ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2023.0428

PLANT DISEASE IDEDNTIFICATION USING MACHINE LEARNING AND IMAGE PROCESSING

Sejal Thakkar[1], Chirag Patel[2] and Ved Suthar[3]
[1]Department of Computer Science and Engineering, Parul University, India
[2]Department of Computer Engineering, Charotar University of Science and Technology, India
[3]Department of Computer Science and Engineering, Indus University, India

Abstract

The primary objective of this study is to investigate the detection and diagnosis of plant diseases using Deep Learning and Digital Image Processing. Previous research has primarily focused on single plant disease scenarios using publicly available datasets, often overlooking the image preprocessing phase. In this study, we propose a model that works with 10 different plants and utilizes approximately 50,000 images for training and testing. We classified 36 distinct classes into healthy or infected types based on disease labels. To enhance the accuracy of disease detection, we recommend employing image processing techniques and considering multiple plant scenarios. We utilized a dual-layer Convolutional Neural Network (CNN) for the publicly available dataset and supplemented it with real-time images captured from various farms in Village Rancharda Near Ahmedabad, Gujarat, India (PIN: 38255). Our research introduces several novel elements in the preprocessing steps. We employed HSV segmentation, flood fills segmentation, and a proposed deep learning model for image segmentation. Additionally, we standardized the resolution of all images to ensure uniformity. These preprocessing techniques refine the image data required for accurate classification and enhance the visibility of diseased portions. For image processing, we employed a sliding window mean average deviation technique and stacked the processed images onto the original image, resulting in six-channel images. Our proposed model demonstrates improved performance on the validation data, achieving an accuracy of up to 97.95%. Furthermore, we transformed this model into a TFLite model, which can be easily integrated into client applications without the need for a server. In our case, we implemented the model on an Android platform. These findings indicate the potential of our proposed model to significantly enhance the detection and diagnosis of plant diseases in real-world scenarios.

Keywords:

Convolutional Neural Network, Image Segmentation, Dual Layered, Sliding Window

1. INTRODUCTION

Plant diseases pose a significant threat to global food security, yet their prompt identification remains a challenge in many regions due to limited resources. However, the combination of widespread smartphone usage and advancements in data science has opened up new possibilities. Modern technologies have enabled us to meet the food demands of over 8 billion people, but food security remains vulnerable to various factors, including climate change and plant diseases. These diseases not only endanger global food security but also have devastating consequences for smallholder farmers who rely on healthy crops for their livelihoods. In developing countries, where more than 80 percent of agricultural production comes from smallholder farmers, reports of yield losses exceeding 50% due to pests and diseases are common.

The agricultural sector has been severely affected by recent global events [1], resulting in a decline in the global economy and particularly impacting developing countries, where agriculture is often the primary source of income. For instance, Ukraine, one of the largest wheat exporters, experienced an unexpected yield decline [2], highlighting the urgent need for effective strategies to prevent crop loss. Moreover, the rising prices of fertilizers in developing countries are anticipated to have a negative impact on future crop yields [3]. The ongoing COVID-19 pandemic has further exacerbated the situation, aggravating the existing challenges faced by the agricultural sector.

To tackle these issues, it is crucial to develop effective solutions that help farmers achieve stable yields and mitigate further complications. This study proposes a solution that focuses on early detection of plant diseases with minimal technical requirements. To accomplish this, we conducted an extensive review of existing research, exploring image processing techniques and deep learning algorithms that offer accurate disease detection. Our study utilized a dataset comprising over 50,000 leaf images from 10 different plant species, sourced from publicly available datasets, as well as real-time images obtained from the Castrol Plants in Village Rancharda near Ahmedabad, Gujarat, India. We personally interviewed the farmers to gain insights into the challenges they face on their farms, and we uploaded the interview video on YouTube.

Our study aimed to accurately predict diseases and provide farmers with a transparent detection process that is not influenced by unrelated artifacts. One of the key novelties of our work is that farmers can utilize the app with or without an internet connection, enhancing its accessibility and usability.

1.1 PLANT DISEASE

Plant diseases encompass a broad range of conditions that impede a plant's optimal growth and development [4]. In our research on plant disease detection, we are utilizing the Plant Village Dataset as a foundation. Additionally, we are capturing real-time images of Castor plants in the field to assess our proposed model, particularly in borderline cases. Our dataset comprises various plant diseases and their corresponding plants, including:

- Tomato: Bacterial spot, Early blight, Late blight, Mold, Septoria leaf spot, Spider mites, Two-spotted spider mites, Target Spot, Mosaic virus, Yellow Curl Virus, and Healthy.
- Potato: Early blight, Late blight, and Healthy.
- Apple: Apple scab, Black rot, Cedar apple rust, and Healthy.
- Grape: Black rot, Esca, Leaf blight, and Healthy.
- Corn: Gray leaf spot, Common rust, Northern leaf blight, and Healthy.

- Cherry: Powdery Mildew and Healthy.
- Pepper Bell: Spots and Healthy.
- Strawberry: Leaf Scorch and Healthy.
- Rice: Unhealthy and Healthy.
- Castor: Spots and Healthy.

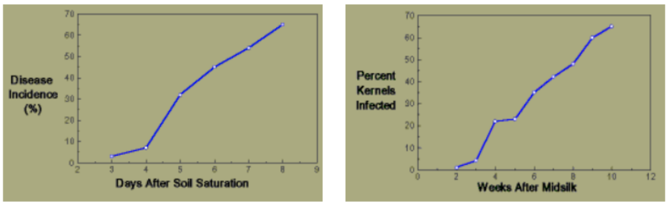
Fig.1. A sample of leaf image from the dataset

Fig.2. A sample image from the real time dataset

By encompassing this diverse range of plants and diseases within our dataset, we aim to develop a comprehensive and robust model for accurate plant disease detection and diagnosis.

1.2 EARLY DETECTION

Plant diseases typically start at a minimal level, initially impacting a small number of plants and tissues, but their significance grows as their severity escalates over time [5]. Through our literature review, we aimed to gain a comprehensive understanding of plant disease detection, including the stages at which diseases can be identified and their progression over time. Achieving an optimal balance in the detection algorithm is crucial, as excessively sensitive detection of inconspicuous or small features can result in reduced accuracy and require extensive training time.



Phytophthora blight of pepper seedlings Fusarium kernel rot of maize

Fig.3. Growth of disease in plant with time

Research conducted by [5] demonstrates that plant diseases initiate from infections or spread within a specific portion of a plant or tissue, eventually extending to the entire plant and potentially affecting neighboring plants. Hence, early detection plays a critical role in safeguarding crop yields during harvest. By identifying and addressing the initial point of disease, potential losses that may have impacted the entire field can be averted. However, manually inspecting every area of a field is a time-consuming and challenging task, especially when identifying minute patterns is arduous. The utilization of an AI model can assist in distinguishing between infected and healthy plants, while identifying the specific disease name can provide valuable information to farmers for targeted remedies and the application of appropriate pesticides to eradicate the problem.

2. LITERATURE REVIEW

In this stage, the analysis of necessary data in the required format is conducted, while simultaneously researching the requirements. India, being an agriculturally based country, cultivates various types of crops. As crop production advances in India, it is crucial to have sustainable methods for maintaining crop health and achieving better crop growth. Traditionally, several techniques such as exclusion, eradication, protection, and immunization [5] have been utilized, but prevention has received comparatively less attention. The implementation of preventive measures on a large scale has been challenging due to obstacles like high labor requirements, specialized expertise, and errors in disease identification. However, leveraging AI and image-based approaches can help overcome these obstacles. Deep learning can be employed to train models that identify diseases from leaf images, and significant progress has been made in this field through various research endeavors.

Dubey and Shanmugasundaram [7] have developed a functional model using the CNN LeNet-5 architecture, achieving 99% accuracy on a validation dataset using a server-based web application. Even without employing image processing techniques, such a model can achieve high accuracy. By incorporating image processing methods, the robustness of the model can be further enhanced.

Another research conducted by Durmuş et al. [8] involves the utilization of AlexNet and SqueezeNet models on tomato plant leaves. Their objective is to develop a robot capable of autonomously capturing images in the field. However, due to the financial constraints faced by farmers in India, deploying a robot for each farmer is nearly impossible.

3. APPROACH

Based on our analysis and review of relevant research papers, several important considerations arise when designing a plant disease identification system that caters to the needs of farmers in developing countries, particularly in India. These considerations are also applicable to other developing nations.

Firstly, given the financial constraints and limited resources faced by farmers in developing countries, it is imperative that any system designed for them is cost-effective and supports the use of low-quality equipment. The algorithms employed should possess low computational complexity and be efficient enough to handle low-quality images captured by inexpensive image sensors or cameras.

Secondly, farmers residing in remote areas often lack internet access, posing a challenge for them to utilize online plant disease identification systems. Consequently, the availability of a compact offline plant disease identification system becomes crucial, enabling farmers in remote regions to detect plant diseases on-site without relying on an internet connection.

Thirdly, the quality of images captured by farmers may be suboptimal due to various factors, such as poor lighting conditions or unsuitable backgrounds. Therefore, effective segmentation and image correction techniques become essential to address these issues and ensure accurate detection of plant diseases.

Once the requirements are fulfilled, the input image can be fed into the detection model for training, enabling it to identify the presence of diseases in plants. It is also vital to experiment with different parameters to achieve a reasonable level of accuracy while maintaining sufficient efficiency, thus enabling the system to perform the task locally on an Android device.

3.1 METHODOLOGY

Data Gathering: The data gathering process involved two main aspects: data collection for the AI model and understanding the requirements for the application.

AI Model Data: Multiple sources were utilized to gather the data required for training the AI model. The primary dataset used was the Plant Village dataset. Additionally, data from rice and castor crops was collected locally from farm fields. This combination of datasets helped in creating a diverse and comprehensive training dataset. Furthermore, feedback from farmers was obtained to understand their specific needs and challenges.

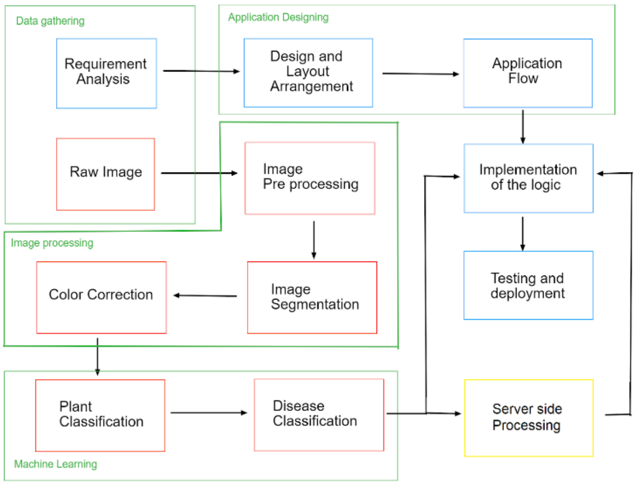


Fig.4. Flow diagram of the approach in order to solve the stated problem

Application Design: The application design process focused on meeting the requirements identified from the farmers' feedback. The key considerations were:

Offline Functionality: The application was designed to be functional even without an internet connection. This ensures that farmers can use the application in remote areas or locations with limited connectivity.

User-Friendly Interface: Farmers often have limited knowledge about modern smartphones and technology. Hence, the application's user interface (UI) and flow were kept simple and easy to understand. The goal was to make it accessible to users with varying levels of technological proficiency.

Image Processing and Disease Identification: The application's workflow was designed to streamline the process for farmers. Upon launching the app, users are directed to a screen where they can either select an image from their device or take a picture. The image is then processed using algorithms for segmentation and correction. The processed image is fed into the plant identification model, which determines the appropriate disease model to use. The output of the disease model is displayed directly on the same screen, providing a clutter-free and hassle-free user experience.

By considering the specific requirements and challenges faced by farmers, the application was designed to be intuitive, accessible, and provide valuable assistance in identifying plant diseases.

3.2 IMAGE PROCESSING

In the context of real-time image processing for plant disease identification, the presence of varying backgrounds in images poses a challenge during model training. To address this issue, image segmentation techniques can be employed to separate the foreground (leaf) from the background. This allows the extraction and utilization of the leaf portion, leading to improved performance and reduced model complexity. Segmentation involves generating a mask that isolates the foreground from the background, specifically focusing on the leaf section. Multiplying this mask with the image facilitates the extraction of the leaf part, enabling more efficient analysis and interpretation.

In real-time image processing, varying backgrounds can hinder model training. To overcome this, a mask is generated through techniques like flood fill algorithm or generative models to isolate the leaf portion of the image. However, the flood fill algorithm may produce jagged edges in the mask, which can be addressed through border smoothening algorithms. Dilation and erosion algorithms can also be employed to fill any holes or remove small closures in the mask, resulting in an accurate mask for extracting the leaf portion.

Image correction is crucial for effective disease detection, as it enhances image features and balances variations in image quality. Histogram equalization is a commonly used technique for balancing image contrast and brightness. By plotting the frequency of color shades in an image, histogram equalization improves contrast and brightness, making images more robust to physical biases that may affect image capture. This approach increases the color difference between healthy leaves and diseased spots, improving the accuracy of disease detection systems.

An alternative approach to image correction is the use of the Sliding Window Mean Absolute Deviation (SW-MAD). Instead of balancing contrast and brightness, SW-MAD calculates the absolute difference between each pixel and its surrounding pixels, highlighting color differences in the image. This method can eliminate the need for image correction and allows for easy visibility of disease spots and leaf edges. SW-MAD also removes regional bias present in the leaf image by processing it with respect to the surrounding pixels.

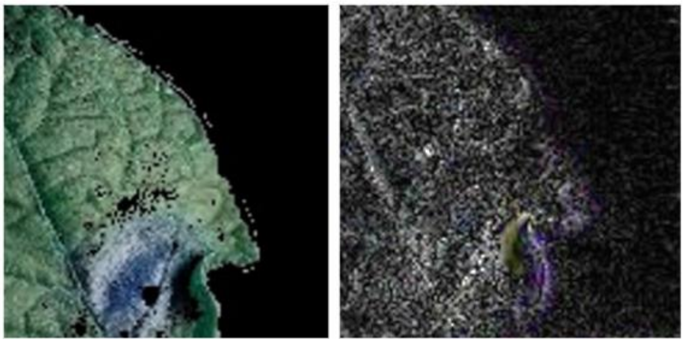


Fig.5. SWMAD Processing

Both image segmentation and correction techniques play crucial roles in enhancing the accuracy and efficiency of plant disease detection systems, ensuring robust performance across a wide range of images with varying backgrounds and qualities.

3.3 DEEP LEARNING

3.3.1 Convolutional Neural Network (CNN):

A Convolutional Neural Network (CNN) is a type of artificial neural network commonly used for image applications. It is designed to process and analyze 2D and 3D arrays, making it well-suited for tasks such as image classification and object detection. CNNs are composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers. These layers work together to extract features from input images and make predictions based on learned patterns.

3.3.2 Dual Layered CNN Model

In the context of plant disease identification, a dual layered CNN model is proposed. Instead of using a single model to classify plants and their associated diseases, a multi-model approach is employed. The plant is first classified using a separate model, and then for each plant, a specific disease model is used to classify the diseases. This approach allows for training on large datasets and facilitates efficient addition of new plants or diseases without requiring retraining of the entire model. The output classes are divided, reducing the model’s error and improving performance. If a new disease needs to be added, only the disease model for that specific plant needs to be retrained, while the plant classification model remains unchanged. This approach provides flexibility and performance gains in handling large and evolving datasets.

3.3.3 Combined Channels for Inputs

During testing, it was discovered that the accuracy of the model trained using the Sliding Window Mean Absolute Deviation (SWMAD) method was behaving inconsistently. To address this issue, a solution was found by combining the processed image with the raw/unprocessed image, creating a 6-channel image. This combined image, containing both the extracted features and the raw data, was used as input for training the model. This approach not only improved accuracy but also allowed for reducing the number of nodes in the hidden layers while maintaining performance.

3.3.4 Model Design

For implementing the CNN model, the Keras library was utilized. The Sequential module from Keras was imported to create the model. The model architecture consisted of a Convolutional 2D layer followed by a Max Pooling layer. The output of these layers was then flattened, and a Dense layer was added for further classification improvement. Different optimizers were experimented with, and the loss was calculated using categorical cross-entropy. The model was trained using various configurations, including training on CPU and GPU, using single batches or multiple random batches, and with or without workers to explore different training setups and evaluate their impact on model performance.

4. RESULTS

The study aimed to improve the accuracy of the model for plant and disease classification. Several experiments were conducted, and the results are summarized below:

Optimizers: Different optimizers were tested to identify the best-suited optimizer for the model. The experiments were performed on Google Colab with GPU runtime. The table shows the training accuracy and validation accuracy achieved with each optimizer:

Table.1. Effects of optimizer

Optimizer	m/ep	Training Accuracy	Validation Accuracy
Adam	4.883	0.9721	0.9292
Adagrad	5.033	0.9608	0.8234
SGD	5.15	0.9752	0.9079
RMSProp	5.167	0.9797	0.8775
FTRL	5.167	0.4318	0.4317
Nadam	5	0.9797	0.944

Based on these results, the Nadam optimizer was selected due to its highest test accuracy. Different layer configurations were tested on Google Colab using GPU runtime. The table presents the minutes per epoch, training accuracy, and validation accuracy achieved with each layer configuration:

Table.2. Layer configuration

Layer Name	Layer Config
Small	CNN 64,32,32,64,32; Flatten; DNN 512,5
Medium	CNN 64,64,128,64; Flatten; DNN 1024,5
Large	CNN 64,256,32; Flatten; DNN 1024,5

Table.3. Performance of the model with different layer configurations

Layer Type	Minutes/ Epoch	Training Accuracy	Valid Accuracy
Medium	5	0.9797	0.944
Large	5	0.9788	0.9744
Small	4.9	0.9723	0.9376

The medium layer configuration exhibited the best balance between accuracy and training time. Due to limited GPU runtime availability on Google Colab, the experiments were switched to a local machine with an Intel i5-7300HQ CPU and Nvidia GTX 1050ti GPU. The table below shows the training accuracy, validation accuracy, and minutes per epoch achieved with different layer configurations:

Table.4. Performance of the model with different layer configurations on the local machine

Layer Configuration	Training Accuracy	Valid Accuracy	Minutes/ epoch
Medium; 1 Worker	0.9822	0.9531	5.33
Medium; 6 Workers	0.9811	0.9501	2.00
Small; 6 Workers	0.9794	0.9388	1.75
Large; 6 Workers	0.9823	0.9515	2.08

The medium layer configuration with 6 workers yielded the best results in terms of both accuracy and training time. Two image processing techniques, HSV segmentation and SWMAD, were compared for their impact on model performance. The table shows the training accuracy and validation accuracy achieved with each technique:

Table.5. Performance of the model with different image processing techniques

Image Processing Method	Training Accuracy	Validation Accuracy
HSV Segmentation	0.9823	0.9515
SWMAD	0.9792	0.9738

The SWMAD technique resulted in slightly higher validation accuracy. Based on the experiments, the final model configuration consisted of a CNN with 16, 32, 64, and 64 nodes, followed by a Flatten layer, and a Dense layer with 64 nodes representing 3 output classes. The model was trained using a combination of raw and processed images generated by SWMAD. The average training time per step was 53ms (1.3s per epoch with 25 steps). Nine random batches were used for training, achieving an average maximum accuracy of 0.9955 for plant classification and 0.983

5. CONCLUSION

After conducting various runs and researching different methods, the following conclusions were drawn. The moving window SWMAD technique, applied to the image along with the combined raw image as input, yielded the best results. This combination provided improved accuracy for plant and disease classification. The recommended model structure consists of 4 CNN layers and 2 Dense layers. This configuration proved to be effective in achieving high accuracy while keeping the training time relatively low. The use of separate models for plant classification and disease classification was found to be beneficial. This approach introduced modularity to the models, allowing for easier maintenance and scalability. By utilizing the moving window SWMAD technique, along with the suggested model structure and separate models for classification tasks, the overall performance and accuracy of the system can be enhanced.

REFERENCES

[1] Pierre-Olivier Gourinchas, "War Dims Global Economic Outlook as Inflation Accelerates", Available at: https://blogs.imf.org/2022/04/19/war-dims-global-economic-outlook-as-inflation-accelerates/, Accessed at 2022.

[2] OEC Wheat in Ukraine, Available at: https://oec.world/en/profile/bilateral-product/wheat/reporter/ukr/, Accessed at 2020.

[3] World Bank Group, "Commodity Markets Outlook: The Impact of the War in Ukraine on Commodity Markets", Available at https://www.worldbank.org/en/news/press-release/2022/04/26/food-and-energy-price-shocks-from-ukraine-war, Accessed at 2022.

[4] Plant Disease: Pathogens and Cycles, Available at https://cropwatch.unl.edu/soybean-management/plant-disease, Accessed at 2022.

[5] Plant Disease Epidemiology: Temporal Aspects, Available at https://www.apsnet.org/edcenter/disimpactmngmnt/topc/EpidemiologyTemporal/Pages/Disease%20Progress.aspx, Accessed at 2021.

[6] O.C. Maloy, "Plant Disease Management", *The Plant Health Instructor*, Vol. 25, pp. 1-13, 2005.

[7] Ankit Dubey and M. Shanmugasundaram, "Agricultural Plant Disease Detection and Identification", *International Journal of Electrical Engineering and Technology*, Vol. 11, No. 3, pp. 354-363, 2020.

[8] H. Durmuş and M. Kirci, "Disease Detection on the Leaves of the Tomato Plants by using Deep Learning", *Proceedings of International Conference on Agro-Geoinformatics*, pp. 1-5, 2016.