1. INTRODUCTION

1.1 BACKGROUND

Modern technologies have given human society the ability to produce enough food to meet the demand of more than 7 billion people. However, food security remains threatened by a number of factors including climate change, the decline in pollinators, plant diseases (Strange and Scott,2005), and others. Plant diseases are not only a threat to food security at the global scale, but can also have disastrous consequences for smallholder farmers whose livelihoods depend on healthy crops. In the developing world, more than 80 percent of the agricultural production is generated by smallholder farmers (UNEP, 2013), and reports of yield loss of more than 50% due to pests and diseases are common. Furthermore, the largest fraction of hungry people (50%) live in smallholder farming households (Sanchez and Swaminathan, 2005), making smallholder farmers a group that's particularly vulnerable to pathogen-derived disruptions in food supply.

It is crucial to prevent unnecessary waste of financial and other resources, thus achieving healthier production in this changing environment, appropriate and timely disease identification including early prevention has never been more important[1]. Various efforts have been developed to prevent crop loss due to diseases. Historical approaches of widespread application of pesticides have in the past decade increasingly been supplemented by integrated pest management (IPM) approaches. Independent of the approach, identifying a disease correctly when it first appears is a crucial step for efficient disease management. Historically, disease identification has been supported by agricultural extension organizations or other institutions, such as local plant clinics. In more recent times, such efforts have additionally been supported by providing information for disease diagnosis online, leveraging the increasing Internet penetration worldwide. Even more recently, tools based on mobile phones have proliferated, taking advantage of the historically unparalleled rapid uptake of mobile phone technology in all parts of the world .

Smartphones in particular offer very novel approaches to help identify diseases because of their computing power, high-resolution displays, and extensive built-in sets of accessories, such as advanced HD cameras. It is widely estimated that there will be between 5 and 6 billion smartphones on the globe by 2020. The combined factors of widespread smartphone penetration, HD cameras, and high performance processors in mobile devices lead to a situation where disease diagnosis based on automated image recognition, if technically feasible, can be made available at an unprecedented scale. Here, we demonstrate the technical feasibility using a deep learning approach utilizing 16,012 images of Tomato species with 9 diseases (or healthy) made openly available through the project PlantVillage (Hughes and Salathé, 2015).

1.2 PROBLEM DEFINITION

Crop diseases are a major threat to food security, but their rapid identification remains difficult in many parts of the world due to the lack of the necessary infrastructure. The combination of increasing global smartphone penetration and recent advances in computer vision made possible by deep learning has paved the way for smartphone-assisted disease diagnosis.

1.3 OBJECTIVES

1. Developing a model that can be used by developer to create smartphones application to detect plant diseases.

2. Accurate and instant diagnosis of particular disease.

3. To provide remedy for disease that is detected.

1.4 SUMMARY

India is a cultivated country and about 70% of the population depends on agriculture. Farmers have large range of diversity for selecting various suitable crops and finding the suitable pesticides for plant. Disease on plant leads to the significant reduction in both the quality and quantity of agricultural products. The studies of plant disease refer to the studies of visually observable patterns on the plants. Monitoring of health and disease on plant plays an important role in successful cultivation of crops in the farm. In early days, the monitoring and analysis of plant diseases were done manually by the expertise person in that field. This requires tremendous amount of work and also requires excessive processing time. The image processing techniques can be used in the plant disease detection. In most of the cases disease symptoms are seen on the leaves, stem and fruit. The plant leaf for the detection of disease is considered which shows the disease symptoms. This project gives the introduction to image processing technique used for plant disease detection.

2. LITERATURE SURVEY

2.1 INTRODUCTION

- Plant disease cause periodic outbreak of diseases which leads to large scale death and famine. In India no estimation has been made but it is more than USA because the preventive steps taken to protect our crops are not even one-tenth of that in USA.
- Losses in million dollars due to epidemics, control becomes difficult due to spreading, threat to food security.
- Eg. epidemics of rice plant in Georgia, Coffee rust epidemic, Banana plat epidemic, Grape powdery epidemic.
- Farmers in rural India have minimal access to agricultural experts, who can inspect crop images and render advice. Delayed expert responses to queries often reach farmers too late.

Ref No.	Title	Technology /Algorithm	Details	Limitation/Future Scope
1	A Fungus Spores Dataset and a Convolutional Neural Networks based Approach for Fungus Detection	Convolutional neural network, deep learning	In this project they used convolutional neural network(CNN)approach. convolutional neural network (CNN) provided state of the art results in many other applications of object detection and classification**.** Optical sensor system was utilized to obtain images. As a result, 40,800 labeled images were used to develop fungus dataset to aid in precise fungus detection and classification. The other main objective of this research was to develop a CNN based approach for the detection of fungus and distinguish different types of fungus. A CNN architecture was designed and it showed the promising results with an accuracy of 94.8%.	This project require proper image acquisition setup for capturing images. So it is not flexible for everyone. We can developed this project by using web based application or mobile application.
2	Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks	Deep learning, convolutional neural networks,image processing,	In this project they used the two different model of convolutional neural network i.e GoogLeNet and Cifar10.To improve the identification accuracy of maize leaf diseases and reduce the number of network parameters, the improved GoogLeNet and Cifar10 models based on deep learning are proposed for leaf disease recognition in this paper. Two improved models that are used to train and test nine kinds of maize leaf images are obtained by adjusting the parameters, changing the pooling combinations, adding dropout operations and rectified linear unit functions, and reducing the number of classiffers. In addition, the number of parameters of the improved models is significantly smaller	These research processing steps are more complex and will introduce unnecessary interference at each step. The method proposed in this paper can directly take the image of the dataset as the input of the convolutional neural networks and let it learn and adjust itself to achieve an

			than that of theVGG and AlexNet structures. During the recognition of eight kinds of maize leaf diseases, the GoogLeNet model achieves a top - 1 average identi_cation accuracy of 98.9%, and the Cifar10 model achieves an average accuracy of 98.8%. The improved methods are possibly improved the accuracy of maize leaf disease, and reduced the convergence iterations, which can effectively improve the model training and recognition efficiency.	effective recognition effect. The recognition accuracy and loss are also in a more satisfactory range, and the training and recognition efficiency has been improved. Meanwhile, in order to enable agricultural producers to make quick and reasonable judgments about crop disease information, the trained model can be combined with mobile devices in a flexible manner.
3	Diagnosis of Diseases on Cotton Leaves Using Principal Component Analysis Classifier	Principle Component Analysis (PCA), Nearest Neighbourhood Classifier (KNN)	In this paper the main emphasis is given to find out mostly occurring diseases on the cotton leaves. Occurrence of diseases on the cotton plant is reflected mainly by its leaves.By using green channel information of RGB image for extracting features because cotton leaves show early symptom of diseases. By choosing appropriate classifier technique like PCA will provide best results to detect the various diseases	It was found that similar pattern diseases are having more cosines distances during KNN classification due to which there will be chance of misclassification

			on leaves of cotton in early stages. The main goal of PCA is to extract the important features, which may vary accordingly with respect to diseases. These features are considered as a set of orthogonal variables i.e. principal components. The PCA therefore includes (i) extraction of the most significant features from database; (ii) compress the size of the data set by keeping only this significant information; (iii) simplify the description of the data set. Principal components are expressed as linear combinations of the original variables. The frequency of detected diseases on cotton leaves are 28% .	some diseases are having similarities in their color patterns due to which disease patterns are not well recognized. In future, they will design more robust Classifier considering features like texture, leaf shape.
4	Classification of Watermelon Leaf Diseases Using Neural Network Analysis	Neural Network Pattern Recognition, Statistical Package for the Social Sciences (SPSS)	This paper mainly discussed the process to classify *Anthracnose* and *Downey Mildew*, watermelon leaf diseases using neural network analysis. A few of infected leaf samples were collected and they were captured using a digital camera with specific calibration procedure under controlled environment. The classification on the watermelon's leaf diseases is based on color feature extraction from RGB color model where the RGB pixel color indices have been extracted from the identified Regions of Interest (ROI). The proposed automated classification model involved the process of diseases classification using Statistical Package for the Social Sciences(SPSS) and Neural Network Pattern Recognition Toolbox in MATLAB. Determinations in this work have shown that the type of leaf diseases achieved 75.9% of accuracy	In order to improve the effectiveness of this classification system for the watermelon leaf diseases, it is recommended to use a high pixel of digital camera to get the best images. Also recommended to increase the number of data for the training and testing to get the best result. In addition, the lighting setup must be in

			based on its RGB mean color component.	proper position because it also can affect the image captured. Besides that, the other color model can be used as the input in order to increase the efficiency such as Cyan, Magenta, Yellow (CMY), Hue Saturation Value (HSV)and Hue Saturation Lightness (HSL).
5	Classification of Rubber Tree Leaf Diseases Using Multilayer Perceptron Neural Network	Artificial neural network(ANN), Principal component analysis(PCA)	This paper presents about classification of rubber leaf diseases through automation and utilizing primary RGB color model. Several rubber tree leaf diseases are been studied for digital RGB color extraction where three sets of rubber tree leaf diseases image are digitally captured under standard and control environment. The identified regions of interest (ROI) these diseases images are then processed to quantify the normalized indices from the RGB color distribution. This system involved the process of image classification by using artificial neural network where 600 samples were used as training while another 200 samples were for testing. The optimized ANN model in this work has two method which based	In order to increase the effectiveness of this classification system, it is recommended to use a high pixel of digital camera for higher accuracy of capturing color images and providing better processor and computer specification since color resolution and processing time

			only on the dominant pixel RGB (mean) and applying principle component analysis (PCA) on the pixel gradation values of each image.	totally relies on processor used. In addition other models can be considered at input for better efficiency.
6	Leaf Disease Detection and Classification using Neural Networks	Artificial Neural Network, Back Propagation Neural Network, Support Vector Machine.	This paper proposes an approach for leaf disease detection and classification on plants using image processing. The algorithm presented has three basic steps: Image Pre-processing and analysis, Feature Extraction and Recognition of plant disease. The plant disease diagnosis is restricted by person's visual capabilities as it is microscopic in nature. Due to optical nature of plant monitoring task, computer visualization methods are adopted in plant disease recognition. The aim is to detect the symptoms of the disease occurred in leaves in an accurate way. Once the captured image is pre-processed, the various properties of the plant leaf such as intensity, color and size are extracted and sent to SVM classifier with Back propagation Neural Network for classification. For classification between the affected leaves, classifiers depend upon the Bayes' theorem and SVM were used for classification and differences between the affected leaves The experimental results obtained using 169 images have shown that the classification accuracy by ANN ranges between 88% and 92%.	Accuracy is improved by the use of different image processing techniques such as image analysis, pre-processing, feature extraction and classification. Speed and accuracy are the two main characteristics of plant disease detection using machine learning method that must be achieved Using the proposed method, the accuracy up to 92% can be achieved. In future accuracy of detection can be increased when using SVM classifier with more number of

				features.
7	Plant Disease Detection using CNN & Remedy	CNN	The proposed system helps in identification of plant disease and provides remedies that can be used as a defense mechanism against the disease. We use Convolution Neural Network(CNN) which comprises of different layers which are used for prediction. A prototype drone model is also designed which can be used for live coverage of large agricultural fields to which a high resolution camera is attached and will capture images of the plants which will act as input for the software, based of which the software will tell us whether the plant is healthy or not. With our code and training model we have achieved an accuracy level of 78% .Our software gives us the name of the plant species with its confidence level and also the remedy that can be taken as a cure.	The proposed system is based on python and gives an accuracy of around 78%. The accuracy and the speed can be increased by use of Googles GPU for processing . The system can be installed on Drones so that aerial survaillances of crop fields can be done.
8	Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification	CNN	The latest generation of convolutional neural networks (CNNs) has achieved impressive results in the field of image classification. This paper is concerned with a new approach to the development of plant disease recognition model, based on leaf image classification, by the use of deep convolutional networks.Novel way of training and themethodology used facilitate a quick and easy system implementation in practice.The developed model is able to recognize 13 different types of plant diseases out of healthy leaves,with the ability to distinguish plant leaves	The main goal for the future work will be developing a complete system consisting of server side components containing a trained model and an application for smart mobile devices with features such as displaying recognized diseases

			from their surroundings. Caffe, a deep learning framework developed by Berkley Vision and Learning Centre, was used to perform the deep CNN training. The experimental results on the developed model achieved precision between 91% and 98%, for separate class tests, on average 96.3%.	in fruits, vegetables, and other plants, based on leaf images captured by the mobile phone camera. This application will serve as an aid to farmers (regardless of the level of experience), enabling fast and efficient recognition of plant diseases and facilitating the decision-making process when it comes to the use of chemical pesticides.
9	Using Deep Learning for Image Based Plant Disease Detection	Machine learning ,Deep learning, CNN.	This paper proposes an approach for leaf disease detection and classification on plants using image processing. Using a public dataset of 54,306 images of diseased and healthy plant leaves collected under controlled conditions,we train a deep convolutional neural network to identify 14 crops pecies and 26 diseases.The trained model achieves an accuracy of 99.35% on a held-out testset ,demonst rating the feasibility of this approach. Overall, the approach of training deep learning model son increasingly large and	There are a number of limitations.First,when tested on a set of images taken under conditions different from the images used for training, the model's accuracy is reduced substantially, to just above 31%. The second

			publicly available image datasets presents a clear path toward smartphone-assisted crop disease diagnosis on a massive global scale.	limitation is that we are currently constrained to the classification of single leaves, facing up, on a homogeneous background.While these are straightforward conditions, a real world application should be able to classify images of a disease as it presents itself directly on the plant. Indeed, many diseases don't present themselves on the upper side of leaves only (or at all), but on many different parts of the plant. Thus, new image collection efforts should try to obtain images from many different perspectives, and ideally from settings that are as realistic as possible.

10	Plant Leaf Disease Detection using Deep Learning and Convolutional Neural Network	CNN, Deep learning, Image processing	This paper is concerned with a new approach to the development of plant disease recognition model, based on leaf image classification, by the use of deep convolutional networks. All essential steps required for implementing this disease recognition model are fully described throughout the paper, starting from gathering images in order to create a database, assessed by agricultural experts, a deep learning framework to perform the deep CNN training. This method paper is a new approach in detecting plant diseases using the deep convolutional neural network trained and fine-tuned to fit accurately to the database of a plant's leaves that was gathered independently for diverse plant diseases. The advance and novelty of the developed model lie in its simplicity; healthy leaves and background images are in line with other classes, enabling the model to distinguish between diseased leaves and healthy ones or from the environment by using deep CNN.	The main goal for the future work will be developing a complete system consisting of server side components containing a trained model and an application for smart mobile devices with features such as displaying recognized diseases in fruits, vegetables, and other plants, based on leaf images captured by the mobile phone camera. This application will serve as an aid to farmers (regardless of the level of experience), enabling fast and efficient recognition of plant diseases and facilitating the decision-making process when it comes to the use of chemical pesticides.

				Furthermore, future work will involve spreading the usage of the model by training it for plant disease recognition on wider land areas, combining aerial photos of orchards and vineyards captured by drones and convolution neural networks for object detection.

3. DESIGN AND DRAWING

3.1 FLOW DIAGRAM

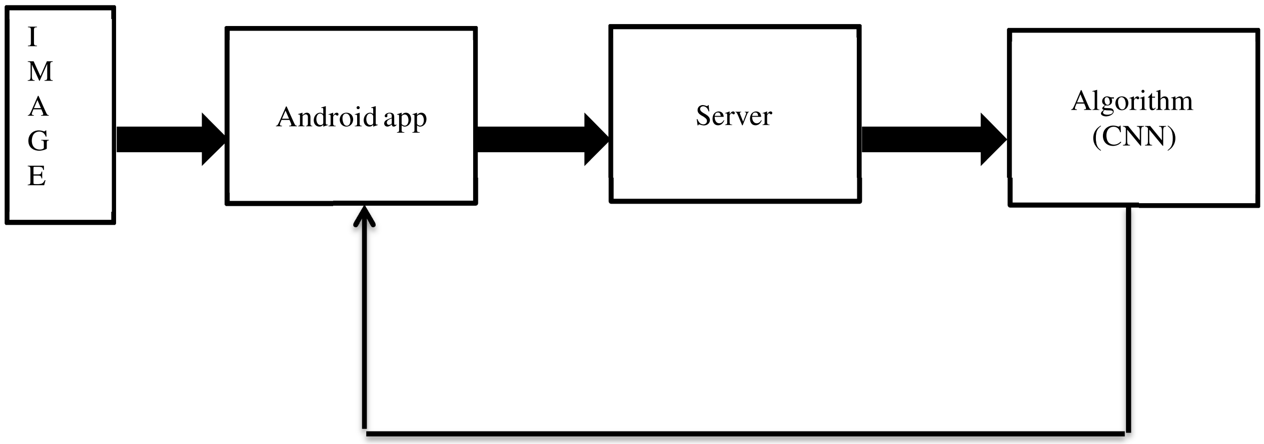
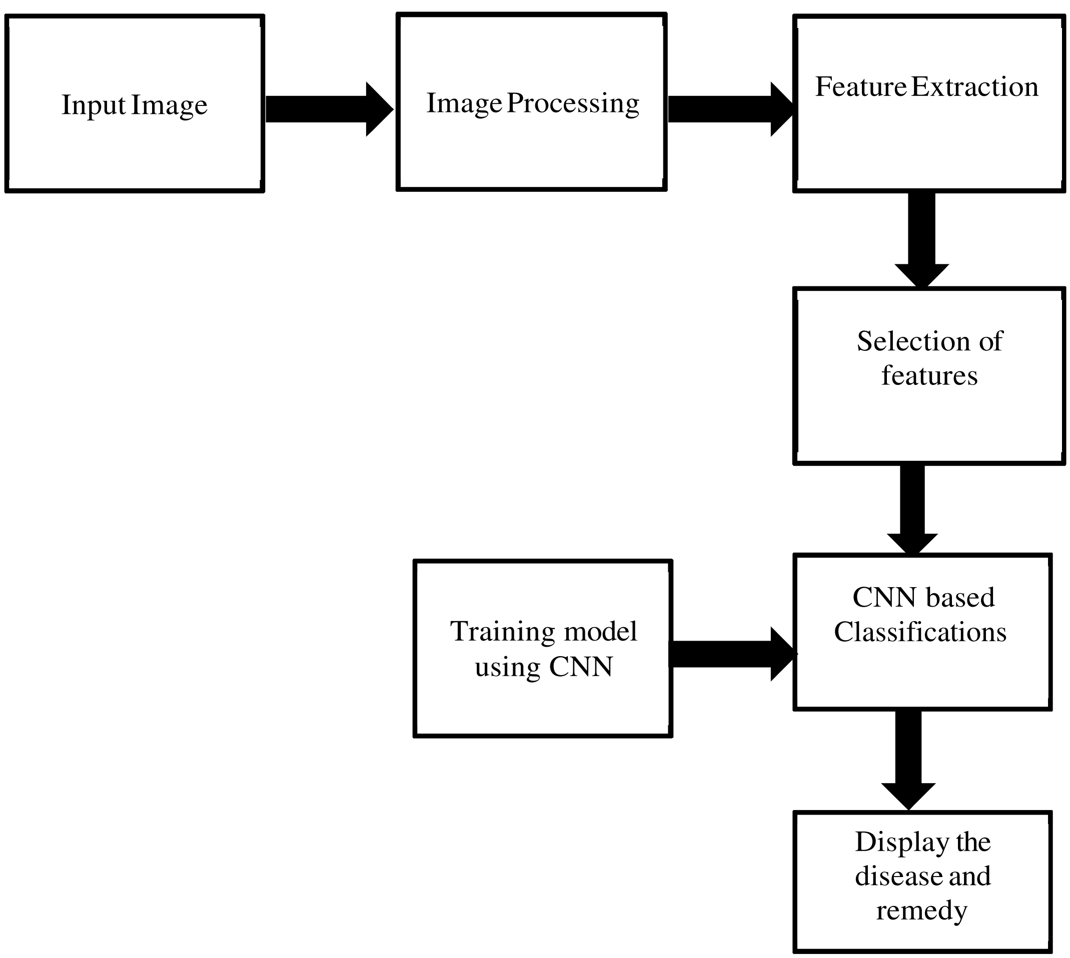


FIG 3.1 FLOW DIAGRAM

3.2 BLOCK DIAGRAM



3.3 BLOCK DIAGRAM DESCRIPTION

- Dataset collection: we have collected dataset from plantvillage. We analyze 16,012 images of tomato leaves. Each class label is a crop-disease pair, and we make an attempt to predict the crop-disease pair given just the image of the plant leaf. In all the approaches described in this report, we resize the images to 256 × 256 pixels, and we perform both the model optimization and predictions on these downscaled images.

- We start with the PlantVillage dataset as it is, in color; then we experiment with a gray-scaled version of the PlantVillage dataset, and the image goes through number of processing steps like preprocessing,feature extraction, selection of feature.

- In preprocessing, noise and distortion is removed. finally we run all the experiments on a version of the PlantVillage dataset where the leaves were segmented, hence removing all the extra background information which might have the potential to introduce some inherent bias in the dataset due to the regularized process of data collection in case of PlantVillage dataset. Segmentation was automated by the means of a script tuned to perform well on our particular dataset. We chose a technique based on a set of masks generated by analysis of the color, lightness and saturation components of different parts of the images in several color spaces (Lab and HSB). One of the steps of that processing also allowed us to easily fix color casts, which happened to be very strong in some of the subsets of the dataset, thus removing another potential bias.
- The model is properly trained using CNN and classification take placed. CNN algorithm contains steps such as convolution, pooling, ReLU, fully connected layer.

- The comparison of test image and trained model takes place. This set of experiments was designed to understand if the neural network actually learns the "notion" of plant diseases, or if it is just learning the inherent biases in the dataset, followed by display of the result.

- If there is detect or disease in the plant software display disease along with remedy.

3.4 METHODS

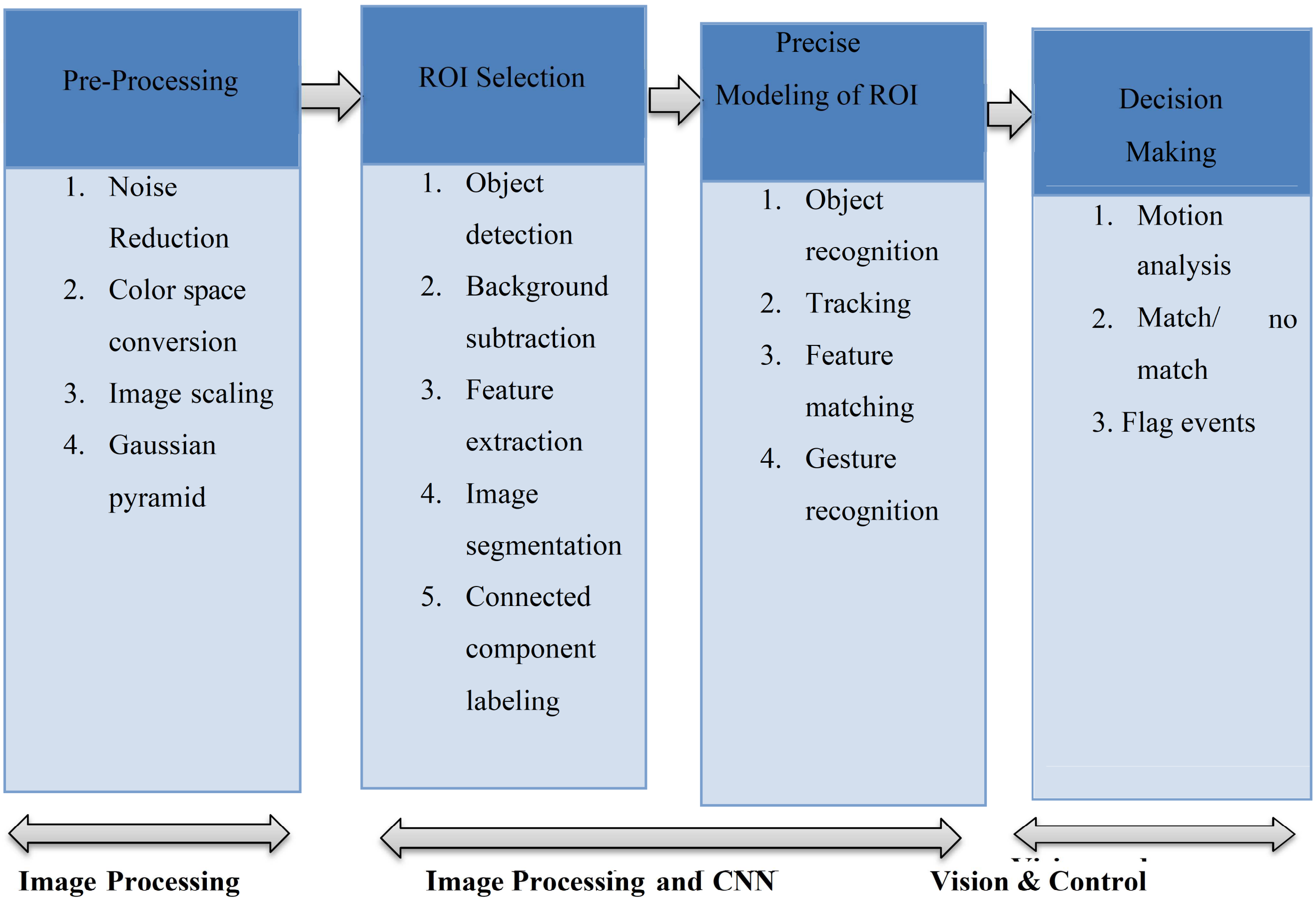


Fig 3.2 Vision algorithm pipeline

Image Pre-Processing: Pre-processing is a common name for operations with images at the lowest level of abstraction -- both input and output are intensity images. The aim of pre-processing is noise reduction, an improvement of the image data that suppresses unwanted distortions or enhances some image features important for further processing. Some of the point processing techniques include: contrast stretching, global thresholding, histogram equalisation, log transformations and power law transformations. Some mask processing techniques include averaging filters, sharpening filters, local thresholding etc.

ROI Selection: It is sometimes of interest to process a single subregion of an image, leaving other regions unchanged. This is commonly referred to as region-of-interest (ROI) processing. Image sub regions may be conveniently specified by using Mathematica Graphics primitives, such as Point, Line, Circle, Polygon, or simply as a list of vertex positions. A region of interest (ROI) is a portion of an image that you want to filter or perform some other operation on. You define an ROI by creating a binary mask, which is a binary image that is the same size as the image you want to process with pixels that define the ROI set to 1 and all other pixels set to 0.More than one ROI in an image can be defined. The regions can be geographic in nature, such as polygons that encompass contiguous pixels, or they can be defined by a range of intensities. In the latter case, the pixels are not necessarily contiguous. Image segmentation technique is used to partition an image into meaningful parts having similar features and properties. The main aim of segmentation is simplification i.e. representing an image into meaningful and easily analyzable way. Image segmentation is necessary first step in image analysis. Divide an image into several parts/segments having similar features or attributes.

Modeling of ROI:

Object detection and object recognition are similar techniques for identifying objects, but they vary in their execution. Object detection is the process of finding instances of objects in images. In the case of deep learning, object detection is a subset of object recognition, where the object is not only identified but also located in an image. This allows for multiple objects to be identified and located within the same image.

Decision Making: Decision making becomes more insightful and accurate because of Deep Learning technology. There is less need for manual data aggregation and document reviewing, and more time can be spent on processing, analyzing, and acting upon the data.

3.5 Convolutional Neural Network (CNN)

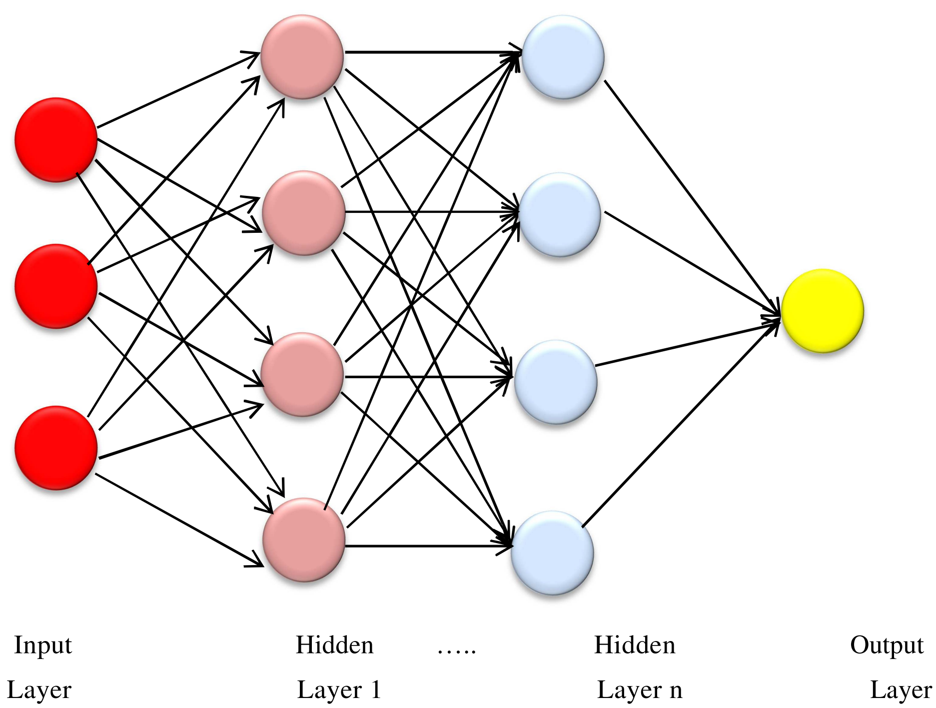


FIG 3.3 CNN ARCHITECTURE

A neural network is a system of interconnected artificial "neurons" that exchange messages between each other. The connections have numeric weights that are tuned during the training process, so that a properly trained network will respond correctly when presented with an image or pattern to recognize. The network consists of multiple layers of feature-detecting "neurons". Each layer has many neurons that respond to different combinations of inputs from the previous layers. As shown in Figure 3.4 the layers are built up so that the first layer detects a set of primitive patterns in the input, the second layer detects patterns of patterns, the third layer detects patterns of those patterns, and so on. Typical CNNs use 5 to 25 distinct layers of pattern recognition. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, ReLU, fully connected layers and normalization layers.

- **Convolutional:** Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli. Each convolutional neuron processes data only for its receptive field. Although fully connected feed forward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10000 weights for each neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using back propagation.
- **Pooling:** Convolutional networks may include local or global pooling layers, which combine the outputs of neuron clusters at one layer into a single neuron in the next layer. For example, max pooling uses the maximum value from each of a cluster of neurons at the prior layer. Another example is average pooling, which uses the average value from each of a cluster of neurons at the prior layer.

- **Non-linear layers:** Neural networks in general and CNNs in particular rely on a non-linear "trigger" function to signal distinct identification of likely features on each hidden layer. CNNs may use a variety of specific functions —such as rectified linear units (ReLUs) and continuous trigger (non-linear) functions—to efficiently implement this non-linear triggering.
- **ReLU:** A ReLU implements the function y = max(x,0), so the input and output sizes of this layer are the same. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer. In comparison to the other non-linear functions used in CNNs (e.g., hyperbolic tangent, absolute of hyperbolic tangent, and sigmoid), the advantage of a ReLU is that the network trains many times faster. ReLU functionality, with its transfer function plotted above the arrow.
- **Fully Connected:** Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). Fully connected layers are often used as the final layers of a CNN. These layers mathematically sum a weighting of the previous layer of features, indicating the precise mix of "ingredients" to determine a specific target output result. In case of a fully connected layer, all the elements of all the features of the previous layer get used in the calculation of each element of each output feature.

4. SOFTWARE REQUIREMENT SPECIFICATION

4.1 PURPOSE

The purpose of this document is to collect and analyse all associated ideas that come up to design the system. Also we will predict how this can be used to prepare the outline of the project which can be later developed and we document the ideas that can be later consider or may be discarded when project being developed. In short SRS provides the detail view of our software product. The SRS the project target audience and its user interface, hardware and software requirement. It defines product and its functionality. It also assists developer during SDLC.

4.2 FUNCTIONAL REQUIREMENT

1. **Performance**: The system should perform efficiently and effectively.

2. **Re-usability**: The system should work properly when used as a standalone system or integrated with other system.

3. **Maintainability**: The system should be maintainable. The architecture and design should be flexible for change and well documented.

4. **Scalability**: The system should be able to handle huge amount of records.

5. **Consistency**: The system should provide accurate results to the user.

4.3 NON FUNCTIONAL REQUIREMENTS

1. Interface Requirements

2. Performance Requirements the project has the following performance requirements:

-The prime requirement is that no error condition causes a project to exit abruptly

-Any error occurred in any process should return an understandable error message.

-The response should be fairly fast, the action participants should not be confused at any point of time about action that is happening.

-The system performance is adequate.

Software quality attributes the difference between an amateur product and a carrier grade product is not much in functionality; it is in Quality. For any serious business to depend on a piece of software to continue to function and evolve as needed, along list of quality attributes or abilities' are required. The list seems to be long, but each ability is vital. If you get stuck with something that doesn't have any one of the required abilities.

4.4 INTERFACES

4.4.1 Software Requirement

1. Android Studio 3.2.1

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

Features

- Gradle-based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations

- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

2. Python Programming Language

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of the Python is large library which can be used for the following

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Text processing

Version Python 3.5.9 - Nov. 2, 2019

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments, and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS, and Linux.

Use of Navigator

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages and use multiple environments to separate these different versions.

The command-line program conda is both a package manager and an environment manager. This helps data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages, and update them – all inside Navigator.

The following applications are available by default in Navigator :-

- JupyterLab
- Jupyter Notebook
- Spyder
- VSCode
- Glueviz
- Orange 3 App
- RStudio

Run Code On Navigator

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images, and interactive interfaces into a single notebook file that is edited, viewed, and used in a web browser.

Conda

Package, dependency and environment management for any language—Python, R, Ruby, Lua, Scala, Java, JavaScript, C/ C++, FORTRAN, and more.

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment.

Anaconda Cloud

Anaconda Cloud is a package management service by Anaconda. Cloud makes it easy to find, access, store and share public notebooks, environments, and conda and PyPI packages. Cloud also makes it easy to stay current with updates made to the packages and environments you are using. Cloud hosts hundreds of useful Python packages, notebooks and environments for a wide variety of applications. You do not need to log in, or even to have a Cloud account, to search for public packages, download and install them.You can build new packages using the Anaconda Client command line interface (CLI), then manually or automatically upload the packages to Cloud to quickly share with others or access yourself from anywhere.

3. Xampp server 7.3.6

XAMPP stands for Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) and Perl (P).

Version 7.3.12

Letter	Meaning
X	as an ideographic letter referring to cross-platform[5]
A	Apache[6] or its expanded form, Apache HTTP Server[5]
M	MariaDB[7] (formerly: MySQL[5][7])
P	PHP[6][5]
P	PERL[6][5]

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends,consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible. Since XAMPP is simple, lightweight Apache distribution it is extremely easy for developers to create a local web server for testing and deployment purposes. Everything you needed is to set up a web server – server application (Apache), database (MariaDB), and scripting language (PHP). XAMPP works equally well on Linux, Mac, and Windows.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. XAMPP has the ability to serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress[further explanation needed]. It is also possible to connect to localhost via FTP with an HTML editor.

4.4.2 Hardware Requirement

1. Processor: Intel Core i3

2. Speed: 1.1 GHz or Higher

3. RAM: 2 GB or Higher

4 .Hard Disk: 100 GB or Higher

5. PROJECT IMPLEMENTATION

5.1 Model

We experimented with two types of images to see how the model work and what exactly it learns, first we take the image as it is with 3 color channels, and then we experimented with 1 color channel images (Gray-Scale). And as expected the model learns different patterns in each approach.

Our model takes raw images as an input, so we used Convolutional Neural Networks (CNNs) to extract features, in result the model would consist of two parts:

- The first part of the model (features extraction), which was the same for full-color approach and gray-scale approach, it consist of 4 Convolutional layers with Relu activation function, each followed by Max Pooling layer.
- The second part after the flatten layer contains two dense layers for both approaches, but in full-color the first has 256 hidden units which makes the total number of network trainable parameters 3,601,478, in the other hand gray-scale approach has 128 hidden units in the first dense layer and 1,994,374 as total trainable parameters, we shrank the size of the gray-scale network to avoid overfitting, for the last layer for both has Softmax as activation and 6 outputs representing the 6 classes.
- We have selected 4000 tomato leaves images for training purpose and 2000 images for testing purpose.

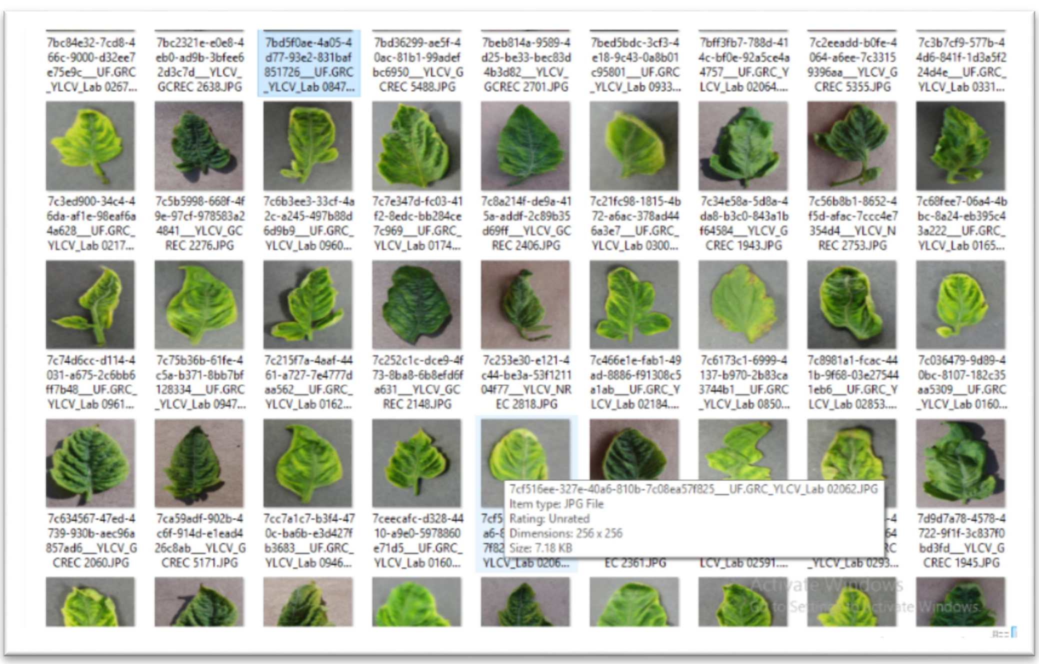


FIG 5.1 TOMATO LEAF DATASET

5.1.1 Server Side Program Flow

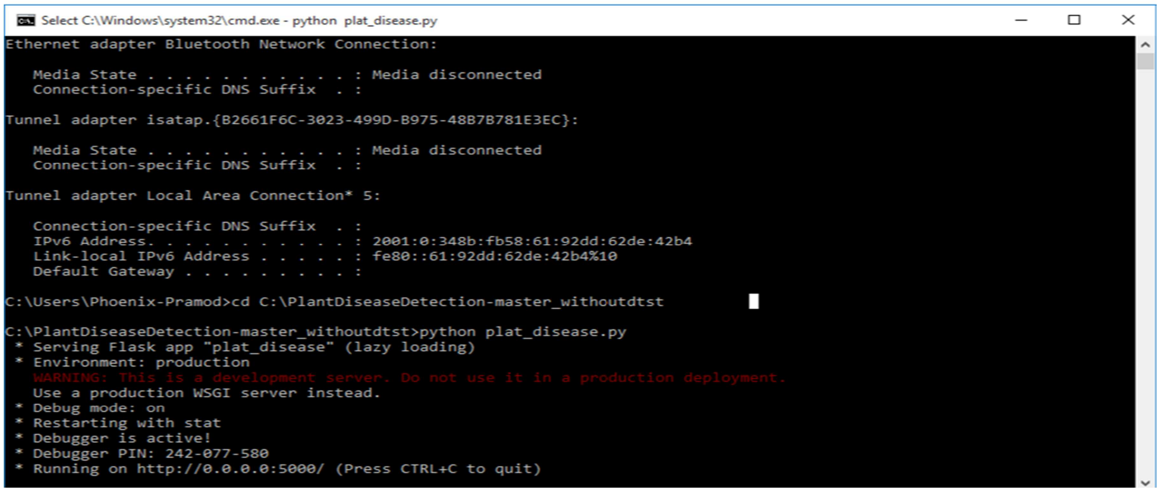


FIG 5.2 SERVER RUN CONFIGURATION

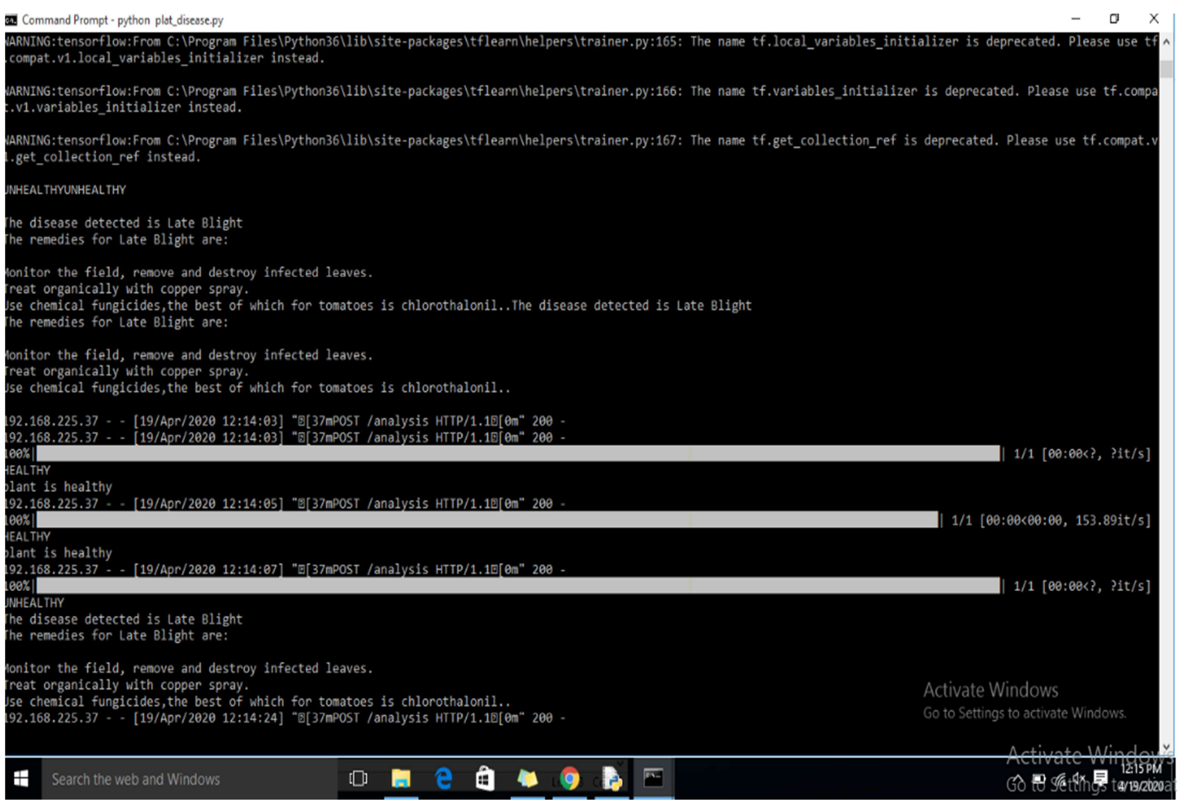


FIG 5.1 SERVER'S REPLY

6. RESULT

6.1 OUTCOME

We have selected 4000 tomato leaves images for training purpose and 2000 images for testing purpose. In image classification, CNNs outperform traditional image processing methods in several applications. According to the results, the recognition rate of our system was above 94% when using the CNN, even when 30% of the leaf was damaged. Our system therefore improves upon previous studies, which achieved a recognition rate of approximately 90%

6.2 SCREENSHOTS

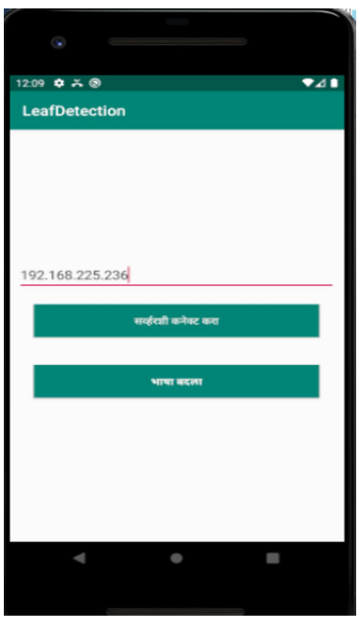


FIG 6.1 FRONT PAGE OF LEAFDETECTION

FIG 6.2 SELECTION OF LANGUAGE

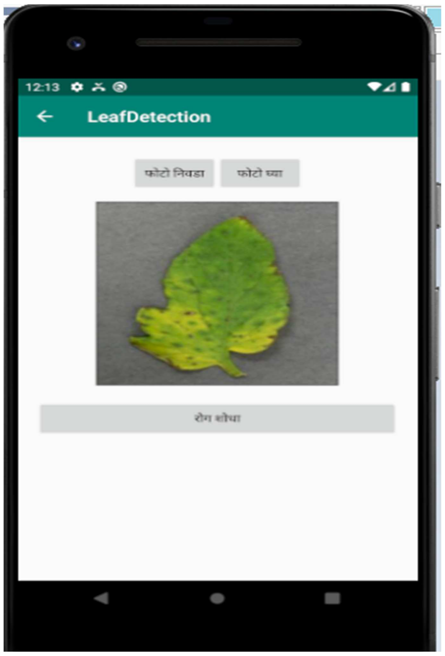


FIG 6.3 LEAF SELECTION

In our Android application anyone can click the picture of leaf using device's camera or may use already existing tomato leaves images in the device.

6.3 LEAF DISEASE DETECTION & SUGGESTION OF PESTICIDES

FIG 6.4 DISEASE DETECTION(MARATHI)

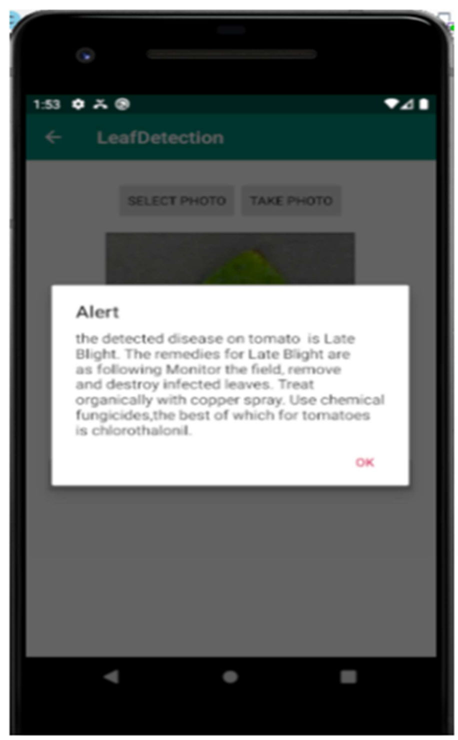


FIG 6.5 DISEASE DETECTION(ENGLISH)

7. FUTURE SCOPE

In automation or computer vision many methods are available for plant infection recognition and categorization process, but still, this research field needs attention. There is lack in commercial explanation as well as solutions in the marketplace, apart from those handling with plants species recognition based on the leaves images.

- Very few method available for disease identification during early stages of crops using deep learning methods
- For Multisperal/Hyperspectral processing, Satellite images are to be used and the main constrain is availability of hyper zoomed images for specific crop or plant.
- Multisperal/Hyperspectral data consist of a massive quantity of narrow and nearby spectral bands. Initial working is necessary on these spectral bands for spectral data analysis and modeling. Which may poses as a challenge

8. CONCLUSION

CONCLUSION

Image processing technique based approach is proposed and useful for plant disease detection. Recognizing the disease is mainly the purpose of the proposed approach that can recognize the leaf diseases with little computational effort. This proposed approach consists of 4 phases. Accuracy is improved by the use of different image processing techniques such as image analysis, pre-processing, feature extraction and classification. Speed and accuracy are the two main characteristics of plant disease detection using machine-learning methods that must be achieved.

In future research we will attempt to recognize leaves attached to branches, in order to develop a visual system that can replicate the method used by humans to identify plant types.

REFERENCES

1) A Fungus Spores Dataset and a Convolutional Neural Networks based Approach for Fungus Detection, Muhammad Waseem Tahir, Nayyer Abbas Zaidi, Adeel Akhtar Rao, Roland Blank, Michael J. Vellekoop and Walter Lang IEEE Transaction On Nanobioscience, Vol. Xx, No. Xx, May 2018

2) Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks Xihai Zhang , (Member, Ieee), Yue Qiao , Fanfeng Meng, Chengguo Fan , And Mingming Zhang ,School of Electronic Engineering and Information, Northeast Agricultural University, Harbin 150030, China

3) Diagnosis of Diseases on Cotton Leaves Using Principal Component Analysis Classifier, Viraj A. Gulhan ,Maheshkumar H. Kolekar,2014 Annual IEEE India Conference (INDICON)

4) Classification of Watermelon Leaf Diseases Using Neural Network Analysis Suhaili Beeran Kutty, Noor Ezan Abdullah, Dr. Hadzli Hashim, A'zraa Afhzan Ab Rahim, Aida Sulinda Kusim,Tuan Norjihan Tuan Yaakub, Puteri Nor Ashikin Megat Yunus, Mohd Fauzi Abd Rahman Faculty of Electrical Engineering, Universiti Teknologi Mara, Shah Alam, Selangor

5) Classification of Rubber Tree Leaf Diseases Using Multilayer Perceptron Neural Network, Noor Ezan Abdullah, Athirah A. Rahim, Hadzli Hashim and Mahanijah Md Kamal, The 5tn Student Conference on Research and Development -SCOReD 2007,11-12 December 2007, Malaysia

6) Leaf Disease Detection and Classification using Neural Networks,V. Ramya1, M. Anthuvan Lydia2 Assistant Professor, Department of ECE, Saranathan College of Engineering, Trichy, India1,2,**ISO 3297:2007 Certified** Vol. 5, Issue 11, November 2016

7) Plant Disease Detection using CNN & Remedy,Adnan Mushtaq Ali Karol1, Drushti Gulhane2, Tejal Chandiwade3,B.E. Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India1,B.E. Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India2,B.E. Student, Dept. of EXTC, Rajiv Gandhi Institute of Technology, Mumbai, India3

8) Deep Neural Networks Based Recognition of,Plant Diseases by Leaf Image Classification Srdjan Sladojevic,1 Marko Arsenovic,1 Andras Anderla,1,Dubravko Culibrk,2 and Darko

Stefanovic1,Department of Industrial Engineering and Management, Faculty of Technical Sciences, University of Novi Sad,Trg Dositeja Obradovica 6, 21000Novi Sad, Serbia,Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 9, Povo, 38123 Trento, Italy

9)UsingDeepLearningforImage-BasedPlantDiseaseDetection
Sharada P.Mohanty1, 2,3, DavidP.Hughes4, 5,6 and MarcelSalathé 1, 2,3* 1 Digital EpidemiologyLab,EPFL,Geneva,Switzerland.

10) Plant Leaf Disease Detection using Deep Learning and Convolutional Neural Network Anandhakrishnan MG Joel Hanson1, Annette Joy2, Jerin Francis3 ,Department of Computer Science Engineering ,SCET, India