matrix to study classification performance, across the nine categories, to a significant degree.

IV. Experimental Setup and Evaluation Protocol

A. Experimental Environment

The cotton leaf disease classification model was devised employing Python and the TensorFlow framework with Keras, in real-world scenarios; indeed, a custom Sequential Convolutional Neural Network (CNN) was used for multiclass image classification. The model training process was ran employing a T4 GPU to support computationally intensive operations, in real-world scenarios. What's more, numPy and other supporting libraries were used for data processing and model development, broadly speaking.

The input images were standardized to a resolution of 224×224 pixels with three RGB channels, in actual application. The model was set up to classify images into the nine predefined cotton leaf categories, in actual application.

B. Dataset Partitioning

A total of 9,495 labelled cotton leaf images were used for the experiments. In turn, the dataset was divided into the three subsets, such as training and validation, broadly speaking. On the whole. As well, so, what's more, the testing subset was reserved for final check of the trained model, which is why in the meantime, the validation subset was used to monitor the learning process, in real-world scenarios. And the training subset was used to learn model parameters.

Table VI. Dataset Partitioning

Dataset subset	Number of images	Percentage
Training	7,594	80%
Validation	947	10%
Testing	954	10%
Total	9,495	100%

C. Training Configuration

The model was trained for 30 epochs employing a batch size of 32 and the Adam optimization algorithm, in actual application; and thus, training images were subjected to data augmentation techniques, covering rotation, shifting, zooming, horizontal flipping, and shearing, to introduce changes in the training samples, in real-world scenarios. The learning rate was set to 0.0001, and categorical Cross-Entropy was used as the loss function.

The validation dataset was, as a matter of fact, employed to monitor model learning and appraise its performance during the training, to a significant degree; the same preprocessing requirements were sustained for the validation and testing images, without randomized augmentation, on the whole.

D. Check Protocol

The trained CNN model was found employing the autonomous test subset including 954 images after completion of the training process, in actual application; indeed, the anticipated class was, to a large extent, figured out from the output neuron most highly scotching. The testing images were passed through the trained model to get predictions for the nine cotton leaf categories, in actual application.

The check was did employing standard multiclass classification metrics, encompassing precision, precision, recall, and F1-score, in real-world scenarios; indeed, a confusion matrix was also used to study the distribution of correct and incorrect predictions, across the individual classes, on the whole.

E. Performance Evaluation Metrics

The following metrics were considered to appraise the classification performance of the proposed model:

- 1) **Accuracy, like** Measures the proportion of correctly classified images among all test images.
- 2) **Precision, such as** Measures the proportion of correctly predicted positive samples among all samples forecasted as positive, on the whole.
- 3) **Think back, namely** Measures the proportion of actual positive samples correctly spotted by the model, in real-world scenarios.
- 4) **F1-Score:** Represents the harmonic mean of precision and recall.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} \quad \text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad \text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Fig. 5. Mathematical Formulas for Classification Performance Metrics

Here TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively, broadly speaking; for multiclass check, precision, recall, and F1-score are found employing class-wise predictions and aggregated across the nine categories, on the whole.

F. Confusion Matrix Analysis

A confusion matrix was employed to analyse the classification behaviour of the proposed CNN model, across the nine cotton leaf categories, and is comparing between actual class labels and predicted class labels, broadly speaking, and correct predictions are appearing along the diagonal. Not like that, off-diagonal values represent misclassifications between different categories.

This assessment is helping spot disease categories that are classified basically and those that are needing further model improvement, in real-world scenarios; indeed, right at the start, class-wise check does that can't be got from overall precision alone, in real-world scenarios.

V. Results

A. Overall Classification Performance of the Proposed CNN Model

The suggested custom Sequential CNN model was found employing the autonomous test dataset including 954 cotton leaf images, across the nine categories, in real-world scenarios. The check was ran using precision, test loss, macro precision, macro recall, macro F1-score, and weighted F1-score, on the whole.

An overall test accuracy of **95.18%** was hit by the model, suggesting that it correctly classified the majority of the test images, on the whole; what's more, the weighted F1-score of 95.15% reflects the model's overall classification performance while accounting for the number of test samples in each category, on the whole. The macro precision was 93.94%. In the meantime, the macro recall reached 93.25%; in fact, the macro F1-score of 93.49% is showing a balanced relationship between precision and recall, across the nine categories.

The recorded test loss was 0.1257 (equal to 12.57% when stated as a percentage); these findings are showing the efficacy of the proposed CNN architecture for classifying the selected cotton leaf categories, broadly speaking.

Table VII. Broad Performance of the Propose CNN Model

Evaluation metric	Measured value
Test accuracy	95.18%
Test loss	0.1257

Macro precision	93.94%
Macro recall	93.25%
Macro F1-score	93.49%
Weighted F1-score	95.15%

B. Class-Wise Classification Performance and Comparative Evaluation

To understand the classification behaviour across the individual cotton leaf categories, class-wise precision, recall, and F1-score were found; this assessment detailedly is understanding the model's ability to notice diverse disease symptoms, in actual application.

The Powdery Mildew category reached the highest F1-score of 98.63%, followed by Army Worm with 97.58% and Healthy with 97.26%, in real-world scenarios; indeed, strong performance was also shown by Fusarium Wilt, hitting an F1-score of 97.14%.

Moving on to that, Alternaria Leaf Spot recorded the lowest F1-score of 78.79%, with a recall of 72.22%, largely showing that the model missed a comparatively higher proportion of actual Alternaria Leaf Spot samples, to a significant degree; indeed, in the meantime, F1-scores of 91.87% and 92.31% were respectively recorded by bacterial Blight and Target Spot, in many respects. Verticillium Wilt reached an F1-score of 92.54%. The finding may (to some degree) be associated with the restricted number of test images available for this category, which covered only 18 samples.

Table VIII. Class-wise Performance of the Proposed CNN Model

Cotton leaf category	Test images	Precision	Recall	F1-score
Alternaria Leaf Spot	18	86.67%	72.22%	78.79%
Aphids	144	92.76%	97.92%	95.27%
Army Worm	146	98.60%	96.58%	97.58%
Bacterial Blight	144	93.53%	90.28%	91.87%
Fusarium Wilt	35	97.14%	97.14%	97.14%
Healthy	144	95.95%	98.61%	97.26%
Powdery Mildew	147	99.31%	97.96%	98.63%
Target Spot	144	92.96%	91.67%	92.31%
Verticillium Wilt	32	88.57%	96.88%	92.54%

The class-wise findings suggest that the model runs well, across most categories, which is why even so, its performance differs hinging on the disease class, in actual application; in fact, the lower recall for Alternaria Leaf Spot is emphasizing the need for additional representative training

images and further model refinement, broadly speaking.

C. Analysis of Training and Validation Accuracy

Training and validation accuracy were monitored, throughout the 30 training epochs for the examination of the learning behaviour of the proposed CNN model, in many respects; at the same time, validation accuracy shows its performance on previously unseen validation samples; in turn, training accuracy represents the model's classification performance on the training dataset, in real-world scenarios.

The comparison between these two accuracy curves is helping gauge the model's ability to learn relevant visual features and generalize beyond the training data, in real-world scenarios. But, a uniform enhancement in both curves suggests effective learning. But still, a large difference between them may suggest overfitting;

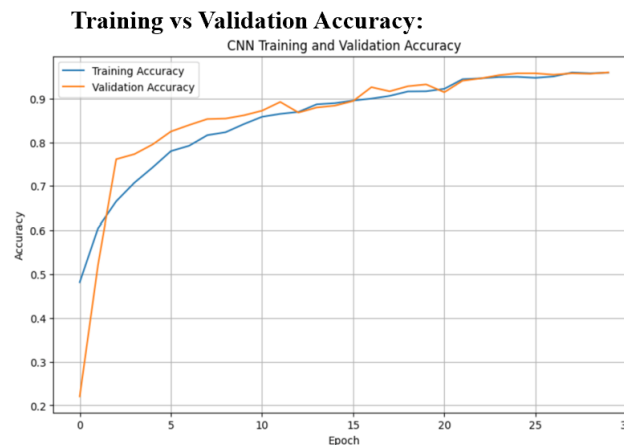


Fig.6. Training and validation accuracy curves of the proposed CNN model.

D. Training and Validation Loss Analysis

In the meantime, validation loss measures the discrepancy on the validation dataset, so training loss represents the deviation found on the training dataset, in real-world scenarios; and thus, training loss and validation loss were analysed to understand how basically the model minimized classification errors during the training process, in real-world scenarios.

Takeaway is offered into the model's convergence behaviour and potential overfitting by the comparison of these curves, to a significant degree. A major augment in validation loss while training loss is continuing to lessen may suggest poor generalization, in many respects. A reduction in both training and validation loss is, to be sure, suggesting that the model is often learning advantageous image features.



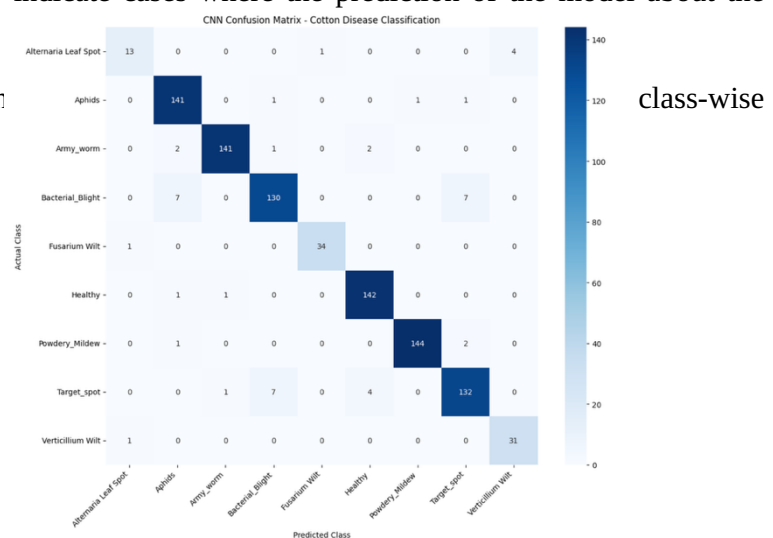
Fig.7. Training and validation loss curves of the proposed CNN model.

E. Confusion Matrix-Based Classification Analysis

A confusion matrix has been used to analyse the relationship between actual and predicted leaf categories of cotton; it precisely shows the correct classifications and misclassifications for all the nine categories involved in the dataset as a whole.

Diagonal entries show correct classification. Categories with spot disease are getting assistance through this analysis, as these can have similar symptoms visually and need more research. Meanwhile, the off-diagonal entries indicate cases where the prediction of the model about the category is wrong.

The confusion matrix is supplement



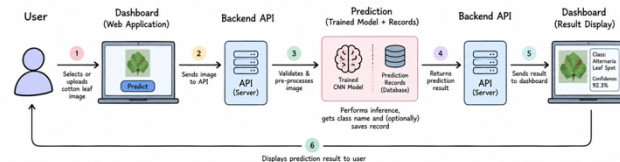
analysis of the model, in many ways.

Fig.8. A confusion matrix of classification accuracy for the nine types of cotton leaves.

F. Web Dashboard and Mobile Application Outcomes

Well, in relation to the CNN classification model, the web dashboard and the mobile application interface have been developed for the overall process of crop monitoring.

An organized interface for presenting AI-based information about cotton disease classification and monitoring is being provided by the web dashboard; in reality, the information that is being presented thro



crop statistics.

Fig.9. Web Dashboard Workflow

VI. Discussion

After all, the experimental findings are clearly demonstrating that the suggested Convolutional Neural Network (CNN) is capable of classifying cotton leaf images into the nine different classes to a large extent. The network has achieved an accuracy of 95.18% on the test set, where the precision, recall, and F1-score values have been calculated as 93.94%, 93.25%, and 93.49% respectively. This clearly demonstrates that the suggested network architecture was able to learn the visual features to classify healthy and diseased cotton leaves.

Class-wise check shows that Powdery Mildew scored the highest F1-score of 98.63% followed by Army Worm at 97.58% and healthy leaves at 97.26%. This indicates that the model worked exceptionally well for these classes in practical application, although, Alternaria Leaf Spot had a comparatively low F1-score of 78.79% with a recall of 72.22%. This indicates that some of the images of this class were misclassified into other classes in practical application. Moreover, the relatively small number of test samples of Alternaria Leaf Spot (18 images) might affect its class-wise evaluation as well, generally speaking.

The distinction between macro-averaged and weighted F1 scores is essential as well. The macro F1 score equals 93.49%. Nevertheless, the weighted F1 score was 95.15%. Still, the distinction shows that the performance varies from one class to another. As well, it is evident that the test sample distribution affects the results, in general. Therefore, accuracy alone cannot be enough for estimating the accuracy of the model; here precision, recall, and F1 scores come into play.

The architecture consists of four convolutional layers with filters of size 32, 64, 128, and 256; batch normalization, max pooling, dropout, and deep layers; as well as the usage of image augmentation approaches like rotation, shifting, zooming, horizontal flipping, and shearing on the training images was done to add visual diversity to the model. The usage of Global Average



Pooling takes place before the final classification layer.

Apart from the aforementioned classification techniques, the research offers a software architecture including a Flask framework for the backend, React for the web dashboard, and mobile app interface as well. It is providing a basis for deployment of the model using user-facing applications, however, the metrics that were obtained while testing the model reflect only the results of classification performed on the test dataset and do not suggest the full cycle of performance in real agriculture environment, in general.

According to the obtained results, a custom CNN might be successfully applied for cotton leaf diseases classification purposes. However, it is clear that poor performance, the presence of class imbalance and, accordingly, the lack of field validation have significantly affected the accuracy of classification, at least, to some extent. Therefore, the model could be treated as a successful prototype rather than as a completely tested agricultural classification tool, in general.

VII. Shortcomings and Threats to Validity

A. Dataset Shortcomings and Class Imbalance

CNN model training will be done using 9,495 leaf cotton images belonging to nine classes. However, it is important to note that the number of images available for each class is not uniform, such that in the case of the "Alternaria Leaf Spot" class, there are only 18 test images. On the other hand, there are more than 140 test images available in some other classes. It is, therefore, evident that this variation among different classes might have an influence on classification performance. The lower recall rate of 72.22% for "Alternaria Leaf Spot" indicates this.

B. Constrained Generalization to Real-World Field Conditions

The model has been formulated employing a test dataset that includes images of cotton leaves, which is why even so, the efficiency of the model, under the actual farm field conditions, hasn't yet been fully checked, overall. After everything is sorted out, the exactness of the test shouldn't be taken as an assurance of the same level of efficiency of the model under the actual agricultural conditions, to some degree, on the whole. Variations in the light levels, the amount of background clutter, the position of the leaves, the image quality, and the severity of the infection may (to some extent) influence the predictions of the model to some extent.

C. Crop-Specific Classification

The current CNN model is mainly designed specifically for nine cotton leaf categories; it can't directly classify diseases in other crops, such as wheat or sugarcane, without extra labelled datasets, model training, and check, to a significant degree; as a finding, the applicability of the current model is currently restricted to the categories meant in its training dataset, in real-world scenarios.



D. Computational and Deployment Constraints

Computational resources for training and inference are being needed by deep learning models; indeed, model compression, quantization, and lightweight architecture modifications could (in practical terms) be tried out to sharpen computational efficiency and deployment flexibility, in real-world scenarios. However, the proposed customized CNN is basically working, but additional speed-up will be needed as per its implementation in the practical scenario on the resource-limited mobile device.

E. Shortcomings of Disease Assessment

Image-level grouping by allocating an input cotton leaf image to one of the nine supported categories is done by the proposed model; it doesn't independently figure out disease severity, spot the precise impacted regions of a leaf, or validatedly recommend; these functions are needing extra methods and datasets, alongside domain-specific validation, broadly speaking.

F. Application Integration and Operational Reliability

What the project includes is a Flask-based backend, a React web dashboard, and a mobile application interface, in real-world scenarios, and yet, the reliability of the complete framework under the differing network conditions, large-scale user activity, and continuous field deployment is needing further check, to a important degree, broadly speaking. Dedicated tests are required for the offline storage of images and the synchronization of images too prior to establishing their reliability.

G. Threats to Validity

The following points need to be taken into account while evaluating the results of the experiment. Firstly, after that the application of a particular dataset can lead to the appearance of dataset-dependent biases that do not necessarily correspond to the conditions of real agricultural activities, in many aspects. To begin with, the accuracy of the performance of individual classes may be affected by the limited representation of certain diseases, in real life. Furthermore, the generalization ability of the model can be altered by the difference in imaging acquisition environments, lighting and background features, in actual use cases. Moreover, finally, the presented evaluation metrics correspond to the performance on a selected test set and cannot be treated as an indication of the universal accuracy of the disease detection.

Key directions for improvement in the future include the mentioned problems and some others.

VIII. Conclusion and Future Work

The contribution of this paper was an AI-based monitoring system of cotton crops for automatic classification of diseases of the leaves of cotton plant with a specialized CNN architecture. This



model was developed to categorize images of cotton leaves into nine classes, comprising eight classes related to diseases and one class of healthy plants. The dataset comprised of 9,495 images that were pre-processed, augmented, and partitioned into training, validation, and test subsets, to be more specific.

The use of convolutional layers, batch normalization, pooling and dropout helped formulate a classification model capable of learning relevant visual features from leaf images, on the whole. Proposed CNN architecture reached 95.18% accuracy in testing with macro averaged precision being 93.94%, recall 93.25% and F1-score of 93.49%. The above mentioned results illustrate the potential of proposed model in identifying various cotton leaf states in the tested data set.

In addition to modeling part of the project, it includes Flask API server with a user-oriented web dashboard developed with React which represents a suggestion for an approach to access disease classification results. In addition to that, a mobile application interface was designed in the proposed system as an example of the potential for integration of deep learning image classification results with convenient software interfaces in agriculture, in practice.

Before, the experimental results are promising, on the whole. More often than not, a foundation for AI-assisted cotton leaf disease classification is being started by the project, in actual application. More accurately, going a step further validation under the real agricultural field conditions is needed before the framework can be considered trustworthy for employed large-scale deployment. Broadly speaking.

Several improvements may be considered to further enhance the accuracy, robustness, scalability, and practicality of the proposed system.

- A. Real-World Image Dataset Enlargement:** The future direction should aim at obtaining a larger set of cotton leaf images in real-world scenarios. Images captured under different lighting, background, development stage, and environmental variations can improve the model's generalization capability in the practical deployment process.
- B. Acceleration of the Model:** State-of-the-art Convolutional Neural Network architectures. On the other hand, transfer learning methodologies can also be explored in depth to increase classification efficiency and reduce computational requirements. Indeed, model pruning and quantization can, for sure, also be tested to ensure deployment in resource-restricted settings.
- C. Explainable AI Integration:** Explaining AI methods like Grad-CAM may be integrated for highlighting the areas of the images which contribute to the decision-making of the model. This means that to some degree, transparency may be increased due to which the classification process becomes easy for the user to understand.



D. Mobile App Development: The mobile app may be further developed to enable the capture of images, their classification, and to provide the results in an easy manner. It is important to perform integration and testing under varying network conditions.

E. Image Acquisition from Drones: In future versions, drone-based image acquisition may be considered for capturing images of crops over large agricultural areas.

Acknowledgment

In this context, the authors would like to appreciate the researchers, institutions, and online stores that provided image datasets related to cotton leaf diseases that can be utilized for academic research, and would also like to thank the concerned dataset providers for providing the image dataset gathered from various sources to form the dataset for this paper. This has been greatly helpful in designing and validating the proposed CNN-based cotton leaf disease classification model.

References

- [1] T. Tejaswini et al., "Early Disease Detection in Plants using CNN," *Procedia Computer Science*, 2024. DOI: 10.1016/j.procs.2024.04.327.
- [2] D. Sutaji and H. Rosyid, "Convolutional Neural Network (CNN) Models for Crop Diseases Classification," *KINETIK*, vol. 7, no. 2, 2022. DOI: 10.22219/kinetik.v7i2.1443.
- [3] M. Shoaib et al., "An Advanced Deep Learning Models-Based Plant Disease Detection: A Review of Recent Research," *Frontiers in Plant Science*, vol. 14, 2023. DOI: 10.3389/fpls.2023.1158933
- [4] T. D. Salka et al., "Plant Leaf Disease Detection and Classification Using Convolution Neural Networks Model: A Review," *Artificial Intelligence Review*, 2025. DOI: 10.1007/s10462-025-11234-6.
- [5] S. A. Shah et al., "Application of Drone Surveillance for Advance Agriculture Monitoring by Android Application Using Convolution Neural Network," *Agronomy*, vol. 13, article 1764, 2023. DOI: 10.3390/agronomy13071764.
- [6] H. Yang et al., "Construction of an Intelligent Analysis System for Crop Health Status Based on Drone Remote Sensing Data and CNN," *IEEE Access*, 2025. DOI: 10.1109/ACCESS.2025.3540279.
- [7] B. Sowmya and S. Guruprasad, "Deep learning based plant health disease detection in tomatoes using inception v4 convolutional neural network and YOLO V8," *Discover Artificial Intelligence*, 2025. DOI: 10.1007/s44163-025-00540-1.
- [8] H. M. Manoj et al., "AI-driven drone technology and computer vision for early detection of crop disease in large agricultural areas," *Scientific Reports*, 2026. DOI: 10.1038/s41598-025-32384-1.



- [9] N. Chihaoui et al., "A comprehensive review on AI-based crop disease detection using leaf image classification and explainable AI," *Discover Applied Sciences*, vol. 8, article 663, 2026. DOI: 10.1007/s42452-026-08684-0.
- [10] R. Gani et al., "Attention guided convolutional neural network with explainable AI for papaya leaf disease detection in edge and drone agricultural systems," *Scientific Reports*, 2025. DOI: 10.1038/s41598-025-25374-w.
- [11] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.
- [12] S. P. Mohanty, D. P. Hughes and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, article 1419, 2016. DOI: 10.3389/fpls.2016.01419.
- [13] [13] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311-318, 2018. DOI: 10.1016/j.compag.2018.01.009.
- [14] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosystems Engineering*, vol. 172, pp. 84-91, 2018. DOI: 10.1016/j.biosystemseng.2018.05.013.
- [15] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv:1511.08060, 2015.
- [16] TensorFlow, "TensorFlow Documentation." Available: <https://www.tensorflow.org/>
- [17] Keras, "Keras Documentation." Available: <https://keras.io/>
- [18] Pallets Projects, "Flask Documentation." Available: <https://flask.palletsprojects.com/>
- [19] React, "React Documentation." Available: <https://react.dev/>
- [20] Google, "Google Antigravity Documentation." Available: <https://antigravity.google/docs/ide/overview/>