Identification of Tomato Disease Detection and Classification with Infected Areas Using Deep Neural Networks

R. Ramya[1], P. Kumar[2], M. Senthil kumar[2], V. Sounthar Raj[2], M. Mohamed Faizal[2]

Assistant Professor[1], Professor[2], Under Graduate students[2]

Department of ECE, K.S. Rangasamy College of Technology

Tamil Nadu, India

srvece@gmail.com

Article Info
Page Number: 434 – 447
Publication Issue:
Vol. 71 No. 3s (2022)

Article History
Article Received: 22 April 2022
Revised: 10 May 2022
Accepted: 15 June 2022
Publication: 19 July 2022

Abstract

Agriculture is indisputably the backbone of our nation. India is the second-largest producer of agricultural products globally. In terms of people to feed, Indian agriculture lags well behind other countries in terms of per hectare production in practically all crops. Plant infections cause major economic and production losses and curtail agricultural production quantity and quality. Producers need to monitor their plants regularly and observe any primary symptoms to prevent plant sickness at a low cost and save a significant part of the production. In recent days, technology has played a vital role in all research fields. So, the help of technology is used to detect plant infection. In India, technology-based modern agriculture is the most required to make more profit in every part of agriculture. Thus, the application of technology in agriculture, such as precision agriculture, may assist enhance productivity, improve the condition of Indian farmers, and preserve their products. Thus, the overall progress of production is obtained. Detection of infections in crop management is peremptory for agriculture to be sustainable. However, because to the cluttered background in today's agricultural economy, automated crop infection diagnosis and prediction is a major difficulty. The Internet of Things (IoT) and deep learning have played a significant role in the recent decade, gathering a tremendous amount of contextual data to recognise agricultural infections. This paper describes a real-time technique for detecting tomato leaf infection based on deep convolutional neural networks such as mobileNet and ResNet CNN models. Tuning the hyper-parameters and altering the pooling combinations on a system enhances deep neural network performance. A neural compute stick comprised of dedicated CNN hardware blocks is used to deploy the pre-trained deep CNN model onto a PIC microcontroller.TTL is used to connect the output from the MATLAB software in the personal computer and PIC microcontroller. The PIC microcontroller is programmed with an

embedded C programme, and the ESP8266 IoT module is used to transfer data from the microcontroller to the cloud server, where it is displayed on a web page. The deep learning model obtains more accuracy in the detection of leaf infections, indicating the practicality of this technology and the treatment of infected leaf infections.
Keywords: Plant infection, Tomato, CNN models, ResNet, MobileNet, Classification, MATLAB, Deep learning

I. INTRODUCTION

The agriculture sector employs a large portion of India's population. Tomatoes are the most commonly utilised vegetable in India. Leaf infections are the farming sector's prime problem, which affect crop production and profit. This proposed work aimed to design and develop a smartphone application to identify tomato leaf infection and its severity. To begin with the login credentials, firstly, insert the high-quality tomato leaf images which are captured on camera. It is classified using the deep convolution neural networks (CNN) method. A CNN is a deep learning algorithm that can take an input image, assign importance to different objects in the image, and classify them. The photos in the database include tomato illnesses such as septoria leaf spot, leaf mould, bacterial spot, early blight, and late blight that were categorised using CNN techniques that were trained and fine-tuned to match precisely to the database. The processed input image and images from the database are compared, detecting the type of infection. Finally, the severity of the tomato leaf infection was estimated as low, medium, and high using python language. Thus, the result will be displayed as the type of infection and its severity. Using this programme increases agriculture output, vegetable growth, and pesticide usage proportionally.

The remaining part of the paper is contained in the following ways; section II presents Literature review work of various research work carried out on infections detection of plants and fruits and other research in classification. Section III mentions the methodology and multiple classifiers tool which is used. Section IV discusses the result, and finally, section V conclusion.

II. LITERATURE REVIEWS

Amit Prakash Singh, Anuradha Chug, and Shradha Verma [1] (2020). The scientists used transfer learning and feature extraction approaches to assess the infection severity in Tomato Late Blight infection using restricted training images and three pre-trained CNN models. With smaller datasets, fine-tuning the models improves results; nonetheless, the feature extraction strategy outperformed transfer learning with less effort. AlexNet outperformed the other two networks in terms of fine-grained picture classification. [2] Sachin B. Jadhav, Vishwanath R. Udupi, and Sanjay B. Patil (2020) proposed a deep learning technique that involved building a classifier model for the defined one non-infection and three illness classes using the AlexNet and GoogleNet CNN architectures (bacterial blight, brown spot, and FLS). The classification accuracy of the AlexNet and GoogleNet CNN models was 98.75 percent and 96.25 percent, respectively. The classification was carried out using the AlexNet and GoogleNet models, with several hyperparameters modified. Daminda T. Andrahanndi, He Dong Jian, and Wang Mei Li [3] (2020), According to statistical analysis, the proposed algorithm for evaluating infection severity is accurate and precise. The tested algorithm demonstrated an average accuracy of 96.12 percent. According to the findings, the proposed method is suitable for evaluating infection severity with minimum computing effort. [4] Azeddine Elhassouny and Florentin Smarandache (2019) suggested a CNN-based algorithm for image recognition that can accurately classify ten prevalent tomato leaf infections. It has the potential to expand the model for fault diagnosis.To improve tomato infection diagnosis accuracy, hundreds of high-quality tomato infection picture samples are still required. After that, think that the simple use of deep convolutional neural networks in computer vision and its applications, particularly smart mobile plant infection recognition, will reduce food shortages caused by illnesses, increase production, and save the lives of many starving people throughout the world. [5] S. Sivagami, S. Mohana Priya (2019) "In this proposed work, two segmentation algorithms, K-means and EM segmentation algorithm, were used to find out deficiency in tomato leaves; when these two algorithms, K-means and EM segmentation algorithm, were compared, the EM segmentation algorithm was more efficient than the K-Means segmented algorithm." When compared to 91 images, these algorithms achieve 87 percent accuracy in K-means and 91 percent accuracy in K Segmentation.

III. METHODOLOGY

Figure 1 depicts the system architecture for this project's plant infection detection system based on deep learning. Precision agriculture may now broaden its computer version scope thanks to deep learning-based automated identification approaches. However, training a deep convolution neural network necessitates huge amounts of data as well as high-performance computational resources. One of the most successful techniques to meeting these stringent requirements on a mobile device is to use cloud computing capabilities to transport input data and acquire inference results. When using more augmented photos, the goal is to change the network by boosting appropriate attributes. The major goal of using augmentation during the training stage is to reduce overfitting.

Several techniques, including affine transformation, perspective transformation, and rotation, are utilised in picture augmentation. This method discusses a method for detecting plant illnesses that uses a deep convolution neural network to distinguish infected leaves from healthy leaves or the environment. CNN is a supervised approach that consists of convolutional layers, ReLU layers, fully connected layers, pooling layers, and activation layers. This study presented a deep learning-based tomato infection detection system that included the following steps: gathering datasets, pre-processing, training the CNN model to identify and detect various leaf infections, and validating the model based on the results.

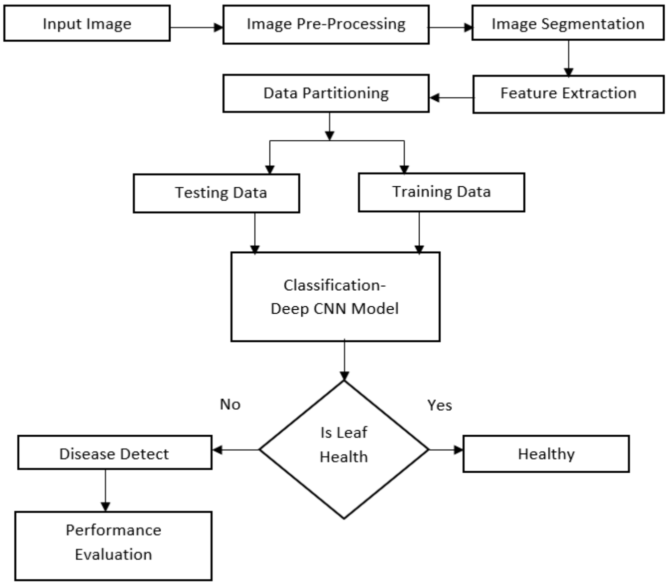


Figure 1: System Architecture for the plant infection detection system using deep learning

Set of tests for predicting whether a leaf is healthy or unwell due to infection in order to evaluate classifier performance. Fine-tuning the training models helps to improve forecast accuracy. Figure 2 depicts the block diagram of the hardware technique used in this work. The component makes use of a PIC microcontroller and cloud storage to accept parametric and picture datasets collected by cameras/sensors. Here, two DHT11 sensors are used to assess temperature and humidity in order to detect rust and blast infections primary.

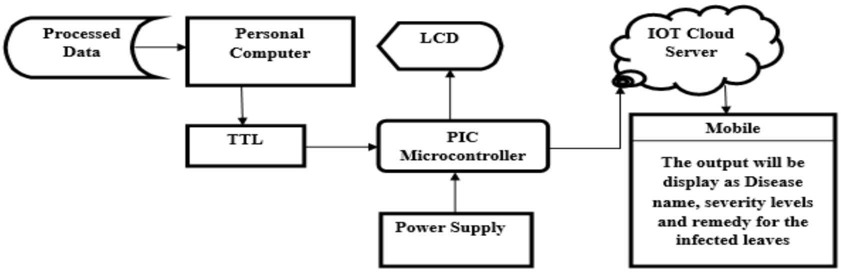


Figure 2: Block Diagram of Hardware Module

The PIC microcontroller is the leading CMOS-based RISC-based microcontroller that practices a distinct bus for training and data, providing simultaneous access to programme and records memory. It is typically used to store critical data that must not be lost if the power source fails unexpectedly. Pic microcontroller interfaces with LCD to display the output from the PC to the microcontroller. Loading the data from the pic microcontroller to the IoT cloud server using the esp8266 IoT module will show the severity level and remedy for that infection. In addition, it will be shown as the web page displayed on the mobile phone. The photos are processed, and the results are delivered to the PIC microcontroller. This makes a decision based on the user's input and displays the SMS/web page to the user through mobile.

Input image - This section obtains input images of the plant leaf from dataset websites like Plant Village and Kaggle. These datasets are used to train the algorithm for this project. Figure 3 shows several sample tomato leaf photos from the dataset for healthy and ill classes.

Healthy Leaf | Early blight Leaf | Leaf Mold | Septoria Leaf Spot | Target Spot

Figure 3: Different Sample images of healthy and unhealthy tomato leaves from the plant village database

Image acquisition- Image acquisition is frequently loosely defined as regaining a picture after some supply, typically a hardware-based supply, and hence it is frequently competent regardless of subsequent operations. Because image acquisition is frequently the principal step in the progression arrangement, no solution is conceivable without a picture. The image that cannot be inherited is unprocessed. As a result, regardless of how the hardware was used to generate it, it may be important for some fields to have a regular baseline from that amount. Table 1 shows the number of leaf photos utilised in the dataset for each class.

Table 1: Diffrent samples of leaf images with various infection and healthy classes

Class	**Unhealthy**					**Healthy**
	Fungi	Bacteria	Mold	Virus	Mite	
Sub Class	Early blight (500)	Bacterial Spot (500)	Late bright mold (500)	Tomato Yellow Leaf Curl Virus (500)	Two spotted spider mite (500)	Healthy (500)
	Septoria leaf spot (500)			Tomato Mosaic Virus (500)		
	Target spot (500)					
	Leaf mold (500)					
Total Tomato Leaf Image (5000)						

Pre-processing- Pre-processing photos typically entails reducing low-frequency background noise, levelling the intensity of individual particles, removing reflections, and masking portions of images. Image pre-processing is the process of improving input images earlier to computer handling.

Image Segmentation

Image segmentation is a technique that, digital image is divided into many subgroups known as Image segments, which aids in reducing the complexity of the image and making any procedure or analysis of the image easier. The distribution of labels to pixels is known as segmentation. A common label is assigned to all image components or pixels that belong to the same class. Take a drag, for example, wherever the image must be delivered as input for object detection. Instead of processing the complete image, the detector will be fed data from an algorithmic segmentation programme. The detector may stop processing the entire image, reducing the activation time.

Feature Extraction

The mining of features begins with an initial collection of measured knowledge. Then, it constructs derived values (features) that are intended to be useful and compressed, hence simplifying the subsequent learning and generalisation phases and, in some situations, resulting in greater human understandings—feature extraction with spatiality reduction. When the input file for a nursing algorithmic programme becomes too large to process and is suspected of being redundant, it is frequently remodelled into a smaller set of possibilities (also named a feature vector). Feature selection refers to the process of determining a collection of beginning alternatives.

Image Classification

Picture classification is the process through which a computer analyses an image and determines which 'class' it belongs to. (Or the likelihood that the image belongs to a 'class.') A class is essentially a label, such as 'flower,' 'leaf,' 'fruit,' and so on. For example, suppose you provide an image of a sheep. The technique of a computer analysing an image and informing you it's a sheep is known as image classification.

Convolutional Neural Networks (CNNs)

CNNs feature an i/p and o/p layer, and additional hidden layers with multiple parameters that allow them to learn multipart objects and patterns. It utilizes convolution and pooling functions to sub-sample the given input before applying an activation function. All of them are middle layers that are connected to a certain extent, with the utterly connected layer at the end of the output layer.

Feature Learning, Layers

Figure 4 depicts a CNN architecture model. A CNN, like other neural networks, has an input layer, an output layer, and several hidden layers in between. These three layers perform operations on the data with the goal of learning data-specific attributes. Convolution activation / ReLU and pooling are the most common layers. The convolution layer converts the input images into convolutional filters, activating specific visual properties. Rectified linear unit (ReLU) training is quicker and further effective since it plots negative values to zero while charge positive values.Because only the activated characteristics are carried on into the next layer, this is frequently referred to as activation. The pooling layer clarifies the output by using nonlinear downsampling to reduce the amount of parameters that the network must learn. These procedures are performed hundreds of times across hundreds of layers, with each layer learning to recognise distinct features.

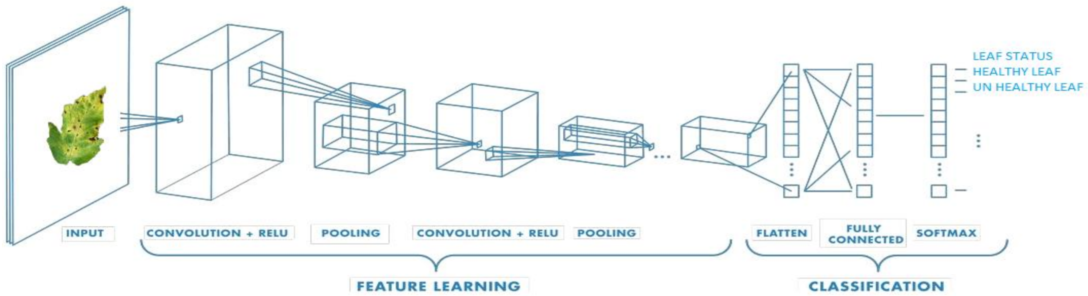


Figure 4: CNN architecture

Pooling is classified into two types: maximum pooling and average pooling. Max Pooling procedure that calculates the supreme value for a feature map's patches and utilises it to produce a downsampled (pooled) feature map. Average Pooling, on the other hand, is a pooling procedure that calculates the average rate for patches of a feature map and utilizes to generate a downsampled (pooled) feature map.

ResNet-50 Architecture

ResNet-50 is a deep network-CNN with fifty association layers. The information is used to import a pre-trained network trained for large images. They stack leftover blocks along to form a network, which will categorise photos into a thousand categories such as plant leaf illnesses, healthy, leaf spot, and so on. As a outcome, the network has trained to create feature illustrations for a extensive range of images. Residual Networks, also known as ResNets, train residual functions based on layer inputs rather than learning unreferenced functions. Instead of assuming that every few stacked layers will directly create a anticipated underlying plotting, residual nets let these layers to supply a residual mapping. They construct a network by stacking residual blocks on top of every alternative: for example, ResNet has 50 layers that use these blocks. Formally, we tend to allow the stacked nonlinear layers work another plotting of F(x): =H(x)x, indicating required underlying plotting as H(x). The initial plotting is transformed into F(x)+x. There is empirical evidence that these networks' area units are easier to optimise and may benefit from the greatly enlarged depth—the design of a ResNet-50 model shown in figure 5 below.

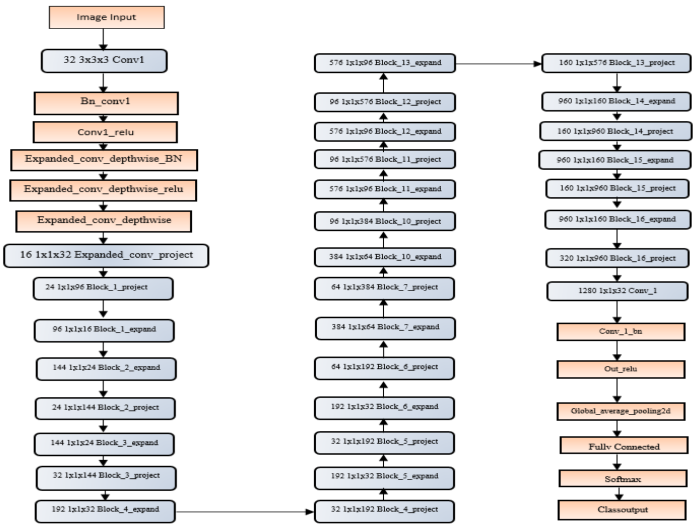


Figure.5: The representation of model architecture image for ResNet-50 and two layered feedforward neural networks

Mobile Net CNN

MobileNet is a 53-layer CNN. Load a pre-trained version of the network that has been educated on additional images from the ImageNet database. The pre-trained network can categorise photos into 1000 different groups, like healthy and sick types. As a outcome, the network has knowledgeable additional feature demonstrations for different types of photos. The network accepts 224-by-224 image input. Figure 6 depicts the proposed MNCNN model. The illustration shows how the feeded plant leaf picture is primarily served into the BF technique to preprocess it and improve the standard. The infectiond sections of the preprocessed plant leaf are then phased using a thresholding approach. Following that, the MNCNN method utilised to produce collection of features, and the EPO algorithmic rule is used to examine the MNCNN model's hyperparameters. Finally, the extracted feature is fed into the ELM, which categorises the infections based on their appropriate class names. MobileNet to be used on mobile and embedded devices; this convolutional neural network relies on deep dissociable convolution operations, that reduce the burden of operations to be administrated within the 1st layers.

Type/Stride	Filter Shape	Input Size
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw/ s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv/s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw/s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw/s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
Conv dw/ s1 5x Conv / s1	3 x 3 x 512 dw 1 x 1 x 512 x 512	14 x 14 x 512 14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x7 x 1024
Avg pool / s1	Pool 7 x 7	7 x 7 x 1024
FC / s1	1024 x 7	1 x 1 x 1024
FC / s1	Classifier	1 x 1 x 5

Figure 6: Architecture of MobileNet CNN.

Using the MNCNN model for feature extraction, the segmentation leaf image region is served into the MNCNN model, which creates a convenient set of feature vectors. MobileNet-economic research idea that uses depth-wise dissociable convolution to create light-weight DCNN that may be used in mobile and laptop vision claims. It is the most compacted, has the simplest calculation, and obtains the highest performance. MobileNet employs a pair of hyper parameters to equalise the trade between the evaluation parameter's performance and accuracy, based on depth-wise dissociable convolution. The most common application of MobileNet is the decomposition of convolutional kernels. They employ depth-wise dissociable convolutions, regular convolution decomposition to depth-wise se convolutions, and point-wise convolutions using a single convolutional kernel. The depth-wise convolutional filter performs convolutions for each channel, and one convolution is used to desegregate the depth-wise convolutional layers' results. As a result, N regular convolutional kernels are victimised by M depth-wise convolutional kernels and N point-wise convolutional kernels. Figure a pair depicts the MobileNet design. A standard convolution filter combines the inputs to produce a new set of outputs, whereas depth-wise dissociable convolution divides the input into second layers (filtering and merging).

IV. RESULTS AND DISCUSSION

The input image is loaded and pre-processed, as seen in figure 4.1. Figure 4.2 depicts the Contrast Enhanced image of the leaf/fruit created with MATLAB. Normalization is another term for contrast stretching. It is a straightforward image enhancement technique. Stretching the range of intensity levels improves image quality.

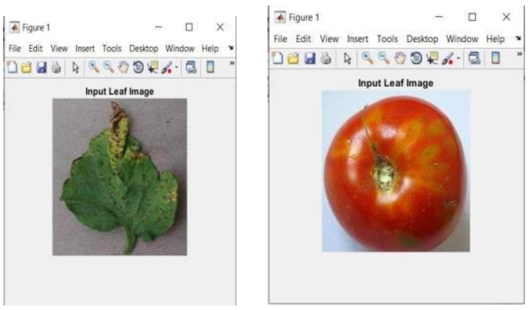


Figure 4.1: Input Image

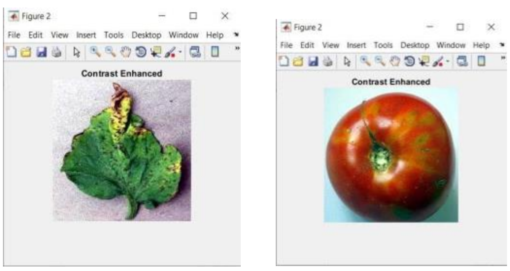
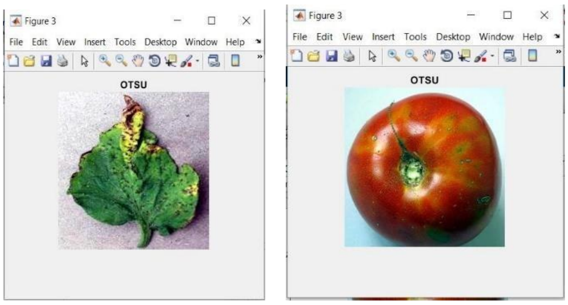


Figure 4.2: Contrast-Enhanced Image

The OTSU (based on the observed distribution of pixel values) model is used to perform image thresholding, and the HSI (Intensity, Saturation, Hue) model, which decouples the intensity factor from the color-carrying data in a colour image segmentation, were used to classify the leaf segments for detecting the status of the leaf in figures 4.3, 4.4.



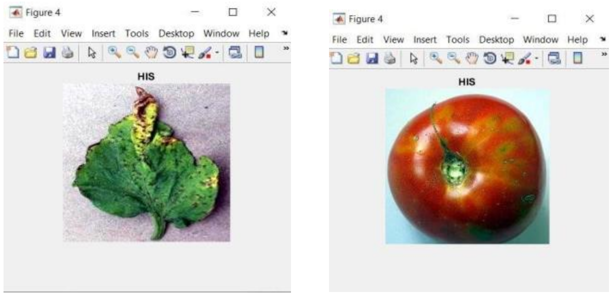


Figure 4.3: OSTU Image

Figure 4.4: HIS Image

Following that, a variety of cluster images are given in figure 4.5, which is useful for infection classification. Users must choose one of these three models to validate the segmented image state, as shown in figure 4.6.

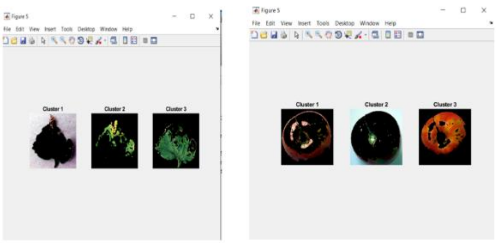
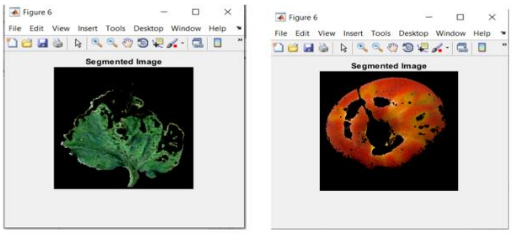
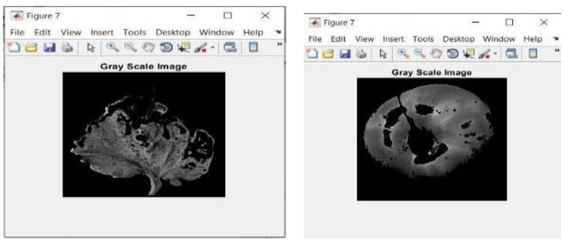



Figure 4.5: Cluster images

Figure 4.6: Segmented image

Figure 4.71 shows a grayscale image that is used to determine whether or not an image is damaged. Following all of these steps, a dialogue box is displayed as an output impacted area in percentage, indicating the condition of the input image.



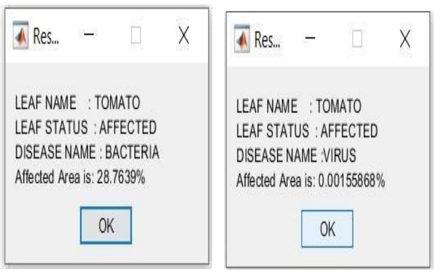


Figure 4.71: Grayscale Image

Figure 4.72: Dialog Box

Because we just teach logistical regression, the training procedure is quite fast. ResNet-50 is a CNN that has been trained on more photos in the ImageNet collection. The network has 50 deep layers in which photos can be classified into 1000 different item categories. Figure. 4.8 illustrates the training accuracy and training loss of ResNet. Figure 4.9 shows the validation accuracy vs. validation loss of ResNet.

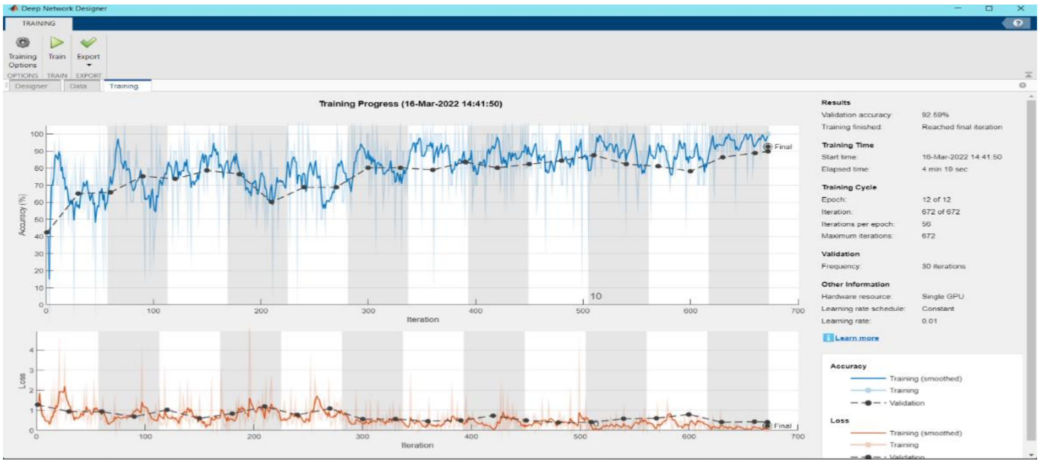


Figure 4.8: ResNet training accuracy vs. loss

Both training and validation loss descend with an increase in the iteration steps. The training and validation accuracy has considerably risen with the highness of the training iterations process. The extreme optimization challenges show that the signal is lost when it is propagated through multiple layers in the ResNet architecture. ResNet, on the other hand, consistently improves the outcomes as the depth increases. The training error for ResNet lowers with increasing depth, demonstrating the success of optimization. The acquired dataset consisted of both infectiond and healthy leaf images undergone during the training and testing process for image classification. Using a deep learning network analyser analyzes the layers of the various CNN model.

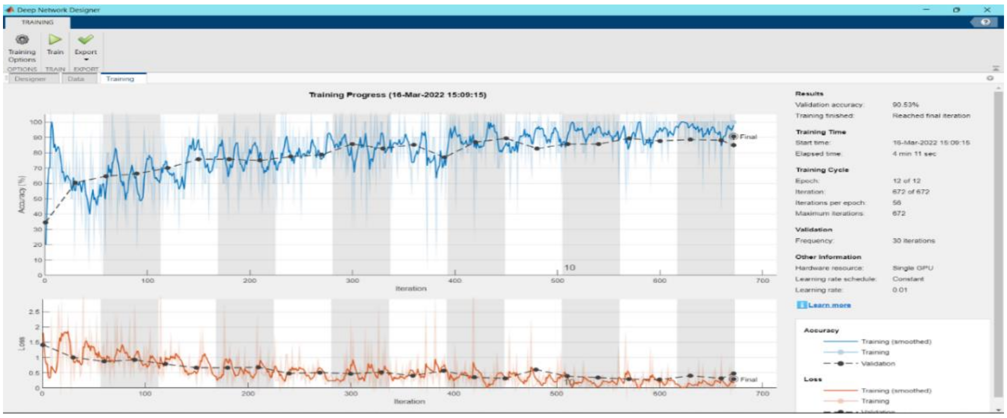


Figure 4.9: ResNet validation accuracy vs. loss

Figure 4.10 illustrates the training accuracy and training loss of MobileNet. Figure 4.11shows the validation accuracy vs. validation loss of MobileNet. The model is then iteratively trained and verified on these various sets using the validation set, where X is a fixed number. There are several approaches to this, which is generally referred to as Cross Validation. Essentially, we use the training set to create various divides of the Train and Validation sets.

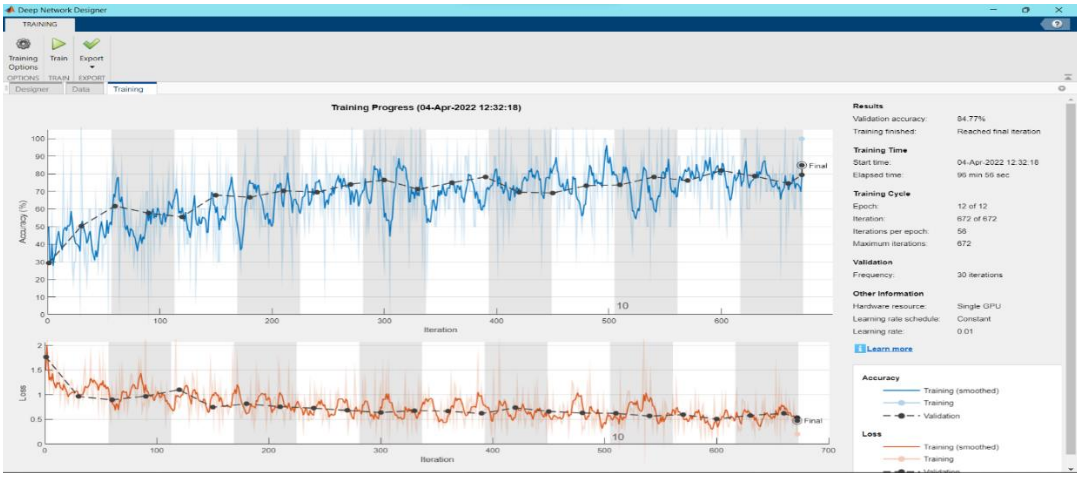


Figure 4.10 MobileNet training accuracy vs. loss

Figure 4.11: MobileNet validation accuracy vs. loss

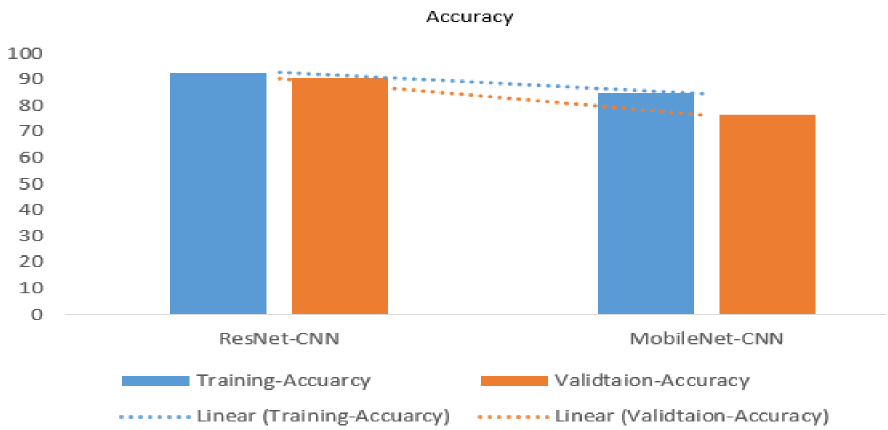
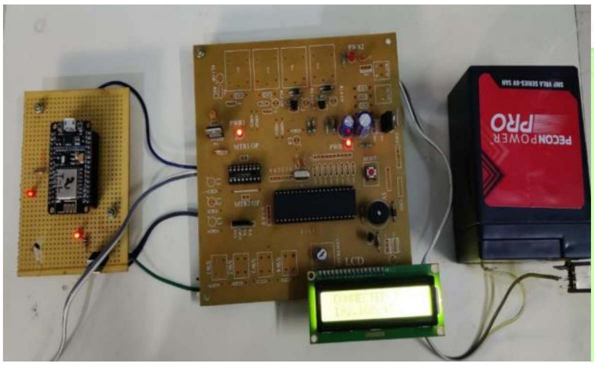


Figure 4.12: Training vs. Validation accuracy

Output for Overall Agricultural System

The processed data from the MATLAB software is interface with TTL. TTL converter is a CMOS Logic-Level Serial port connection between a host computer and a development board of PIC microcontroller. voltage regulators, rectifiers, and Filters are utilized to construct power supply circuits. A stable DC voltage is obtained by fixing the AC voltage, then filtering to a DC level, then regulate to targeted DC voltage.



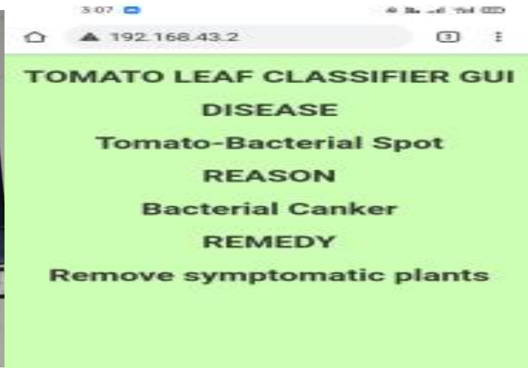


Figure 4.13: Hardware Output   Figure 4.14: Webpage Display

PIC microcontrollers are typically used to store critical data that must not be lost if the power source fails unexpectedly. Figure 4.13 shows the hardware setup, PIC microcontroller interfaces with LCD to display the output from PC to the microcontroller. Loading the data from the PIC microcontroller to the IoT cloud server by using the ESP8266 IoT module and will show the severity level and remedy for that infection as an output and it will be shown as the web page display in the mobile phone is shown in figure 4.14.

V. CONCLUSION

This proposed work has presented automated detection and classifying the tomato plant leaf infections with the help of various CNN techniques and MATLAB. An application that detects leaf and fruit infections and monitors plant infection helps farmers save time and effort. This application assists farmers in reducing their effort while increasing farm output. The proposed method enables the detection of plant infection utilising multiple neural network designs and the construction of a new model.After tracking the infection, this application helps the farmer choose an appropriate medicine for a particular infection. Finally, the analysis of the plant infection is implemented and processed in MATLAB.

VI. REFERENCES

1. Bhushan V. Patil and Pravin S. Patil, “Computational Method for Cotton Plant Infection Detection of Crop Management Using Deep Learning and Internet of Things Platforms”, Lecture Notes on Data Engineering and Communications Technologies, page no. 875-885, Springer (2021).
2. Junde Chena, Jinxiu Chena, Defu Zhanga, Yuandong Sun, "Using deep transfer learning for image-based plant infection identification”, Computers and Electronics in Agriculture, Published by Elsevier, (2020).
3. Jalal Sadoon Hameed Al-bayati, Burak Berk Ustundag (2020), “Evolutionary Feature Optimization for Plant Leaf Infection Detection by Deep Neural Networks”, International Journal of Computational Intelligence Systems Vol. 13(1), pp. 12–23, (2020).
4. Parul Sharma, Yash Paul Singh Berwal, Wiqas Ghai, “Performance analysis of deep learning CNN models for infection detection in plants using image segmentation”,

Information Processing in Agriculture, page no. 566-574, Elsevier, (2019).

5. Qimei Wang, Feng Qi, Minghe Sun, Jianhua Qu, "Identification of Tomato Infection Types and Detection of Infected Areas Based on Deep Convolutional Neural Networks and Object Detection Techniques", Computational Intelligence and Neuroscience, Volume 2019, Article ID 9142753.
6. Disha garg, Samiya Khan and Manshf Alam , "Integrative Use of IoT and Deep Learning for Agricultural Applications", Proceedings of ICETIT 2019 (pp.521-531)
7. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks", Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 1, page no.91-99, 2017.
8. Nidhi Kundu, Geeta Rani and Vijaypal Singh Dhaka "IoT and Interpretable Machine Learning Based Framework for Infection Prediction in Pearl Millet", MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations, MDPI Article, 2021.
9. Ravesa Akhter, Shabir Ahamed sofi, "Precision agriculture using IoT data analytics and machine learning", Journal of King Saud Univerisity-Computer and Information Science, 2021.
10. Tanha Talavia Dhara Shah and Nivetia Patel, "Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides", Artificial Intelligence in Agriculture, 2020.