International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

AI AND ANDROID APP BASED TOMATO PLANT DISEASE PREDICTION SYSTEM

PUNITH B[1], SIDDESH K T[2], KOTRU SWAMY S M[3]

[1] STUDENT DEPARTMENT OF MCA, BIET, DAVANGERE
[2]ASSISTANT PROFESSOR, DEPARTMENT OF MCA BIET DAVANGERE
[3]ASSISTANT PROFESSOR, DEPARTMENT OF MCA BIET DAVANGERE

ABSTRACT

Plant diseases are one of the grand challenges that face the agriculture sector worldwide. In the United States, crop diseases cause losses of one-third of crop production annually. Despite the importance, crop disease diagnosis is challenging for limited-resources farmers if performed through optical observation of plant leaves' symptoms. Therefore, there is an urgent need for markedly improved detection, monitoring, and prediction of crop diseases to reduce crop agriculture losses. Computer vision empowered with Machine Learning (ML) has tremendous promise for improving crop monitoring at scale in this context. This paper presents an ML-powered mobile-based system to automate the plant leaf disease diagnosis process. The developed system uses Convolutional Neural networks (CNN) as an underlying deep learning engine for classifying 38 disease categories. We collected an imagery dataset containing 96,206 images of plant leaves of healthy and infected plants for training, validating, and testing the CNN model. The user interface is developed as an Android mobile app, allowing farmers to capture a photo of the infected plant leaves. It then displays the disease category along with the confidence percentage. It is expected that this system would create a better opportunity for farmers to keep their crops healthy and eliminate the use of wrong fertilizers that could stress the plants. Finally, we evaluated our system using various performance metrics such as classification accuracy and processing time. We found that our model achieves an overall classification accuracy of 94% in recognizing the most common 38 disease classes in 14 crop species.

Keywords: plant leaf diseases; agriculture; mobile app; convolutional neural networks (CNN); deep learning

1. INTRODUCTION

Agriculture is the backbone of many economies, particularly in developing countries where it provides livelihood and food security for a large portion of the population. Tomato plants, a staple crop, are particularly vulnerable to a range of diseases that can significantly impact yield and quality. Traditional methods of disease detection, which rely on expert knowledge and manual inspection, are often slow, subjective, and inaccessible to many smallholder farmers. In recent years, advancements in artificial intelligence (AI) and mobile technology have opened new avenues for enhancing agricultural practices. Deep learning, a subset of AI, has shown remarkable success in image recognition tasks and can be applied to identify plant diseases from images with high accuracy. Coupled with the widespread use of smartphones, this technology can be harnessed to develop powerful diagnostic tools that are both affordable and easy to use.

This paper presents an innovative system that integrates AI with an Android application to detect diseases in tomato plants. The core of the system is a convolutional neural network (CNN) trained on an extensive dataset of images depicting healthy and diseased tomato plants. The trained model is capable of identifying and classifying several common tomato plant diseases, including early blight, late blight, and leaf mold, with high accuracy. The Android application is designed to be intuitive, allowing farmers to simply capture images of their tomato plants using their smartphone cameras. The app processes these images and provides immediate feedback on the presence and type of disease, along with recommended actions. This real-time diagnostic capability empowers farmers to take prompt and informed measures to protect their crops, thereby enhancing productivity and reducing losses.

By leveraging AI and mobile technology, this system aims to democratize access to advanced agricultural diagnostics, particularly benefiting smallholder farmers who may lack access to traditional expert resources. The integration of AI into everyday farming practices represents a significant step forward in modernizing agriculture, improving crop management, and ensuring food security.

Plant diseases , pest infestation , weed pressure , and nutrient deficiencies are some of the grand challenges for any agricultural producer, at any location and for whatever commodities or size of the operation is dealing daily. It is crucial that farmers would know the existence of such challenges in their operations on a timely basis. Nevertheless, it would be tremendously helpful to agricultural producers to have access to readily available technology to instruct them on how to deal with each of these threats for agricultural production to enhance crop production and operation profitability.

For instance, in the United States, plant disease causes losses of between 20 and 40 percent of the agricultural crop production annually . Therefore, farmers must promptly diagnose the different types of plant diseases to stop their spread within their agricultural fields. Traditionally, underserved farmers try to diagnose plant diseases through optical observation of plant leaves' symptoms, which incorporates a significantly high degree of complexity . Any misdiagnosis of crop decreases will lead to the use of the wrong fertilizers that could stress the plants and lead to nutrient deficiencies in the agricultural field.

2. RELATED WORK

Sinha and Shekhawat (2020) provide a comprehensive review of image processing techniques for detecting plant diseases, covering methods such as thresholding, edge detection, segmentation, and feature extraction. They discuss the application of these techniques in real-time disease monitoring systems and emphasize the importance of machine learning in enhancing the accuracy and efficiency of these systems. The review highlights the potential for integrating these image processing approaches into mobile applications, making them accessible for on-field disease detection and management, which is particularly beneficial for crops like tomatoes.[1]

Ai et al. (2020) develop a deep learning model for recognizing crop diseases and insect pests, specifically in challenging environments. Their model leverages convolutional neural networks (CNNs) to maintain high accuracy despite variations in lighting, occlusions, and plant growth stages. This robustness is critical for real-world agricultural applications. Their research underscores the importance of creating reliable and resilient disease detection systems that can be implemented in mobile apps, providing farmers with effective tools for managing tomato plant health in diverse conditions.[2]

Zeng et al. (2020) utilize Generative Adversarial Networks (GANs) for data augmentation to improve the performance of deep learning models in detecting the severity of citrus diseases. By generating additional training data, their approach enhances model generalization and accuracy. This technique addresses the challenge of limited labeled data in agricultural datasets. Applying this method to tomato plant disease detection can significantly boost the effectiveness of AI- based mobile applications, helping farmers identify and manage diseases more accurately.[3]

Thomas et al. (2018) highlight the advantages of hyperspectral imaging for plant disease detection. Hyperspectral imaging captures detailed spectral information across a wide range of wavelengths, enabling early and accurate identification of disease symptoms. Although traditionally complex and costly, their study suggests potential pathways for integrating hyperspectral technology into more accessible systems. This can inform the development of mobile apps that offer advanced diagnostic capabilities for tomato plant diseases, improving early detection and crop protection.[4]

The Crop Protection Network provides a comprehensive online resource for managing plant diseases and pests. The website offers up- to-date information, tools, and strategies for disease prevention and management, catering to various crops including tomatoes. This platform can be a valuable reference for developing an AI and Android app-based tomato plant disease detection system, offering reliable data and resources that can be integrated into the app to enhance its functionality and user support.[5]

Jiang et al. (2019) introduce a real-time apple leaf disease detection system based on improved convolutional neural networks (CNNs). Their enhancements focus on increasing detection speed and accuracy, making the system practical for real-time applications. The feasibility of deploying such models on mobile devices is demonstrated, providing immediate diagnostic feedback to users. This approach is highly relevant for developing an AI and Android app for tomato plant disease detection, ensuring quick and precise disease identification to enable timely intervention.[6]

Chen et al. (2020) present Ricetalk, an IoT and AI-based system for detecting rice blast disease. Their system integrates IoT sensors with AI algorithms to monitor and diagnose disease conditions in real-time. The application of IoT technologies enhances data collection and environmental monitoring, providing comprehensive insights into disease dynamics. This approach can be adapted for tomato plants, integrating IoT and AI in an Android app to offer advanced, real-time disease monitoring and management.[7]

He et al. (2017) introduce Mask R-CNN, a stateof-the-art deep learning framework for instance segmentation. Mask R-CNN extends Faster RCNN by adding a branch for predicting segmentation masks on each Region of Interest (RoI), enabling precise object detection and segmentation. This framework can be leveraged in developing an AI-based tomato disease detection app, offering detailed and accurate segmentation of diseased areas on
leaves, which enhances the diagnostic capabilities of the app.[8]

Ahmed et al. (2020) discuss a distributed system for smart irrigation using IoT technology. Their system optimizes water usage by monitoring soil moisture levels and environmental conditions in real-time, making irrigation more efficient and sustainable. The integration of IoT with AI for data analysis can be extended to plant disease detection, where similar real-time monitoring and smart decision-making capabilities can be applied to develop an effective tomato plant disease detection system.[9]

Sun et al. (2020) focus on detecting Northern maize leaf blight in complex field environments using deep learning. Their approach addresses challenges such as varying light conditions, background noise, and occlusions by employing advanced deep learning techniques. The robustness and adaptability of their model to real-world conditions are crucial for developing a reliable disease detection system. Applying these principles to tomato plants can enhance the accuracy and reliability of an AI and Android app-based detection system, ensuring it performs well in diverse agricultural environments.[10]

3. METHODOLOGY

The methodology for developing the AI and Android app-based tomato plant disease detection system involves several key steps, encompassing data collection, model training, Android app development, and system evaluation. Here is a detailed outline of each phase:

1. Data Collection and Preprocessing:

- Dataset Acquisition: Gather a diverse dataset of tomato plant images, including both healthy plants and those affected by common diseases such as early blight, late blight, leaf mold, and others.
- Annotation: Annotate the images to label the presence of diseases, specifying the type of disease present in each image. This annotated dataset serves as the training data for the deep learning model.

2. Model Training:

- Deep Learning Model Selection: Choose a suitable deep learning architecture for the task of tomato plant disease detection. Convolutional Neural Networks (CNNs) are commonly used due to their effectiveness in image classification tasks.
- Data Augmentation: Augment the training dataset by applying transformations such as rotation, flipping, and scaling to increase its diversity and robustness.
- Model Training: Train the selected CNN architecture using the annotated dataset. Utilize transfer learning by fine-tuning a pre- trained model (e.g., ResNet, MobileNet) on the tomato plant disease dataset to expedite training and improve performance.

3. Android App Development:

- User Interface Design: Design an intuitive and user-friendly interface for the Android application, allowing users to capture images of tomato plants using their smartphone cameras.
- Integration with Deep Learning Model: Integrate the trained deep learning model into the Android app to enable real-time inference of tomato plant diseases directly on the device.
- Feedback Mechanism: Implement a feedback mechanism to provide users with immediate diagnostic results, indicating the presence of diseases and potentially suggesting treatment or mitigation strategies.

3.1 DATASET USED

Although standard object detection datasets exhibit volume and variety of examples, they are not suitable for plant disease detection as they annotate a set of object categories not include plant diseases. Therefore, we collected more than labeled 96k images of healthy and infected plant leaves for training the CNN model from different sources such as Kaggle, Plant Village and Google Web Scraper . Many images in our dataset are in their natural environments because object detection is highly dependent on contextual information. To increase the training accuracy and minimize training loss of the CNN model, we applied a series of image preprocessing transformations to the training dataset. Particularly, we altered the contrast of image colors, added Gaussian noise, and used image desaturation, which makes pixel colors more muted by adding more black and white colors. The primary purpose of these transformations is to weaken the influence of the background factor during the training process. This had a better effect on learning the 38 disease classes more effectively and increased our CNN model's stability. We had to normalize the range of pixel intensity values of leaf images in the dataset before training the CNN model. This step was necessary because all dimensions of feature vectors extracted from input images should be in the same intensity range.

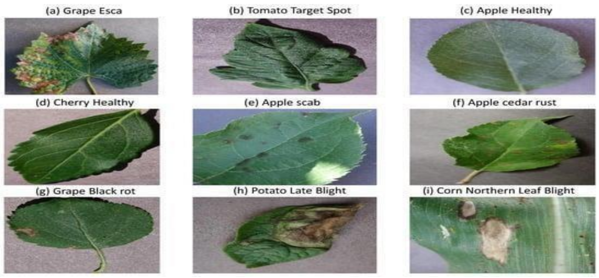

Figure 3.1. Samples from our Imagery Dataset that Show Different Types of Healthy and Diseased Plant Leaves.

3.2 DATA PRE PROCESSING

Data preprocessing is a crucial step in developing an AI and Android app-based tomato plant disease detection system, ensuring that the training data is clean, accurate, and suitable for analysis. Initially, image acquisition involves gathering a diverse set of tomato plant images from various sources such as field data, online databases, and research publications. Each image must be accurately labeled with the corresponding disease or health condition, often requiring manual annotation by experts or the use of pre- labeled datasets. Once collected, the data undergoes cleaning to remove duplicates and enhance diversity. Noise reduction techniques are applied to improve image quality, addressing issues like blurriness and irrelevant background information. Standardization and normalization of images follow, resizing them to a consistent format and adjusting pixel values to a common scale to ensure uniformity.

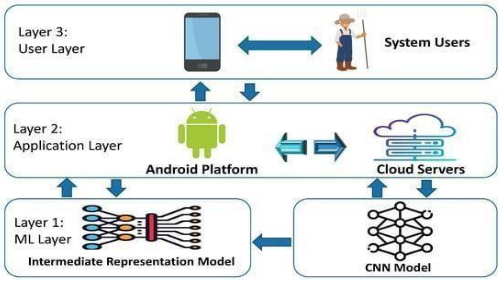

Figure 3.2: system architecture

3.3 ALGORITHAM USED

The CNN network has two types of layers: convolution and pooling. Each layer has a group of specialized neurons that perform one of these operations. The convolution operation means detecting the visual features of objects in the input image such as edges, lines, color drops, etc. The pooling process helps the CNN network to avoid learning irrelevant features of objects by focusing only on learning the essential ones. The pooling operation is applied to the output of the convolutional layers to downsampling the generated feature maps by summarizing the features into patches. Two common pooling methods are used: average- pooling and max-pooling. In this paper, we used the max-pooling method, which calculates the maximum value for each patch of the
feature map as the dominant feature. As shown in Figure 3, the output of every Conv2D and MaxPooling2D layer is a 3D form tensor (height, width, channels). The width and height dimensions tend to shrink as we go deeper into the network. The third argument (e.g., 16, 32 or 64) controls the number of output channels for each Conv2D layer. During the training phase, the CNN model generated around 4 million trainable parameters.

Layer (type)	Output Shape	Param #
rescaling_1 (Rescaling)	(None, 180, 180, 3)	0
conv2d (Conv2D)	(None, 180, 180, 16)	448
max_pooling2d (MaxPooling2D)	(None, 90, 90, 16)	0
conv2d_1 (Conv2D)	(None, 90, 90, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 45, 45, 32)	0
conv2d_2 (Conv2D)	(None, 45, 45, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 22, 22, 64)	0
flatten (Flatten)	(None, 30976)	0
dense (Dense)	(None, 128)	3965056
dense_1 (Dense)	(None, 10)	1290

Total params: 3,989,930
Trainable params: 3,989,930
Non-trainable params: 0

Figure 3.3: the structure of the CNN model

3.4 TECHNIQUES

The CNN model is implemented using Keras development environment. Keras is an open- source neural network library written in Python, which uses TensorFlow 02 as a back- end engine. Keras libraries running on top of TensorFlow make it relatively easy for developers to build and test deep learning models written in Python. For instance, we used the keras.preprocessing.image.ImageDataGenera tor library to augment some images in our dataset via several geometric transformations; therefore, our model would never see twice the same image. This helps to avoid overfitting and helps the model generalize better. The training images must have the same size before feeding them as input to the model. Our model was trained with colored (RGB) images with resized dimensions of 200 × 200 pixels. We set the batch size and number of epochs to be 150 images and 10 epochs, respectively. The model training was carried out using a server computer equipped with a 4.50 GHz Intel Core™ i7-16MB CPU processor, 16 GB of RAM, and RTX-3060 CUDA GPU 3584-cores with a base clock speed of 1320 MHz. The training phase took approximately 2 days to run 10 epochs. We took a snapshot of the trained weights every 2 epochs to monitor the progress. The training error and loss are calculated using this equation:

$M=1n\sum i=1n(yi-xi)2$

where M is the mean square error of the model, y is the predicted class calculated by the model, and x is the actual class. M represents the error in object detection.

4. RESULTS

4.1 GRAPHS

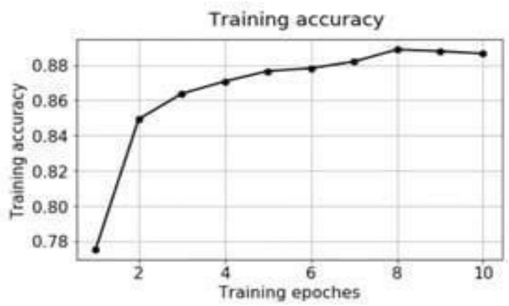

Figure 4.1.1: The Training Accuracy of the CNN Model.

Figure 4.1.2: The Training Loss of the CNN Model.

4.2 SCREENSHOTS

The plant disease detector's user interface is implemented as a self-contained mobile app developed using Kotlin Multiplatform Mobile [23]. Kotlin is a mobile framework that allowed us to write a single codebase for the system's business logic, and then deploy it as an iOS or Android app. In this paper, we deployed the app as an Android app using the Android SDK (Software Development Kit) and XML (Extensible Markup Language) to build the front-end activities. We also built a middleware between the app and the cloud server using Python 3.9.

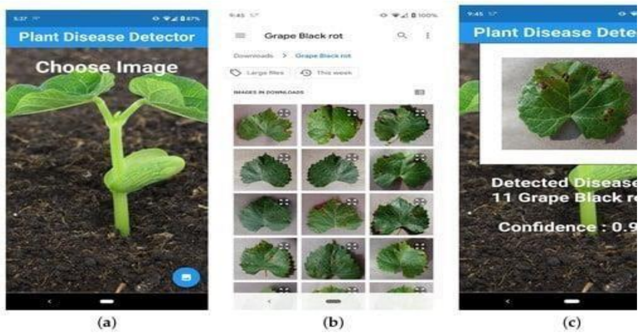

Figure 4.2: Screenshots Detecting Plant Leaf Diseases.

5. REFERENCES

1. Sinha, A.; Shekhawat, R.S. Review of image processing approaches for detecting plant diseases. *IET Image Process.* **2020**, *14*, 27–39.
2. Ai, Y.; Sun, C.; Tie, J.; Cai, X. Research on recognition model of crop diseases and insect pests based on deep learning in harsh environments. *IEEE Access* **2020**, *8*, 686–693.
3. Zeng, Q.; Ma, X.; Cheng, B.; Zhou, E.; Pang, W. Gans-based data augmentation for citrus disease severity detection using deep learning. *IEEE Access* **2020**, *8*, 882– 891.
4. Thomas, S.; Thomas, S.; Kuska, M.T.; Bohnenkamp, D.; Brugger, A.; Alisaac, E.; Wahabzada, M.; Behmann, J.; Mahlein, A.K. Benefits of hyperspectral imaging for plant disease detection and plant protection: A technical perspective. *J. Plant Dis. Prot.* **2018**, *125*, 5–20.
5. Crop Protection Network. 2021. Available online: https://cropprotectionnetwork.o rg/ (accessed on 25 June 2021).
6. Jiang, P.; Chen, Y.; Liu, B.; He, D.; Liang, C. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. *IEEE Access* **2019**, *7*, 69–80.
7. Chen, W.; Lin, Y.; Ng, F.; Liu, C.; Lin, Y. Ricetalk: Rice blast detection using internet of things and artificial intelligence technologies. *IEEE Internet Things J.* **2020**, *7*, 1001–1010.
8. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the

IEEE International Conference on Computer Vision, ser. ICCV '17, Venice, Italy, 22–29 October 2017; pp. 2980– 2988

9. Ahmed, A.A.; Omari, S.A.; Awal, R.; Fares, A.; Chouikha, M. A distributed system for supporting smart irrigation using iot technology. *Eng. Rep.* **2020**, 1– 13
10. Sun, J.; Yang, Y.; He, X.; Wu, X. Northern maize leaf blight detection under complex fielenvironment based on deep learning. IEEE Acce