Crop Disease Detection Using CNN And Yield Prediction Model with Web Integration

Jasleen Kaur Multani[1], Dhiraj Jadhav[2], Divyanshi Nagrale[3], Abhiram Nair[4], Girish Nalkar[5]
[1,2,3,4,5]*Department of Computer Engineering Vishwakarma Institute of Technology Pune, India*

***Abstract*—In recent years, the combination of artificial intelligence (AI) and computer vision has continued to provide good results in dealing with the problems of agriculture. The authors of the research pointed out that one of the future areas of crop management, early disease detection, has been one of the main problems. A deep learning-based method powered by Convolutional Neural Networks (CNN) has been suggested with MobileNetV2 architecture as a means to successfully diagnose plant diseases. The mobile camera that captures the plants at periodic intervals is the hardware component of the solution that is subject to the CNN model trained on publicly available datasets of crop diseases. The application is hosted in the form of a web-based system whose frontend is developed in Bootstrap while backend integration is done using FastAPI/Flask, MySQL, and Weather APIs. The system is primarily designed for disease detection but it also has a machine learning-based yield prediction module in its architecture that estimates the crop yield based on agricultural inputs and prevailing environmental factors such as humidity, temperature, and precipitation. The results indicate that the system is capable of attaining high accuracy while keeping the computational cost low, thus making it useful for both real-time and edge-based applications. Farmers are provided with instant feedback through a user-friendly interface of the solution which makes it possible to control outbreaks of diseases in advance and promotes organic farming methods.**

***Index Terms*—Precision Agriculture, Crop Disease Detection, Yield Prediction, Deep Learning, Machine Learning, Computer Vision**

I. INTRODUCTION

Agriculture functions as the main economic base which pro- vides essential food supplies and employment opportunities to about 50 percent of people worldwide. The agricultural sector today most commonly encounters difficulties when it needs to detect and control crop diseases at their earliest stages. Crop diseases generate economic losses through yield reductions, which diminish both the quality and quantity of harvests thereby jeopardizing food sustainability. Traditional methods of disease identification depend on agricultural experts who need to perform manual inspections, which creates a process that requires extensive time and physical work and leads to potential mistakes. The agricultural monitoring processes have started to use automated systems because of the recent progress in Artificial Intelligence (AI) and Computer Vision technologies.

The research project aims to develop an AI agricultural dis- ease detection system which uses MobileNetV2 CNN model to detect plant diseases through mobile device leaf image analysis. The system offers a web application which allows users especially farmers to upload infected crop pictures and obtain instant diagnostic results. Bootstrap is used by the system to create its UI, while FastAPI and Flask are used for the backend while MySQL is used for the system data storage.

A dependable yield prediction system was built using Random Forest Regressor to assess potential crop yields based on essential agricultural and environmental and geographical factors which include planted area and fertilizer application and pesticide usage and location. The implementation of a structured data preprocessing pipeline through Column Trans- former enables reliable predictions because it applies One-Hot Encoding to categorical location data and Standard Scaler to numerical data, which helps the model achieve better stability and generalization. The system uses Weather APIs to obtain current weather data which includes temperature and humidity and precipitation, thus improving yield predictions through better environmental condition monitoring [10]. Additionally, integrating external environmental factors with deep learning- based monitoring can further enhance crop disease-related decision-making [8].

II. RELATED WORK

In recent years, the use of artificial intelligence for the detection and classification of crop diseases has been the subject of serious research, with Convolutional Neural Networks (CNNs) being the first choice owing to their performance superiority in image-based pattern recognition tasks.

In an early study, Mohanty *et al.* (2016) reported the use of a deep CNN, which was trained on the PlantVillage data set and achieved an accuracy of 99.3% in the classification of 26 crop and disease sets. It was found that deep learning has the power to outperform traditional feature-based methods. Also, Ferentinos (2018) reports having used transfer learning, which was facilitated by architectures like AlexNet, VGG16 and GoogLeNet for plant disease recognition, which reportedly performed quite well with them achieving over 97% classification accuracy on many crop species. But these models proved to be very compute intensive and not at all suitable for real time mobile or field-based use.

Following studies examined how moving to lightweight CNN architectures reduced model complexity while maintaining accuracy. Too *et al.* (2019) for example, achieved model size and accuracy (over 95%) mobile deployment for leaf disease classification using MobileNetV2. Also, Ramcharan *et al.* (2020) using TensorFlow Lite, brought to light the possibility of mobile deployment of CNN models for cassava disease detection, in a resource constrained environment.

Image classification has many potential applications in agriculture, and recent research has shown the incorporation of data fusion approaches, combining external and internal environmental data with meteorological data, in enhancing the accuracy of disease prediction. Islam *et al.* (2021) showed that the incorporation of weather data i.e., where the model addresses the disease conducive environmental conditions such as temperature and humidity, improved the results of CNN significantly. Such hybrid approaches show the potential of reconciling visual symptoms and contextual data, strengthening the overall understanding of plant health.

III. METHODOLOGY

The suggested system for identifying crop diseases em- ploys a methodical approach consisting of data collection and preprocessing, training a model with MobileNetV2, and real- time deployment by integrating with a web application. The overall process illustrates the workflow from image acquisition through diagnosis to the communicating results to the users.

A. Data Collection

The training and testing datasets consisted of multiple sources, one of which was the PlantVillage dataset containing annotated images of different plant leaves showing healthy and diseased states, and this is an example of publicly available agricultural image repositories.

B. Data Preprocessing

The resizing process was done first and every image was resized to a pixel of $224 \times 224 \times 3$, thus permitting it to be an input for MobileNetV2 architecture. Then the pixel intensity values were normalized to [0, 1], hence making the whole dataset uniform. One of the methods used to alleviate overfitting was data augmentation which included random rotation, flipping, zooming, and brightness adjustment among others.

C. CNN Model Architecture (MobileNetV2)

Lightweight architecture of MobileNetV2 made it the first choice along with its responsive adaptability to mobile and embedded applications.

The model was fine-tuned employing transfer learning and weights that had previously been trained on the ImageNet dataset. A custom classifier for the crop disease categories was used to replace the final layers. The classification head was made up of a global average pooling layer and a fully connected layer with softmax activation for multi-class classification in succession. The Adam optimizer was used for training with a learning rate of 0.0001 and categorical cross- entropy loss.

D. Model Training and Validation

The data were split into three segments: the training set, the validation set, and the test set in the ratio 70:15:15. The model was trained with 50 epochs with a batch size of 32. The techniques of early stopping and dropout were applied to curb the model learning the training data too well. The performance of the model was measured through precision, recall, accuracy, and F1-score metrics. The weights of the model that performed the best were saved and used later for inference.

E. Yield Prediction Module

The yield prediction module utilizes a machine learning method where the Random Forest Regressor was the selected option due to its reliability and effectiveness in working with difficult agricultural data sets. The model inputs comprised important agricultural and regional features such as field size, the usage of fertilizers and pesticides, and geographical area. For the categorical variables, One-Hot Encoding was implemented to create binary representations of the variables, while the numerical features underwent standardization using a Column Transformer so that the entire dataset would have the same preprocessing.

To be able to train the model, a dataset containing more than 10,000 samples was made available, which included both local products and the public agricultural records. One of the evaluation metrics for the model was the coefficient of determination (R^2), and it received a score of 0.984 indicating that the model accounts for a substantial part of the variability in the crop yields.

Through Weather APIs, environmental factors like temper- ature, soil moisture, and rainfall were collected in order to consider climatic differences. The predictions of the yield are shown together with the outputs of disease recognition, giving the farmers a complete tool for the smart management of crops.

IV. SYSTEM DESIGN AND IMPLEMENTATION

This section presents the architectural design, software components, and implementation details that together support the system.

A. System Architecture

As shown in *fig 1*, the proposed architecture system follows a three-tier design, comprising the presentation layer, application layer, and data layer. The presentation layer uses Bootstrap to create web interfaces which allow users to interact with the system through its responsive interfaces that deliver user-friendly navigation. The application layer which developers built using Flask and FastAPI controls vital system operations which include request handling and MobileNetV2 model communication and agricultural input processing which executes the yield prediction model and external API connection. MySQL handles the data layer which stores user data and prediction results that include yield estimation outputs and environmental information obtained from Weather API sources.

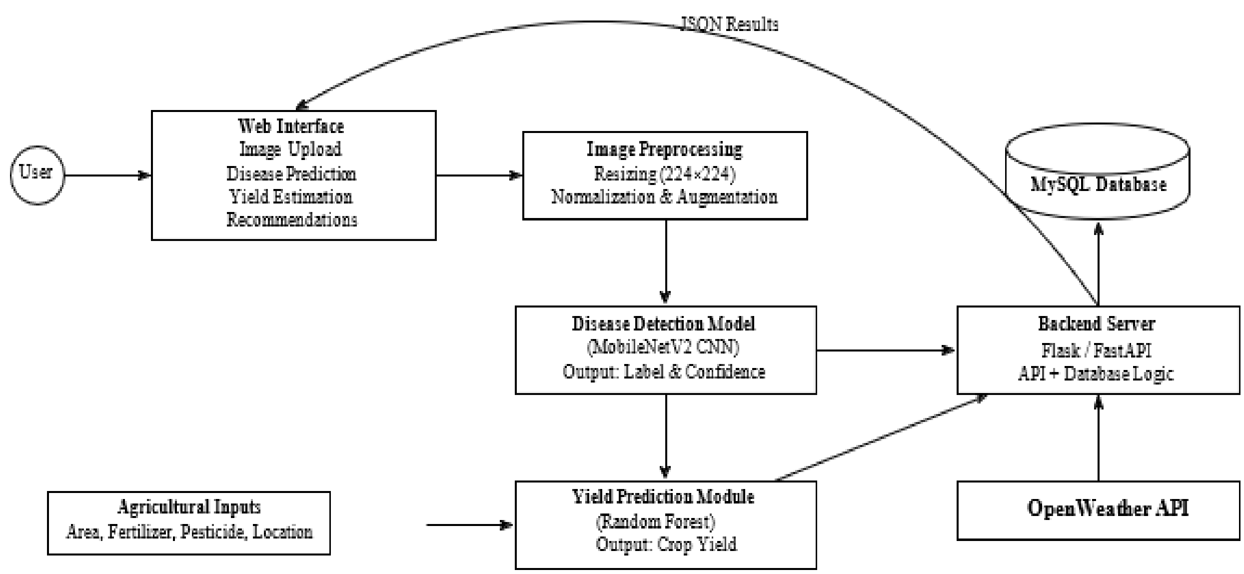

Fig. 1. System Architecture for Crop Disease Detection and Yield Prediction

The system workflow is described as follows:

1) The user uploads the image through the web interface which is then sent to the backend server.
2) The backend system processes the image to send it to the MobileNetV2 model which will perform inference.
3) The model identifies the disease type through its classified results which include prediction certainty.
4) The backend system can obtain weather information and farming data to forecast crop production while it keeps disease forecasts and crop yield predictions in the database.
5) The system shows users their combined results together with disease information and the recommended treatment methods.

B. *Software Components*

1) Frontend:

The system uses HTML5, CSS3, Bootstrap and JavaScript to create a user interface that delivers both visual appeal and responsive design.

2) *Backend:*

The backend uses Flask and FastAPI because both frameworks have lightweight design and asynchronous request processing capabilities. The backend system processes image uploads while performing inference with the trained CNN model and executing agricultural input data analysis to forecast yield and handle database functions and Weather API requests.

3) Database:

The MySQL database stores user data and historical predictions together with environmental variables.

4) APIs and External Services:

The system uses Open- Weather APIs to obtain current temperature and humidity and rainfall information through its integration with external services. The yield prediction model uses these factors which have a direct relationship with disease occurrence to increase contextual accuracy of the disease prediction model.

C. *Database Schema*

The database schema is designed to ensure efficient storage, retrieval, and scalability. Core tables are shown in *table 1.*

TABLE I Database Schema Description

Table Name	Key Attributes	Description
User	user_id, username, location	Stores user information
Image	image_id, user_id, timestamp, image_path	Metadata of uploaded images
Prediction	prediction_id, image_id, disease_confidence_score	Stores CNN prediction output
Weather	weather_id, temperature, humidity, rainfall, timestamp	Stores environmental readings

D. *Implementation Details*

The model inference module was developed using Python programming language together with PyTorch framework. The system used for real-time inference at the backend was built with the trained MobileNetV2 model which had been exported. The system created Flask routes to process HTTP POST image requests which were then sent to the model after preprocessing. The system returned disease predictions through responses which used JSON format. The backend system includes extra routes which manage agricultural input data needed for crop yield prediction and return estimated yield results through JSON format.

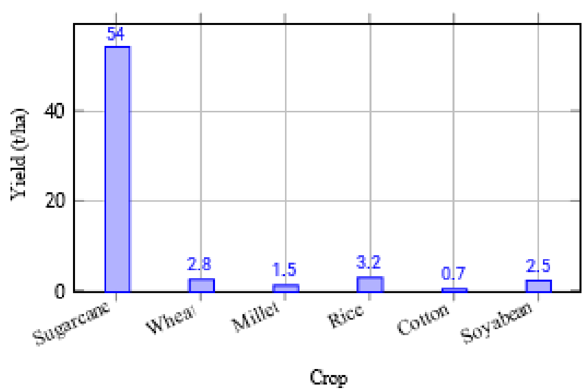

Fig. 2. Average Yield by Crop Type

Users can take pictures using their mobile device cameras or upload video files which support JPG and PNG and JPEG for- mats through the frontend. The system uses RESTful Python requests to integrate weather APIs which show retrieved data on the dashboard in real time.

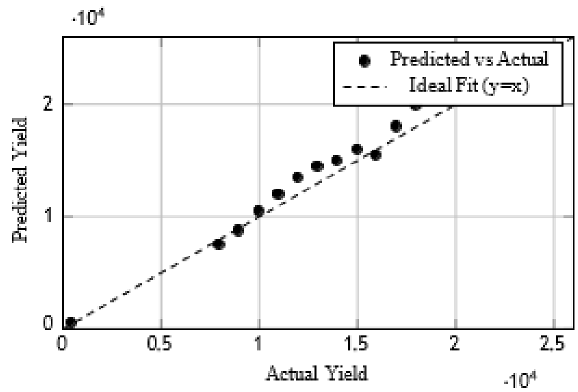

Fig. 3. Actual vs. Predicted Yield Comparison

E. Security and Optimization

The system uses HTTPS encryption to protect the data exchanged between its frontend and backend components. The model inference system reached its optimal performance through the combination of pruning and quantization techniques which decreased computational requirements to enable operation on edge devices and maintained stable performance during low bandwidth scenarios.

F. System Workflow Summary

The system operates as a closed-loop platform which achieves continuous improvement through user interactions. The system stores new images together with prediction data to enable model retraining and refinement which improves dis- ease identification and yield prediction accuracy. The average yield by crop type is shown in *fig 2*. The system provides a powerful and expandable agricultural solution which enables disease detection and yield estimation through its real-time weather integration and mobile accessibility and cloud-based storage features.

V. RESULTS AND DISCUSSION

The proposed crop disease detection system showed positive results through its dual testing of classification accuracy and system usability. The evaluation used a dataset which included various crop species and their associated diseases. The experiments tested both MobileNetV2 CNN model performance and web platform real-time prediction capabilities.

A. Model Performance Evaluation

The researchers used transfer learning to enhance MobileNetV2 model performance which required fewer computation resources while achieving high accuracy results with their testing on PlantVillage dataset and enhanced field images. The training and validation curves showed stable convergence which proved that the model could generalize its performance across different environmental conditions. The researchers assessed model performance using accuracy precision recall and F1-Score metrics.

The system achieved an overall accuracy of 96.8% according to the results. The comparison between the actual and predicted yield can be seen from *fig 3*. Nearly no incorrect predictions were made, and almost all the true values were detected by the system due to almost perfect recall and precision metrics.

The confusion matrix analysis revealed that most misclassifications occurred between diseases which shared visual traits because their shared symptoms made it difficult to achieve precise diagnosis. The MobileNetV2 system showed its capability to manage various visual transformations because 95 percent of its classification results reached correct performance.

B. Comparative Analysis with Other Models

The MobileNetV2 framework evaluated its performance through three ground-tested CNN models which included VGG16 and ResNet50 and InceptionV3. The results in Table II show that MobileNetV2 achieved comparable accuracy to other systems while delivering better performance through its smaller model size and faster web deployment inference times. The yield prediction model established better performance than baseline regression models because it improved both stability and computational efficiency. The comparison between the actual and predicted yield is shown in *fig 3* through which the analysis can be done clearly.

C. Integration and Real-Time Performance

The system used Flask/FastAPI as its backend to test the model in real-time after its final tuning. The system processed each prediction request in 0.6 seconds which included both preprocessing and execution tasks that meet requirements for user-centered applications. The system scalability test required multiple users to use the system at once which demonstrated that the system maintained its performance in all situations.

TABLE II Comparative Analysis of CNN Architectures

Model	Accuracy (%)	Parameters (M)	Inference Time (ms)
VGG16	97.1	138.3	315
ResNet50	97.5	25.6	210
InceptionV3	96.2	23.9	195
MobileNetV2	96.8	3.4	88

The system achieved improved accuracy results because it used Weather API data. The system proved its capacity to monitor environmental changes through its yield prediction system which correctly predicted actual climatic conditions of humidity and rainfall.

D. User Interface Evaluation

The researchers conducted their web interface tests with farmers and agricultural students. The Bootstrap-based frontend received positive feedback because it enabled users to upload leaf images and get predictions within a short time. The platform provided users with disease information which included symptoms and causes and recommended treatments which made the platform more accessible. The pilot test showed that users were 90% satisfied with the system because its design was easy to use and it provided quick results.

E. Discussion

The proposed system demonstrates that lightweight CNN architectures combined with modern web frameworks can effectively bridge the gap between advanced AI models and real-world agricultural use. The system uses weather-based contextual data together with visual disease recognition to create an environmental yield prediction system which delivers complete agricultural information. The system encountered performance problems when it operated under actual field-testing conditions which included extreme variations in light and background sound and camera system performance. The system will achieve better adaptability through continuous retraining which uses multiple datasets and includes new local disease categories. The modular system design enables upcoming implementation of IoT sensors and satellite imagery and advanced agricultural analysis tools which will improve disease detection and yield prediction abilities.

VI. CONCLUSION AND FUTURE SCOPE

The proposed system demonstrates the successful application of deep learning techniques for automated crop disease detection through an accessible web platform. The system achieves high classification accuracy through MobileNetV2 architecture which enables it to function effectively in environments that have limited computational resources. The system uses Bootstrap-based user interfaces together with FastAPI/Flask back-end services and MySQL data management to provide users with a system that enables real-time data processing and uninterrupted data movement. The system now includes Weather API data which improves its contextual understanding by enabling crop yield prediction that depends environmental conditions thus creating an advanced agricultural decision-support system.

The model achieves outstanding results which reach more than 96% accuracy to show that MobileNetV2 functions as an effective solution for agricultural live monitoring systems which connect advanced scientific research with practical field applications.

The research delivers its benefits to smart agriculture re- search through its development of an efficient scalable agricultural program which uses intelligent technology to identify crop diseases and forecast agricultural output.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Computer Engineering, Vishwakarma Institute of Technology, Pune, for providing the necessary infrastructure and academic environment to carry out this research. The authors also thank the faculty members for their valuable guidance, technical insights, and continuous encouragement throughout the development of this work.

The authors acknowledge the use of publicly available datasets such as the PlantVillage dataset and external services including the OpenWeather API, which contributed significantly to the experimentation and evaluation of the proposed system.

REFERENCES

[1] Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, pp. 1097–1105, 2012.
[2] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.
[3] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 1251–1258.
[4] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.
[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 4510–4520.
[6] E. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.
[7] R. Ramcharan, K. Baranowski, P. McCloskey, M. Ahmed, J. Legg, and D. Hughes, "Deep learning for image-based cassava disease detection," Frontiers in Plant Science, vol. 10, p. 272, 2020.
[8] M. M. Islam, M. A. Hasan, and M. M. Rahman, "Integration of environmental data for crop disease detection using deep learning," Journal of Agricultural Informatics, vol. 12, no. 2, pp. 45–55, 2021.
[9] TensorFlow Documentation, "Transfer Learning and Fine-tuning." Available: TensorFlow Documentation
[10] OpenWeatherMap *API*, "Current Weather Data API." Available: OpenWeatherMap API