VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI, KARNATAKA-590018

A Project Dissertation on

"Crofting Hybrid Robot"

Submitted in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science & Engineering during the year 2021-2022

By

Archana K	**4MN18CS004**
Geetha R S	**4MN18CS012**
Meghana C	**4MN18CS022**

Under the guidance of

Dr. Ranjit K N
Associate Professor
Dept of CS&E,
MITThandavapura

2021-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY, THANDAVAPURA

JUST OFF NH 766, NANJANAGUDU TALUK, MYSORE DISTRICT - 571302

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MAHARAJA INSTITUTE OF TECHNOLOGY THANDAVAPURA

MYSORE-571302

CERTIFICATE

This is to certified that the project work titled "**Crofting Hybrid Robot"** has been successfully carried out by **ARCHANA K [4MN18CS004], GEETHA R S [4MN18CS012], MEGHANA C [4MN18CS022]** bonafide students of **Maharaja Institute of Technology Thandavapura in partial fulfilment of requirements of Degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum** during the academic year 2021-22. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report deposited in the Department library. The project report has been approved has it satisfies the academic requirements with respect to the project work prescribed for Bachelor of Engineering Degree.

Signature of guide	Signature of the HOD	Signature of the Principal
Dr. Ranjit K N	**Dr. Ranjit K N**	**Dr. Y. T. Krishne Gowda**
Associate Professor	Associate Professor	Principal
Dept. of CS&E,	HoD of CS&E	MIT Thandavapura
MIT Thandavapura	MIT Thandavapura	

External Viva

Name of the Examiners **Signature with date**

1. ______________________________

2. ______________________________

ACKNOWLEDGEMENT

It gives us immense pleasure to bring our Project report entitled **"Crofting Hybrid Robot"** in the Eighth Semester Engineering Course.

We are thankful to **Dr. Y T Krishne Gowda, Principal, MITT** for having supported us in our academic endeavours.

We would like to extend our thanks to **Dr. H K Chethan, Professor and Mentor, CS&E, MITT**, **Dr. Ranjit K N, Associate Professor and Head, Department of CS&E, MITT** for providing us timely suggestions, encouragement and support to complete this work.

We would like to sincerely thank our project guide **Dr. Ranjit K N, Associate Professor, Dept. of CSE, MITT** for providing relevant information, valuable guidance and encouragement to complete this report.

We would also like to thank all our teaching and non-teaching staff members of the Department for their support. We always thankful to our parents for their valuable support and guidance in every step.

We express our deepest gratitude and indebted thanks to **MITT** which has provided us an opportunity in fulfilling our most cherished desire of reaching the goal.

Archana K 4MN18CS004

Geetha R S 4MN18CS012

Meghana C 4MN18CS022

ABSTRACT

Our project focuses on detection of N, P and K contents, soil by using sensors and also monitor the temperature and sunlight in the agricultural field. India is such a country which has capacity to produce three crops in a year. Indian agriculture is characterized by agriculture ecological diversities in soil, rainfall, temperature, and cropping system. Indian agricultural productivity is very less compared to world standards due to, use of absolute farming technology. Also due to, lack of understanding of the need for sustainability in the poor farming community has made things worse. Fertilizer has been the key input in augmenting food production in India. However, fertilizer use in India is skewed, high in a few states having adequate irrigation & dismally low in the NE states. There is also imbalanced use of N, P, and K. Due to the imbalanced use of plant nutrients is considered as the main cause for decline in crop yield and crop response ratio.

The proposed system helps in identification of plant disease and provides remedies that can be used as a defence mechanism against the disease. The database obtained from the Internet is properly segregated and the different plant species are identified and are renamed to form a proper database then obtain test-database which consists of various plant diseases that are used for checking the accuracy and confidence level of the project. Then using training data. We will train our classifier and then output will be predicted with optimum accuracy.

We use Convolution Neural Network (CNN) which comprises of different layers which are used for prediction. A prototype rover 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.

LIST OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

India is mainly an agricultural country. Agriculture is the most important occupation for most of the Indian families. Over 60% of India's land area is arable making it the second largest country in terms of total arable land. Most of the farming in India is monsoon dependent–if monsoon is good, the entire economy is upbeat and when the monsoon fails, everyone everywhere takes a hit to some extent. Green revolution began in India with an objective to give greater emphasis on Agriculture. Significant increase in the production of food crops, the productivity of land increased tremendously giving huge economic boost to the nation Irrigation which consumes more than 80% of the total water use in the country needs a proper overall if the country has to improve agricultural output and boost the overall economy. Irrigation water is becoming a scarce commodity. Thus, proper harvesting and efficient utilization of water is of great importance.

The primary occupation in India is agriculture. India ranks second in the agricultural output worldwide. Here in India, farmers cultivate a great diversity of crops. Various factors such as climatic conditions, soil conditions, various disease, etc. affect the production of the crops. The existing method for plants disease detection is simply naked eye observation which requires more man labor, properly equipped laboratories, expensive devices, etc. And improper disease detection may lead to inexperienced pesticide usage that can cause development of long-term resistance of the pathogens, reducing the ability of the crop to fight back. The plant disease detection can be done by observing the spot on the leaves of the affected plant. The method we are adopting to detect plant diseases is image processing and using Convolution neural network to detect plant disease.

1.2 Overview with Problem Identification

There is no automated version to detect plant disease and to measure NPK of soil. To determine the amount of NPK [N-nitrogen, P-phosphorus, K-potassium] in soil and can give suggestions to improve soil fertility.

Farmers face difficult in identifying if any kind of disease infects the plants and also adding adequate and type of fertilizers to soil is crucial. Finding the amount of NPK in soil requires time consuming process of collecting soil sample and taking it to nearest city LAB to get tested there is a requirement for a modern robot which can do both the task of finding the plant disease and also finding the NPK conversation of soil.

1.3 Objective

To build system which can detect plant disease using machine learning, image processing and can give the information of what are the precautionary measures have to be taken by farmer. To build rover which is compact and will be controlled by the joystick and which can carry soil testing kit to the farming area or fields and also which consist of Ph testing, pesticide and water spray, weed cutter, seed sowing, soil color sensing along with the camera which can rotate 306 degrees night mode and also which can detect humans. To build system which can determine the amount of N-nitrogen, P- phosphorus, K- potassium [NPK] values in soil and gives the suggestions to improve the soil fertility and what are the different types of crops can grow on that particular soil. To build system which can predict the yield of a crops which will be calculated and determined based on the land size or area of a land. The output will be displayed on the lcd screen which will be placed on the rover and in the system also with some attributes which will be displayed on the lcd screen.

1.4 Scope

India is a land of agriculture and mainly known for growing variety of crops. Around half of the population in India depend on agriculture. Diseases to the crops may affect the livelihood of the farmers. In order to overcome this major problem, a robot that detects the leaf disease using image processing and Machine learning is deployed. This robot also monitors the field condition such as soil moisture, quality of crops and sprays the required amount of water and pesticides for achieving the good yield in agriculture. In this situation the yield of the crops must be high and of good quality which leads to a good amount of income in agriculture. Diseases to crops may affect both quality as quantity of the crops. Crop diseases are of mainly three types namely bacterial, fungal and spots. Traditional methods were used to detect the diseases which lead to the use of large number of pesticides harming the fertile soil and also the nature. A solution to this is to use modern methods in agriculture that helps the farmers to detect the diseases faster and increase the crop yield.

1.5 Existing System

Existing system depends on professionals' analysis of plants to identify disease. It involves inspection of plant by a medical professional this is time consuming and also it is expensive and also farmers do predict diseases of plants based on their experience but they are not right all the time and they are only familiar with very common diseases. Measuring the concentration of NPK can only be done by testing labs in major cities.

1.6 Proposed System

Proposed system consists of rover which is compact and can wireless controlled using mobile, laptop or joystick. It is powered using batteries. The rover has inbuilt camera which is used for navigation and also to take pictures of plant leaf to detect if any infection has occurred to the plant. Soil testing for NPK and soil color testing is done through Arduino and results are displayed on LCD screen.

If any infection occurred to the plant the precautions will be displayed at the result of google collab note book. The robot also consists features like Crop yield prediction, Human detection, Ph testing, Pesticide spray, Weed cutter and Seed sowing.

1.7 Applications

- It can be used to detect plant disease and to measure NPK of soil also can be used in small and large farm field. To know the kind of crops which can be grown.
- It also detects the humans in the farm fields. It also calculates the Ph level in the soil and pesticides spray, weed cutter and seed sowing.

CHAPTER 2

LITERATURE SURVEY

2.1 Related Works

Author Name &Year	Title	Method	Dataset	Results
Xulang Guan **2021**	A Novel Method of Plant Leaf Disease Detection Based on Deep Learning and Convolutional Neural Network	Convolution Neural Network (CNN), Residual Network (ResNet)	AI Challenger	91.7%
Fatma MARZOUGUI, Mohamed ELLEUCH, Monji KHERALLAH **2020**	A Deep CNN Approach for Plant Disease Detection	Convolution Neural Network (CNN)	Created own dataset of 500 Samples	96.7%
S. Santhana Hari, Mr. M. Sivakumar, Dr. P.Renuga, S. karthikeyan, S. Suriya **2019**	Detection of plant disease by leaf image using convolutional neural network	Convolutional Neural Network	Plant village website	86%
Suma V, R Amog Shetty, Rishab F Tated, Sunku Rohan, Triveni S Pujar **2019**	CNN based Leaf Disease Identification and Remedy Recommendation System	Artificial Neural Networks algorithm, Machine Learning algorithm	Plant village website	95.32%
Ishrat Zahan Mukti, Dipayan Biswas	Transfer Learning Based Plant Diseases Detection Using ResNet50	CNN, Artificial intelligence, Transfer Learning, Pattern	Research of Salathe Group	92.80%

2019		recognition, ResNet50.		
Melike Sardogan, Adem Tuncer, Yunus Ozen **2018**	Plant Leaf Disease Detection and Classification #ased on CNN with LVQ Algorithm	Convolution Neural Network (CNN), Learning vector Quantization	Plant village website	86%

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATIONS

3.1 Functional Requirements

Functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be technical details, data manipulation, encryption and decryption of data, and processing and other specific functionality that define what a system is supposed to accomplish.

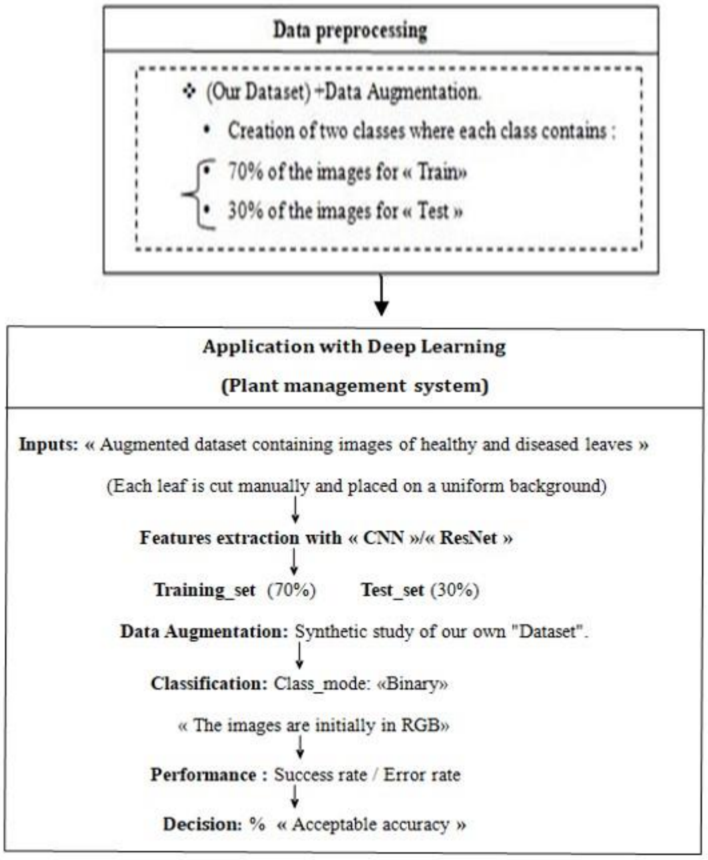


Figure 3.1: Overview of Proposed Framework

3.2 Non-Functional Requirements

Non-functional requirements are requirements which impose constraints on the design or implementation. This requirement includes performance engineering requirements, quality standards, or design constraints. They ensure the usability and effectiveness of the entire system. Failing to meet any one of them can result in systems that fail to satisfy internal business, user, or market needs, or that do not fulfill mandatory requirements imposed by regulatory or standards agencies. In some cases, non-compliance can cause significant legal issues.

Non-functional requirements are often mistakenly called the quality attributes of a system, however there is a distinction between the two. Non-functional requirements are the criteria for evaluating how a software system should perform and a software system must have certain quality attributes in order to meet non-functional requirements. So, when we say a system should be secure, highly-available, portable, scalable and so on, we are talking about its quality attributes. Other terms for non-functional requirements are qualities, quality goals, quality of service requirements, constraints, non-behavioral requirements or technical requirements.

Informally these are sometimes called the "ilities", from attributes like stability and portability. Qualities that is non-functional requirements can be divided into two main categories:

1. Execution qualities, such as safety, security and usability, which are observable during operation (at run time).
2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

Some of the non-functional requirements are**:**

- **Accuracy:** The model will predict accurately by extracted features in the training dataset.
- **Robustness:** The system is relying on predicting by training dataset which guarantees robustness.
- **Scalability:** The system should work for large number of given datasets.
- **Performance:** The performance of the system is measured based on the efficiency of the training dataset and testing dataset obtained.

3.3 Hardware Requirements

- Arduino Mega
- GPS Module
- PH Sensor
- Camera
- Motors
- Robot Wheels
- Soil testing
- Battery (Li-ion)
- Transmitter and Receiver

3.4 Software Requirements

- Code Editor: Google collab
- Language : Python, C++, java

3.5 Requirement Traceability Matrix

Serial No	Requirement Id	Requirement Brief	Requirement Description
1	RID-1	Validation for camera check	To verify the start of camera
2	RID-2	Validation for image capture	To verify that image is captured or not
3	RID-3	Validation for image capture rejection	No image should be captured when leaf is not in frame
4	RID-4	Validation for storing the captured image into respective folder	The captured image is stored as test.jpg
5	RID-5	Validation for image cropping	Identify leaf data in a image
6	RID-6	Embedding extraction	Identifying leaf landmarks in a leaf data and extracting its embedding
7	RID-7	Pushing into pretrained model	All structured embedding data will be pushed into pretrained model
8	RID-8	Disease Detection	To identify the disease of the plant leaf
9	RID-9	Precautions	Precautions based on disease
10	RID-10	Back to the Plant leaf detection	After the completion of each disease detection the flow goes back to the plant leaf detection

Figure 3.2: Requirement Traceability Matrix

CHAPTER 4

DATASETS

4.1 Datasets

This dataset consists of about 87K RGB images of healthy and diseased crop leaves which is categorized into 38 different classes. The total dataset is divided into 80/20 ratio of training and validation set preserving the directory structure. A new directory containing 33 test images is created later for prediction purpose.

The total number of images given as input at once in forward propagation of the CNN. Basically, batch size defines the number of samples that will be propagated through the network.

For instance, let's say you have 1050 training samples and you want to set up a batch size equal to 100. The algorithm takes the first 100 samples (from 1st to 100th) from the training dataset and trains the network. Next, it takes the second 100 samples (from 101st to 200th) and trains the network again. We can keep doing this procedure until we have propagated all samples through of the network.

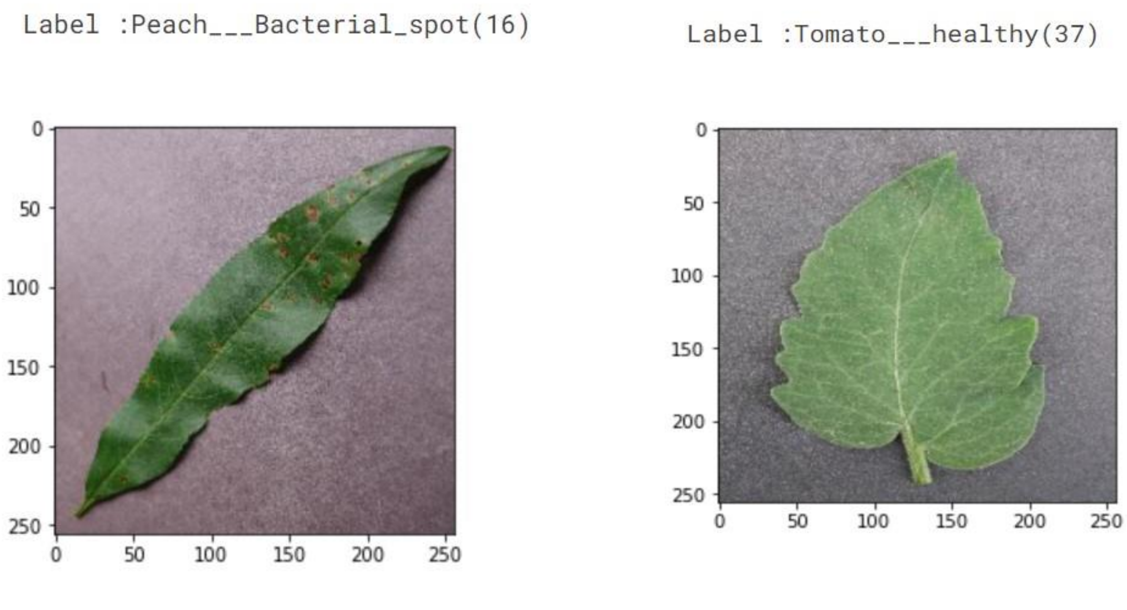


Figure 4.1: Leaf image

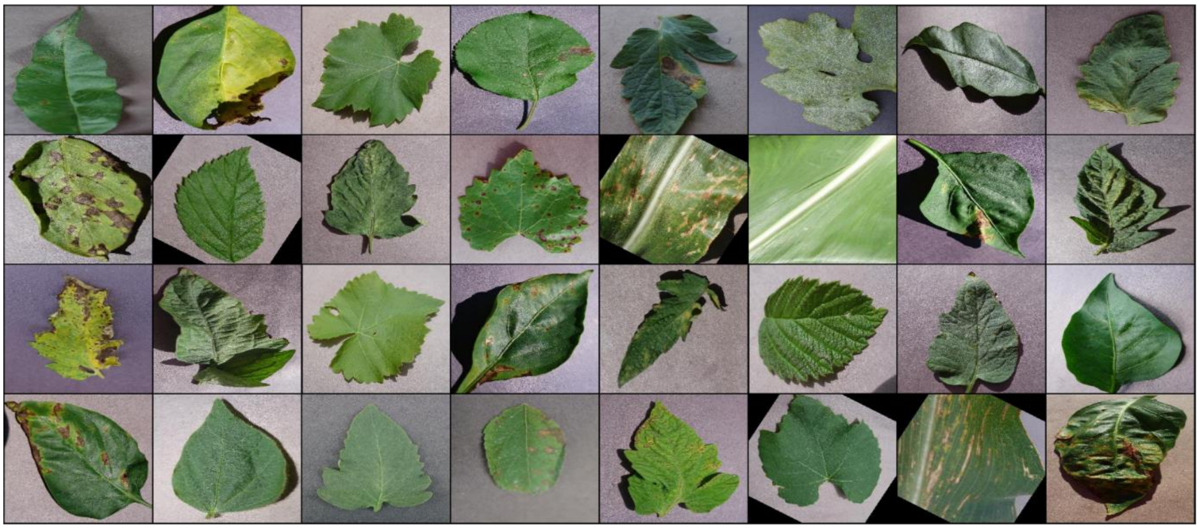


Figure 4.2: Different types of leaf images

	no. of images
Tomato___Late_blight	1851
Tomato___healthy	1926
Grape___healthy	1692
Orange___Haunglongbing_(Citrus_greening)	2010
Soybean___healthy	2022
Squash___Powdery_mildew	1736
Potato___healthy	1824
Corn_(maize)___Northern_Leaf_Blight	1908
Tomato___Early_blight	1920
Tomato___Septoria_leaf_spot	1745
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	1642
Strawberry___Leaf_scorch	1774
Peach___healthy	1728
Apple___Apple_scab	2016
Tomato___Tomato_Yellow_Leaf_Curl_Virus	1961
Tomato___Bacterial_spot	1702
Apple___Black_rot	1987
Blueberry___healthy	1816
Cherry_(including_sour)___Powdery_mildew	1683
Peach___Bacterial_spot	1838
Apple___Cedar_apple_rust	1760
Tomato___Target_Spot	1827
Pepper,_bell___healthy	1988
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	1722
Potato___Late_blight	1939
Tomato___Tomato_mosaic_virus	1790
Strawberry___healthy	1824
Apple___healthy	2008
Grape___Black_rot	1888
Potato___Early_blight	1939
Cherry_(including_sour)___healthy	1826
Corn_(maize)___Common_rust_	1907
Grape___Esca_(Black_Measles)	1920
Raspberry___healthy	1781
Tomato___Leaf_Mold	1882
Tomato___Spider_mites Two-spotted_spider_mite	1741
Pepper,_bell___Bacterial_spot	1913
Corn_(maize)___healthy	1859

Figure 4.3: Datasets showing total no of images used to training and testing

CHAPTER 5

SYSTEM ANALYSIS AND DESIGN

5.1 System Analysis

The system design process builds up a general framework building design. The programming outline includes speaking to the product framework works in a shape that may be changed into one or more projects. The prerequisite indicated by the end client must be put systematically. An outline is an inventive procedure; a great configuration is a way to the viable framework. The framework "Outline" is characterized as "The procedure of applying different systems and standards with the end goal of characterizing a procedure or a framework inadequate point of interest to allow its physical acknowledgment". Different configuration components are taken after to add to the framework. The configuration detail portrays the components of the framework, the segments or components of the framework, and their appearance to end clients.

5.2 System Architecture

The architectural configuration procedure is concerned with building up a fundamental basic system for a framework. It includes recognizing the real parts of the framework and the interchange between these segments. The beginning configuration procedure of recognizing these subsystems and building up a structure for subsystem control and correspondence is called construction modeling outline and the yield of this outlined procedure is a portrayal of the product structural planning. The proposed architecture for this system is given below. It shows the way this system is designed and the brief working of the system.

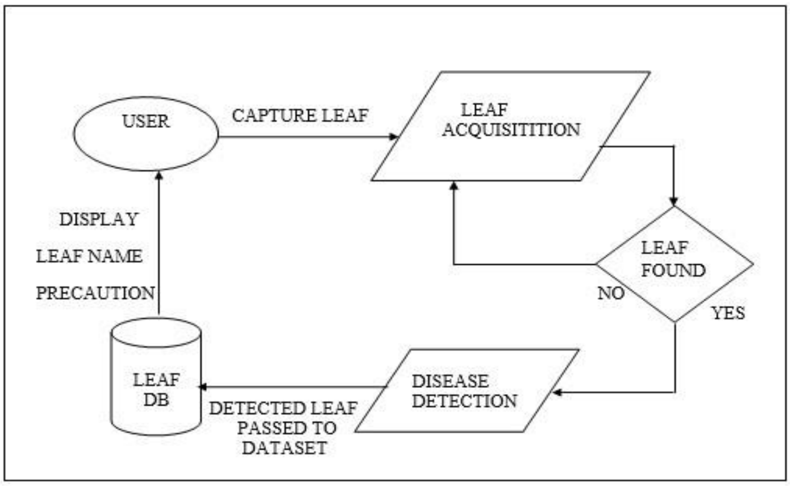


Figure 5.1: System Architecture

5.3 High-Level Design

High-Level Design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for their interfaces. The HLD uses possibly non-technical to mildly technical terms that should be understandable to the administrators of the system. In contrast, low-level design further exposes the logical detailed design of each of these elements for programmers.

5.3.1 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

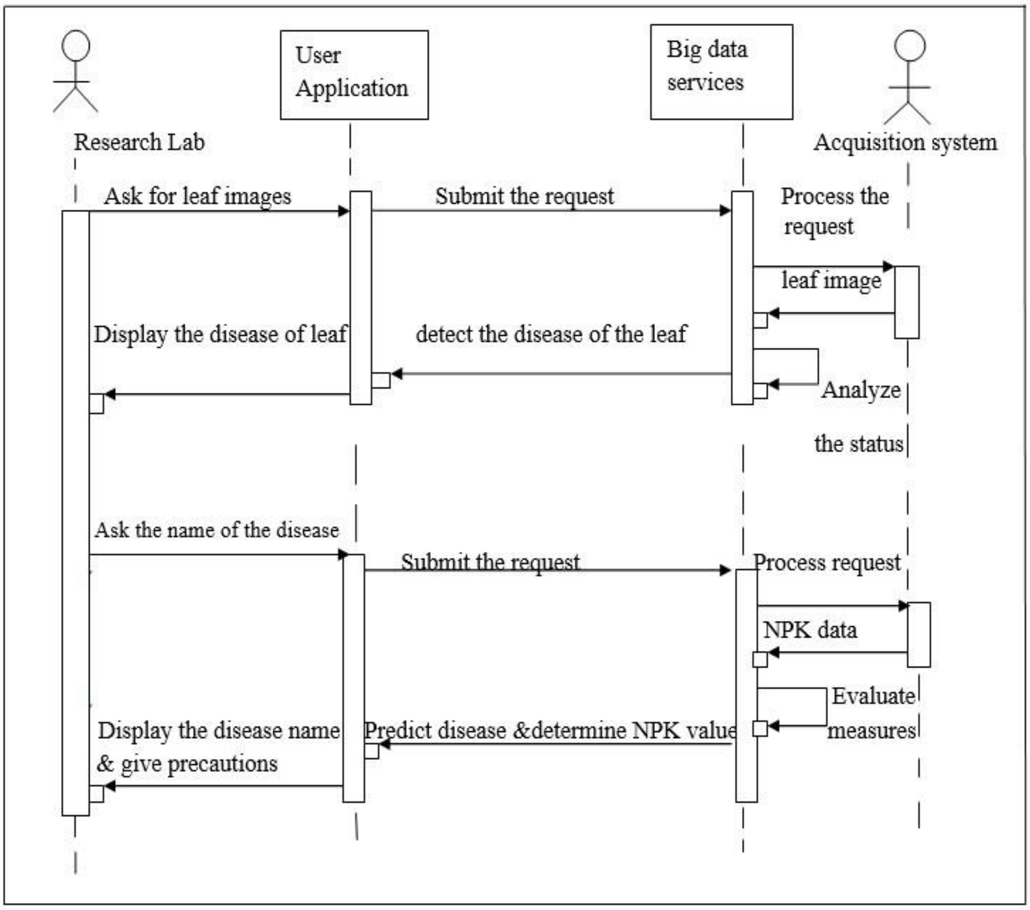


Figure 5.2: Sequence Diagram

5.4 Low-Level Design

Low-Level Design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. During the detailed phase, the logical and functional design is done and the design of the application structure is developed during the High-Level Design.

5.4.1 Flowchart

A flowchart is a diagram that represents a set of instructions. Flowcharts normally use standard symbols to represent the different types of instructions. These symbols are used to construct the flowchart and show the step-by-step solution to the problem.

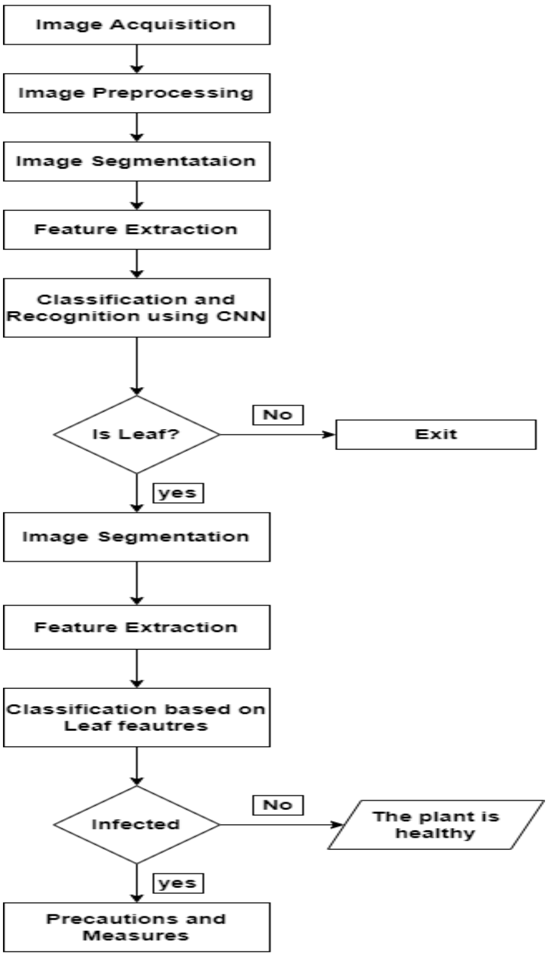


Figure 5.3: Flow Chart

5.5 Data Flow

Figure shows the data flow for the proposed system. In the proposed method the model identifies the modules that should be built. For developing the system and the specifications of these modules. Data preprocessing is a data mining technique which is used to transform the raw data in a useful and efficient format. Image preprocessing improves the image data necessary for image classification task. In preprocessing methods, there are geometric transformations of images such as image rotation, image scaling, and translation of images Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups.

segmentation techniques and extracted features from leaf images and then implemented neural network as a classifier with back propagation. The classification is a necessary step as it compares the values received after the feature extraction step is compared with a pre-calculated set of data. Feature Extraction Feature Extraction is a method used for dimensionality reduction which helps in representing into a compact feature vector, the parts of the image which are interesting.

This process results to be very helpful when the sizes of images are large and for the faster image matching and retrieval, reduced feature representations are required to complete tasks quickly. Depth analysis is used to estimate depth from an image. The task requires an input image and outputs a depth image. The depth image includes information about the distance of the objects in the image from the viewpoint, which is usually the camera taking the image. Error on the validation set is used to determine when to stop training so that the model does not overfit. The validation error that comes out of this process can also be used to estimate generalization performance of the model. The model will predict the required output.

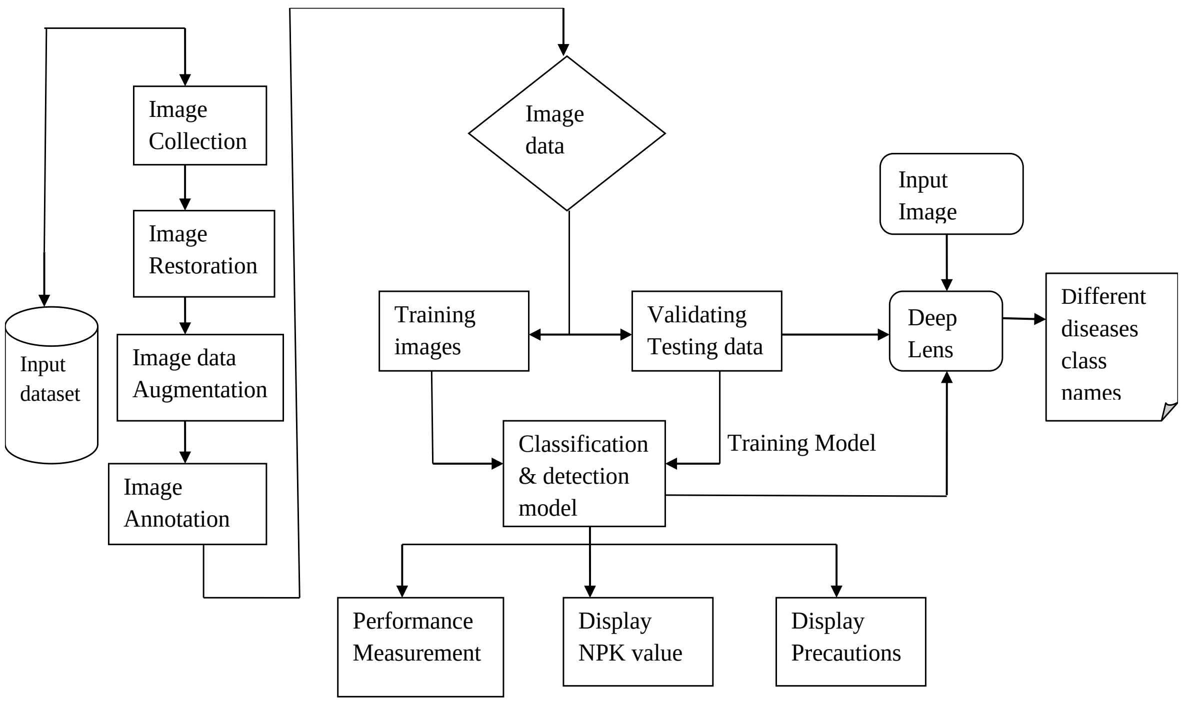


Figure 5.4: Data Flow

CHAPTER 6

IMPLEMENTATION

6.1 Image Acquisition

Image Acquisition is the first step in any image processing system. The general aim of any image acquisition is to transform an optical image (real-world data) into an array of numerical data which could be later manipulated on a computer. Image acquisition is achieved by suitable cameras. We use different cameras for different applications. In this Project, the webcam is used for the image acquisition where the user face is captured.

6.2 Image Preprocessing

Image pre-processing is the term for operations on images at the lowest level of abstraction. These operations do not increase image information content but they decrease it if entropy is an information measure. The aim of pre-processing is an improvement of the image data that suppresses undesired distortions or enhances some image features relevant for further processing and analysis tasks.
There are 4 different types of Image Pre-Processing techniques and they are listed below.

1. Pixel brightness transformations/ Brightness corrections
2. Geometric Transformations
3. Image Filtering and Segmentation
4. Fourier transform and Image restauration

Brightness transformations modify pixel brightness and the transformation depends on the properties of a pixel itself. In PBT, output pixel's value depends only on the corresponding input pixel value. Examples of such operators include brightness and contrast adjustments as well as colour correction and transformations.

Contrast enhancement is an important area in image processing for both human and computer vision. It is widely used for medical image processing and as a pre-processing step in speech recognition, texture synthesis, and many other image/video processing applications.

There are two types of Brightness transformations and they are below:

1. Brightness corrections
2. Gray scale transformation

Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions, often based on the characteristics of the pixels in the image. Image segmentation could involve separating foreground from background, or clustering regions of pixels based on similarities in color or shape.

6.3 Machine Learning Techniques

This section discusses of various machine learning techniques used for classification and has been used for this system. Results of all of these models has been compared to find the best model for the system.

6.3.1 Logistic Regression

Logistic Regression is a classification not a regression formula. it’s wont to estimate distinct values (Binary values like 0/1, yes/no, true/false) supported given set of freelance variables. In straightforward words, it predicts the chance of prevalence of an occurrence by fitting information to a log it performs. Hence, it’s additionally called logistic regression. Since, it predicts the chance, its output values lie between zero and one. Logistic regression is employed for prediction of output that is binary, as declared higher than. as an example, if a master card company goes to make a model to determine whether or not to issue a master card to a client or not, it’ll model for whether or not the client goes to Default or Not Default on this master card. Firstly, simple regression is performed on the link between variables to urge the model. the brink for the classification line is assumed to be at zero.5 Logistic operate is applied to the regression to urge the chances of its happiness in either category. It provides the log of the chance of the event occurring to log of the chance of it not occurring. In the end, its categories the variable supported the upper chance of either class.

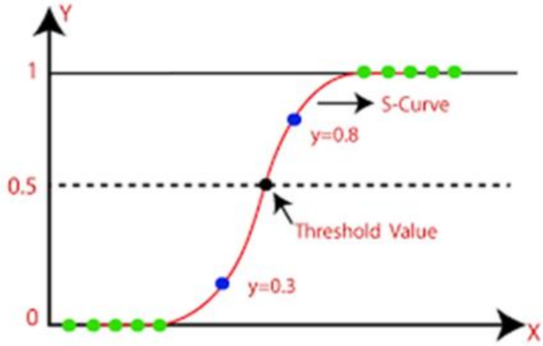
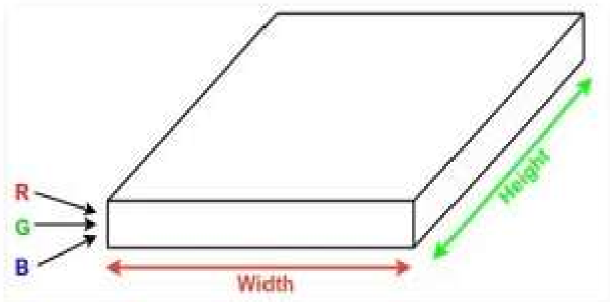


Figure 6.1: Logistic Regression

6.3.2 Convolution Neural Networks

A CNN convolutional neural network is perhaps the most widely applied method for extracting reasonable information from huge datasets. The convolution and pooling layers are then passed through in several steps to obtain global features from the input data. Finally, the extracted characteristics are passed to the fully connected layer where classification is performed in this layer.



Now imagine taking a small patch of this image and running a small neural network on it, with say, k outputs and represent them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different width, height, and depth. Instead of just R, G and B channels now we have more channels but lesser width and height. his operation is called Convolution.

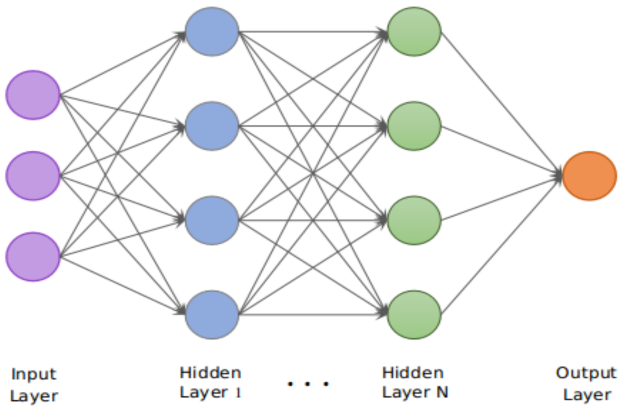


Figure 6.2: Convolution Neural Networks

6.3.3 Neural Network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

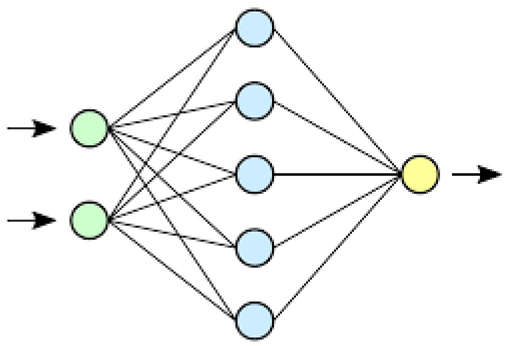


Figure 6.3: Neural Network

6.3.4 Visual Geometry Group 19

VGG19 is a variant of VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layer, 5 Max Pool layers and 1 SoftMax layer). There are other variants of VGG like VGG11, VGG16 and others. VGG19 has 19.6 billion FLOPs.

AlexNet came out in 2012 and it improved on the traditional Convolutional neural networks, so we can understand VGG as a successor of the AlexNet but it was created by a different group named as Visual Geometry Group at Oxford's and hence the name VGG, it carries and uses some ideas from its predecessors and improves on them and uses deep Convolutional neural layers to improve accuracy.

6.4 User Interface Design

In computer or software design, user interface (UI) design is the process of building interfaces that are aesthetically pleasing. Designers aim to build interfaces that are easy and pleasant to use. UI design refers to graphical user interfaces and other forms of interface design. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals.

User interface design plays an important role in determining how people interact with a software product. This is because UI design has to do with how different visual elements, such as colors, typography, and imagery, work together to create a seamless user experience.

The overarching goal of every UI design project is to create a product that is both aesthetically appealing and easy to interact with. In other words, the product needs to create a pleasant experience for users. This will not only create a strong first impression, but it also keeps people coming back for more. As you can imagine, user retention is an important metric for any app or website, and UI design plays a huge role in boosting that figure.

6.4.1 GUI

Designing the visual composition and temporal behavior of a GUI is an important part of software application programming in the area of human–computer interaction. Its goal is to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline named usability. Methods of user-centered design are used to ensure that the visual language introduced in the design is well-tailored to the tasks.

A GUI uses a combination of technologies and devices to provide a platform that users can interact with, for the tasks of gathering and producing information.

A series of elements conforming a visual language have evolved to represent information stored in computers. This makes it easier for people with few computer skills to work with and use computer software.

The graphical user interface for an Android app is built using a hierarchy of View and View Group objects. View objects are usually UI widgets such as buttons or text fields and View Group objects are invisible view containers that define how the child views are laid out, such as in a grid or a vertical list.

Android UI Controls are those components of Android that are used to design the UI in a more interactive way. It helps us to develop an application that makes user interaction better with the view components. Android provides us a huge range of UI controls of many types such as buttons, text views, etc.

As we know UI is the only thing that a user interacts with within an application. This is the reason that we make our application look aesthetic and, more and more connective. To do so, we need to add the UI controls or we say Input controls in the respective application.

Moving forth we will see some important Android UI controls for our applications:

- TextView
- EditText
- Button
- ImageButton
- ToggleButton
- RadioButton
- RadioGroup
- CheckBox
- AutoCompleteTextView
- ProgressBar
- Spinner
- TimePicker
- DatePicker
- SeekBar

- AlertDialog
- Switch
- RatingBar

6.5 Methodology and Techniques

Training of machine learning module with data sets: The neural machine learning module is created using keras library and python in google collab editor The CNN model is built on vgg19 architecture.

For training the module 1000's of images of plant leaves with disease are collected and stored in a file. The file is divided into two parts training datasets and testing data sets training data sets is used to train the machine learning module. It consists 38 classes of plant disease images labeled respectively. These images will be fed into machine learning module, and trained module is extracted.

The detection of plant disease starts with the collection of plant leaf image sample at the farm, the image is then passed into trained neural module for diagnosis. The trained machine learning module will give output whether the sample image contains any disease and if disease is detected a precaution for that disease will also be displayed.

6.5.1 SYSTEM ARCHITECTURE

The system is machine learning based approach to detect the plant disease. It uses convolution neural network built on vgg 19 architecture.to detect disease.

● A fixed size of (224 * 224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3).

● The only preprocessing that was done is that they subtracted the mean RGB value from each pixel, computed over the whole training set.

● Used kernels of (3 * 3) size with a stride size of 1 pixel, this enabled them to cover the whole notion of the image.

● spatial padding was used to preserve the spatial resolution of the image.

● max pooling was performed over a 2 * 2-pixel windows with side 2.

● This was followed by Rectified linear unit (ReLU) to introduce non-linearity to make the model classify better and to improve computational time as the previous models used tanh or sigmoid functions this proved much better than those.

● Implemented three fully connected layers from which first two were of size 4096 and after that a layer with 1000 channels for 1000-way ILSVRC classification and the final layer is a SoftMax function.

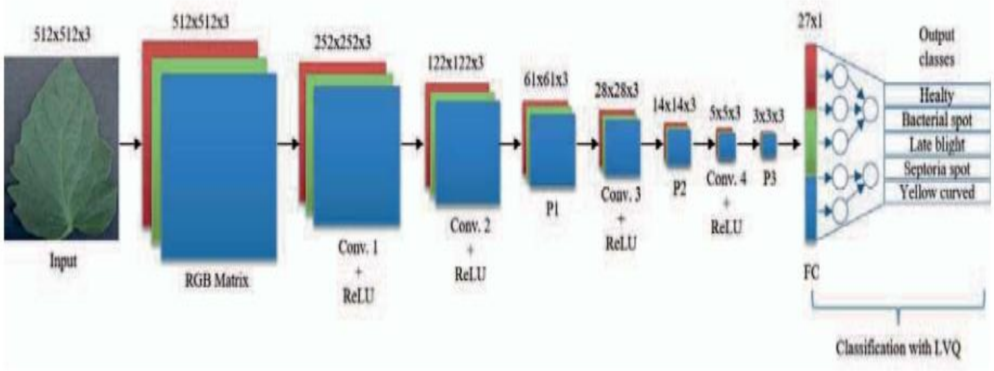


Figure 6.4: Architecture of the proposed method

6.5.2 CONVOLUTIONAL NEURAL NETWORK ALGORITHM

Input layer is the one which accepts features as the input, in other words, images are given as input through this layer. The Middle layer consists of the desired number of nodes based on the application. Output layer produces an output.

A. Convolutional Layer - It performs a convolutional operation over the pixel values in it along with the kernel matrix. The kernel matrix is slid over the pixel matrix, and value is determined.

B. Max Pooling Layer - This is used for reducing the final size of the filter map produced as an output. This helps in prevention overfitting issues.

C. ReLU Activation Function - Rectified Linear Units, this function will simply replace all the negative values present, by zero (0) which are present in the output matrix and retain all the positive values.

D. Fully Connected Layer - In this layer, every node in the previous hidden layer is connected with the next set of nodes in the following hidden layer, shown in figure. There may be desired numbers of nodes present in the FC layer. FC layer is otherwise called as Dense Nodes. All layers will be connected to all other previous and next layers by means of edge connectivity between the neurons present in it.

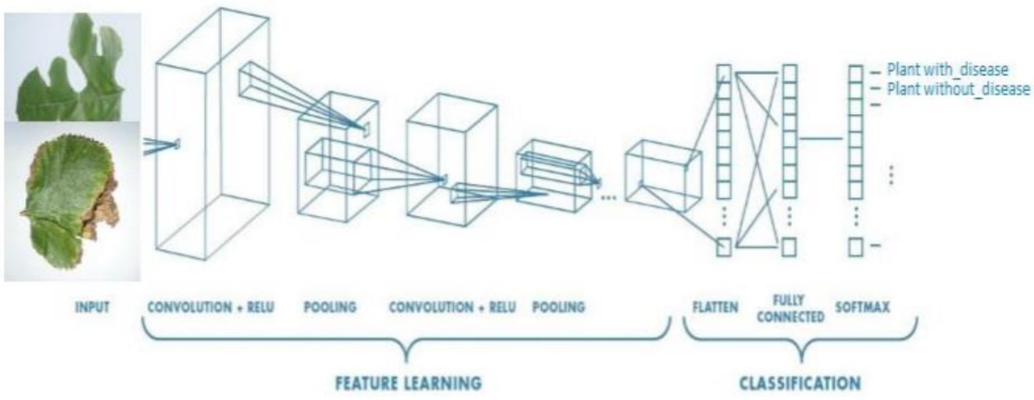


Figure 6.5: Image showing how CNN process take place

Input Image

In the figure, we have an RGB image which has been separated by its three-color planes - Red, Green, and Blue. There are a number of such color spaces in which images exist - Grayscale, RGB, HSV, CMYK, etc.

You can imagine how computationally intensive things would get once the images reach dimensions, say 8K (7680×4320). The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

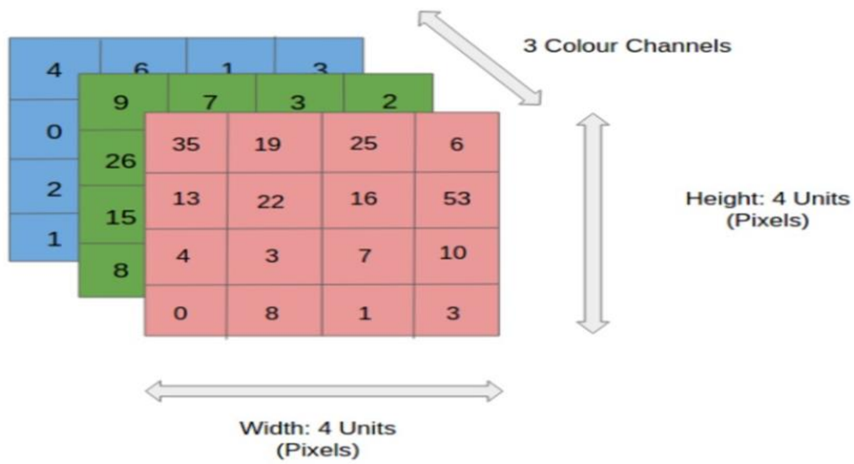


Figure 6.6: Input image

Convolution Layer - The Kernel

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, e.g., RGB)

In the above demonstration, the green section resembles our 5x5x1 input image, I. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix.

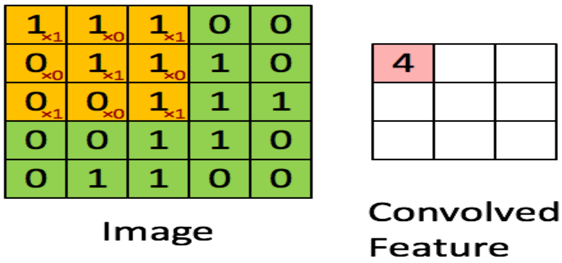


Figure 6.7: Convolution layer

Pooling Layer

Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling.

1. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand.
2. Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

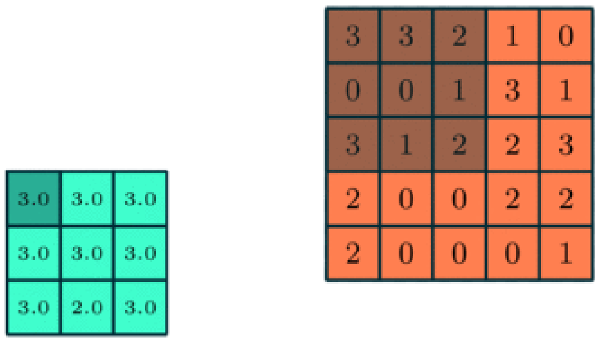


Figure 6.8: Pooling layer

Classification - Fully Connected Layer (FC Layer):

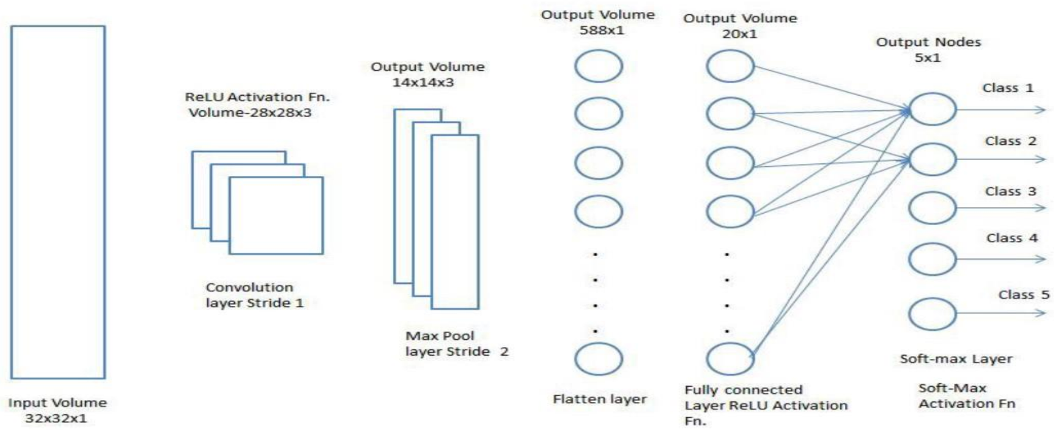


Figure 6.9: Classification of fully connected layer

Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

Now that we have converted our input image into a suitable form for our Multi-Level Perceptron, we shall flatten the image into a column vector. The flattened output is fed to a feed-forward neural network and backpropagation applied to every iteration of training. Over a series of epochs, the model is able to distinguish between dominating and certain low-level features in images and classify them using the SoftMax Classification technique.

- The robot is designed using an advanced processor which is integrated with machine learning model. The machine learning model with Image processing is trained with feature extraction, Using Convolution Neural Network [CNN] algorithm.
- The robot is wi-fi controlled using the joystick and also in autonomous mode robot can also move and complete the tasks in autonomous mode with the help of GPS tracker. It is powered using Li-ion battery.

  The robot also consists of several features like:
- **Crop yield prediction**: crop yield for a given land size will be calculated and will be displayed on screen.
- **Soil color sensing**: color of soil is sensed using soil color sensor, this will give an idea about the type of soil. the values will be displayed on screen.
- **Human detection**: Camera installed in robot is equipped with human detection which

useful in guarding the farm and gives alert message.

- **Ph testing**: The Ph of the soil can be calculated using this sensor and shows the amount of Ph in the soil and measures.
- **Pesticide spray**: pesticide and water can be sprayed on the plants with the help of inbuild spray in robot. And also watering alert.
- **Weed cutter**: robot is equipped with electric weed cutter. Which helps in cutting weeds in the land area.
- **Seed sowing**: The robot also equipped with seed sowing and soil leveler which can plant seeds in the soil and level the area of land.

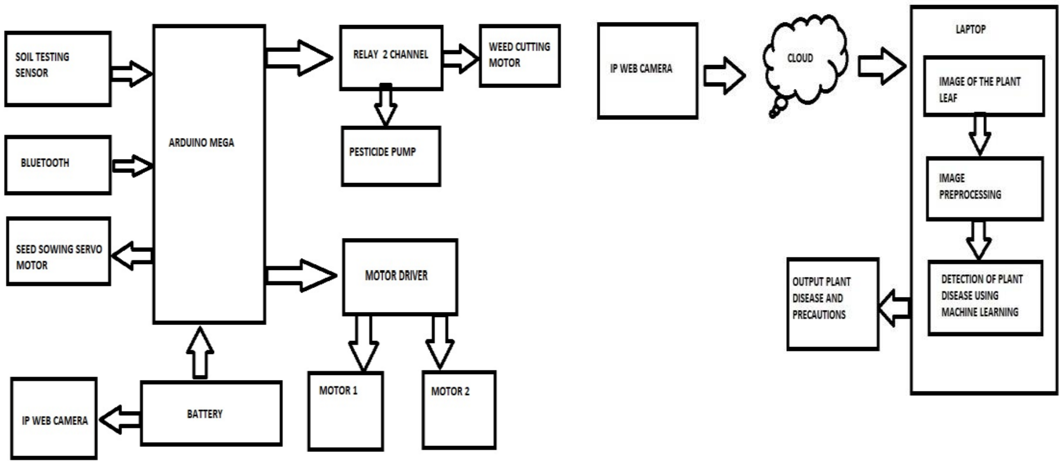


Figure 6.10: Overview of the robot model

CHAPTER 7

TESTING

7.1 Design of test cases

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.2 Types of Testing

7.2.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.3 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests areidentified and the effective value of current tests is determined.

7.2.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.2.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledgeof the inner workings, structure and language of the software, or at least its purpose. It is used to test areas that cannot be reached from a black box level.

7.2.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.2.7 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7.3 Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

7.3.1 Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

7.3.2 Features to be tested

- Verify that the entries are of the correct format.
- No multiple images should be allowed
- All links should take the user to the correct page.

7.4 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

7.5 Test Cases

Serial No	Requirement Id	Test Case Reference	Input	Output	Test Result	Remarks
1	RID-1	TC-1	Initiating Camera	Initiated Camera	Pass	Started successfully
2	RID-2	TC-2	Identify the Image	Identified the leaf Images	Pass	Image captured when found
3	RID-3	TC-3	Leaf is not in frame	No leaf image is found	Pass	Image not captured
4	RID-4	TC-4	Capture the leaf image	Stored in the Dataset	Pass	Image saved as test.jpg
5	RID-5	TC-5	Given the leaf image	Identify the leaf image in the dataset	Pass	leaf data in a image is found
6	RID-6	TC-6	Given the Extracted leaf image	Identified the lead landmark and extract the feature.	Pass	Landmarks identified
7	RID-7	TC-7	The image cannot be identified correctly	Disease not found	Fail	The leaf image was not detected For Pre-processed image.
8	RID-8	TC-8	The infected plant can identified correctly	Name of the plant disease can be displayed	Pass	Identify the disease of the plant leaf.
9	RID-9	TC-9	Collect the Soil to be tested	Determine the NPK value	Pass	NPK value will be displayed.
10	RID-10	TC-10	Infected plant image can be given	Give the solution to grow the plants	Pass	Precautions will be displayed.

11	RID-11	TC-11	Leaf image not in frame	Leaf image was not detected	Fail	The flow didn’t return application had to be closed and opened again.

Figure 7.1: Test Cases Table

CHAPTER 8

SNAPSHOTS AND RESULTS

8.1 Snapshots

Figure 8.1: Model of crofting hybrid robot

```
[ ] path="/content/drive/MyDrive/livetest/999.jpg"
    prediction(path)

    9
    The plant diagnosed asCorn_(maize)__Northern_Leaf_Blight
    ⍰     Com_(maize)_Northern_Leaf_Blight

    1.    Crop rotation with a non host crop like legumes to reduce the disease severity of Turcicum leaf blight by providing the time for infected resi
    2.    Destruction of crop debris.
    3.    Use of resistant/tolerant hybrids
    4.    Seed treatment with Azospririllum (ACD 15 or ACD 20 Stran) @ 25 g and Trichoderma @ 6g per kg seeds or Thiram 75 WP @ 2 g or Capatn 50 WP @ 2 g
    5.    Spray 2.5g Mancozeb 75 WP or 1 ml Hexaconazole 5 EC or 1 ml combiproduct fungicide Azoxystrobin 18.2% + Difenconazole 11.4% per liter of water
    6.    Two sprays of Mancozeb spray (2-3gm/litre) has to be taken at 15 days interval immediately after disease appearance.
```

Figure 8.2: Plant disease identification and precautions

```
soiltest=input("Soil test for n \n 1:Adequate \n 2:Deficient \n 3:Depleted \n")
if(soiltest == "1"):
  path="/content/drive/MyDrive/npk/n/ade.txt"
if(soiltest == "2"):
  path="/content/drive/MyDrive/npk/n/def.txt"
if(soiltest == "3"):
  path="/content/drive/MyDrive/npk/n/dep.txt"
f=open(path)
print(f.read())

... Soil test for n
 1:Adequate
 2:Deficient
 3:Depleted
```

Figure 8.3: Soil testing for Nitrogen

```
Soil test for n
 1:Adequate
 2:Deficient
 3:Depleted
1
N - Adequate
Lawn [3.75 - 4.0]
Fruit [3.75 - 4.0]
Flower [3.75 - 4.0]
Shrubs (Flowering) [3.75 - 4.0]
Shrubs (Foilage) [3.75 - 4.0]
Veggies (Root) [3.75 - 4.0]
Veggies (Leafy) [7.75 - 8.0]
Tree [3.75 - 4.0]
General Feed [3.75 - 4.0]
```

Figure 8.4: Resultant for the Nitrogen content in the given soil

```
soiltest=input("Soil test for p \n 1:Adequate \n 2:Deficient \n 3:Depleted \n")
if(soiltest == "1"):
  path="/content/drive/MyDrive/npk/p/ade.txt"
if(soiltest == "2"):
  path="/content/drive/MyDrive/npk/p/def.txt"
if(soiltest == "3"):
  path="/content/drive/MyDrive/npk/p/dep.txt"
f=open(path)
print(f.read())
```

```
Soil test for p
 1:Adequate
 2:Deficient
 3:Depleted
```

Figure 8.5: Soil testing for Phosphorous

```
Soil test for p
 1:Adequate
 2:Deficient
 3:Depleted
2
P - Deficient
Lawn [1.0 - 1.5] = [0]
Fruit [4.0 - 4.25] = [2.25]
Flower [4.0 - 4.25] = [2.25]
Shrubs (Flowering) [4.0 - 4.25] = [1.0 - 1.25]
Shrubs (Foilage) [5.25 - 5.5] = [2.25]
Veggies (Root) [5.25 - 5.5] = [3.0]
Veggies (Leafy) [5.25 - 5.5] = [2.25]
Tree [5.25 - 5.5] = [2.25]
General Feed [4.0 - 4.25] = [1.0 - 1.25]
```

Figure 8.6: Resultant for the Phosphorous content in the given soil

```
soiltest=input("Soil test for k \n 1:Adequate \n 2:Deficient \n 3:Depleted \n")
if(soiltest == "1"):
  path="/content/drive/MyDrive/npk/k/ade.txt"
if(soiltest == "2"):
  path="/content/drive/MyDrive/npk/k/def.txt"
if(soiltest == "3"):
  path="/content/drive/MyDrive/npk/k/dep.txt"
f=open(path)
print(f.read())
```

```
Soil test for k
 1:Adequate
 2:Deficient
 3:Depleted
```

Figure 8.7 Soil testing for Potassium

```
Soil test for k
 1:Adequate
 2:Deficient
 3:Depleted
3
K - Depleted
Lawn [4.75 - 5.0] = [0]
Fruit [13.5 - 14.0] = [4.75 - 5.0]
Flower [13.5 - 14.0] = [4.75 - 5.0]
Shrubs (Flowering) [13.5 - 14.0] = [4.75 - 5.0]
Shrubs (Foilage) [8.75 - 9.0] = [2.25 - 2.5]
Veggies (Root) [8.75 - 9.0] = [2.25 - 2.5]
Veggies (Leafy) [8.75 - 9.0] = [2.25 - 2.5]
Tree [8.75 - 9.0] = [2.25 - 2.5]
General Feed [8.75 - 9.0] = [2.25 - 2.5]
```

Figure 8.8: Resultant for the Potassium content in the given soil

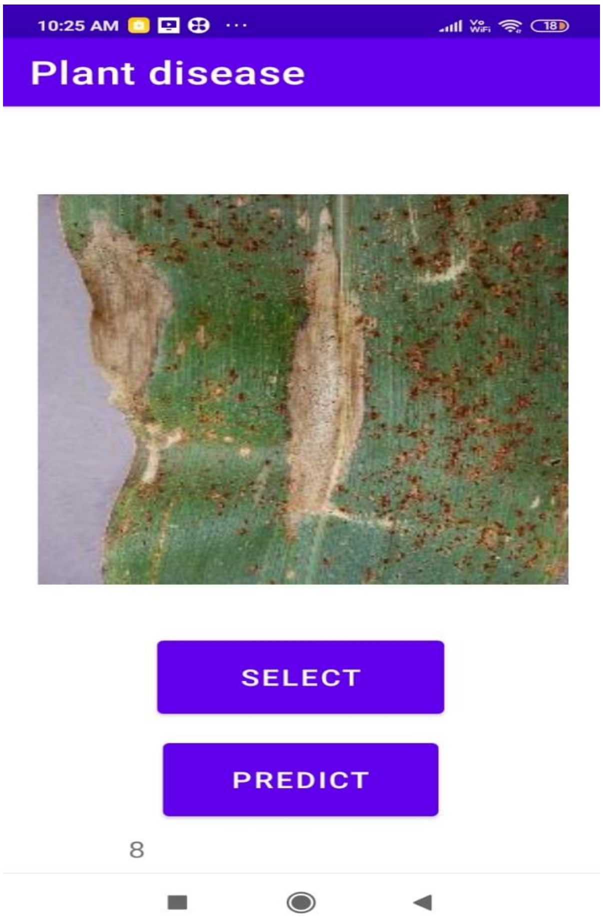


Figure 8.9: Showing Plant disease in then android app

CONCLUSION AND FUTURE ENHANCEMENT

In this project, a Convolutional Neural Network was constructed to identify plant diseases automatically. Our developed model can distinguish between 38 classes of both healthy leaves and disease affected leaves. The entire method has been narrated specifically. Model performances have been analyzed through comparison with few other transfer learning models along with suitable graphs. With the increasing depth of CNN model, more image data is required for the best generalization of the model. For that reason, after preprocessing the data, we enlarged the dataset through the process of augmentation. Finally, It’s probably due to the nature of the leaves. The images of some species are very close to each other and share very similar shape, color, and texture. So, sometimes it is very hard for networks to correctly predict the true labels. Nowadays, smart mobile devices facilities us in terms of easy access and the flexibility in uses. Mobile camera can be used to capture these leaves of plant. A precise plant diseases detection model implemented in those smart phones will help farmers to recognize the plant diseases in a very short time through a convenient way. Farmers will be able to make decision own self. And, it will have a golden impact on the progress of agriculture.

For the future enhancement, providing more functionalities and control to the application. Developing the robots in large number so that farmers will be benefited by our project. Alerting the farmer when the disease crop is formed. Perdition of the crop yield based on the area and also considering climatic changes, soil fertility and other parameters.

BIBLIOGRAPHY

[1] A Novel Method of Plant Leaf Disease Detection Based on Deep Learning and Convolutional Neural Network Xulang Guan.

[2] 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.

[3] Transfer Learning Based Plant Diseases Detection Using ResNet50 Ishrat Zahan Mukti, Dipayan Biswas.

[4] CNN based Leaf Disease Identification and Remedy Recommendation System Suma V R Amog Shetty, Rishab F Tated, Sunku Rohan, Triveni S Pujar.

[5] A Deep CNN Approach for Plant Disease Detection Fatma MARZOUGUI, Mohamed ELLEUCH, Monji KHERALLAH.

[6] Plant Leaf Disease Detection and Classification based on CNN with LVQ Algorithm Melike Sardogan, Adem Tuncer, Yunus Ozen.