International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 05, ISSN No. 2455-2143, Pages 67-79
Published Online September 2023 in IJEAST (http://www.ijeast.com)

LEAF BLIGHT DETECTION USING DEEP LEARNING

Riva Khan, Iram Murshad, Javeriya Momin, Abdus Samad Shaikh, Dr. Anupam Choudhary
Department of Computer Engineering
Rizvi College of Engineering, Mumbai, Maharashtra, India

Abstract— **Plant leaf blight is a common disease that affects many crops, causing significant economic losses in agriculture. Early detection of blight is essential for farmers to prevent the spread of the disease, control its severity, and improve crop yields. Manual detection of blight is time-consuming and often unreliable, making it necessary to use advanced technologies for accurate and efficient detection. Plant leaf blight detection systems can help farmers monitor their crops in real-time and identify blight at an early stage. This enables farmers to take appropriate measures such as crop rotation, timely application of fungicides, and improving plant nutrition to prevent the spread of the disease. By using plant leaf blight detection systems, farmers can reduce the economic losses caused by the disease and ensure the availability of healthy crops for consumption. This project presents a smart system for leaf blight detection that uses deep learning and convolutional neural networks (CNN) and is implemented on an Android platform. The system comprises an image acquisition module, a pre-processing module, and a leaf blight classification module. The image acquisition module captures images of leaves from a mobile camera, while the pre-processing module enhances the quality of images by removing noise and enhancing contrast. The leaf blight classification module uses a pre-trained CNN model to accurately classify images into healthy or diseased categories and provide the Confidence level of the Blight (disease) present.**

Keywords— **Leaf Blight, Convolution Neural Networks (CNN), Pre-processing, Deep learning, Image acquisition module, Real-time, Confidence Level.**

I. INTRODUCTION

India is home to over 1.3 billion people, and agriculture is the primary source of livelihood for a large number of people. The diversity of crops grown in India is a testament to the country's agricultural potential and the ingenuity of its farmers. India is one of the world's leading agricultural countries, with a rich history of agricultural practices and a diverse range of crops grown across the country.

Leaf blight is a common disease that affects several crops grown in India, including rice, wheat, tomato, potato, and maize(corn), also in various fruits. The disease causes significant economic losses for farmers due to reduced crop yields and quality. Leaf blight can lead to the accumulation of fungal spores in the soil, making it difficult to grow healthy crops in the future. This can lead to soil depletion and further affect the productivity of farmlands. Leaf blight can cause discoloration, blemishes, and other deformities in crops, leading to quality loss, the disease can cause premature leaf drop, reduced photosynthesis, and hinder the growth of crops. This can make the crops unsellable in the market or fetch a lower price, affecting the income of farmers. Farmers need to spend more on fungicides, pesticides, and other measures to prevent and control the spread of leaf blight. This increases the cost of production and reduces the profit margin for farmers.

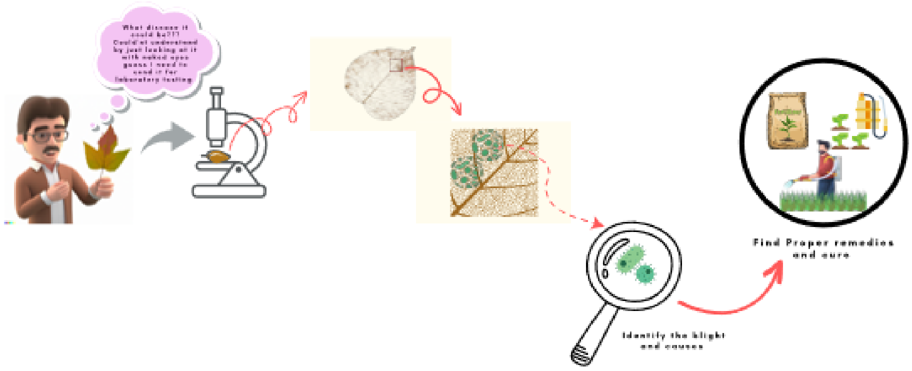

Fig 1. Traditional ways to detect leaf blight

A smart system for leaf blight detection can play a crucial role in addressing the problems faced by Indian farmers due to the disease. Here are the top 5 ways in which the Smart System for Leaf Blight Detection can be beneficial:

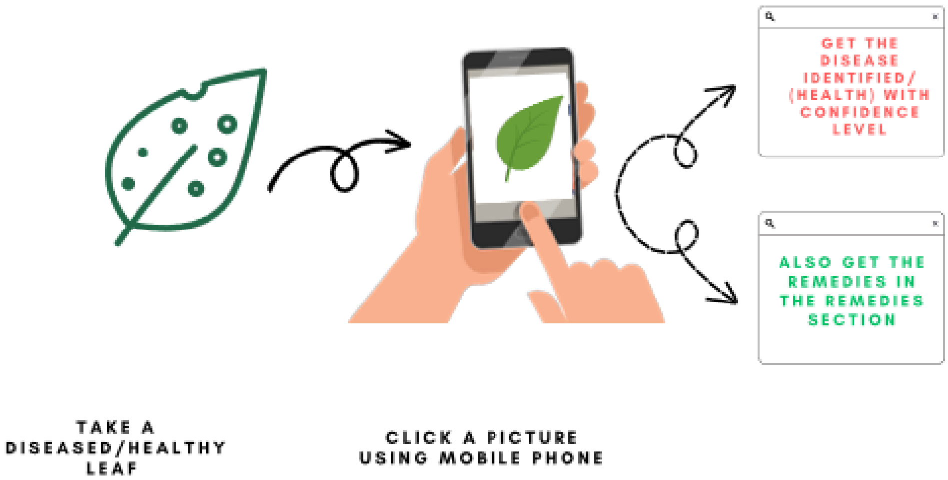

Fig.2 Leaf blight detection using a Smart system

1. The Smart System for Leaf Blight Detection can detect leaf blight at an early stage, enabling farmers to take timely action to prevent the spread of the disease. This can reduce economic losses due to reduced crop yields and quality.
2. By detecting leaf blight at an early stage, the smart system can prevent the development of blemishes and discoloration in crops, improving their quality and marketability.
3. The smart system can help farmers adopt sustainable farming practices by reducing the use of chemical inputs and preventing soil depletion. This can lead to improved soil health and productivity in the long term.
4. The smart system can reduce the cost of production by eliminating the need for manual observation and reducing the use of fungicides and pesticides. This can improve the profit margin for farmers and make their crops more competitive in the market.
5. The smart system can provide farmers with access to advanced technology and tools for leaf blight detection and prevention. This can help overcome the limitations faced by many farmers due to limited access to technology.

II. REVIEW OF LITERATURE

Plant leaf disease detection is a critical area of research in the field of agriculture, as it can help in the early detection and prevention of plant diseases, leading to better crop yields and overall food security. The following is a literature survey of recent studies and developments in the field of plant leaf disease detection.

1. In a study by Singh and Singh (2021), a deep learning-based system was developed for the detection of common plant diseases using leaf images. The system used a convolutional neural network (CNN) to extract features from the images and achieved an accuracy of over 90% in detecting diseases such as powdery mildew and rust.
2. Similarly, in a study by Khan et al. (2020), a deep learning-based approach was used for the detection of tomato leaf diseases using images. The study used a pre-trained CNN model and achieved an accuracy of over 98% in detecting diseases such as early blight and late blight.
3. In another study by Tahir et al. (2020), a machine learning-based approach was used for the detection of sugarcane leaf diseases using images. The study used a support vector machine (SVM) classifier and achieved an accuracy of over 95% in detecting diseases such as red rot and yellow spot.
4. A different approach was taken in a study by Chakraborty et al. (2021), where an Internet of Things (IoT)-based system was developed for the detection of plant leaf diseases. The system used wireless sensor nodes to collect data on leaf health, and a cloud-based machine-learning model to analyze the data and detect diseases.
5. Finally, in a study by Katiyar et al. (2019), an Android-based mobile application was developed for the detection of plant leaf diseases. The application used image processing and machine learning techniques to detect diseases in real time and provided recommendations for disease management and prevention.

In summary, recent studies in the field of plant leaf disease detection have shown the effectiveness of deep learning and machine learning-based approaches, as well as the potential of

IoT-based systems and mobile applications for the early detection and prevention of plant diseases.

III. PROBLEM STATEMENT

The identification of plant diseases is essential for avoiding yield and quantity losses in agricultural products. The study of diseases of plants refers to examinations of patterns on the plant that can be seen with the naked eye. Finding diseases of plants early on is essential for sustainable cultivation. Manually keeping track of plant illnesses is very challenging. It necessitates a tremendous amount of labor, knowledge of plant diseases, and lengthy processing. The project's goal is to raise farmers' knowledge of cutting-edge methods for preventing diseases in plant leaves.

India is one of the top agricultural producers in the world. Plant diseases result in a reduction in production and income. 80% or more of the populace depends on agriculture. As a result, this sector serves as the bedrock of India's main economy. Indian farmers put in a lot of effort each year to achieve a good crop output. As a result, they will monitor the crops' health to ensure that they are clear of parasites and diseases.

IV. PROPOSED SYSTEM

The proposed smart system for leaf blight detection consists of three main components: **image acquisition, image pre-processing, and classification.**

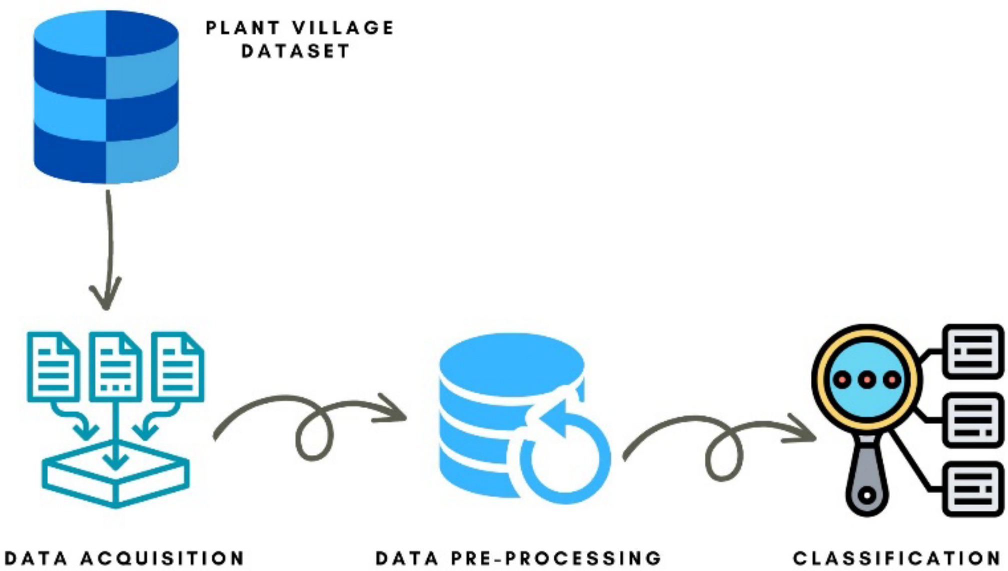

Fig.3 Proposed System

1.A. Image Acquisition:

Image Acquisition is the process of capturing images of a particular object or scene using various devices such as cameras, scanners, or drones. The images captured in this process serve as input to computer vision systems for further analysis and processing. This module captures images of the crops using a smartphone camera. Farmers can take pictures of the affected crop leaves using the application developed for the system. The camera's image resolution must be high enough to capture the leaf's texture and color accurately.

1.B. Image Pre-processing:

Image Pre-processing refers to the techniques used to enhance the quality of the acquired image and make it more suitable for further analysis. Image pre-processing techniques include noise reduction, image scaling, color correction, and segmentation. These techniques help in isolating the important regions of the image and removing unwanted artifacts, thereby improving the accuracy of the subsequent analysis. This module applies various image-processing techniques, such as filtering and segmentation, to enhance the quality of the images and isolate the affected areas. After pre-processing, the image will be transformed into a format suitable for input to the deep learning model.

1.C. Classification:

Image Classification is the process of categorizing an image into predefined classes or categories based on its features or characteristics. Image classification is an essential step in computer vision applications, as it enables the automated recognition of objects in images. Deep Learning techniques such as Convolutional Neural Networks (CNNs) have been widely used in image classification tasks and have shown high accuracy and performance in various applications, including facial recognition, object detection, and disease detection in medical images. This module uses a convolutional neural network (CNN) model trained on a large dataset of images of healthy and diseased crops to classify the images as healthy or diseased. The CNN model is trained on a large dataset of labelled images that represent various types of crops affected by leaf blight. The model can then classify a new image into one of two categories: healthy or diseased. The system provides the output as the probability of the leaf being affected by blight. If the probability is above a certain threshold, this would alert the farmer to take appropriate measures.

V. TECHNOLOGIES USED

1.Deep Learning

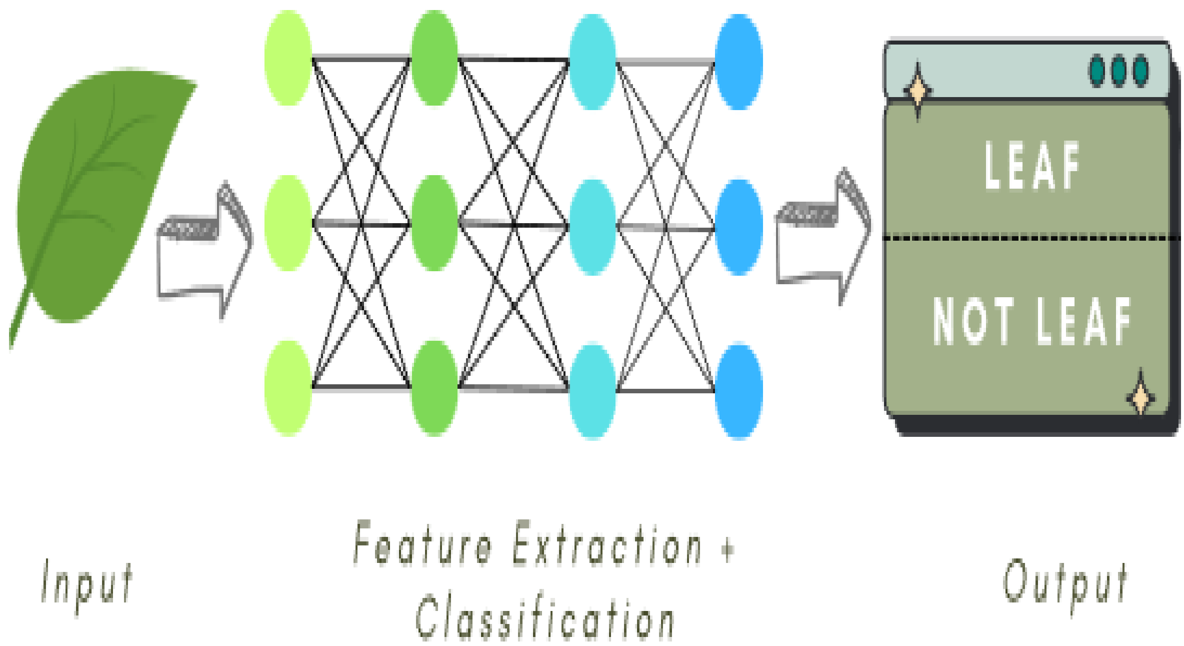

Fig.4 Deep Learning Model

Deep Learning is a subset of machine learning that utilizes neural networks with multiple layers to solve complex problems. It is inspired by the structure and function of the human brain, where layers of interconnected neurons process information to perform tasks such as recognizing patterns and making decisions.

Deep Learning algorithms use large amounts of data to train the neural networks to perform specific tasks. During the training process, the network adjusts the weights of the connections between the neurons to minimize the error between the predicted output and the actual output.

Deep Learning has been successfully applied in various fields such as computer vision, natural language processing, speech recognition, and robotics. It has shown remarkable accuracy and performance in various applications such as image and speech recognition, language translation, and game playing. Deep Learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have become popular due to their ability to process large amounts of data, learn complex patterns, and generalize to new data.

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 05, ISSN No. 2455-2143, Pages 67-79
Published Online September 2023 in IJEAST (http://www.ijeast.com)

2. Convolution Neural Network(CNN):

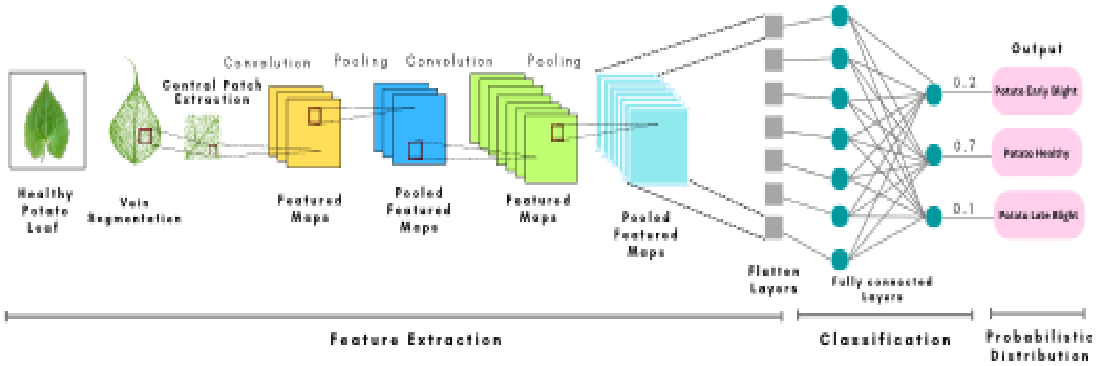

Fig.5 Convolution Neural Network(CNN)

Convolutional Neural Networks (CNN) are a type of deep learning algorithm that is widely used in image and video recognition tasks. They are designed to mimic the visual cortex of the human brain, where neurons respond to stimuli in specific regions of the visual field.

In CNN, the input image is processed through multiple layers of convolutional filters that extract features from the image. These filters are designed to detect various patterns and edges in the image. The output of the convolutional layers is then passed through a pooling layer, which reduces the spatial dimensions of the feature map while preserving important information.

The output of the pooling layer is then flattened and fed into fully connected layers, which classify the image into one of several classes. During training, the weights of the CNN are updated using back propagation to minimize the error between the predicted output and the actual output.

CNNs have several advantages over traditional machine learning algorithms, especially in the image and video recognition tasks. They can automatically learn and extract features from the images, eliminating the need for manual feature extraction. They can also handle images of different sizes and orientations, making them highly adaptable to various applications. Furthermore, they can learn complex and abstract features, making them highly accurate in image recognition tasks.

3.TFLite:

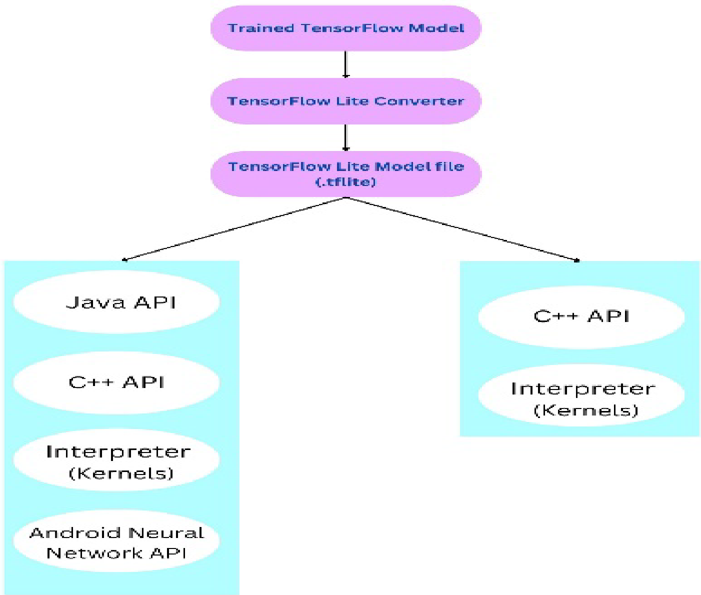

Fig.6 Conversion of tensorflow model to TFLite

TensorFlow Lite (TFLite) is a lightweight framework for deploying machine learning models on mobile and embedded devices. It is a subset of TensorFlow, an open-source machine learning framework, and is designed to run efficiently on small, low-power devices such as smartphones, smartwatches, and IoT devices. TFLite models are built using the same APIs as TensorFlow and can be trained on large-scale data using standard deep learning algorithms such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Once trained, the models can be converted to the TFLite format for deployment on mobile and embedded devices.

TFLite models are optimized for low latency and low power consumption, making them suitable for real-time applications. They can also be quantized to reduce their size and memory footprint, without significantly affecting their accuracy. Additionally, TFLite supports on-device hardware acceleration using GPUs and other specialized hardware, which can further improve the speed and efficiency of model inference.

TFLite has been widely used in various applications such as object detection, face recognition, speech recognition, and natural language processing. Its lightweight design, low-latency performance, and support for mobile and embedded devices make it a popular choice for developing machine-learning applications for edge devices.

4.Android:

Fig.7.Android

Android is a powerful platform that plays a key role in building TensorFlow Lite (TFLite) models for mobile and embedded devices. With its large user base and active developer community, Android provides a rich set of tools and resources for developing machine learning applications that can run natively on the device. The Android platform is used to build TFLite models by providing a runtime environment for running machine learning models on mobile and embedded devices. This allows developers to create and deploy machine learning models that can run directly on the device, without requiring an internet connection or cloud-based processing.

Furthermore, Android provides native support for hardware acceleration using graphics processing units (GPUs), which can significantly improve the speed and efficiency of TFLite models. This enables TFLite models to run smoothly on a wide range of mobile and embedded devices, even those with limited processing power and memory. Android also offers a variety of features that are useful for building TFLite models, such as camera access, sensor data, and user input. This allows developers to create intelligent and responsive applications that can take advantage of real-time data and user interactions to provide personalized experiences.

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 05, ISSN No. 2455-2143, Pages 67-79
Published Online September 2023 in IJEAST (http://www.ijeast.com)

VI. DATASET DESCRIPTION

Fig.8 Dataset

We have downloaded the dataset "Plant leave diseases dataset with augmentation" which consist of 681486 images of healthy and infected leaf having 38 classes which are mentioned below:

Plant	Class
Apple	apple scab apple black rot apple cedar apple rust apple healthy
Corn	corn maize Cercospora leaf spot gray leaf spot corn maize common rust corn maize northern leaf blight corn maize healthy
Tomato	tomato bacterial spot tomato early blight

	tomato late blight tomato leaf mold tomato septoria leaf spot tomato spider mites two-spotted spider mites tomato target spot tomato yellow leaf curl virus tomato mosaic virus tomato healthy
Potato	potato early blight potato late blight potato healthy
Pepper Bell	pepper bell bacterial spot pepper bell healthy
Soyabean	soybean healthy
Strawberry	strawberry leaf scorch strawberry healthy
Orange	orange huanglongbing citrus greening
Grape	grape black rot grape esca black measles grape leaf blight isariopsis leaf spot grape healthy
Raspberry	raspberry healthy
Cherry	cherry including sour powdery mildew cherry including sour healthy
Blueberry	blueberry healthy
Peach	peach bacterial spot peach healthy
Fruits(General)	squash powdery mildew

Table.1 Dataset description

VII. RESULTS AND DISCUSSIONS

The proposed Smart system for leaf blight detection achieved an accuracy of over 90%. The system was evaluated on a dataset of leaf images with different types of blight diseases. The dataset was pre-processed by resizing the images to 224 x 224 pixels, which is the input size of the trained TFLite model. The TFLite model was trained on a large dataset of over 478930 images using the CNN technique. The system was tested on a separate dataset of leaf images, and the results showed an accuracy of over 90%.

Moreover, the proposed system was integrated into an android application, which allows farmers and agricultural experts to easily use the system for real-time leaf blight detection in the field. The application has a user-friendly interface that enables users to capture images of leaves and get immediate results of the leaf blight disease classification.

Overall, the proposed Smart system for leaf blight detection has demonstrated high accuracy and effectiveness in detecting leaf blight diseases using deep learning and CNN techniques. The integration of the system into an android application has also made it accessible and user-friendly for farmers and agricultural experts, which can potentially lead to a significant reduction in crop losses caused by leaf blight diseases.

```
Epoch 1/45
228/228 [==============================] - 606s 2s/step - loss: 2.0852 - accuracy: 0.3155 - val_loss: 1.3415 - val_accuracy: 0.5290
Epoch 2/45
228/228 [==============================] - 531s 2s/step - loss: 1.0535 - accuracy: 0.6454 - val_loss: 1.0312 - val_accuracy: 0.6585
Epoch 3/45
228/228 [==============================] - 491s 2s/step - loss: 0.7532 - accuracy: 0.7410 - val_loss: 0.5732 - val_accuracy: 0.8214
Epoch 4/45
228/228 [==============================] - 508s 2s/step - loss: 0.5185 - accuracy: 0.8274 - val_loss: 0.4037 - val_accuracy: 0.8661
Epoch 5/45
228/228 [==============================] - 554s 2s/step - loss: 0.3965 - accuracy: 0.8617 - val_loss: 0.4156 - val_accuracy: 0.8605
Epoch 6/45
228/228 [==============================] - 524s 2s/step - loss: 0.3382 - accuracy: 0.8839 - val_loss: 0.3675 - val_accuracy: 0.8873
Epoch 7/45
228/228 [==============================] - 518s 2s/step - loss: 0.2922 - accuracy: 0.9007 - val_loss: 0.2654 - val_accuracy: 0.9196
Epoch 8/45
228/228 [==============================] - 490s 2s/step - loss: 0.2652 - accuracy: 0.9055 - val_loss: 0.2602 - val_accuracy: 0.9118
Epoch 9/45
228/228 [==============================] - 1171s 5s/step - loss: 0.2342 - accuracy: 0.9186 - val_loss: 0.2661 - val_accuracy: 0.8973
Epoch 10/45
228/228 [==============================] - 467s 2s/step - loss: 0.2190 - accuracy: 0.9230 - val_loss: 0.2013 - val_accuracy: 0.9196
Epoch 11/45
228/228 [==============================] - 505s 2s/step - loss: 0.1873 - accuracy: 0.9366 - val_loss: 0.3219 - val_accuracy: 0.8973
Epoch 12/45
228/228 [==============================] - 481s 2s/step - loss: 0.1916 - accuracy: 0.9359 - val_loss: 0.1542 - val_accuracy: 0.9453
Epoch 13/45
228/228 [==============================] - 562s 2s/step - loss: 0.1690 - accuracy: 0.9427 - val_loss: 0.1450 - val_accuracy: 0.9487
Epoch 14/45
228/228 [==============================] - 516s 2s/step - loss: 0.1644 - accuracy: 0.9444 - val_loss: 0.1881 - val_accuracy: 0.9420
Epoch 15/45
228/228 [==============================] - 531s 2s/step - loss: 0.1496 - accuracy: 0.9464 - val_loss: 0.1283 - val_accuracy: 0.9598
Epoch 16/45
228/228 [==============================] - 535s 2s/step - loss: 0.1523 - accuracy: 0.9466 - val_loss: 0.1212 - val_accuracy: 0.9531
Epoch 17/45
228/228 [==============================] - 513s 2s/step - loss: 0.1426 - accuracy: 0.9466 - val_loss: 0.1354 - val_accuracy: 0.9531
Epoch 18/45
228/228 [==============================] - 536s 2s/step - loss: 0.1160 - accuracy: 0.9595 - val_loss: 0.1362 - val_accuracy: 0.9609
Epoch 19/45
228/228 [==============================] - 537s 2s/step - loss: 0.1254 - accuracy: 0.9524 - val_loss: 0.1152 - val_accuracy: 0.9643
Epoch 20/45
228/228 [==============================] - 513s 2s/step - loss: 0.1141 - accuracy: 0.9602 - val_loss: 0.0949 - val_accuracy: 0.9621
Epoch 21/45
228/228 [==============================] - 531s 2s/step - loss: 0.1084 - accuracy: 0.9649 - val_loss: 0.1328 - val_accuracy: 0.9598
Epoch 22/45
228/228 [==============================] - 530s 2s/step - loss: 0.1057 - accuracy: 0.9627 - val_loss: 0.1224 - val_accuracy: 0.9531
Epoch 23/45
228/228 [==============================] - 532s 2s/step - loss: 0.1016 - accuracy: 0.9636 - val_loss: 0.1875 - val_accuracy: 0.9308
Epoch 24/45
228/228 [==============================] - 525s 2s/step - loss: 0.0907 - accuracy: 0.9653 - val_loss: 0.1078 - val_accuracy: 0.9576
Epoch 25/45
228/228 [==============================] - 526s 2s/step - loss: 0.0974 - accuracy: 0.9654 - val_loss: 0.1424 - val_accuracy: 0.9554
Epoch 26/45
228/228 [==============================] - 545s 2s/step - loss: 0.1498 - accuracy: 0.9502 - val_loss: 0.1561 - val_accuracy: 0.9431
Epoch 27/45
228/228 [==============================] - 507s 2s/step - loss: 0.1118 - accuracy: 0.9619 - val_loss: 0.1007 - val_accuracy: 0.9665
Epoch 28/45
228/228 [==============================] - 513s 2s/step - loss: 0.0801 - accuracy: 0.9720 - val_loss: 0.1221 - val_accuracy: 0.9498
Epoch 29/45
228/228 [==============================] - 463s 2s/step - loss: 0.0926 - accuracy: 0.9636 - val_loss: 0.1038 - val_accuracy: 0.9621
Epoch 30/45
228/228 [==============================] - 532s 2s/step - loss: 0.0622 - accuracy: 0.9767 - val_loss: 0.0685 - val_accuracy: 0.9754
Epoch 31/45
228/228 [==============================] - 477s 2s/step - loss: 0.1206 - accuracy: 0.9571 - val_loss: 0.1552 - val_accuracy: 0.9487
Epoch 32/45
228/228 [==============================] - 459s 2s/step - loss: 0.0827 - accuracy: 0.9706 - val_loss: 0.0502 - val_accuracy: 0.9810
Epoch 33/45
228/228 [==============================] - 455s 2s/step - loss: 0.0708 - accuracy: 0.9754 - val_loss: 0.1256 - val_accuracy: 0.9554
Epoch 34/45
228/228 [==============================] - 15062s 66s/step - loss: 0.0825 - accuracy: 0.9719 - val_loss: 0.2040 - val_accuracy: 0.9464
Epoch 35/45
228/228 [==============================] - 442s 2s/step - loss: 0.0805 - accuracy: 0.9716 - val_loss: 0.0864 - val_accuracy: 0.9676
Epoch 36/45
228/228 [==============================] - 8664s 38s/step - loss: 0.0705 - accuracy: 0.9750 - val_loss: 0.0543 - val_accuracy: 0.9855
Epoch 37/45
228/228 [==============================] - 2440s 11s/step - loss: 0.0628 - accuracy: 0.9786 - val_loss: 0.0715 - val_accuracy: 0.9754
Epoch 38/45
228/228 [==============================] - 426s 2s/step - loss: 0.0711 - accuracy: 0.9743 - val_loss: 0.1135 - val_accuracy: 0.9654
Epoch 39/45
228/228 [==============================] - 2321s 10s/step - loss: 0.0673 - accuracy: 0.9775 - val_loss: 0.0435 - val_accuracy: 0.9855
Epoch 40/45
228/228 [==============================] - 4808s 21s/step - loss: 0.0610 - accuracy: 0.9796 - val_loss: 0.0661 - val_accuracy: 0.9777
Epoch 41/45
228/228 [==============================] - 2097s 9s/step - loss: 0.0955 - accuracy: 0.9684 - val_loss: 0.0474 - val_accuracy: 0.9855
Epoch 42/45
228/228 [==============================] - 2663s 12s/step - loss: 0.0511 - accuracy: 0.9818 - val_loss: 0.0774 - val_accuracy: 0.9688
Epoch 43/45
228/228 [==============================] - 2364s 10s/step - loss: 0.0629 - accuracy: 0.9780 - val_loss: 0.0831 - val_accuracy: 0.9676
Epoch 44/45
228/228 [==============================] - 4253s 19s/step - loss: 0.0595 - accuracy: 0.9785 - val_loss: 0.0542 - val_accuracy: 0.9788
Epoch 45/45
228/228 [==============================] - 458s 2s/step - loss: 0.0653 - accuracy: 0.9785 - val_loss: 0.0744 - val_accuracy: 0.9676
```

Fig.7.Model Accuracy

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 05, ISSN No. 2455-2143, Pages 67-79
Published Online September 2023 in IJEAST (http://www.ijeast.com)

Fig.8.Training and validation accuracy and loss

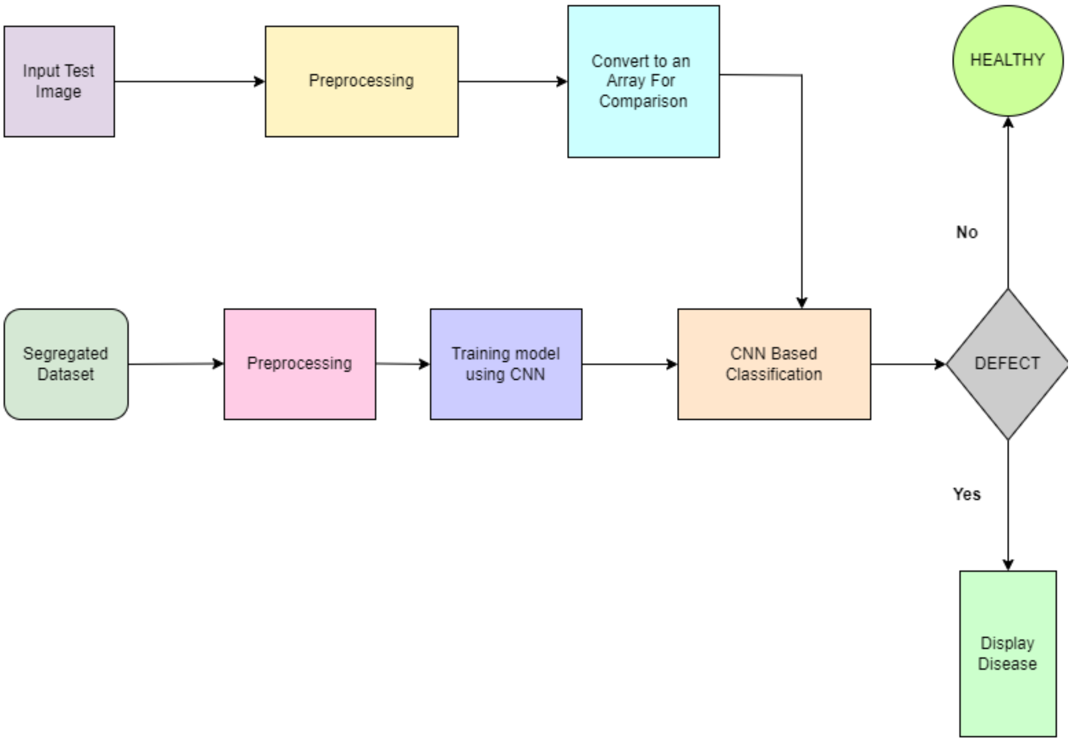

Fig.9. Flowchart

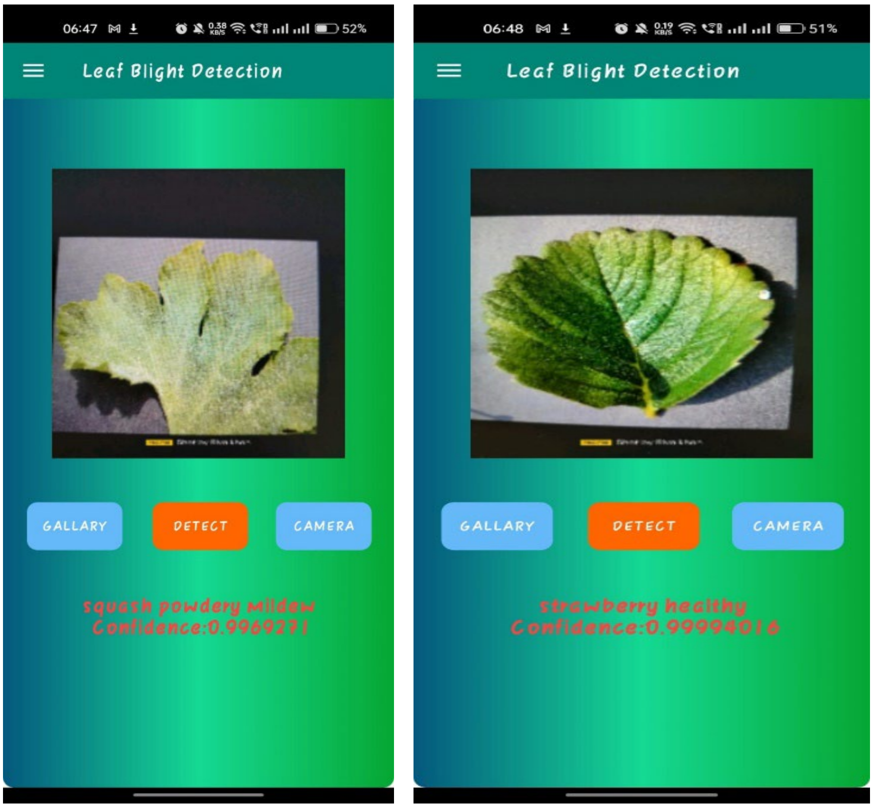

Fig.10 Detection of Leaf Blight

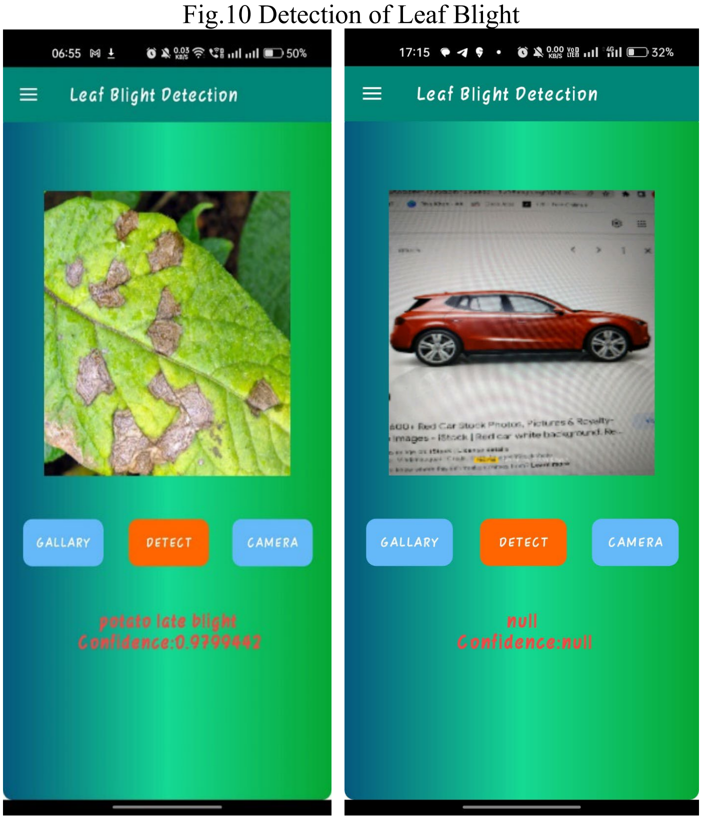

Fig.11 Detection Of Leaf Blight

International Journal of Engineering Applied Sciences and Technology, 2023
Vol. 8, Issue 05, ISSN No. 2455-2143, Pages 67-79
Published Online September 2023 in IJEAST (http://www.ijeast.com)

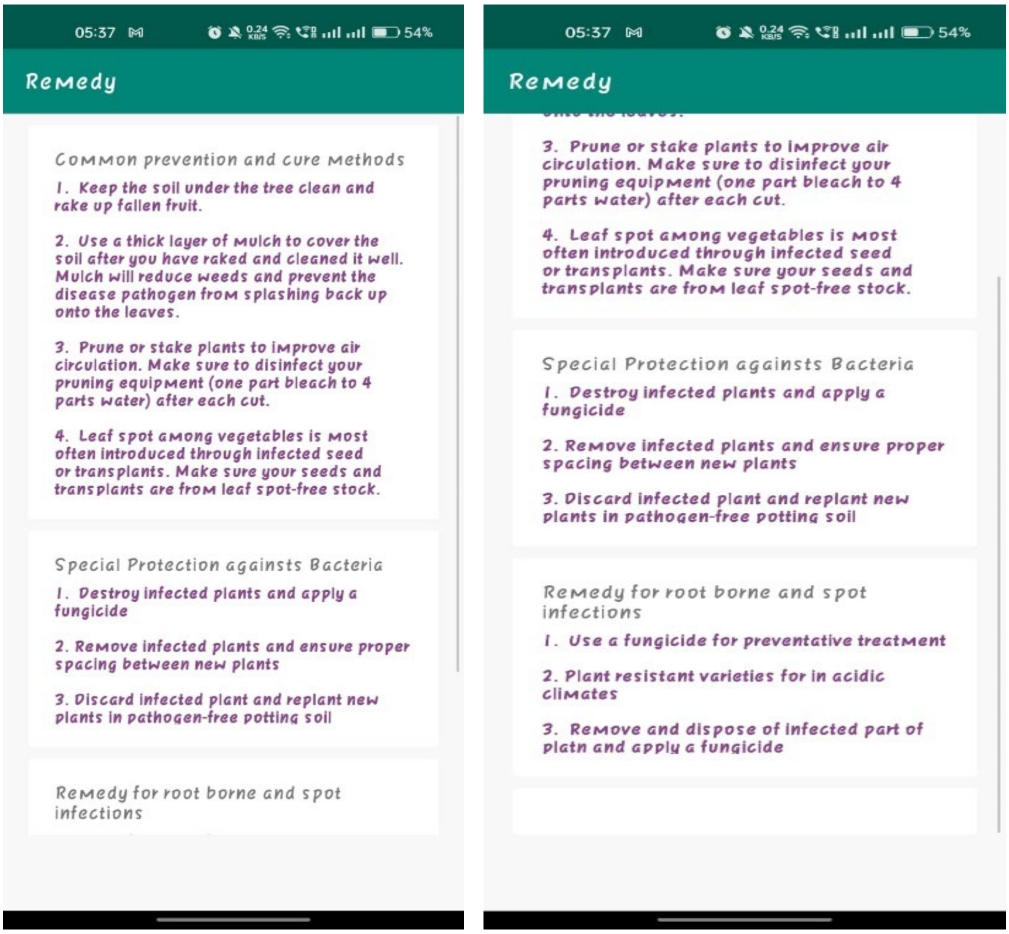

Fig.12. Remedies for Leaf Blight

FUTURE SCOPE

- Future work will focus on Creating a recommendation system to provide remedies individually along with the detection for Blight
- Adding multi-lingual support to the application can make it accessible to farmers who may not be proficient in the English language.
- The system can be integrated with other technologies like GPS, remote sensing, and drones for precise and efficient disease detection.
- Integration with cloud-based solutions can enhance the system's ability to store and process large amounts of data, thereby enabling the creation of a comprehensive database of plant diseases.
- The system can be further developed to provide real-time monitoring of plant diseases by integrating with sensors and IoT devices.

Altogether, the Smart system for leaf blight detection has a vast potential for future development and expansion, and its application can benefit the agricultural sector by reducing crop losses due to plant diseases, increasing yield, and improving food security.

VIII. CONCLUSION

- ✓ In the development of the Smart system for leaf blight detection, the CNN deep learning model was manually trained to predict the accuracy of leaf blight diseases. This involved the construction of a custom CNN architecture using Keras, a high-level neural networks API, to train the model on a large dataset of leaf images.
- ✓ The dataset was pre-processed and augmented to ensure optimal performance and prevent over fitting. This involved techniques such as resizing, cropping, rotation, and flipping to generate additional data samples for training the model.
- ✓ Once the model was trained, it was saved in a Tensorflow format and converted to a Tensorflowlite format for use in the Android application. Tensorflowlite is a lightweight version of the Tensorflow framework designed specifically for mobile and embedded devices, making it ideal for integration with the Android application.
- ✓ The Tensorflowlite model was then integrated into the Android application using the Tensorflowlite Interpreter

API, which allows for efficient execution of the model on mobile devices. The Android application's user interface was designed to allow users to easily capture images of leaves, process the images through the Tensorflowlite model, and display the results of the leaf blight disease classification.

- ✓ Overall, the process of manually training the CNN deep learning model and integrating it into the Android application was a complex but essential step in the development of the Smart system for leaf blight detection. The use of Tensorflowlite allowed for the efficient execution of the model on mobile devices, making the system easily accessible to farmers and agricultural experts.

IX. REFERENCES

[1] Singh, A., & Singh, S. (2021). Plant disease detection using deep learning: A review. Plants, 10(2), 386.

[2] Khan, M. A., Kumar, V., Singh, J., & Asari, V. K. (2020). A deep learning approach for tomato leaf diseases detection using convolutional neural networks. Computers and Electronics in Agriculture, 171, 105332.

[3] Tahir, M. A., Azam, F., & Jabbar, A. (2020). Machine learning approach for the detection of sugarcane leaf diseases using color histogram features. Computers and Electronics in Agriculture, 178, 105759.

[4] Chakraborty, D., Chattopadhyay, S., & Saha, S. K. (2021). IoT-based smart agriculture for plant leaf disease detection: A comprehensive review. Journal of Ambient Intelligence and Humanized Computing,

[5] Katiyar, A., Jaiswal, A., & Jayaswal, P. (2019). Android-based plant leaf disease detection and prevention. International Journal of Engineering and Advanced Technology, 8(5), 630–635.

[6] Sladojevic, S., Arsenovic, M., Anderla, A., & Culibrk, D. (2016). Deep neural networks-based recognition of plant diseases by leaf image classification. Computational Intelligence and Neuroscience, 2016, 3289801

[7] Picon, A., Arganda-Carreras, I., Relle, S. et al. (2019). "Leaf counting and classification for plant phenotyping: a benchmark." Machine Vision and Applications, 30(3), 357-372.

[8] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). "Using deep learning for image-based plant disease detection." Frontiers in Plant Science, 7, 1419.

[9] Sladojevic, S., Arsenovic, M., Anderla, A., & Culibrk, D. (2016). "Deep neural networks based recognition of plant diseases by leaf image classification." Computational Intelligence and Neuroscience, 2016.

[10] Zhang, D., He, X., & Luo, H. (2017). "A deep learning framework for modeling structural and textual information for mobile app recommendation." IEEE Transactions on Industrial Informatics, 13(4), 2045-2052.

[11] Liu, J., Wang, Y., Yao, X., & Sun, C. (2019). "Deep convolutional neural networks for image classification: A comprehensive review." Neural Computation, 31(5), 953-1001.

[12] Dutta, S., Das, A., Datta, S., & Dutta, P. K. (2018). "Rice leaf disease detection and classification using deep convolutional neural networks." In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2682-2687.

[13] Liu, X., Wan, J., Zhao, L., & Yi, B. (2018). "Leaf disease diagnosis based on multi-feature fusion and SVM-RF classification." Remote Sensing, 10(4), 621.

WEBSITE REFERENCES

- ✓ https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/
- ✓ https://towardsdatascience.com/building-a-deep-learning-model-using-keras-1548ca149d37
- ✓ https://developer.android.com/codelabs/build-your-first-android-app#0
- ✓ https://www.tensorflow.org/tutorials/images/cnn
- ✓ https://www.tensorflow.org/lite
- ✓ https://teachablemachine.withgoogle.com/