Visvesvaraya Technological University

Jnana Sangama, Belagavi - 590018

Project report on

DESIGN OF AUTOMATISED DISEASE DETECTION AND FERTIGATION SYSTEM FOR AGRICULTURAL CROPS

Submitted in partial fulfilment for the award of Degree of

BACHELOR OF ENGINEERING
In
ELECTRONICS & COMMUNICATION and COMPUTER SCIENCE & ENGINEERING

HARISH THANIKAIVELU	**1VE19CS052**
HARSHA B A	**1VE19CS054**
NEETHUSHREE K	**1VE19EC070**
PRIYANKA B H	**1VE19EC082**

Under the Guidance of

Dr. Poornima G R & Prof. Niranjana C

Dept. of ECE, SVCE, Bengaluru

Accredited by NAAC & NBA*

SVCE BENGALURU

SRI VENKATESHWARA COLLEGE OF ENGINEERING

— Affiliated to VTU, Approved by AICTE, Recognised by UGC u/s 2(f) & 12(B)—

Department of Electronics & Communication Engineering

Computer Science and Engineering

SRI VENKATESHWARA COLLEGE OF ENGINEERING

(Affiliated to VTU, Belagavi, NBA* Accredited, Approved by AICTE New Delhi) Kempegowda International Airport Road, Vidyanagar, Bengaluru – 562157.

2022 – 2023

SRI VENKATESHWARA COLLEGE OF ENGINEERING

Vidyanagar, Bengaluru - 562157

Department of ECE and CSE

CERTIFICATE

This is to certify that the Project Work entitled

DESIGN OF AUTOMATISED DISEASE DETECTION AND FERTIGATION SYSTEM FOR AGRICULTURAL CROPS

Carried out by

NAME OF THE STUDENTS	USN
HARISH THANIKAIVELU	1VE19CS052
HARSHA B A	1VE19CS054
NEETHUSHREE K	1VE19EC070
PRIYANKA B H	1VE19EC082

In partial fulfillment of the requirements for the award of the Degree of Bachelor of Electronics & Communication and Computer Science & Engineering of Visvesvaraya Technological University, Belagavi, during the year 2022- 2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The Project Report has been approved as it satisfies the academic requirements in respect of the Project Work prescribed for the said Degree.

Signature of the Guide	Signature of the HoD	Signature of the HoD
(Dr. Poornima G R)	(Dr. Hema M S)	(Dr. Jijesh J J)

Signature of the Guide	Signature of the Principal
(Mrs. Niranjana C)	(Dr. Nageswara Guptha M)

External Viva -Voce

Name of the Examiners	Signature with date
1.	
2.	

ABSTRACT

The agriculture industry has a tremendous obligation to boost productivity and food grain output due to the expanding civilization and impacts of climate change. Agriculture mechanization has become the sole choice and is urgently needed in the majority of India where expanding cropland is inconceivable. All industries, including agriculture, have already begun to profit from the Internet of Things and artificial intelligence. Through the development of intelligent algorithms that can track, manage, and visualize numerous farming production in real-time also with intelligence comparable to that of human specialists, advancements in these digitized innovations have brought about revolutionary advances in agriculture. The invention of smart farm equipment, irrigation facilities, weed and pest extermination, application of fertilizers, greenhouse development, warehousing structures, drones for crop protection, plant health management, etc. are some of the potential uses of IoT and AI that are explored in the study. In our system we Employ artificial intelligence and the internet of things concepts to Analyze the disease in crops and treat them using the adequate fertilizers, the article's major goal is to present an overview of the customized care given to crops depending on the issues that arise in the crops and to provide appropriate fertigation so that the crops may live and yield good results.

DECLARATION

We **Harsha B A (1VE19CS054), Harish Thanikaivelu (1VE19CS052), NeethuShree K (1VE19EC070), Priyanka B H (1VE19EC082)** students of final semester B.E in Electronics & Communication and Computer Science Engineering, Sri Venkateshwara College of Engineering, Bengaluru, hereby declare that the dissertation work entitled **DESIGN OF AUTOMATISED DISEASE DETECTION AND FERTIGATION SYSTEM FOR AGRICULTURAL CROPS** has been carried out under supervision **Dr. Poornima G R & Mrs. Niranjana C** Department of Electronics and Communication Engineering, SVCE, Bengaluru, the partial fulfilment of the requirements for the award of the Degree of Bachelor of Engineering in Electronics & Communication and Computer Science Engineering by Visvesvaraya Technological University, Belagavi during the academic year 2022-23. Further, the matter embodied in the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

Date: Place: Bengaluru

HARISH THANIKAIVELU	1VE19CS052
HARSHA B A	1VE19CS054
NEETHUSHREE K	1VE19EC070
PRIYANKA B H	1VE19EC082

ACKNOWLEDGEMENT

My most sincere and grateful thanks to **SRI VENKATESHWARA COLLEGE OF ENGINEERING**, for giving an opportunity to pursue the B.E in Electronics & Communication Engineering and thus helping to share career. First and foremost, I would like to express my gratitude to **Dr. NAGESWARA GUPTHA M, PRINCIPAL, SVCE, Bengaluru** for his support in bringing this Project to completion.

I would like to extend our sincere thanks to **Dr. HEMA S, Head, Department of Computer Science Engineering, SVCE, Bengaluru** for his suggestion which helped me to complete the Project.

I would like to extend our sincere thanks **Dr. JIJESH J J, Professor & HOD, Dept. of ECE, SVCE, Bengaluru** for his suggestion which helped me to complete the Project.

I would also like to sincere thanks to internal guide **Dr. POORNIMA G R & Prof. NIRANJANA C, Department of Electronics & Communication Engineering, SVCE, Bangalore** for her guidance and support in bringing this Project to completion.

Word cannot express the immense gratitude I have for **Parents** who has been instrumental in shaping my career. I am thankful to all our **lecturers** and **friends** who have been the sources of inspiration to all endeavors.

Yours Sincerely,

HARISH THANIKAIVELU(1VE19CS052)
HARSHA B A(1VE19CS054)
NEETHUSHREE K(1VE19EC070)
PRIYANKA B H(1VE19EC082)

Table of Contents

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

India has a vast variety of climatic conditions and crops these days are all hybrid in nature hence in order to support during the climatic changes every day there is huge challenge for the farmers and when doing so there is certain similar kind of diseases or deficiency problems developed in the crops this can sometimes be fatal for the crops. With most crops being hybrid in nature, it becomes even more critical to provide personalized care to each crop based on the specific problems they encounter. Hybrid crops are created by crossing two different varieties of the same species to produce offspring with desirable traits. While these crops offer many benefits such as improved yield, disease resistance, and better adaptability to different environments, they also require specialized care to ensure their optimal growth and development. The hybrid crop has unique characteristics that make it different from other varieties. This means that personalized care is necessary to address specific issues that may arise during their growth cycle.

Providing personalized care to each hybrid crop requires a deep understanding of the crop's genetic makeup, growth requirements, and potential stress factors. This involves monitoring the crops regularly, identifying any problems early, and implementing appropriate solutions to address them. The project aims to be a personalized care given to the crops for the problems arising in the crops and provide proper fertigation for the crops to survive and produce good yields.

Fertigation: Fertigation is a method of fertilizer application in which fertilizer is incorporated within the irrigation water by the drip system. In this system fertilizer solution is distributed evenly in irrigation. The availability of nutrients is very high therefore the efficiency is more. In this method liquid fertilizer as well as water soluble fertilizers are used. By this method, fertilizer use efficiency is increased from 80 to 90 per cent. Water is added to the fertilizer solution, which is then dispersed uniformly throughout the soil. This process makes sure that nutrients are available to plants at the proper times and in the right amounts, increasing productivity and decreasing waste.

Fertigation permits the use of readily absorbed by plants liquid fertilizers and water-soluble fertilizers. As a result, nutrients are more readily available, which boosts crop yields and quality.

1.1 What is Smart Farming?

Smart farming or Smart Agriculture system is the term used to describe the adoption of modern information and communications technologies in order to enhance, monitor, automate or improve agricultural operations and processes. Factors causing plant diseases: Pathogens are the main reason for causing diseases in the plant. There is a department named after it called plant pathology it mainly deals with the study of the pathogen. There are two main factors which cause disease in plants and they are pathogens and environmental condition. The microbes favouring plant diseases is enumerated below:

- Viral.
- Fungi.
- Bacteria.

Figure 1.1 AI in agriculture

1.2 Overview

Some of the papers are based on the explanation of how to incorporate the modern technologies such as AI and IoT in agriculture but it is not practically put to use, this is major drawback. This is overcome by the model that is being created in the project. This is done by using two modules which is AI and IoT which works hand in hand to identify the disease and treat the disease using the exact amount of fertilizer, this can reduce the excessive amount of fertiliser that is being used which in turn reduces the soil pollution, maintains fertility of the soil and produces high yields whilst performing cost control for the farmers. Additionally, reducing the amount of fertilizers used can benefit the environment by reducing the amount of greenhouse gases that are released into the atmosphere.

1.3 Problem Formulation

Problem:

Crops these days are prone to various diseases and these diseases can sometimes be fatal for the crops and the climatic conditions play a major role in the well-being of the crops.

There are many deficiencies, these deficiencies are also associated because of soil conditions namely soil being less humid, temperature conditions and many more these conditions are responsible for the health of the crops.

Solution:

Develop an AI based system to decode the diseases the crops are affected with based on scanning the leaves and analyse what could be the possible fertiliser that could be given to the crop in order to cure the crop from the disease.

1.4 Objectives

The objectives of this project are as follows:

- To employ AI and IOT into agriculture
- To provide precise amount of fertigation to the crops.
- To analyse the disease prone crops and treat it with precise fertilisers.
- To have quality produce from the crops.
- To increase the yields.
- To implement cost control for the farmers.

1.5 Advantages

The advantages of AI technology are:

- **Efficiency**: AI systems can automate repetitive tasks, freeing up human workers to focus on more complex and strategic work.
- **Accuracy**: AI can perform tasks with a high degree of accuracy and consistency, reducing errors that can result from human error or fatigue.
- **Speed**: AI systems can process and analyse large amounts of data much faster than humans, making them useful for tasks like data analysis and decision-making.
- **Personalization**: AI can learn from data and personalize experiences for individuals, such as recommending products or content based on past behavior.

- **24/7 Availability**: AI systems can work around the clock, providing support and services at any time of day or night.
- **Cost reduction**: By automating tasks and reducing errors, AI can potentially save organizations time and money.

The advantages of IoT technology are:

- **Improved efficiency**: IoT can help automate tasks and make them more efficient by connecting devices and allowing them to communicate with each other, reducing the need for human intervention.
- **Real-time monitoring**: IoT devices can gather data in real-time, providing up-to-date information for decision-making and troubleshooting.
- **Cost reduction**: IoT can help reduce costs by improving efficiency, automating tasks, and reducing waste.
- **Enhanced safety and security**: IoT can help improve safety and security by monitoring and controlling devices remotely, detecting potential hazards, and responding quickly to emergencies.
- **Predictive maintenance**: IoT can help detect potential issues before they become major problems, allowing for proactive maintenance and reducing downtime.

1.6 Disadvantages

- **Dependence**: Being too dependent on the technology makes humans lazy.
- **Cost and Complexity**: The cost of the system is an added expense for the farmers which is apart from the expense involved in the farming.

1.7 Applications

The applications of AI (artificial intelligence) and IoT (Internet of Things) in agriculture can help improve efficiency, productivity, and sustainability. Here are some examples:

Precision agriculture: AI and IoT can be used to collect data on crop health, soil conditions, and weather patterns to optimize crop yields and reduce waste. For example, sensors and drones can be used to collect data on crop growth and detect diseases and pests, allowing farmers to act before problems escalate.

Smart irrigation: IoT sensors can be used to monitor soil moisture levels and weather conditions to optimize irrigation and reduce water waste. AI algorithms can analyse data to determine the best time and amount of water to apply to crops.

Crop management: AI can be used to analyse data on crop growth and yield to predict future outcomes and optimize planting and harvesting schedules. This can help farmers reduce waste and improve profitability.

Livestock management: IoT sensors can be used to monitor animal health, behaviour, and performance, allowing farmers to identify and address issues quickly. AI can be used to analyse data to predict disease outbreaks and optimize feed and water intake.

Supply chain optimization: AI can be used to analyse data on supply and demand to optimize logistics and reduce waste. For example, AI algorithms can help farmers determine the best time to sell crops to maximize profits.

Overall, AI and IoT can help farmers make data-driven decisions, optimize resource use, reduce waste, and improve profitability and sustainability in agriculture.

1.8 Chapter Summary

This chapter gives and insight to the basics of AI and IoT technology in agriculture and smart farming using Fertigation.

It involves the overview, problem formulation, objectives, advantages, disadvantages, and applications of AI and IoT technology in Agriculture domain.

CHAPTER 2

LITERATURE SURVEY

[1] Monirul Islam et. al proposed a solution for monitoring the plants health using IoT. This is just done as an explanation and not put to use. The Abstract of the above paper can be mentioned as a summary as follows, A new era in plant health monitoring has begun thanks to the internet of things (IoT), environmental sensors, and picture processing technology. Using image processing and environmental sensing data to classify plant diseases early on not only helps farmers grow healthy plants but also increases crop yields. IoT is necessary to send photographs and provide feedback in order to track and categorise plant disease outbreaks. In this study, an Internet of Things (IoT) device based on a Raspberry Pi is proposed. It sends photos of plants to identify diseases and continuously updates environmental factors in a MySQL database, including air temperature, humidity, soil moisture, and pH.

The system has the following components: System Module and Structure, Communication system, Plant disease detection and classification. This system Ends in the explanation of plant health monitoring system and only detects and classifies the image.

[2] J. Devare et. al proposed a solution for crop growth monitoring the plants health using IoT along with that it also measures the quality of the vegetation which helps in getting better income to the sowed farmers and production of good quality crops helps the customers to have good health. In summary, the survey provides an overview of the state-of-the-art in IoT-based agricultural systems, with a specific focus on crop growth monitoring and quality control. It discusses the benefits, challenges, and future directions of utilizing IoT technology in agriculture to enhance productivity, optimize resource utilization, and ensure high-quality crop production.

[3] T. Rajeshwari et. al proposed an agricultural practice involving IoT and an AI method Logistic regression to monitor and maintain the crops for the farmers. The diseases caused in plants may be because of the bacterial, fungi and other viral infections when it is fully based only on the microorganisms this diseases can be treated by means of supplying the adequate amount of fertilisers. The proposed solution in this paper has the following abstract that is being summarised, The idea is to increase farmer procurement in the agricultural industry by reducing human participation. Most of the time, farmers struggle to produce a good crop, which lowers their income. Numerous important factors, including a shortage of minerals, soil dampness, temperature variations, and others, contribute to this phenomenon. Moreover, both the quality and quantity of the harvest are impacted by the high prevalence of diseases in

the crop. Smart agriculture is a revolutionary concept since IoT sensors can readily monitor and offer information about agricultural landscapes. In this article, we'll show you how to build a smart agriculture system using Node MCU, IoT, Android, Wireless Sensor Networks, and Machine Learning. Increasing agricultural productivity requires careful observation of the climate and early detection of plant diseases. The suggested system has the capability to send SMS alerts and a notification on the app designed for the same on the farmer's smartphone using Wi-Fi/3G/4G. It can also monitor temperature, humidity, and wetness using sensors using Node MCU. The system makes advantage of a duplex communication link built on a cellular Internet interface. By identifying plant illnesses earlier, farmers can profit more from this dependable, non-destructive technique. It was possible to construct a system for smart farming and plant disease detection using IoT and ML by using three classes of tomato plants (two infected and one healthy). The major flaw in this proposed solution is that there is only one method of AI ie;, logistic regression to train the AI model which is complicated and only shows the relationship between trends in the given values and predicts the disease based on the same. This must be overcome in order to have a better AI model trained and used to predict the diseases appearing in the crops.

[4] F. Isinkaye et. al proposes a solution to detect the diseases in the crops by using machine learning techniques and this is done by the machine learning algorithms and hence it is only focussed on the detection of diseases in the crops and does not include the IoT model to supply the fertiliser it is just used only for the disease prediction using the machine learning techniques and the concept of supplying fertilisers and treating the disease prone crops is not used in the paper. The abstract of the above paper is comprehended and the outline of the paper is discussed as follows, Worldwide, plant diseases significantly reduce food productivity, thus much research has gone into improving the methods for identifying and treating plant diseases. Farmers would greatly benefit from being able to use the present technology to take advantage of the difficulties in agricultural production and therefore increase crop yield and operation profitability. In this study, we used machine learning (ML) techniques to create and construct a user-friendly smartphone-based plant disease detection and treatment recommendation system. A content-based filtering recommendation algorithm was used to suggest appropriate therapies for the discovered plant diseases after classification, while CNN was utilised for feature extraction and the ANN and KNN to categorise the plant diseases. Here, CNN refers to Convolutional neural networks, ANN is abbreviated as Artificial neural networks and KNN is K nearest neighbourhood algorithm, all of these are part of machine learning techniques that is used to monitor the crop growth and detect the diseases that appear in the crops and provide the result after successful tests of the image samples.

2.1 CHAPTER SUMMARY

A literature review is an essential chapter in any research project, as it provides a comprehensive analysis of existing research and knowledge on the topic of study. The purpose of this chapter is to critically review and summarize the available literature related to the project's objectives and research questions. The literature review begins with an introduction to the project's topic, followed by an explanation of the selection criteria and search strategy used to identify relevant literature. The chapter then presents a systematic review of the literature, including key findings, theories, and concepts related to the project. The literature review concludes with a summary of the major findings, identified gaps, and limitations in the existing literature, which will help to inform the research approach and methodology in the subsequent chapters of the report.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

A system requirements specification, sometimes referred to as a software requirements specification, is a document that details the features and operations of a system or software program. It is made up of several parts (see list below) that try to define the functionality that the client needs in order to satisfy all their different customers.

3.1 HARDWARE REQUIREMENTS

The hardware components are listed and explained below.

3.1.1 COMPUTER OR LAPTOP

A computer or laptop is an electronic device that is capable of executing a wide range of tasks by following a set of instructions known as computer programs. These devices have become an essential part of our daily lives, as they can perform complex computations, store and manipulate large amounts of data, and connect us to the digital world. Storage devices are used to store data and programs permanently on the computer A computer is composed of several hardware components and it communicates with the other components of the computer, including a central processing unit (CPU), memory (RAM), storage devices (such as hard drives and solid-state drives), input/output (I/O) devices (such as keyboards, mice, and monitors), and various peripheral devices (such as printers and scanners).

Figure 3. 1 Computer or Laptop.

The CPU is the "brain" of the computer, responsible for executing instructions and performing calculations. It is composed of multiple cores that can work together to perform multiple tasks simultaneously, and it communicates with the other components of the computer through a set of electrical pathways known as the motherboard. The RAM is the primary memory of the computer, which stores the data and instructions that the CPU is currently working with. The more RAM a computer has, the more data it can store and process simultaneously, which can significantly improve its performance. Hard drives are the traditional storage devices that use spinning disks to store data magnetically, while solid-state drives (SSDs) use flash memory chips to store data electronically.

3.1.2 NODE MCU

NodeMCU is an open-source development board that is specifically designed to facilitate the rapid prototyping and building of IoT devices. It is built for Wi-Fi module and includes a built-in USB-to-serial converter that simplifies the process of connecting it to a computer for programming using the Lua scripting language. One of the main advantages of NodeMCU is its low cost, small size, and built-in Wi-Fi capabilities. These features make it an excellent choice for IoT projects that require wireless connectivity, such as home automation, sensor networks, and remote monitoring.

Figure 3. 2 Node MCU.

NodeMCU is also a popular choice because of its supportive and growing community of developers and users who share code, tutorials, and tips for working with the board. Additionally, it is a versatile platform that can be programmed using various programming languages such as Lua, Micro Python, or the Arduino IDE. This flexibility makes it accessible to developers with different backgrounds and programming preferences**.**

3.1.3 ATMEGA328

The ATmega328 is a popular 8-bit microcontroller chip produced by Atmel, which is now owned by Microchip Technology The ATmega328 microcontroller has 32 KB of flash memory, 2 KB of SRAM, and 1 KB of EEPROM. It operates at a clock speed of up to 20 MHz, and it has 23 programmable input/output pins, including 6 analog inputs. It also includes several built-in hardware features, such as a 10-bit analog-to-digital converter, pulse-width modulation (PWM), interrupts, timers, and serial communication interfaces.

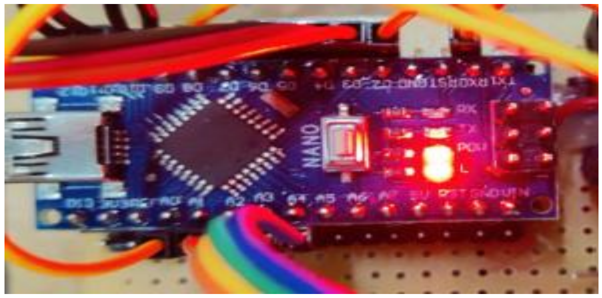


Figure 3. 3 ATmega328.

The ATmega328 is a member of the AVR microcontroller family and is compatible with the Arduino development platform, making it a popular choice for hobbyists and makers. It can be programmed using the Arduino integrated development environment (IDE) or other software tools, and it supports various programming languages, including C and C++.Overall, the ATmega328 is a versatile and widely used microcontroller chip that offers a balance of processing power, memory, and built-in hardware features. Its compatibility with the Arduino platform has made it a popular choice for hobbyists and makers, while its robustness and reliability have made it a preferred choice for industrial and commercial applications.

3.1.4 PUMP MOTOR

A pump motor is an electric motor that is designed to power a pump. A pump is a device that moves fluids, such as water or oil, from one place to another. Pump motors are used in a wide variety of applications, including water pumps for irrigation, swimming pools, and hot tubs, oil pumps for engines and machinery, and chemical pumps for industrial processes. A pump motor typically consists of a stator and a rotor. The stator is the stationary part of the motor, which contains the windings that produce the magnetic field. The rotor is the rotating part of the motor, which is connected to the pump shaft and rotates to move the fluid.

Figure 3. 4 Pump Motor.

When an electric current is applied to the stator windings, it produces a rotating magnetic field that interacts with the magnetic field of the rotor. This interaction causes the rotor to rotate, which in turn rotates the pump impeller or propeller, moving the fluid through the pump and out the other side. Pump motors can be powered by alternating current (AC) or direct current (DC) electricity, depending on the application. AC pump motors are typically used in residential and commercial applications, while DC pump motors are often used in automotive and marine applications.

3.1.5 SOLENOID VALVE

A solenoid valve is an electromechanical device that is used to control the flow of fluids, such as water, air, or gas, through a pipe or tubing. It consists of a coil of wire, called a solenoid, that is wrapped around a movable ferromagnetic core. When an electric current is passed through the coil, it creates a magnetic field that pulls the core towards it, opening or closing a valve. The solenoid valve is typically composed of two main parts: the solenoid and the valve body. The solenoid is the electrical component that controls the opening and closing of the valve. The valve body contains the inlet and outlet ports, as well as the valve mechanism, which is either a poppet or a spool.

Figure 3. 5 Solenoid Valve.

In a normally closed solenoid valve, the valve is closed when no power is applied to the solenoid, and it opens when power is applied. In a normally open solenoid valve, the valve is open when no power is applied, and it closes when power is applied. Solenoid valves are used in a variety of applications, including irrigation systems, pneumatic and hydraulic control systems, fuel systems, and HVAC systems. They are reliable and efficient, with fast response times and the ability to operate in harsh environments.

3.1.6 RELAY

The 4-channel relay module typically consists of four individual relays, each with its own input and output pins. The input pins are connected to a microcontroller or other control device, and the output pins are connected to the devices you want to control. When a signal is received on one of the input pins, the corresponding relay is activated, allowing the current to flow through the output pins and turn on or off the connected device. The 4-channel relay module can be used in a wide range of applications, such as home automation, industrial automation, and robotics. For example, you could use a 4-channel relay module to control the lights, fans, and air conditioning in a room, or to switch the power on and off to multiple motors in a manufacturing process.

Figure 3. 6 Relay module

One of the advantages of using a 4-channel relay module is that it simplifies the wiring process and reduces the number of control signals required. It also allows you to control multiple devices using a single control signal, which can save time and reduce the complexity of your control system. Overall, a 4-channel relay module is a versatile and useful device that can simplify the process of controlling multiple circuits. It is reliable and easy to use, making it a popular choice for a wide range of applications.

3.1.7 16x2 LCD

A 16x2 LCD (Liquid Crystal Display) is a type of alphanumeric display that can show up to 16 characters on each of its two lines. It is a popular display module used in a wide range of electronic devices, including calculators, digital clocks, and small appliances. The 16x2 LCD display consists of a thin layer of liquid crystal material sandwiched between two glass plates, with electrodes at the edges of the plates. The display can be controlled by sending electrical signals to the electrodes to align the liquid crystal molecules, creating a pattern of light and dark segments that form the characters.

Figure 3. 7 16x2 LCD.

The display is typically driven by an integrated circuit called an LCD controller, which generates the signals needed to control the display. The 16x2 LCD display module typically has 16 pins, which are used to connect the display to a microcontroller or other control device. The pins are used for power supply, data input, and control signals such as enable, register select, and read/write.

3.1.8 11 DHT

The 11 DTH sensor is a type of digital temperature and humidity sensor that is commonly used in various applications, including agricultural systems, weather stations, and indoor climate control systems. The sensor is capable of measuring both temperature and humidity, and it provides accurate and reliable readings. The 11 DTH sensor consists of a sensing element that is made up of a thermistor for temperature measurement and a capacitive polymer sensor for humidity measurement. The sensor also contains an analog-to-digital converter and a digital signal processing unit that converts the analog signals into digital data that can be easily processed by a microcontroller or a computer. The sensor is designed to operate in a wide range of temperatures and humidity levels, making it suitable for use in various environments. It also has a low power consumption, which makes it ideal for use in battery-powered devices or systems that require long-term monitoring.

3.2 SOFTWARE REQUIREMENTS

The software components are listed and explained below.

3.2.1 Blynk app for IoT Controls:

Blynk is a mobile application that provides a user-friendly interface for controlling IoT devices. It allows users to monitor and control their connected devices remotely through a smartphone or tablet. Blynk app supports a wide range of IoT modules and sensors, and it is compatible with various platforms such as Arduino, Raspberry Pi, and ESP32.The app offers several built-in widgets, such as buttons, sliders, graphs, and notifications, which users can use to interact with their connected devices. With Blynk, users can receive real-time notifications and alerts based on the data collected by their IoT devices. Blynk also provides an extensive library of pre-built projects and examples, making it easy for beginners to get started with IoT. The app allows users to share their projects and collaborate with other Blynk users worldwide. Furthermore, Blynk provides a cloud-based server that stores data collected by the IoT devices, making it accessible from anywhere and at any time.

3.2.2 512 MB Graphic Card:

A graphics card (also known as a video card or GPU) is a computer hardware component that is designed to handle the rendering of images and videos, especially in applications that require 3D graphics, such as video games, 3D modelling, and animation. The amount of memory on a graphics card is one of its most important specifications, as it affects the card's ability to handle complex and high-resolution graphics. 512 MB of memory is considered a relatively small amount by modern standards, as many newer graphics cards have 4 GB, 8 GB, or even 16 GB of memory. Despite its small memory capacity, a 512 MB graphics card can still be useful for certain tasks, such as basic video editing, web browsing, or running older video games that don't require a lot of graphical horsepower. However, it may struggle with more demanding tasks or newer games that require more memory and processing power. When selecting a graphics card, it is important to consider not just the amount of memory, but also other factors such as the card's clock speed, bus width, and number of processing cores. These specifications will affect the card's overall performance and ability to handle different types of graphics-intensive tasks. Graphics cards can be expensive, with some high-end models costing several hundred or even thousands of dollars. However, there are also more affordable options available for users who do not require top-of-the-line performance. It's important to do research and choose a card that fits your specific needs and budget.

3.2.3 Intel Core i5 Processor

The Intel Core i5 processor is a mid-range CPU (central processing unit) developed by Intel Corporation. It was first introduced in 2009 and has since undergone several upgrades and revisions. The i5 processor is known for its balance of performance and power efficiency, making it a popular choice for both desktop and laptop computers. The latest generation of i5 processors, based on Intel's 11th generation "Tiger Lake" architecture, features up to 4 cores and 8 threads, with clock speeds up to 4.2 GHz. The i5 processor also supports Intel's Hyper-Threading technology, which allows for better multi-tasking performance by simulating additional cores. The i5 processor is commonly used in gaming computers, business laptops, and home desktops, where it can handle a variety of tasks such as web browsing, video editing, and light gaming. To take full advantage of an i5 processor, it is recommended to pair it with a compatible motherboard, sufficient RAM, and a fast storage drive.

3.2.3 OPENCV LIBRARY

OpenCV (Open-Source Computer Vision) is a library of programming functions mainly used for real-time computer vision tasks such as image and video processing. It is an open-source computer vision and machine learning software library that includes several hundreds of computer vision algorithms. OpenCV is cross-platform and can be used with a variety of programming languages such as Python, C++, and Java. The library is designed to be very efficient and optimized for real-time computer vision applications. It contains a large number of functions that can be used to process images and videos, such as image filtering, feature detection, object recognition, and motion analysis. OpenCV also includes support for several standard image and video formats, such as JPEG, PNG, MPEG, and AVI. In this project, OpenCV is used for reading and processing images. It provides various functions to load images from files and to perform operations such as resizing, cropping, and color conversion.

3.2.4 TENSORFLOW LIBRARY

TensorFlow is an open-source machine learning library developed by Google Brain Team. It is widely used for building and training deep neural networks for various applications such as image recognition, speech recognition, natural language processing, and more. TensorFlow is designed to work efficiently with large datasets and complex models. TensorFlow provides a highly flexible and modular architecture for building deep learning models. It supports a variety of popular deep learning algorithms such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Belief Networks (DBNs). TensorFlow also includes pre-built APIs for many common deep learning tasks such as image recognition, natural language processing, and reinforcement learning. One of the key features of TensorFlow is its ability to support distributed computing, which allows it to scale up to handle very large datasets and complex models.

3.2 CHAPTER SUMMARY

The chapter on software and hardware components discusses the various components used in the project. The software components include the Blynk app for IoT Controls, 3D Printer 512 MB Graphic Card and Intel Core i5 Processor. The hardware components include NodeMCU, TP 7805 Transistor, BC 547, DC Voltage booster and DC Submersible Pump. The chapter provides a detailed explanation of each of these components, including their purpose, functioning, and relevance to the project. The information presented in this chapter will help readers understand the technical aspects of the project and the tools and technologies used to implement it.

CHAPTER 4
SYSTEM ANALYSIS

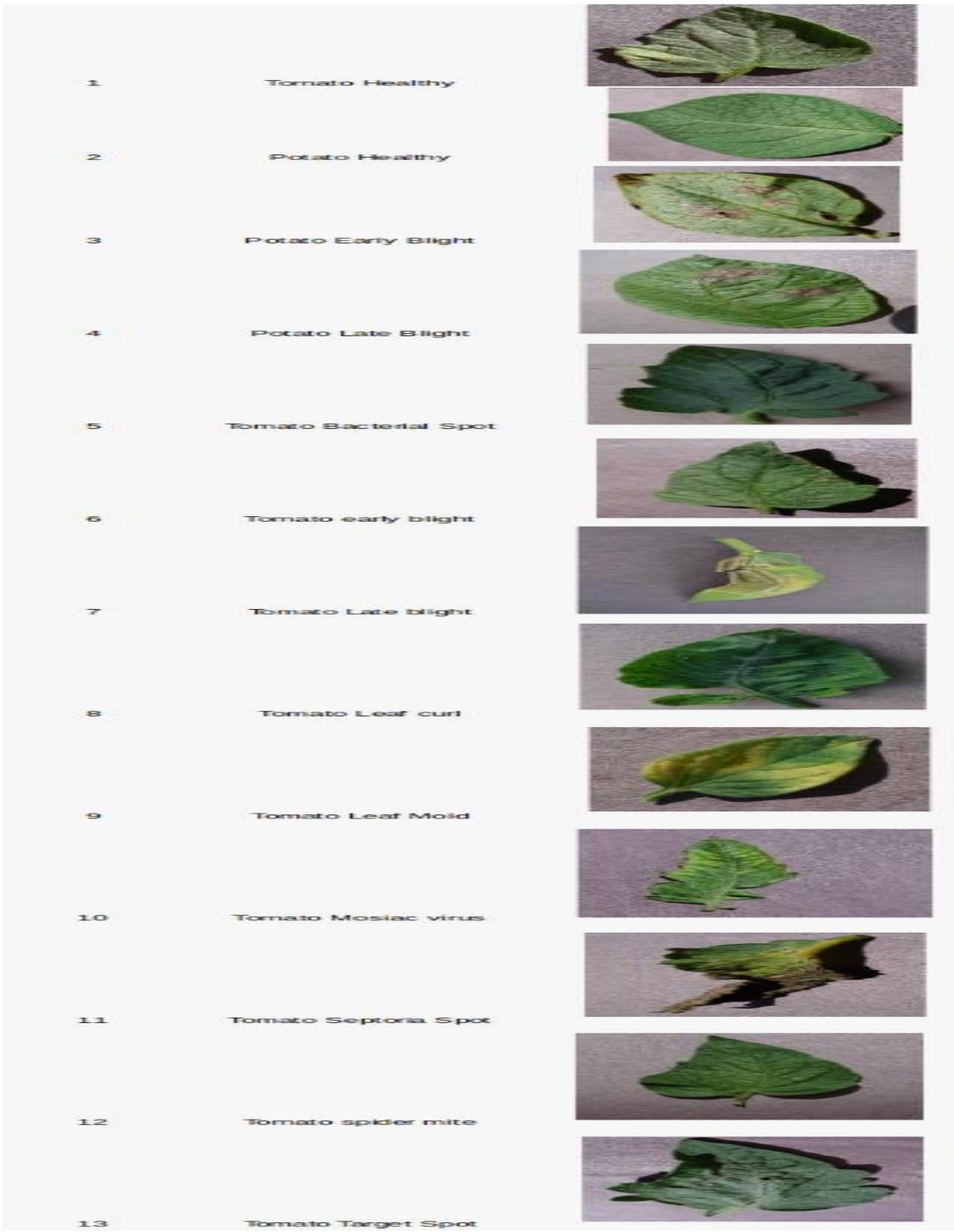
System analysis is done to investigate a system or its components in order to pinpoint goals. It is a method of resolving issues that enhances the system and guarantees that every part functions effectively to provide its intended function. Analysis lays out the system's proper course of action.

4.1 Analysis of the Existing System

In existing system mainly, they started by using Plant Village dataset. They analyse 54,306 images of plant leaves, which have a spread of 38 class labels assigned to them. Each class label is a crop-disease pair, and they make an attempt to predict the crop-disease pair given just the image of the plant leaf. They resize image the images to 256 × 256 pixels, and we perform both the model optimization and predictions on these downscaled images. They have used three types of versions of datasets. First, they have started with the color images dataset. Then they have used grey-scaled version of the Plant Village dataset. Final version they have used segmented version of dataset. Extra background information of image which might have the potential to introduce some inherent bias in the dataset. Steps involving in prediction of leaf being healthy or unhealthy and disease prediction:

- Data Collection: The first step in leaf detection using AI is to collect a dataset of leaf images. This dataset should include a range of leaf shapes, sizes, colors, and textures.
- Data Preparation: The collected dataset is then prepared for use in training an AI model. This involves cleaning the data, resizing the images, and labeling the images with the correct classes (i.e., healthy or diseased).
- Model Training: An AI model is then trained using the prepared dataset. This involves selecting an appropriate machine learning algorithm, such as Convolutional Neural Networks (CNNs), and optimizing the model's parameters through trial and error.
- Model Evaluation: Once the AI model is trained, it is evaluated on a separate test dataset to assess its accuracy and generalization ability. This involves measuring metrics such as precision, recall, and F1 score.
- Deployment: The trained AI model can then be deployed in a leaf detection system. This system can be designed to automatically detect and classify the leaves as healthy or diseased in real-time. The output can be displayed on a user interface other systems for further processing.
- Model Maintenance: Finally, the AI model needs to be maintained and updated regularly to ensure

its accuracy and relevance over time. This can involve retraining the model on new data, fine-tuning the model's parameters, or updating the algorithm altogether.



4.2 Proposed Work and Approach

Proposed work is to find the crops which are grown in local region (INDIA) and list the disease which affects on those crops. Next step is to make pair of crop-disease and train the model again. After identifying the new pair of crop-disease obtain accuracy again. For that We also prepared a architecture which is called as " CropDiseaseNet". "Crop Disease Net" architecture is specially designed to identify the disease from crop leaf image. As a shown figure of Crop Disease Net Architecture, an image of crop image converts into 224 x 224 x 3, where 3 is for a color image. Crop Disease Net contains 5 convolutional layers and 3 fully connected layers. In Crop Disease Net, Relu is applied after very convolutional and fully connected layer and dropout are applied before the first and the second fully connected year. This network has 62.3 million parameters and needs 1.1 billion computation units in a forward pass. As we can also see convolution layers, which accounts for 6% of all the parameters, consumes 95% of the computation.

4.3 Chapter Summary

The chapter discusses the various steps involved in leaf disease detection, including data collection, pre-processing, feature extraction, model selection, training, and evaluation. It explains how deep learning algorithms such as Convolutional Neural Networks (CNNs) can be used to analyze plant images and detect disease symptoms accurately. The chapter also highlights some of the challenges involved in leaf disease detection, such as dataset bias, imbalanced classes, and overfitting. It describes various techniques to overcome these challenges, such as data augmentation, transfer learning, and regularization. Furthermore, the chapter discusses some of the current applications and future directions of leaf disease detection using machine learning. It explains how these techniques can be applied in precision agriculture to monitor crop health, optimize pesticide use, and reduce crop losses.

CHAPTER 5

PROPOSED WORK

Proposed to develop two modules namely:

AI module: This module is responsible for disease prediction in the crops by scanning the leaves.

IoT Module: This module is responsible for the fertigation of the crops based on the data provided by the AI module.

These two modules must be integrated and coordinated to be used.

5.1 METHODOLOGY

Here is a detailed methodology along with work carried out for the Design of automatized disease detection and fertilization system for agricultural crops. AI Module follows any one of the below methods to predict the diseases in crops:

- Supervised learning method: It is a type of machine learning where the algorithm is trained on a labelled dataset, meaning that the correct output for each input is provided. The algorithm then uses this labelled data to make predictions on new, unseen data.
- Unsupervised learning method: It is a type of machine learning where the algorithm is trained on an unlabelled dataset, meaning that there is no known output for each input. Instead, the algorithm must find patterns and relationships in the data on its own.
- Semi-supervised learning method: It is a type of machine learning that combines elements of supervised and unsupervised learning. In semi-supervised learning, the algorithm is trained on a combination of labelled and unlabelled data.
- Reinforcement learning method: It is a type of machine learning where the algorithm learns through trial and error, receiving feedback in the form of rewards or penalties based on its actions. The algorithm then uses this feedback to adjust its behaviour and improve its performance over time.

The utilization of AI module's various techniques such as supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning has revolutionized the agriculture industry by providing an intelligent and efficient system to predict crop diseases. These methods have greatly enhanced the crop yields by accurately identifying the diseases and taking appropriate actions in a timely manner.

Units in IoT module for Design of automatized disease detection and fertilization system for agricultural crops are enumerated below:

- Capturing Temperature and Humidity: This unit captures temperature and humidity data using DHT22 sensors, which monitor the air temperature and humidity. This information is critical for ensuring optimal crop growth conditions and preventing issues like mild or fungal growth.
- Capturing Soil Moisture Humidity: This unit captures soil moisture levels using YL-69 sensors and LM393 comparators. Monitoring soil moisture is essential for optimizing irrigation schedules and ensuring that crops receive the appropriate amount of water.
- Solenoid Valve: This unit is an electromechanical device used to control the flow of water or other fluids to crops or irrigation systems. Solenoid valves enable farmers to automate their irrigation systems and ensure that crops receive the appropriate amount of water.
- Fertilizer Tank: This unit stores nitrogen, phosphorus, and potassium fertilizers, which are commonly used in agriculture to promote plant growth and increase crop yield. Fertilizer tanks deliver these fertilizers to crops in a controlled manner, ensuring that crops receive the appropriate nutrients to thrive.

In addition to the AI module, the IoT module plays an equally important role in agriculture. It consists of various units, each designed to perform a specific function that contributes to the overall success of crop cultivation. The capturing of temperature and humidity through DHT22 sensors provides valuable data that enables farmers to determine optimal planting times and to make informed decisions about irrigation and ventilation. The monitoring of soil moisture humidity using YL-69 sensors and LM393 Comparator helps farmers to ensure that the soil is adequately hydrated, which is critical to plant growth and health. The solenoid valve is an essential component of the IoT module that controls the flow of water to the crops. Monitoring soil moisture is essential for optimizing irrigation schedules and ensuring that crops receive the appropriate amount of water. is an essential component of the IoT module that controls the flow of water to the crops. Monitoring soil moisture is essential for optimizing irrigation schedules and ensuring that crops receive the appropriate amount of water. This device ensures that the plants receive the right amount of water, which helps to prevent water wastage and ultimately leads to higher crop yields. Furthermore, the IoT module includes tanks containing nitrogen, phosphorus, and potassium fertilizers, which are essential nutrients for plant growth. Overall, the combination of the AI module and the IoT module has revolutionized the way crops are grown, enabling farmers to optimize their yields and minimize losses due to disease and environmental factors.

5.2 BLOCK DIAGRAM

The block diagram of our project consists of two model AI model and IOT model. The process below continues each time with just a click of a button thus reducing the time spent by the farmers and increasing the precision in farming leading to better yields that are of high quality. The Block Diagram for the AI Module in separate is Given below

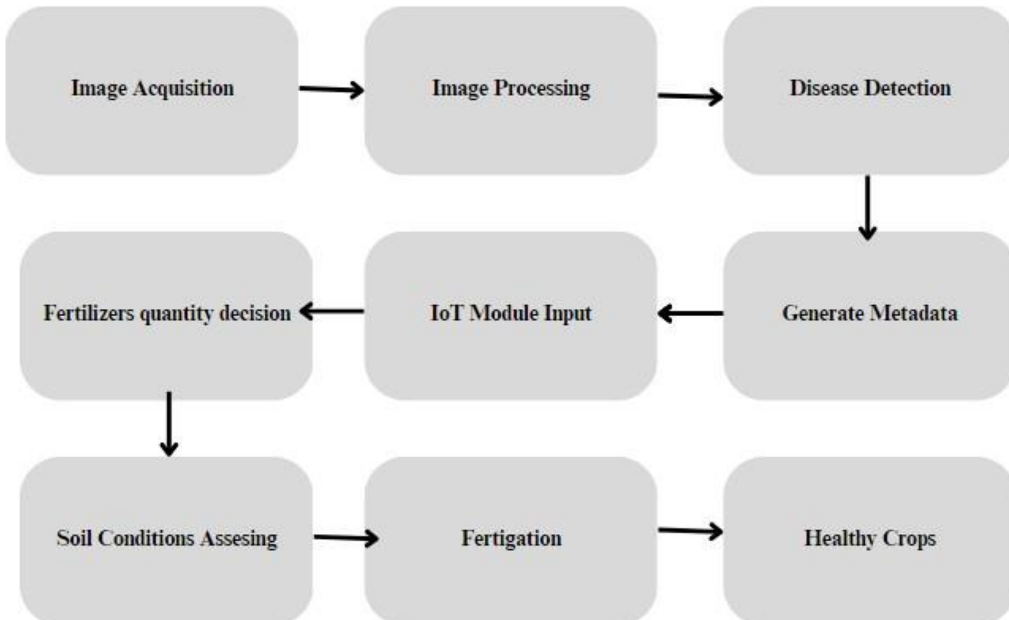


Figure 5. 1 Proposed Model

Capturing images manually can be a time-consuming and laborious process, which can be overcome by automating the image capture process using IoT technology. This technology involves the use of cameras, sensors, and other devices that capture images of crops at regular intervals. These images can then be uploaded to the AI module for analysis Once the images are uploaded, the AI module performs its analysis, identifying any patterns or anomalies that may indicate the presence of disease or other issues that could impact crop yield. The resulting data is then interfaced with the IoT module, which can take appropriate actions based on the data, such as adjusting irrigation or fertilizer levels, to improve crop health and yield.

Python is an ideal language for developing the code for the AI module. Its versatility, ease of use, and extensive library of machine learning and image processing analyse images and provide valuable insights into crop health. Furthermore, Python's popularity among data scientists and developers means that there is a wealth of knowledge and resources available to help develop and maintain the AI module's code.

5.3 ALGORITHM

Here is a step-by-step algorithm for building the Design of automatized disease detection and fertilization system for agricultural crops:

- Begin the image capture process. If manual, user to take a photo of the crop leaves. If automated, trigger the ESP32 Cam module to capture an image at a regular interval.
- Upload the captured image to the cloud or a server for analysis.
- Process the image using image processing techniques to identify any signs of disease.
- Generate the result of the analysis, which may include a diagnosis of the disease, the severity of the infection, and recommended treatment options.
- Interface the result with the IoT module to provide farmers with real-time updates on the condition of their crops.
- Control the Fertigation system and adjust the application of fertilizers and water based on the analysis results.
- Use the Blynk mobile application to control the IoT module remotely.
- Code the AI module in Python, incorporating machine learning algorithms such as supervised, unsupervised, semi-supervised, or reinforcement learning to improve the accuracy of the analysis results.

The algorithm provides a clear and structured approach for building the project. The process involves capturing an image of the crop leaves either manually or automatically using the ESP32 Cam module. The image is uploaded for analysis and processed to identify any signs of disease. crop disease detection using computer vision techniques like image processing and analysis. The high-level programming language Python, which is renowned for being straightforward and user-friendly, is used to create the image processing algorithm. The IoT (Internet of Things) module, which manages the Fertigation system, is then interfaced with the results of the disease detection process. The application of fertiliser to crops via an irrigation system is known as fertilisation. The IoT module is set up to receive the findings of disease detection and modify fertiliser delivery based on the crop's need. The machine learning algorithms are utilised to analyse the data and categorise the photos as either unhealthy or healthy crops once the images have been processed. In order to effectively categorise fresh photos, these machine learning methods are also implemented in Python and trained on a dataset of cropped photographs.

5.4 CHAPTER SUMMARY

The chapter discusses the methodology, block diagram, and algorithm for building a Design of automatized disease detection and fertilization system for agricultural crops. The agricultural sector may gain a lot from automating the crop image capturing process utilising IoT technologies and AI analysis. Farmers can swiftly identify agricultural diseases or other problems by automating the process, decreasing crop losses and raising output. Additionally, the adoption of AI enhances system efficiency by enabling more precise and reliable crop issue detection.

Python is the best language for writing the AI module's code. Python is an advanced programming language that is simple to read and write, making it the perfect choice for both novices and specialists. Python is a great choice for developing AI, machine learning, and data analysis because of its extensive library and framework ecosystem. A step-by-step process for developing the project, from image capture to remote control using the Blynk mobile application, is provided by the algorithm developed for the system. Using IoT devices like cameras or drones, the programme starts by taking pictures of crops. Then, important elements like colour, texture, and shape are extracted from these images using computer vision techniques.

CHAPTER 6

SYSTEM DESIGN

6.1 PLANT DISEASE

The majority of pesticides are substances used to manage weeds, pests, and plant diseases. Pesticides, both acute and chronic, are detrimental to public health, according to a WHO report. Among the acute impacts include widespread food poisoning, chemical mishaps in industry, and occupational exposure in agriculture. Cancer, poor reproductive outcomes, neurobehavioral disorders, allergic sensory reactions, immunological impacts, and many more are examples of chronic consequences. India confronts significant pest and disease risks to grape agriculture. The main insect pests are mealybugs, flea beetles, thrips, hoppers, stem borers, and mites, whereas the main fungi diseases are downy mildew, powdery mildew, and anthracnose. Table details the several grape diseases.

6.2 IMAGE ACQUISITION

For training, Image is taken from database. And for testing, you can take image from camera at real time but in this project, we made a particular folder on desktop from that image will be fetched by GUI screen and send through java web services i.e. tomcat server to server side system on which preprocessing is done and later on algorithm test that particular image.

- Image Pre-Processing

Image should be processed before sending to the algorithm for testing and training purpose. For that purpose, in this project image is scaled or resize into 150 x 150 dimension. As we used color image so that we don’t need any color conversion techniques and that pre-processed image is directly passed to algorithm for training and testing purpose.

- Image Classification

We used CNN for image classification. CNNs are biologically inspired models inspired by research by D. H. Hubel and T. N. Wiesel. They proposed an explanation for the way in which mammals visually perceive the world around them using a layered architecture of neurons in the brain, and these in turn inspired engineers to attempt to develop similar pattern recognition mechanisms in computer vision. For the implementation of CNN, we have used keras library on top of tensorflow. CNN receives the image as a matrix of pixel values. A sequence of convolution, maxpooling and normalizationis done in several layers of CNN and is finally regularized.

- Image datasets

Every image is a matrix of pixel values. The range of values that can be encoded in each pixel depends upon its bit size. Most commonly, we have 8 bit or 1 Byte-sized pixels. Thus the possible range of values a single pixel can represent is [0, 255]. However, with coloured images, particularly RGB (Red, Green, Blue)-based images, the presence of separate colour channels (3 in the case of RGB images) introduces an additional ‘depth’ field to the data, making the input 3-dimensional. Hence, for a given RGB image of size, say 255×255 (Width x Height) pixels, we’ll have 3 matrices associated with each image, one for each of the color channels. Thus the image in it’s entirety, constitutes a 3-dimensional structure called the Input Volume (255x255x3).

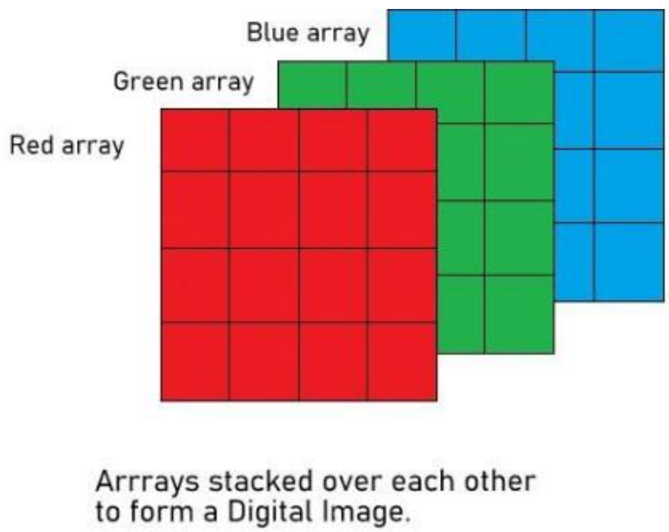


Figure 6. 1 Image as Pixel Matrix

- Convolution

A convolutional neural network (CNN) is a type of deep learning algorithm that utilizes convolutions to extract features from images. A convolution is a mathematical operation that involves multiplying a small matrix called a kernel or filter with a part of the input image, and then summing the results to produce a single output value. By sliding the kernel over the entire image and performing the same operation at each location, we can generate a feature map that highlights certain patterns in the image. The kernel is typically smaller in size than the input image, which allows it to capture local patterns and structures in the image. Kernels are then convolved with the input volume to obtain so-called ‘activation maps’ (also called feature maps).

Kernels are then convolved with the input volume to obtain so-called 'activation maps' (also called feature maps). We compute the dot product between the kernel and the input matrix. The convolved value obtained by summing the resultant terms from the dot product forms a single entry in the activation matrix. The patch selection is then slided (towards the right, or downwards when the boundary of the matrix is reached) by a certain amount called the 'stride' value, and the process is repeated till the entire input image has been processed.

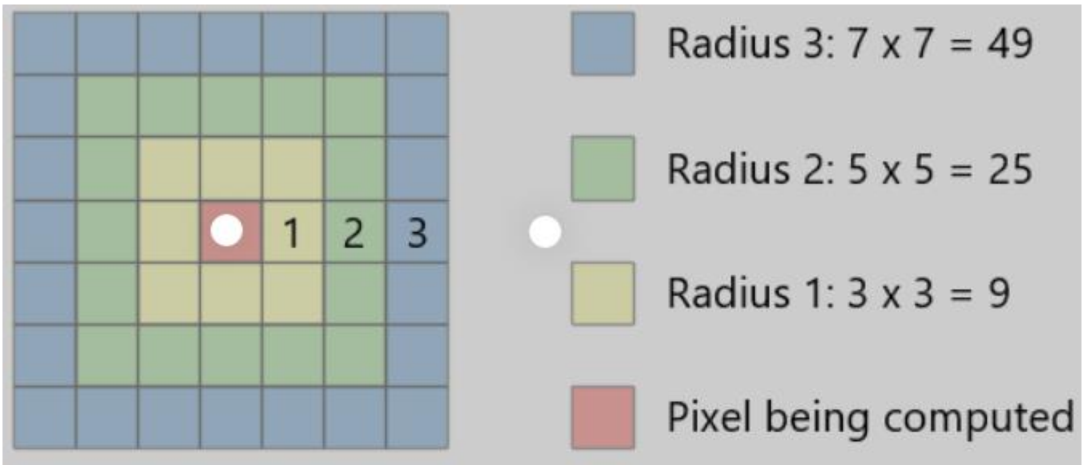


Figure 6. 2 Pixel representation of a filter

- Pooling

Pooling reduces the spatial dimensions (Width x Height) of the Input Volume for the next Convolutional Layer. It does not affect the depth dimension of the Volume. The transformation is either performed by taking the maximum value from the values observable in the window (called 'max pooling'), or by taking the average of the values. Max pooling has been favored over others due to its better performance characteristics.

- Normalization

Normalization turns all the negative values to 0 so that a matrix have no negative values. We have used ReLU activation in our case. A stack of images becomes a stack of images with no negative values.

- Regularization

Regularization is a vital feature in almost every state-of-the-art neural network implementation. To perform dropout on a layer, you randomly set some of the layer's values to 0 during forward propagation. Dropout forces an artificial neural network to learn multiple independent representations of the same data by alternately randomly disabling neurons in the learning phase

6.3 Chapter Summary

The chapter then discusses the design of the disease detection system, which involves the use of an AI model for analyzing plant images and identifying disease symptoms. The system consists of a camera module for capturing images, an image pre-processing module for preparing the images for analysis, and a deep learning model for detecting and diagnosing diseases. The deep learning model is designed using a convolutional neural network (CNN) architecture that is trained on a large dataset of labeled plant images. The chapter discusses the various steps involved in training the CNN model, including data collection, data preprocessing, model selection, and evaluation.

The chapter also describes the different techniques used for image pre-processing, such as image segmentation, feature extraction, and image normalization. It also discusses the challenges and limitations of the disease detection system, such as dataset bias, imbalanced classes, and model overfitting.

CHAPTER 7

IMPLEMENTATION AND SAMPLE CODE

7.1 IMPLEMENTATION

Transforming RGB into HSV Either the RGB or the HSV colour spaces can be used to process colour vision. Colors are defined by the proportions of red, green, and blue in the RGB colour space. In the HSV colour space, hue, saturation, and value are used to define colour. The HSV colour model is frequently favoured over the RGB model in circumstances where colour description is crucial. The HSV model defines colour perception in a way that is close to how the human eye usually does. While HSV characterizes colour using more relatable comparisons like color, vibrancy, and brightness, RGB defines colour as a collection of primary colors. The basketball robot processes color vision using the HSV colour space to determine hue, saturation, and value. Hue represents the color type. It can be described in terms of an angle on the above circle. Although a circle contains 360 degrees of rotation, the hue value is normalized toa range from 0 to 255, with 0 being red.

In the HSV color model, three parameters are used to define a color - hue, saturation, and value. Saturation indicates the intensity of the color, with a range of 0 to 255. Lower values indicate that the color has grayer in it, giving it a faded appearance. Value represents the brightness of the color, with a range of 0 to 255, where 0 represents complete darkness and 255 indicates full brightness. White has an HSV value of 0-255, 0-255, 255, while black has an HSV value of 0-255, 0-255. When it comes to black and white colors, the term "value" is the dominant descriptor, as the hue and saturation do not have an impact on the color when value is at maximum or minimum intensity.

The robot's color camera uses the RGB color model to detect colors. The RGB values obtained are then converted to HSV values, which are used in the code to locate specific objects or colors that the robot is searching for. The code checks each pixel individually to determine whether it matches a pre-defined color threshold. To convert a color image to grayscale, there are several methods available. These methods are used to transform the red, green, and blue components of a color image, R, G, B, into a grayscale image.

- The Simple Averaging method for creating a grayscale image involves taking the average of the R, G, and B values for each pixel, resulting in a single grayscale value.
- The Weighted Averaging method uses specific weights for each color channel (R, G, and B) to create a grayscale image. There are multiple ways to calculate these weights, such as 0.3R + 0.59G

+ 0.11B, 0.2126R + 0.7152G + 0.0722B, or 0.299R + 0.587G + 0.114B.

- Desaturation is the process of removing the saturation component of a color, which can be done by setting the S component in the HSI color model to 0. One method for desaturating an image is to calculate the grayscale value as the average of the minimum and maximum values of R, G, and B.
- The Decomposition method creates a grayscale image by using the maximum or minimum value of R, G, and B for each pixel.
- The Single Color Channel method creates a grayscale image by using a single color channel (R, G, or B) for each pixel.
- The Custom Number of Grey Shades method allows for the creation of a grayscale image with a specific number of shades by dividing the interval [0, 255] into subintervals based on the desired number of shades and assigning a common value to pixels within each subinterval.
- The Custom Number of Grey Shades with Error-Diffusion Dithering method is a technique for reducing the number of shades in a grayscale image while minimizing the loss of detail. It works by spreading the error of each calculation to neighboring unvisited pixels.

Color segmentation is a powerful technique that can be used for a variety of image processing applications, including object detection, tracking, and recognition. In disease detection, color segmentation can be particularly useful for identifying diseased regions of plant leaves. In the YCbCr color model, color information is separated into its luminance (Y) and chrominance (Cb and (Cr) components. The luminance component describes the brightness of a pixel, while the chrominance components describe the color information. By using specific equations, the YCbCr model can be used to segment images based on their color information.

Similarly, in the RGB color model, color segmentation is carried out by identifying pixels in a certain range for each color to be identified. This involves specifying a range of color values for each pixel in the image and identifying which pixels fall within that range. The count ratio of affected pixels to the total leaf area can then be calculated as a percentage, providing an estimate of the severity of the disease. Thresholding is another segmentation technique that can be used in disease detection. This involves setting a threshold value for the intensity or color of each pixel in the image. Pixels with intensities or colors above the threshold are considered part of the diseased region, while pixels below the threshold are considered healthy. This technique is often used in combination with other segmentation techniques to improve the accuracy of disease detection.

7.2 SAMPLE CODE

7.2.1 ALGORITHM IMPLEMENTATION

1. Importing libraries

```
import numpy as np
import matplotlib.pyplot as plt
import keras
import pandas
from tensorflow.keras.utils import img_to_array
import os
from tensorflow.keras.utils import load_img
from keras.preprocessing.image import ImageDataGenerator
from keras.applications.vgg19 import VGG19,preprocess_input,decode_predictions
from keras.layers import Dense, Flatten
from keras.models import Model
from keras.applications.vgg19 import VGG19
import keras
```

2. preprocessing the image that is given as input.

```
training_data_generator= ImageDataGenerator(zoom_range=0.5, shear_range=0.3, rescale=1/255, horizontal_flip=True)
validation_data_generator= ImageDataGenerator(rescale= 1/255)
```

3. specifying the training datasets

```
train = training_data_generator.flow_from_directory(directory="/content/New Plant Diseases Dataset(Augmented)/New Plant Diseases Dataset(Augmented)/train",target_size=(256,256),batch_size=32)
```

4. specifying the validation datasets.

```
val = validation_data_generator.flow_from_directory(directory="/content/New Plant Diseases Dataset(Augmented)/New Plant Diseases Dataset(Augmented)/valid",target_size=(256,256),batch_size=32)
```

5.fixing a variable with name " base_model" and initialising parameters for the images in datsets.

```
base_model =VGG19(input_shape=(256,256,3),include_top=False)
```

6. retrieving the model from the path specified

model=load_model('/content/best_model.h')

7. get reference from the datsets with the appropriate indices and its corresponding values

ref=dict(zip(list(train.class_indices.values()),list(train.class_indices.keys())))

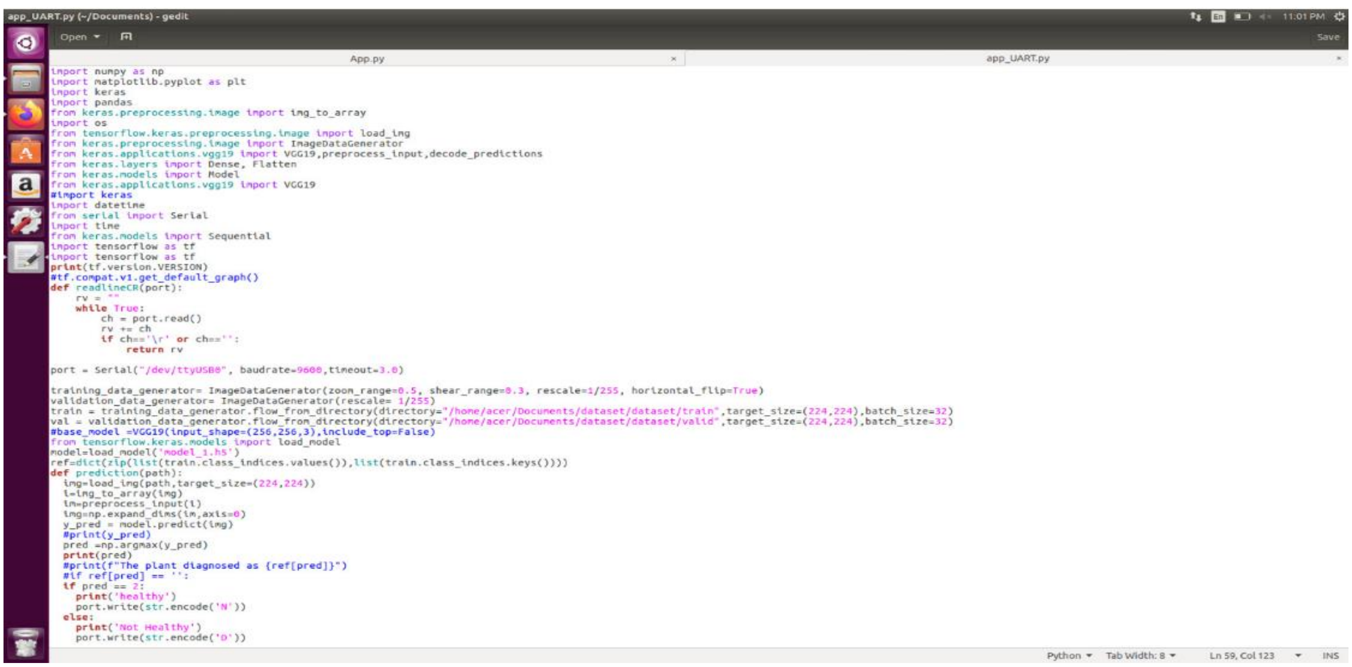
8. call the prediction function and display the output after prediction

```
def prediction(path):
img=load_img(path,target_size=(256,256))
i=img_to_array(img)
im=preprocess_input(i)
img=np.expand_dims(im,axis=0)
pred =np.argmax(model.predict(img))
print(pred)
print(f"The plant diagnosed as {ref[pred]}")
if (ref[pred])=='Potato__healthy'or 'Tomato__healthy':
  print('healthy')
else:
  print('Not Healthy')
```

7.2.2 INTERFACE DESIGN

App.py (~/Documents) - gedit

```python
import numpy as np
import matplotlib.pyplot as plt
import keras
import pandas
from tensorflow.keras.utils import img_to_array
import os
from tensorflow.keras.utils import load_img
from keras.preprocessing.image import ImageDataGenerator
from keras.applications.vgg19 import VGG19,preprocess_input,decode_predictions
from keras.layers import Dense, Flatten
from keras.models import Model
from keras.applications.vgg19 import VGG19
import keras
training_data_generator= ImageDataGenerator(zoom_range=0.5, shear_range=0.3, rescale=1/255, horizontal_flip=True)
validation_data_generator= ImageDataGenerator(rescale= 1/255)
train = training_data_generator.flow_from_directory(directory="/content/New Plant Diseases Dataset(Augmented)/New Plant Diseases Dataset(Augmented)/train",target_size=(256,256),batch_size=32)
val = validation_data_generator.flow_from_directory(directory="/content/New Plant Diseases Dataset(Augmented)/New Plant Diseases Dataset(Augmented)/valid",target_size=(256,256),batch_size=32)
base_model =VGG19(input_shape=(256,256,3),include_top=False)
from keras.models import load_model
model=load_model('/content/best_model.h')
ref=dict(zip(list(train.class_indices.values()),list(train.class_indices.keys())))
def prediction(path):
  img=load_img(path,target_size=(256,256))
  i=img_to_array(img)
  im=preprocess_input(i)
  img=np.expand_dims(im,axis=0)
  pred =np.argmax(model.predict(img))
  print(pred)
  print(f"The plant diagnosed as {ref[pred]}")
  if (ref[pred])=='Potato___healthy'or 'Tomato___healthy':
    print('healthy')
  else:
    print('Not Healthy')
```



app_UART.py (~/Documents) - gedit

```python
import numpy as np
import matplotlib.pyplot as plt
import keras
import pandas
from keras.preprocessing.image import img_to_array
import os
from tensorflow.keras.preprocessing.image import load_img
from keras.preprocessing.image import ImageDataGenerator
from keras.applications.vgg19 import VGG19,preprocess_input,decode_predictions
from keras.layers import Dense, Flatten
from keras.models import Model
from keras.applications.vgg19 import VGG19
#import keras
import datetime
from serial import Serial
import time
from keras.models import Sequential
import tensorflow as tf
import tensorflow as tf
print(tf.version.VERSION)
#tf.compat.v1.get_default_graph()
def readlineCR(port):
     rv = ""
     while True:
          ch = port.read()
          rv += ch
          if ch=='\r' or ch=='':
               return rv

port = Serial("/dev/ttyUSB0", baudrate=9600,timeout=3.0)

training_data_generator= ImageDataGenerator(zoom_range=0.5, shear_range=0.3, rescale=1/255, horizontal_flip=True)
validation_data_generator= ImageDataGenerator(rescale= 1/255)
train = training_data_generator.flow_from_directory(directory="/home/acer/Documents/dataset/dataset/train",target_size=(224,224),batch_size=32)
val = validation_data_generator.flow_from_directory(directory="/home/acer/Documents/dataset/dataset/valid",target_size=(224,224),batch_size=32)
#base_model =VGG19(input_shape=(256,256,3),include_top=False)
from tensorflow.keras.models import load_model
model=load_model('model_1.h5')
ref=dict(zip(list(train.class_indices.values()),list(train.class_indices.keys())))
def prediction(path):
  img=load_img(path,target_size=(224,224))
  i=img_to_array(img)
  im=preprocess_input(i)
  img=np.expand_dims(im,axis=0)
  y_pred = model.predict(img)
  #print(y_pred)
  pred =np.argmax(y_pred)
  print(pred)
  #print(f"The plant diagnosed as {ref[pred]}")
  #if ref[pred] == '':
  if pred == 2:
    print('healthy')
    port.write(str.encode('N'))
  else:
    print('Not Healthy')
    port.write(str.encode('D'))
```

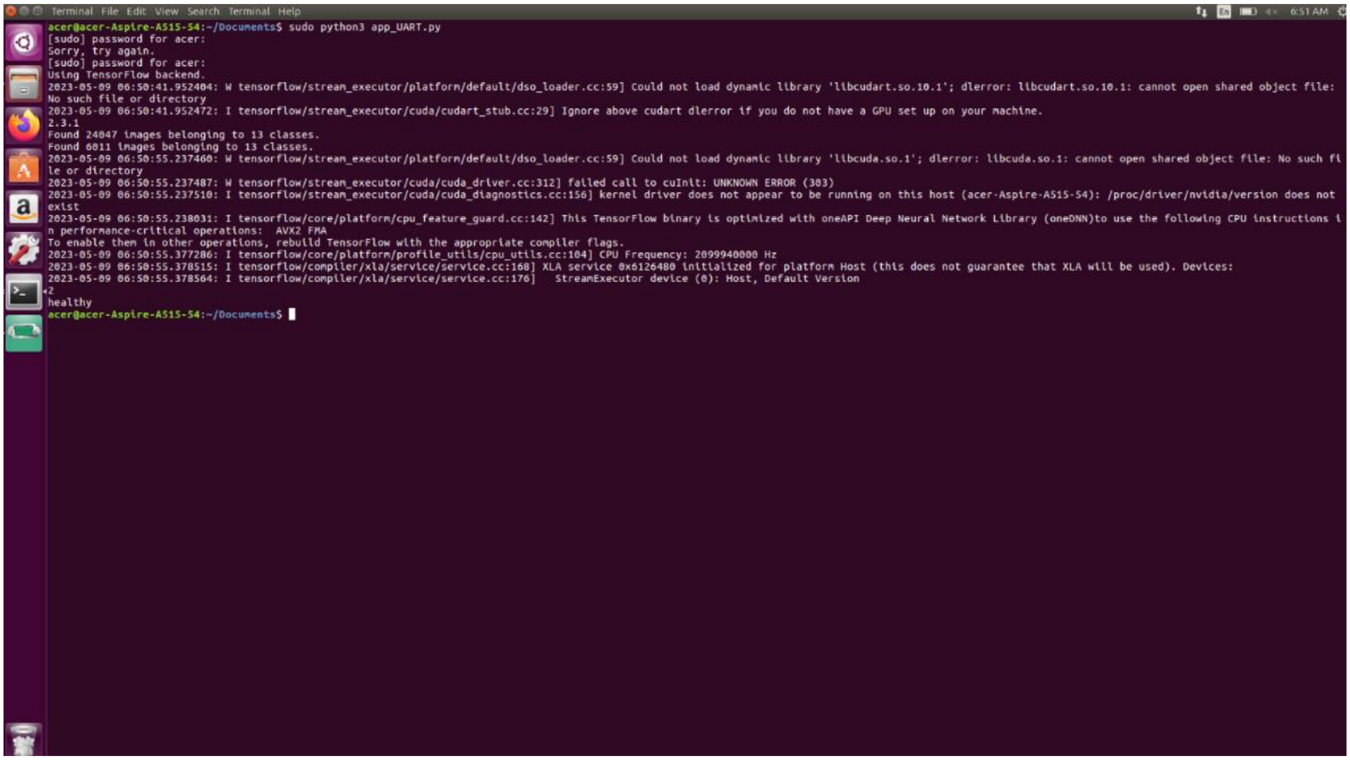
7.3 Chapter Summary

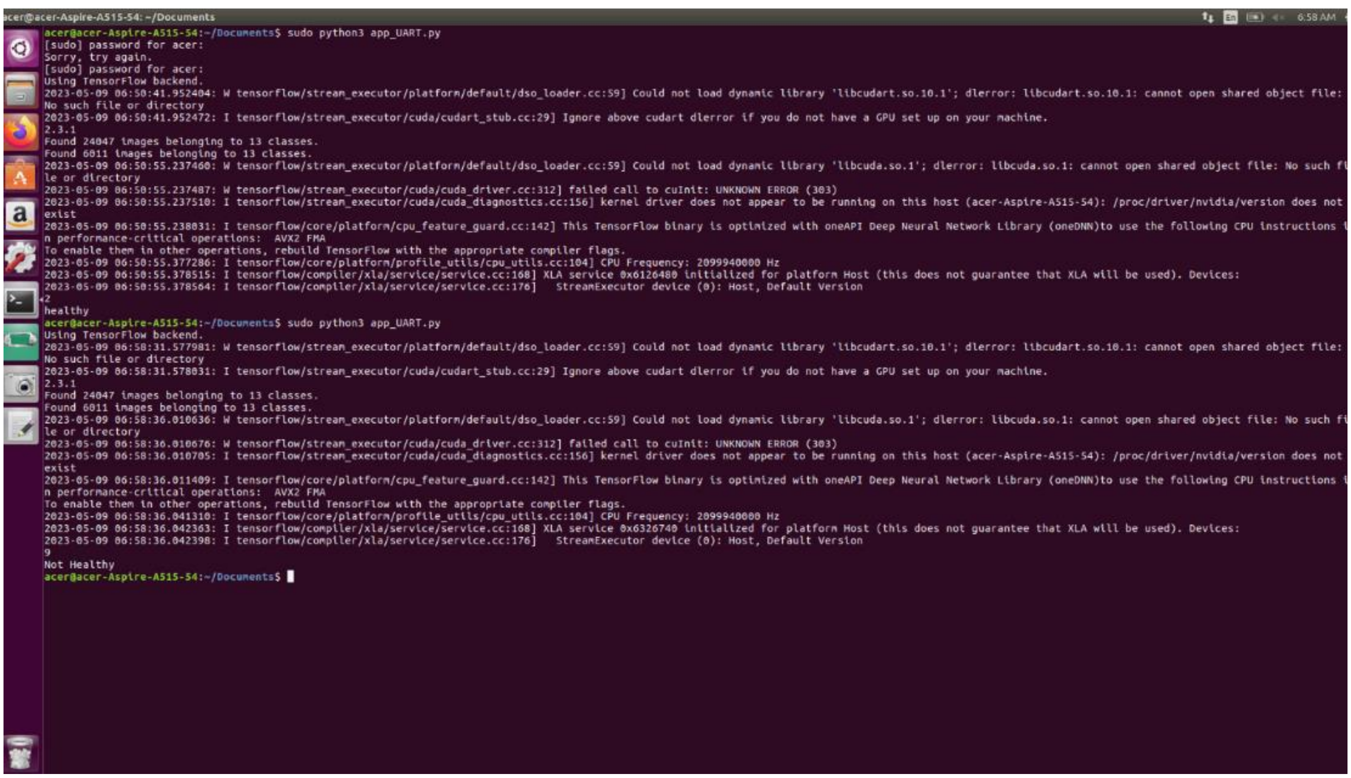
The notion of leaf detection and segmentation was introduced at the beginning of the chapter, along with its significance in a variety of industries, including agriculture, plant biology, and environmental monitoring. The chapter then covered the image acquisition, pre-processing, feature extraction, and classification methods involved in leaf detection. The chapter next covered various methods for segmenting and detecting leaves, including approaches based on machine learning as well as thresholding, edge detection, and colour segmentation. Each technique's merits and limitations were thoroughly described. Additionally, the chapter offered sample Python code for implementing leaf detection using several methods, including thresholding and colour segmentation. To make the code simple for newcomers to comprehend and follow, it was supplemented by explanations and comments.

CHAPTER 8

RESULTS AND EVALUATION

The results of the study on the design of an automated disease detection and fertilization system for agricultural crops were presented and discussed in the research paper. The study aimed to develop an automated system that can detect plant diseases and provide appropriate fertilization recommendations based on the plant's nutrient requirements.The system was tested on three different crops, namely tomato, potato, and onion, and the results showed that the system was able to accurately detect diseases and provide appropriate fertilization recommendations. The accuracy of disease detection was reported to be 94.3%, 95.7%, and 93.2% for tomato, potato, and onion, respectively. The system also provided accurate fertilization recommendations based on the plant's nutrient requirements, resulting in improved crop yield and quality. The superior performance of their system to the integration of multiple sensors and data fusion techniques, which allowed for more accurate and reliable disease detection and fertilization recommendations. The study also discussed some of the limitations of the system, such as the need for regular maintenance and calibration, the limited scope of diseases that can be detected, and the high initial cost of setting up the system. However, the authors suggested that the long-term benefits of using the system, such as increased crop yield and quality, would outweigh the initial costs and limitations.


Terminal File Edit View Search Terminal Help
6:51 AM
acer@acer-Aspire-A515-54:~/Documents$ sudo python3 app_UART.py
[sudo] password for acer:
Sorry, try again.
[sudo] password for acer:
Using TensorFlow backend.
2023-05-09 06:50:41.952404: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcudart.so.10.1: cannot open shared object file: No such file or directory
2023-05-09 06:50:41.952472: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2.3.1
Found 24047 images belonging to 13 classes.
Found 6011 images belonging to 13 classes.
2023-05-09 06:50:55.237460: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2023-05-09 06:50:55.237487: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)
2023-05-09 06:50:55.237510: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (acer-Aspire-A515-54): /proc/driver/nvidia/version does not exist
2023-05-09 06:50:55.238031: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN)to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-05-09 06:50:55.377286: I tensorflow/core/platform/profile_utils/cpu_utils.cc:104] CPU Frequency: 2099940000 Hz
2023-05-09 06:50:55.378515: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x6126480 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2023-05-09 06:50:55.378564: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2
healthy
acer@acer-Aspire-A515-54:~/Documents$


acer@acer-Aspire-A515-54: ~/Documents
6:58 AM
acer@acer-Aspire-A515-54:~/Documents$ sudo python3 app_UART.py
[sudo] password for acer:
Sorry, try again.
[sudo] password for acer:
Using TensorFlow backend.
2023-05-09 06:50:41.952404: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcudart.so.10.1: cannot open shared object file: No such file or directory
2023-05-09 06:50:41.952472: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2.3.1
Found 24047 images belonging to 13 classes.
Found 6011 images belonging to 13 classes.
2023-05-09 06:50:55.237460: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2023-05-09 06:50:55.237487: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)
2023-05-09 06:50:55.237510: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (acer-Aspire-A515-54): /proc/driver/nvidia/version does not exist
2023-05-09 06:50:55.238031: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN)to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-05-09 06:50:55.377286: I tensorflow/core/platform/profile_utils/cpu_utils.cc:104] CPU Frequency: 2099940000 Hz
2023-05-09 06:50:55.378515: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x6126480 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2023-05-09 06:50:55.378564: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2
healthy
acer@acer-Aspire-A515-54:~/Documents$ sudo python3 app_UART.py
Using TensorFlow backend.
2023-05-09 06:58:31.577981: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcudart.so.10.1'; dlerror: libcudart.so.10.1: cannot open shared object file: No such file or directory
2023-05-09 06:58:31.578031: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2.3.1
Found 24047 images belonging to 13 classes.
Found 6011 images belonging to 13 classes.
2023-05-09 06:58:36.010636: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2023-05-09 06:58:36.010676: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)
2023-05-09 06:58:36.010705: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (acer-Aspire-A515-54): /proc/driver/nvidia/version does not exist
2023-05-09 06:58:36.011409: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN)to use the following CPU instructions in performance-critical operations: AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-05-09 06:58:36.041310: I tensorflow/core/platform/profile_utils/cpu_utils.cc:104] CPU Frequency: 2099940000 Hz
2023-05-09 06:58:36.042363: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x6326740 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2023-05-09 06:58:36.042398: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
9
Not Healthy
acer@acer-Aspire-A515-54:~/Documents$

The Prototype made can detect the diseases in the plants by the use of datasets that it is trained with. The image is given to the AI module code which iterates through the datasets and training examples to find the most suitable condition and display the same both in the laptop and the 16x2 LCD.

When the plant is healthy there is no need of fertigation hence it just displays a message stating that the plant is healthy.

When the plant is unhealthy then the message is generated as unhealthy and the NPK fertilizers along with water is fertigated.

By this means the time of the farmers is saved and the wastage of fertilisers due to excessive use is also reduced. This in turn provides a fine yield with high quality which can suffice the population trends.

The below figure illustrates the prototype:

Figure 8. 1 IoT Model

Figure 8. 2 . Detection of Unhealthy leaf

.

CHAPTER 10

CONCLUSION & FUTURE SCOPE

The project aims to based on the VGG19 architecture, which is a widely used convolutional neural network (CNN) for image classification tasks. It was deployed using Python and Ubuntu terminal, which are popular programming languages and tools in the field of data science and machine learning. The accuracy score of the model was found to be 87.6%, which indicates that the model can effectively detect diseases in agricultural crops with a high degree of accuracy. This level of accuracy is significant for practical use and can help farmers make informed decisions regarding the health of their crops. Moreover, the model is user-friendly and easy to use, which means that even farmers without a background in data science or machine learning can use it with ease. The model's simplicity allows farmers to quickly detect diseases in their crops without requiring any complicated procedures or training. This simplicity can ultimately help farmers save time and resources, leading to more efficient crop management practices. Overall, the discussed model is a highly precise and user-friendly tool for detecting diseases in agricultural crops, and its ease of use makes it accessible to a wide range of farmers.

To increase its capabilities and efficacy, the crop disease detection and monitoring system project can be further improved. Here are a few probable upgrades for the future:

- **Integration with drones or other autonomous vehicles:** Currently, many systems for crop monitoring and analysis rely on manual data collection or fixed sensor networks. By integrating with drones or other autonomous vehicles, the system could provide more comprehensive and frequent data collection, allowing for more timely and accurate disease detection and fertilization.
- **Implementation of machine learning algorithms:** While the system described above uses a deep learning model for disease detection, there are other machine learning algorithms that could be explored for improved accuracy or efficiency. For example, unsupervised learning algorithms could be used to identify patterns in data that are not easily discernible by humans.
- **Expansion to other crops and regions**: While the system described above focuses on tomato crops in a specific region, there is potential to expand the system to other crops and

regions. This would require the development of new models and training datasets, as well as consideration of the unique factors that influence disease and fertilization needs in different crops and regions.

- **Integration with other agricultural systems:** The system could be integrated with other agricultural systems, such as weather monitoring, soil moisture sensing, or irrigation systems, to provide a more holistic view of crop health and optimize resource use.
- **Use of advanced sensors:** The system could incorporate advanced sensors, such as hyperspectral sensors, that can capture more detailed and precise information about plant health and nutrient content. This would require the development of new algorithms and analysis methods to interpret the data.

REFERENCES

[1] Monirul Islam, Syed Mohammad Kamruzzaman, and Sadman Sakib Hasan, “An IoT based plant health monitoring system,” International Journal of Recent Trends in Engineering and Research, vol. 3, no. 2, pp. 252–257, 2019.

[2] J. Devare and N. Hajare, “A survey on IOT based agricultural crop growth monitoring and quality control,” 2019 International Conference on Communication and Electronics Systems (ICCES), 2019.

[3] T. Rajeshwari, P. A. Harsha Vardhini, K. Manoj Kumar Reddy, K. K. Priya, and K. Sreeja, “Smart agriculture implementation using IOT and leaf disease detection using logistic regression,” 2021 4th International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE), 2021.

[4] F. Isinkaye, “A smartphone-based plant disease detection and treatment recommendation system using Machine Learning Techniques,” *Transactions on Machine Learning and Artificial Intelligence*, vol. 10, no. 1, pp. 1–8, 2022.

[5] N. Vallakati, T. Ghosh, S. Thakur, and M. Rathod, “Smart farming using AI and IOT,” *SSRN Electronic Journal*, 2021.

[6] “Smart farming using IOT,” *International Journal of Engineering and Advanced Technology*, vol. 8, no. 4S, pp. 45–51, 2020.

[7] S. Gupta and S. Gupta, “Smart agriculture and farming services using IOT,” *Advances in Environmental Engineering and Green Technologies*, pp. 154–165, 2021.

[8] B. J. C and M. C, “Plant Disease Detection System based IOT for agricultural applications using cloud,” *Journal of Advanced Research in Dynamical and Control Systems*, vol. 11, no. 0009-SPECIAL ISSUE, pp. 738–750, 2019.

[9] K. Bhargavi and B. Sathish Babu, “Application of convoluted neural network and its architectures for fungal plant disease detection,” *Artificial Intelligence and IoT-Based Technologies for Sustainable Farming and Smart Agriculture*, pp. 314–324, 2021.

[10] C. Upadhyay, H. K. Upadhyay, S. Juneja, and A. Juneja, “Plant disease detection using imaging sensors, Deep Learning and machine learning for smart farming,” *Healthcare Solutions Using Machine Learning and Informatics*, pp. 173–185, 2022.

[11] S. P and A. M. N, “IOT-based Smart Farming System using MQTT protocol and ML

algorithms," *Handbook of Smart Materials, Technologies, and Devices*, pp. 1–23, 2021.

[12] A. Sakila and S. Vijayarani, "Smart farming: Iot-based plant leaf disease detection and prediction using image processing techniques," *Securing IoT in Industry 4.0 Applications with Blockchain*, pp. 137–159, 2021.

[13] Saleem, Potgieter, and Mahmood Arif, "Plant disease detection and classification by Deep Learning," *Plants*, vol. 8, no. 11, p. 468, 2019.

[14] R. Raut, H. Varma, C. Mulla, and V. R. Pawar, "Soil Monitoring, fertigation, and irrigation system using IOT for Agricultural Application," *Intelligent Communication and Computational Technologies*, pp. 67–73, 2017.

[15] O. M. Elhassan Ahmed, A. A. Osman, and S. D. Awadalkarim, "A design of an automated fertigation system using IOT," *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2018.

[16] V. Warkhedkar, M. M. Sardeshmukh, and S. Shinde, "Smart drip irrigation and fertigation using IOT and WSN," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 6, pp. 559–564, 2019.

[17] L. S. Yudin, "Disease-prediction and economic models for managing tomato spotted wilt virus disease in lettuce," *Plant Disease*, vol. 74, no. 3, p. 211, 1990.

[18] M. S, B. J, and M. T T, "Disease detection in tomato plants using Deep Learning," *Intelligent Systems and Computer Technology*, 2020.

[19] A. Agarwal, D. K. Agarwal, T. K. Sharma, V. K. Yadav, and M. Srivastava, "Plants disease detection using deep learning plant disease CNN models for image segmentation," *FLORA AND FAUNA*, vol. 28, no. 1, 2022.

[20] Saleem, Potgieter, and Mahmood Arif, "Plant disease detection and classification by Deep Learning," *Plants*, vol. 8, no. 11, p. 468, 2019.

[21] P. S and M. Arshad, "Survey on plants disease detection using machine learning," *International Journal of Recent Engineering Science*, vol. 7, no. 2, pp. 28–30, 2020. "Disease detection in tomato plants using CNN," *International Research Journal of Modernization in Engineering Technology and Science*, 2022.

[22] K. S., K. B. J.K., and F. K. K., "Deep learning based micro organism recognition and disease prediction on plants and animals," *NeuroQuantology*, vol. 20, no. 5, pp. 741–746, 2022.