Artificial Intelligence based Cable Driven Robot for plant health assessment in Vertical Farming Environment

A BS Final Year Project by

Aqsa Qayyum
554-FET/BSEE/F19

Maryam Waqar
552-FET/BSEE/F19

Sidra Bibi
591-FET/BSEE/F19

Supervised by
Dr. Aleem Khaliq

Co-supervised by
Mr. Imran Qureshi

Department of Electrical and Computer Engineering
Faculty of Engineering and Technology
International Islamic University, Islamabad

June,2023

Certificate of Approval

It is certified that we have checked the project presented and demonstrated by **Aqsa Qayyum-554-FET/BSEE/F19, MaryamWaqar-552-FET/BSEE/F19, SidraBibi-591-FET/BSEE/F19** and approved it.

External Examiner	*Internal Examiner*
Name here…	**Name here…**
Designation	Designation

Supervisor	*Co-supervisor*
Dr. Aleem khaliq	**Mr. Imran Qureshi**
Lab Engineer	Hardware Manager

بسم الله الرحمن الرحيم

In the name of Allah (SWT), the most beneficent and the most merciful
A BS Final Year Project submitted to the

Department of Electrical and Computer Engineering

International Islamic University, Islamabad

In partial fulfillment of the requirements

For the award of the degree of

Bachelor of Science in Electrical Engineering

Declaration

We hereby declare that this work, neither as a whole nor as a part thereof has been copied out from any source. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning. We further declare that the referred text is properly cited in the references.

Aqsa Qayyum
554-FET/BSEE/F19

Maryam Waqar
552-FET/BSEE/F19

Sidra Bibi
591-FET/BSEE/F19

Acknowledgments

This BS thesis in Electrical Engineering has been conducted at Department of Electrical and Computer Engineering, Faculty of Engineering and Technology, International Islamic University, as part of the degree program. We would like to thank Engr. YYY for providing us an opportunity to work on this project, under his supervision and guidance throughout the project. We would also like to thank Engr. ZZZ for his help, efforts and dedicated support throughout the project. Further we are particularly thankful to Almighty Allah and grateful to our parents, brothers and sisters who always supported and encouraged us during our project and studies at IIUI.

Aqsa Qayyum

Maryam Waqar

Sidra Bibi

Project Title: **Artificial Intelligence based Cable Driven Robot for plant health assessment in Vertical Farming Environment**

Undertaken By:

Aqsa Qayyum	554-FET/BSEE/F19
Maryam Waqar	552-FET/BSEE/F19
Sidra Bibi	591-FET/BSEE/F19

Supervised By: **Dr Aleem Khaliq**
Lab Engineer

Co-Supervised By: **Mr. Imran Qureshi**
Hardware Manager

Date Started: September,2022

Date Completed: June,2023

Tools Used:

- Proteus 8 professional
- Autodesk Fusion 360
- Arduino IDE
- G-code sender
- Google collab
- Kaggle

Abstract

Cable-driven parallel robots (CDPR), a branch of parallel kinetic robots, are an emerging field of robotics, where the movement of robots is controlled through flexible cables and a motorized pulley system. Cables are much lighter than the rigid links of a traditional robot, and very long cables can be

used without creating a huge mechanism. The particular property of the cables gives these types of robots several advantages over traditional robots, such as adaptable environments, flexible working spaces and lower manufacturing costs. Vertical farming is the practice of growing crops in vertically stacked layers. It often incorporates controlled-environment agriculture, which aims to optimize plant growth, and soilless farming techniques such as Hydroponics, Aquaponics, and Aeroponics. Hydroponics is the cultivation of plants in nutrient-enriched water, skipping soil. Plants grown hydroponically are known to have higher nutrient content, require far less space, conserve water and allow for 30-50% faster all year-round growth compared to traditional farming. Root rot, mold growth, and plant leaf issues are the most common problems in hydroponics. Conventionally, methods like manual scouting, ladders, drones, and labor are hired for the health assessment of plants which remains expensive and time-consuming. In some cases, ground based robots and fixed cameras are installed, but these solutions provide limited and short-range vision. This project aims at contributing to the aforementioned cutting-edge technology by introducing 2D cable-driven robot for plant health surveillance. Precisely, cable coupled with motors and pulleys mounted on the vertical stand makes it a flexible space robot capable of maneuvering payloads. With the help of computer vision and machine learning techniques, the proposed system will detect non-healthy plants based on the asymptotic appearance of plants. This will serve as a cost-efficient, fully automated system with scheduled monitoring of plants. On detection of specific appearance-wise abnormalities, it shall generate an alert to the Agronomist on duty.

Table of Contents

List of Figures

List of Abbreviations

VFE	Vertical Farming Enviroment
CDPR	Cable-driven parallel Robot
OBE	Outcome Based Education

Chapter 1

1.1 Motivation

The gradual rise in climate change and growing population has led to a progressive growth in world hunger, water scarcity, extreme weather events, land degradation, desertification, and rising sea levels.

According to the UN's report "By 2050, the global population is expected to hit 10 billion people. This means that feeding everyone, will take 56% more food than is produced in the world today''

To undermine the growing world hunger and global climate change crisis, we need to adopt modern engineering solutions that promise climate-friendly sustainable food production systems.

There are many motivations for using AI models for plant health assessment in vertical farming. One of the main benefits is that AI models can provide more accurate and precise assessments of plant health, by analyzing large amounts of data and identifying patterns and trends that may be difficult for humans to detect. This can help farmers and growers to identify and prevent problems before they become serious, which can reduce crop losses and increase productivity. Additionally, AI models can help to reduce the need for harmful pesticides and herbicides, by providing more targeted and precise treatments for pests and diseases. This can help to reduce the environmental impact of agriculture and horticulture, and can help to promote sustainable farming practices. Finally, AI models can help to increase the efficiency and productivity of vertical farms, by automating many of the tasks involved in plant health assessment and management. This can reduce labor costs and increase yields, which can help to make vertical farming more economically viable. Overall, AI models for plant health assessment have the potential to revolutionize the way we grow and produce food, by making agriculture and horticulture more efficient, sustainable, and productive.

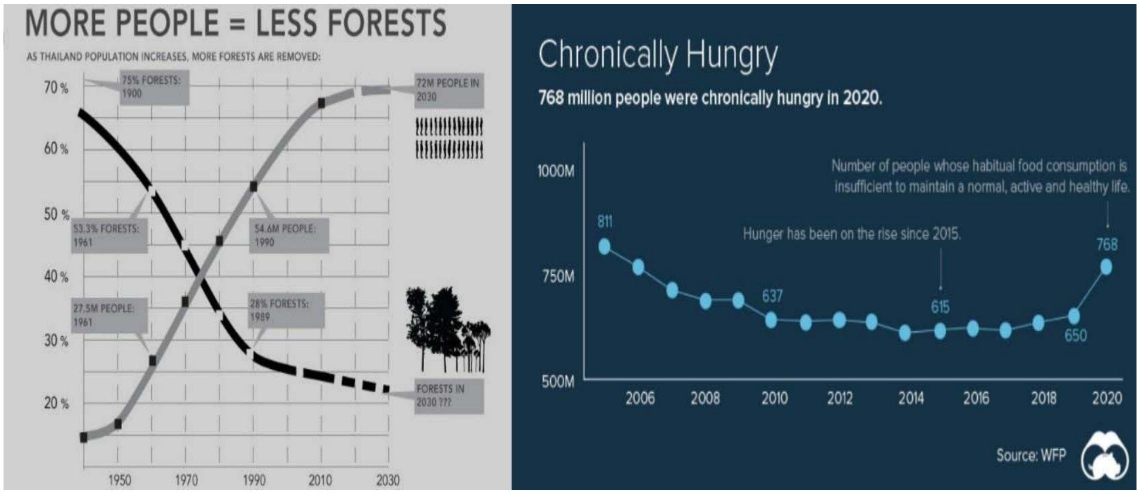


Fig1.1; Statistical data showing relation between population growth, deforestation and world hunger

1.1.1Future-proofing Agriculture

One such solution is vertical farming as shown in fig. which is a form of controlled environment agriculture where greens can grow hydroponically, aquaponically or even aeroponically. Plants have their roots suspended in nutrient-rich water. The filtered and purified water is packed with these nutrients, such as calcium, phosphorus and nitrogen, and is then given to the plants by the growing system. LED panels simulate sunlight, thus food is cultivated under ideal conditions 24 hours a day. The process is so efficient that it consumes 95 per cent less water and 99 per cent less land than usual agricultural practices. In addition, water is recycled and evaporated water is reclaimed, resulting in virtually no waste. The output of vertical farms can be significantly increased by using fertilizers specifically designed for certain plants and the appropriate lighting conditions in a strictly controlled indoor environment, or by adding value to the plants by influencing their nutritional content, flavor, and appearance. Because the cultivating environment is strictly controlled and not affected by climatic fluctuations or unfavorable weather circumstances, vertical farms offer yearround farming and several planting "seasons," considerably increasing a farm's overall annual yield and minimising crop failures.

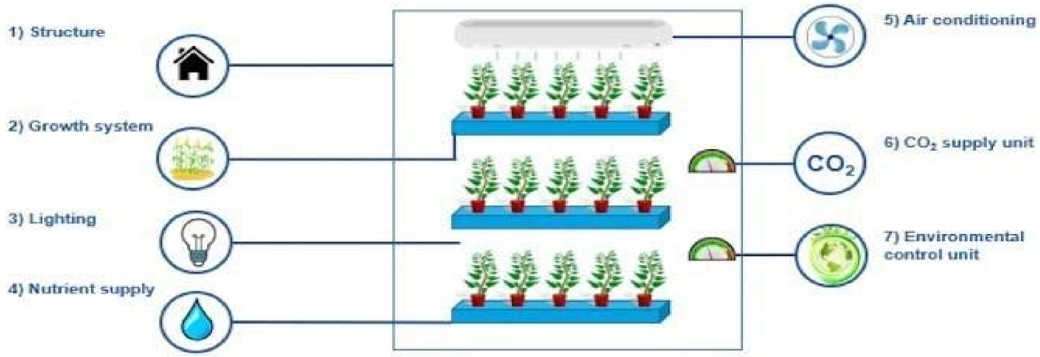


Fig1.2; Example of vertical farming in controlled environment

1.2 Project Overview

Considering the exigency for an effective indoor crop monitoring system, our work shall focus on designing and implementing of a fully autonomous intelligent system to monitor plants health in a vertical farming environment. The mechanical part of the proposed system employs flexible cables, a motorized pulley system, and a camera unit fitted at the end effector, having cable of movement in two dimensions, it will be programmed to move across each shelve one by one. An AI model will be trained for the health assessment of plants which will acquire live data from the camera unit, and certain noise reduction and removal filters will be applied to the incoming data. As the model will be trained and tested with extensive data set of healthy and non-healthy plant images, any appearance-wise abnormality will be detected by the model, as a result, an alert will be sent to the concerned body of the farm

An effective indoor crop monitoring system, an Machine Learning based plant health monitoring system will be developed comprising of a stepper motor-driven camera positioning system, an image acquisition system, and a host computer running the collection, processing, storage, and analysis functions. An alert system will be developed, on detection of unhealthy plants, an alert will be generated and sent to the concerned body of the farm.

1.2.1 Vertical Farming

The process of producing vegetables in layers that are piled vertically. The technique can make use of soil, hydroponic, or aeroponic growing techniques. In order to grow food in difficult conditions, vertical farms are used. The majority of the setup for the crops is done in a contained space with a specially designed environment for plant growth.

A cutting-edge technique for growing crops in controlled settings and vertically stacked layers is called vertical farming. It is a space-constrained metropolitan area's sustainable and effective method of food production. In comparison to conventional farming, vertical farming requires less water and fertiliser since it grows crops without soil using cutting-edge technologies like hydroponics, aeroponics, and aquaponics. It also requires less storage and transportation, making it a more environmentally responsible choice. Regardless of the weather, vertical farming can produce a variety of crops all year long and can be tailored to each crop's particular requirements. It is a potential answer to the problems of feeding a growing world population while minimising the negative environmental effects of agriculture.

Vertical farming is an innovative method of growing crops in vertically stacked layers and controlled environments. It is a sustainable and efficient way of producing food in urban areas, where space is limited. Vertical farming uses advanced technologies such as hydroponics, aeroponics, and aquaponics to grow crops without soil, using less water and fertilizers than traditional farming methods. It also reduces the need for transportation and storage, making it a more environmentally friendly option. Vertical farming can produce a variety of crops year-round, regardless of weather conditions, and can be customized to meet the specific needs of each crop. It is a promising solution to the challenges of feeding a growing global population while reducing the impact of agriculture on the environment

Certainly! Vertical farming methods can be used to raise a wide range of crops. Popular options include herbs like basil, mint, and parsley as well as leafy greens like lettuce, kale, and spinach. Strawberries, tomatoes, cucumbers, peppers, and even flowers can also be cultivated in vertical farms. The secret is to select crops that can flourish in a regulated environment with constrained light and space. Because temperature, humidity, and light can be precisely controlled in vertical farming, crops can be cultivated all year long, no matter the weather. This enables the production of locally grown, fresh produce even in severe climates or regions with little arable land.

1.2.2 Cable Driven Robot

Cable-driven robots, also known as cable robots or wire-driven robots, are a type of robot that uses cables or wires to move and manipulate objects. The cables are attached to motors or winches that control the robot's movements. Manufacturing, construction, and entertainment are just a few of the industries that use cabledriven robots. They are especially helpful when a robot needs to handle large or bulky goods across significant distances. Additionally, cable-driven robots have a reputation for being extremely accurate and precise, which makes them advantageous for jobs that call for a high level of control, like surgery or assembly work. In general, cable-driven robots are an adaptable and creative sort of robot that are revolutionising the way we think about automation and robotics.

Cable-driven robots can also be used to manage and monitor crops because they can control humidity, temperature, and light levels as well as detect and get rid of diseases and pests. This can reduce the need for harmful pesticides or herbicides and assist to ensure that crops continue to grow in a healthy and productive manner. Overall, by increasing the production, sustainability, and efficiency of vertical farming, cable-driven robots have the potential to change how we grow and produce food.

1.2.3 Plant Health Assessment

The robot monitors the field. The plant health is judged based on its leaf colour. Plant health assessment is the process of evaluating the health and condition of plants, in order to identify and prevent diseases, pests, or other problems that can affect their growth and productivity. Plant health assessment can involve a variety of techniques, including visual inspection, laboratory analysis, and remote sensing. Visual inspection involves examining plants for signs of damage, discoloration, or other symptoms that may indicate a problem. Laboratory analysis involves testing plant samples for pathogens or other contaminants that may be affecting their health. Remote sensing involves using satellites or other technologies to monitor plant health from a distance, by detecting changes in color, temperature, or other indicators. Plant health assessment is an important part of modern agriculture and horticulture, as it helps to ensure that crops are healthy and productive, and that farmers and growers can identify and prevent problems before they become serious.

1.3 Problem Statement

The ability to feed world population very much depends on three factors: availability of arable land, accessible water and population pressures. According to FAO’s prediction nearly 670 million people (8 percent of world population) will be facing hunger in 2030 – even if a global economic recovery is taken into consideration. We can say that as population grows, arable land shrinks and world hunger rises exceptionally. To promise a sustainable future, sustainable development strategies in agriculture should be undertaken. One of the advancement in agriculture is vertical farming where the crops are stacked in vertical layers minimizing extra use of resources such as land, water etc. and increases crop yield significantly. However, increasing crop yield takes into account various factors in crops life cycle that is preventing plant diseases, pruning and infections etc. Plant diseases can be detrimental if not properly attended to. Depending on the type of disease or virus your crops have, it is possible that the disease will destroy your entire crop as it moves from one crop to another. Root rot, mold growth, and plant leaf issues are the most common problems in vertical farming. Conventionally, methods like manual scouting, ladders, drones, and labour are hired for the health assessment of plants which remains expensive and time-consuming. In some cases, ground-based robots and fixed cameras are installed, but these solutions provide limited and short-range vision. However, the proposed system introduces a novel method that uses flexible cables and a motorized pulley system capable of maneuvering payloads for the monitoring of plants health in vertical farming. Cable driven robot is ought to be vertical farming adaptable, robust, fully autonomous and cost-efficient. By effectively monitoring the crops, crop yield can be increased significantly which increases food production overall.

- **Project Objectives**

- Development of an intelligent robotic machine-vision system for noninvasive plant/crop surveillance.
- To develop a robust and adaptable system cable of maneuvering payloads. Maneuvering of payload should be fast and stable.
- Integration of optimized hardware components and flexible mechanical design shall make a flexible working space robot.
- To achieve motion stability in the robotic camera positioning system.
- To test various machine learning algorithms considering the goals of the project and selecting the most optimized one.
- To achieve cost-effectiveness.
- To achieve more accuracy than the previous existing solutions.
-

1.4 Brief Project Methodology

We start with the model training. After the training of the model, this model is deployed in the real time working in a vertical farming environment.

- It starts with the stabilization of the camera, then this camera takes a certain position in front of a certain plant or crop.

- The camera then captures the image.
- Preprocess the image by removing the unnecessary background.
- Then the system predicts weather the plant image is healthy or unhealthy
- If the plant is healthy the camera will move to the next position
- If the plant is unhealthy, an alert is sent to the host commuter with the location and the image of the plant.

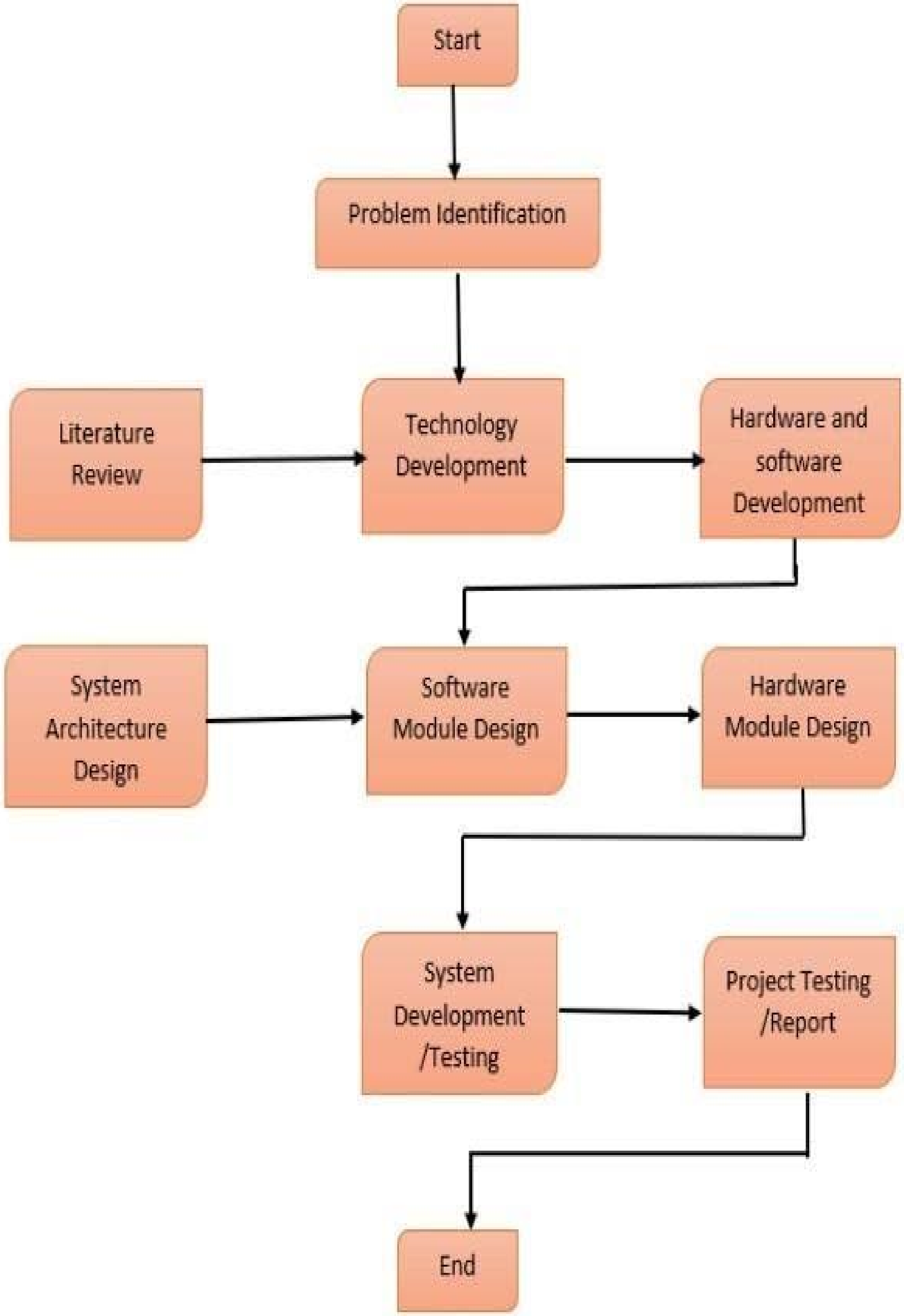


Fig1.3; model training and testing

Chapter 2

Literature Review

This section aims to provide the valuation of the literature available to classify and identify plant diseases and health. Research can be divided into two categories: ML-based approaches and DL-based approaches. The previous proposed solutions were not stable and could not be pragmatic in the real life magnificently. These solutions comprises of DCNN model, Crop segmentation, plant blob extraction and 3D reconstruction, leaf segmentation and geometric surface modeling.

2.1 Background of Project

A machine-learning tactic for identifying plant diseases must prioritize accuracy and speed. Developing techniques such as automated disease detection and categorization using image processing is necessary. Farmers will find that this is an appreciated strategy that will enable them to be informed in time to greatly reduce the risk of the disease spreading over a huge area and to prevent dying of plants due to insufficient nutrients. The land farming occupies a wide area of the agriculture land but still due to this over growing population that is not enough. A vertical farming comprises of a vertical farm in which plants are grown vertically staked over each other and it occupies 30% of the area as compare to land farming and gives more crop production.

It is not possible for a human ideally to look after the crops that are grown above hundreds of feet, so for the plants health assessment the most suitable solution is an AI robot that is placed vertically and it moves to each plant or crop for its health recognition.

There is a camera placed that is connected to pulleys, these pulleys are directly connects to the stepper motors. These motors moves the camera vertically by the Machine Learning techniques.

2.1.1 Role of AI and Robotics

With the integration of AI and robotics into the vertical farming industry, farms are becoming increasingly tech-powered, leading to increased precision, sustainability, and overall high-quality food production. With the aid of AIpowered robots, from monitoring and management to predictive analytics in indoor farming can be done very effectively, at highly reduced costs. Heretofore, a significant variation of intelligent bots and high-tech machinery is being deployed at vertical farms today that includes drones, autonomous sprayer systems, harvesting systems, and disease remediation robots. In order to move plants equally through their growth cycles, timed conveyor belts are also employed. Some vertical farming systems use static beds to adapt the environment to a plant's life cycle, while others are designed to transport plants between stations as they grow.

Fig2.1; ground based robot in VF

2.1.2 Challenges in vertical farming

The indoor farming industry undoubtedly has benefits, but it is not without its challenges such as high initial start-up cost, controlling environmental factors such as humidity, temperature, maintaining air circulation throughout and food safety etc. .Effective monitoring of crops during their life cycle counts for timely pest, mold, or disease detection and remediation when identified and treated at the right time leads to increased yield which in return boosts production and business profitability. For such a purpose, Drones serve as a great crop scouting tool and are widely used but since they are not environment adaptable, have limited flight time, and require licensing and expertise for flying, thus, remain unfavorable for indoor farming. Ground robots are also used at some farms but their coverage is limited and require additional electric ladders to reach the top shelves of the vertical stands, henceforth, do not serve as a very effective solution.

2.2 Related Work/Projects

Mellit et al. 2021 developed an Internet of Things (IoT)-based smart greenhouse in Algeria (a nation in North Africa) to detect infections in tomato plants [1]. By correctly classifying numerous tomato diseases, the experiment's results showed the usefulness of the proposed system. In order to remotely monitor tomato plants in real time, a prototype has been constructed. For diseases that would boost tomato yield, the proposed approach has an accuracy rate of 88%.

Franchetti et al. 2019 proposed an automatic method for extracting phenotype features of plants, based on CV, 3D modeling and deep learning from the extracted features, height, weight and leaf area were predicted and validated with ground truths obtained manually [2]. The results show that the plant height, leaf area and weight obtained using inexpensive RGBD cameras matched closely with the detailed measurements. The ability to obtain detailed information on the plant weight and therefore yield, without employing destructive techniques, facilitates the process of automation of the growth of vegetables in indoor VFAL conditions and consequently can substantially diminish the costs of production.

Story & Kacira, 2015 proposed a machine vision-guided plant sensing and monitoring system was constructed to continuously monitor color, morphological, textural, and spectral (crop indices and temperature) features from a crop canopy [3]. The machine vision system extracted these identified plant features which can be used to determine the overall plant growth and health status, but is capable of analyzing a much larger range of parameters for other plant phenotyping applications (i.e., perimeter, centroid, diameter etc.). Combining the systems capability with a decision support system can assist in dealing with identifying complexity of the crop stress symptoms and an increased control of the overall plant growth environment, which can potentially improve resource use efficiency in controlled environment crop production systems.

Hwang et al. 2022 developed an image-based crop growth monitoring system for vertical farming in this study [4]. As a result of the effective crop arrangement on vertical farms, the images obtained from the vertical farms contain too many regions of interest which attract the focus of the crop segmentation model. SCMix was suggested in order to guarantee that the crop segmentation model in our system performs well even when confronted with this confusing input. All results depict the performances of models trained with fixed data for cross comparison. However, with its real-world monitoring phase, our system will significantly benefit from extra unlabeled data that may be available for the retraining of the model.

Zane Zaik et al. 2020 proposed the use of trajectory planning and tracking with 2½D Visual Serving for the control of Cable-Driven Parallel Robots [5]. Perturbations and errors in the robot model. Furthermore, a Control Stability Workspace (CSW) was defined and computed for a CDPR prototype ACROBOT, based servoing control. The effect of perturbations on CSW size was the improvement of robustness due to the use of trajectory planning and tracking was clearly shown in experimental the trajectory produced by the former is clearly affected by. A further improvement would be developing a control law instead of increasing robustness to these errors.

Similarly, an autonomous assistant robot arms for monitoring temperature, humidity, pressure and light of soil in strawberry farm. This assistant robot uses sensor technology and the data collected from the sensors is gathered and updated on a LCD display.

Fig2.2; Autonomous assistant robot for monitoring temperature, humidity, pressure in strawberry farm

Although the mentioned robot is a good tool for controlling environmental factors in controlled environment agriculture, it has a few drawbacks as it necessitates extra human assistance and monitoring with it which demands more employee hiring and eventually higher resource usage. The proposed system takes into account the limitations of current crop surveillance practices. A fully autonomous system suitable for the vertical agricultural environment will be prepared for programmed crop monitoring in accordance with the requirements. It will be a low-cost, flexible working space intelligent robot that will analyze the overall health of the crops/plants, and detect healthy and unhealthy plants along with an alert system.

Fig2.3; on site crop monitoring

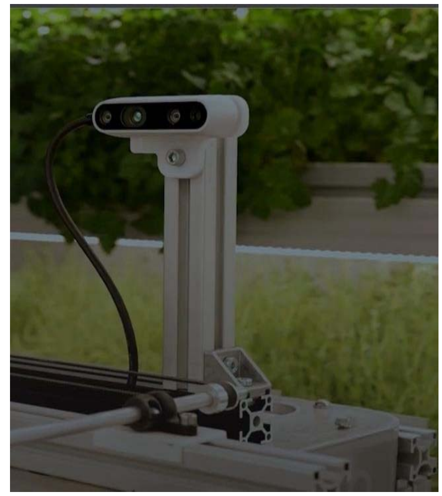


fig2.4; ground base robot

Qian, S., Bin, Z.2018 introduced the history of CDPR development and presents various examples of successful recent CDPR applications. In order to give readers a thorough and concise understanding of the design and analysis of CDPRs, the development of CDPRs is described with a focus on design, performance analysis, and control theory. The advantages of CDPRs over traditional rigid-link parallel robots are discussed in the paper, including a higher load-weight ratio, the potential for fast speed and acceleration, and a larger working area. The unilateral actuation nature of cables, however, presents difficulties in the design and implementation of CDPRs. The study areas for CDPRs, including as design, modelling, performance optimisation, control, and planning, are highlighted in the report.

Mattioni, E., & Mattioni, V. 2022 outline the most recent developments in servowinch design for cable-driven robots. From an application standpoint, they present a fresh design concept and critically evaluate it in comparison to current and suggested architectures. The study shows that while being historically the first developed, the rototranslating-drum concept does not have many advantages. The translating-motor concept is found to be the best option for applications that are low-cost, not very dynamical, and do not have stringent installation orientation requirements. On the other hand, the authors recommend the Spline Winch as the best choice for situations requiring high dynamics and vertical winch axis installations. A spooling helper solution is additionally introduced to optimise the amount of cable kept in relation to the winch footprint. However, caution is advised in highly dynamical operations when a load cell is embedded in the helper for measuring cable tension.

2.3 Project Contribution

Our AI model overcomes all the previous existing solutions as it is more stable in recognizing the plant health and diseases. This solution can be applied in real in a vertical farming environment with a great success. This solution includes the designing of an autonomous computer vision guided system comprising of a stepper motor-driven camera positioning system, an alert system and a host computer running the collections. This model has a programmable scheduled monitoring of plants.

The main task was the controlling of the motors and their movement and the stability of the camera positioning.

In the previous solutions like manual scouting the large number of labor was required to heir. They went up on the ladder for the crop assessment but it was not safe as there was always danger of falling from such a large height. Other than this was the fix cameras which was also not stable as it could only see the plants that comes to its vision. The other option proposed was ground based robots, but the issue was these robots could only attend the plants in the bottom and the plants in the upper shelves were left unattended.

Timely detection of plant diseases is a great challenge. If a disease is identified early it boosts the crop productivity. If a single plant gets a disease it spreads in the entire crop in no time and can destroy the entire production in no time. This problem has been solved with machine learning techniques using an automated method for detecting plant diseases in a vertical farming environment which is beneficial because it reduces monitoring time.

An artificial intelligence cable-driven robot for plant health assessment in a vertical farming environment is a robotic system designed to autonomously monitor and assess the health of plants in a vertical farming environment. The robot is equipped with a range of sensors, including cameras and environmental sensors that allow it to collect data on the plants' growth and health. The robot is also equipped with a cable-driven mechanism that allows it to move vertically through the vertical farming environment, providing it with access to all areas of the plants.

Chapter 3

System Design and Implementation Details/Design Procedures

To set up an AI model the machine learning techniques, hardware modules, software setup and a 3D prototype is required. When the model is all set up it is then trained until it meets the required accuracy. Once the model is properly stable it is deployed in to a vertical farming environment.

3.1 System Design

The system is designed integerating cable driven robot technology with computer vision and machine learning techniques. Plant diseases can be categorized using machine learning based on a range of factors. Before effectively extracting features, preprocessing is necessary, such as image improvement, colour alteration, and segmentation. Deep learning is required for smart farming, which makes use of modern agricultural technology, technology, and algorithms. Deep learning is frequently used to find solutions to issues with picture categorization, feature extraction, transformation, and pattern analysis. System is planned in such a way that it must be cost effectiveness so that it can be casted-off easily in the vertical farming environment. The designing of the model is prepared by making sure that it helps in the recognition of disease timely and alert about the health of the plant so that urgent action can be taken for the health of the plants. The design is completely autonomous there is no indulge of a human except the check on the computer to receive the alert. The camera is connected to the accessible computer through the camera and it send the alert as it detects any health problem in the plant. The monitoring of the crops/plants is programmable.

3.1.1 System Architecture/Flow Diagram

A cable driven robot is designed. The stepper motors are designed to move the pulleys so these pulleys can move the camera that is attached to the end of these pulleys. A power supply of 12V is connected to the structure. Arduino is connected to the CNC shield that run the motor drivers so that the drivers can move the motors.

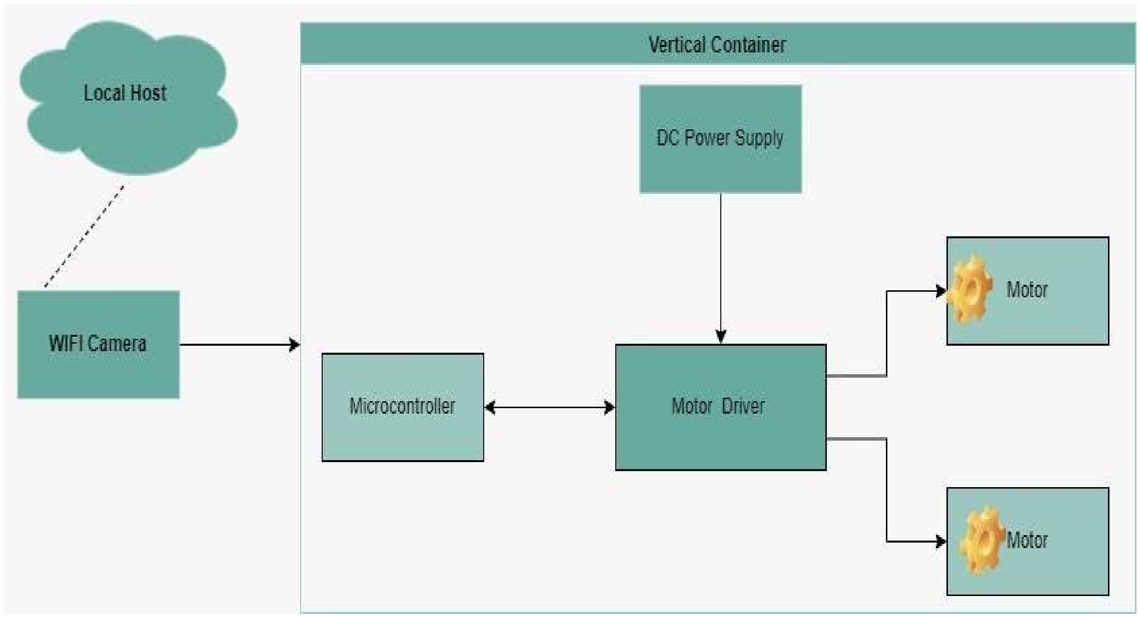


Fig3.1; block diagram

3.1.2 Requirements/Requirements Analysis

One of the most important thing in a model is to accomplish the purpose for which it is prepared, the purpose of machine learning based plant health assessment robot is to monitor each plant and its health and on uncovering of any type of health issue it should send an attentive to the available host.

Hence in this AI model the most important thing is the stable movement of the motors that are controlled by the machine learning algorithms. Other one is the existing of the proper data set. The timely detection of ant fault in plants health. Timely alert to the computer system for quick analysis. Excellence in these areas make this model diverse and efficacious that the prior existing models.

3.2 Methodological/Implementation/Experimental Details

The microcontroller is programmed for the motors movement, as the entire assessment depends on the moving of motors that moves the pulleys and the camera moves with the pulleys as it is connected to the end of the pulleys. The camera moves in the vertical 2d direction from one plant to the other.

It is trained for the detection of health. The model is trained by providing the data set. This data set includes the healthy and the non-healthy plants images. These images are preprocessed by the system. After several training when the model reach the required accuracy then this model is implemented to the vertical farm.

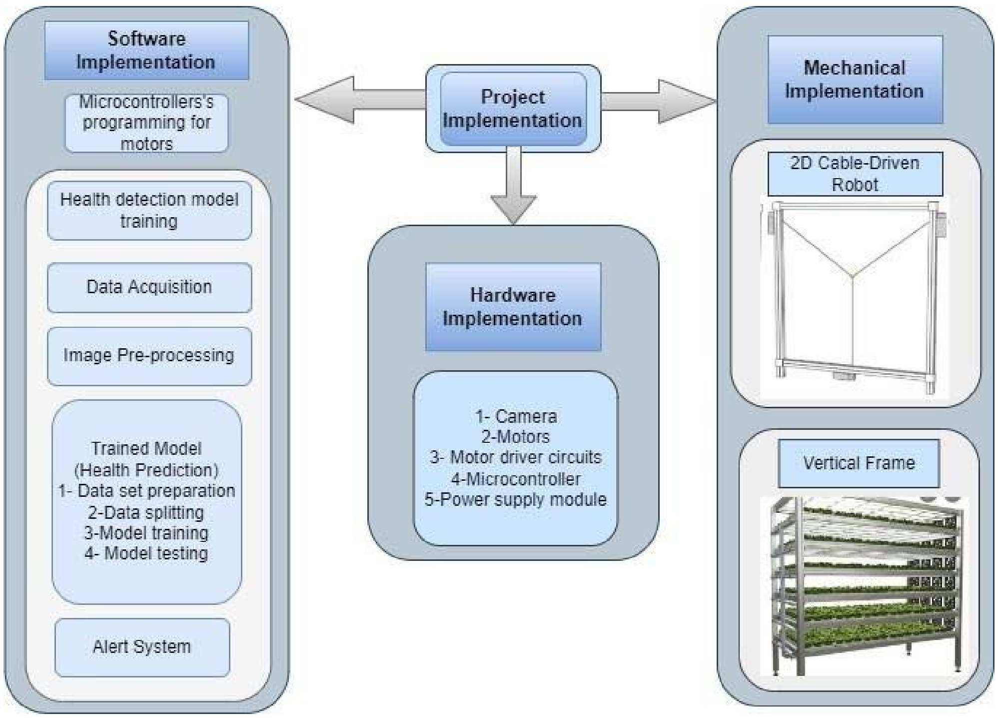
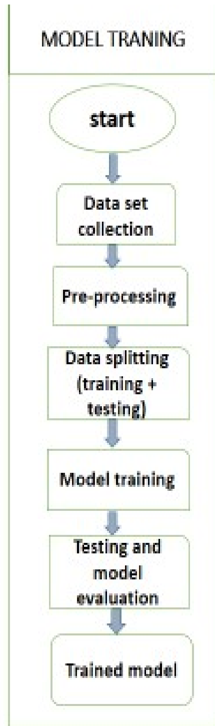
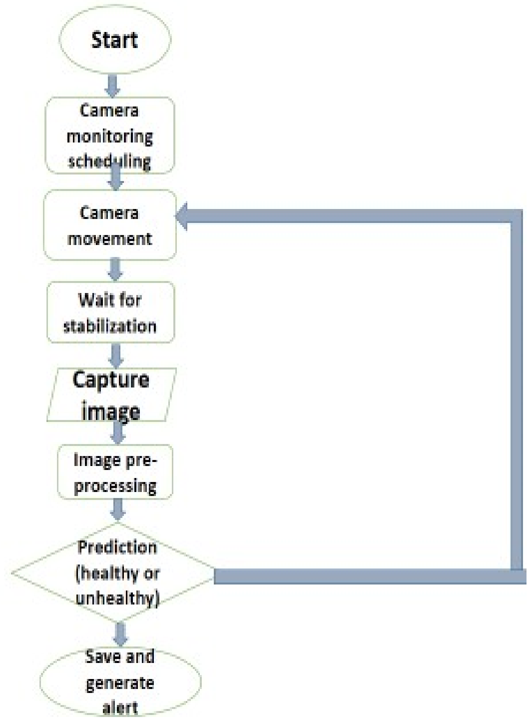


Fig3.2

3.2.1 Software/Development Setup





3.2.2 Hardware Details

3.2.3 Bipolar Stepper motors; Nema 17

NEMA 17 stepper motors are known for their exact control, which is significant in applications, for example, advanced mechanics where precision is vital. These engines have a stage point of 1.8 degrees, which takes into consideration extremely fine developments and control. This additionally implies that they can move in tiny augmentations, making them ideal for situating undertakings.

Nema 17 stepper motors require less power contrasted with DC engines, and that implies the robot can work on low power, making it energy productive.

One more benefit of NEMA 17 stepper motors is that they can give high force, which is significant for applications that require a great deal of force and strength. This is especially significant in vertical cultivating conditions where the robot needs to go all over and convey gear to evaluate the soundness of plants.

Moreover, NEMA 17 stepper motors are broadly utilized in the business, and that implies that they are promptly accessible and savvy contrasted with different kinds of engines.

By and large, utilizing NEMA 17 stepper motors in a link driven robot for plant wellbeing evaluation in vertical cultivating conditions gives exact control, high force, and cost-viability, settling on them a famous decision for the overwhelming majority mechanical technology applications.

Fig3.4; stepper motor

Motor Driver; A4988 Drivers

The A4988 is a complete micro stepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ±2 A. The A4988 is not intended for controlling DC motors. A stepper motor is driven by a DC voltage applied through a driver.

A4988 motor drivers give high-goal motor control, which prompts smooth and exact development of the robot. This is great for plant wellbeing evaluation where exactness is central. Utilizing A4988 motor drivers with Nema 17 stepper engines offer high accuracy, low power utilization, improved on wiring, bipolar stepper motors, and cost-viability, making them a reasonable choice for link driven robots for plant wellbeing evaluation in the upward cultivating climate.

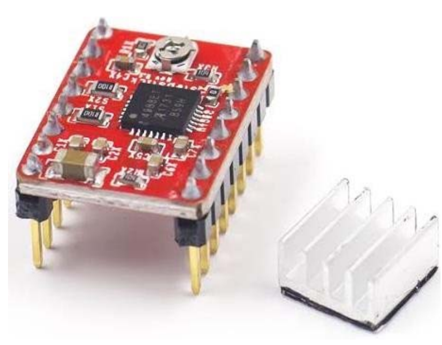


Fig3.5; motor driver

Microcontroller; Arduino UNO

Arduino is a famous decision for mechanical technology and robotization projects in light of its usability, flexibility, and moderateness. An open-source stage gives an extensive variety of microcontrollers and improvement sheets that are not difficult to program utilizing the Arduino programming language.

This stage is exceptionally viable with an immense range of sensors and electronic parts which settles on it an optimal decision for planning a link driven robot for plant wellbeing evaluation. With the various libraries and code scraps accessible, it is not difficult to carry out complex calculations, for example, PC vision or AI, on an Arduino-based microcontroller.

Besides, Arduino is very much upheld by the local area, with an abundance of instructional exercises, gatherings, and models accessible on the web. This makes it simple for engineers to investigate issues and speed up their improvement cycle.

All in all, Arduino is a flexible and savvy choice for mechanical applications. It offers a large number of highlights, similarity, and local area support, making it ideal for planning a link driven robot for plant wellbeing evaluation in vertical cultivating.

Fig3.6; microcontroller

CNC shield V3

CNC safeguard is a typical extra board utilized with Arduino that empowers the simple control of stepper motors. It permits the client to associate various stepper motors, end stops, and different peripherals to a similar control board, working on the wiring system. The CNC safeguard likewise considers more exact movement control which is significant in the improvement of a link driven robot for plant wellbeing evaluation in vertical cultivating.

Here are a few benefits of utilizing a CNC considering our project:

1. Different stepper engine control: The CNC safeguard have some control over up to four stepper engines at the same time, which is significant while working a link driven robot. This empowers various engines to work in synchronization and guarantees framework soundness.

2. Simple wiring: With the utilization of CNC safeguard, different engines and sensors can be effectively associated with a solitary board, working on the wiring system, and decreasing the quantity of links utilized in the framework.

3. Exact movement control: CNC safeguard gives a more exact and exact movement control that is fundamental in the improvement of a link driven robot for plant wellbeing evaluation in vertical cultivating. It gives smoother and more reliable motor operation, which diminishes movement mistakes and guarantees the exact development of the robot.

4. Intensified usefulness: CNC safeguard has various extra elements that empower more intricate movement control that can be utilized to further develop the plant wellbeing appraisal precision. It can peruse limit switches, control shaft speed, and even produce signs of high recurrence, that can work with the improvement of the robot's functionalities.

Generally, the CNC safeguard gives a reasonable stage to controlling and coordinating the developments of a link driven robot, which is fundamental for fostering a dependable and exact plant wellbeing evaluation framework for vertical cultivating.

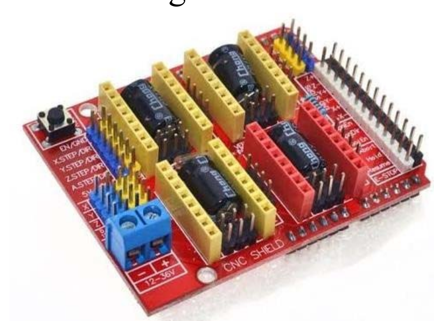


Fig3.7; cnc board

3.2.4 Software/Tools Google colab

A cloud-based development environment that allows users to write and run Python code in a web browser, using Google's powerful hardware and software infrastructure. Google Colab provides many features that are useful for data scientists and machine learning practitioners, including access to powerful GPUs, pre-installed libraries for machine learning and data analysis, and the ability to share and collaborate on code with others. Google Colab is free to use, and requires only a Google account to get started. Overall, Google Colab is a powerful and convenient tool for anyone who wants to develop and run Python code in the cloud.

Proteous

Proteus is a software tool used for simulating, designing, and testing electronic circuits. It is widely used by engineers, students, and hobbyists for designing and testing electronic circuits before building them in real life. Proteus provides a virtual environment where users can design and test circuits using a wide range of electronic components, including microcontrollers, sensors, motors, and displays. Proteus also includes a powerful simulation engine that allows users to test their designs under various conditions, such as different voltages, temperatures, and loads. Proteus is easy to use, and provides a wide range of features for designing and testing electronic circuits, including schematic capture, PCB layout, and 3D visualization. Overall, Proteus is a powerful tool for anyone who wants to design and test electronic circuits, whether for professional or personal use.

AutoCAD

AutoCAD is a computer-aided design (CAD) software tool used for creating 2D and 3D designs, models, and drawings. It is widely used by architects, engineers, designers, and other professionals for designing buildings, products, and other objects. AutoCAD provides a wide range of features and tools for creating and editing designs, including drawing tools, editing tools, and dimensioning tools. AutoCAD also supports a wide range of file formats, making it easy to share designs with others. Additionally, AutoCAD can be customized using programming languages such as Auto LISP and Visual Basic for Applications (VBA), allowing users to create their own tools and automate repetitive tasks. Overall, AutoCAD is a powerful and versatile tool for anyone who needs to create and edit 2D and 3D designs.

3.3 Algorithms/Simulation Details/Codes

3.3.1 Simulation on proteus

Circuit simulation is crucial for verifying and improving circuit designs. we used proteous 8 professional, a proprietary software tool suite used primarily for electronic design automation. In our project, we used an Arduino Uno, stepper motors, a CNC shield, and A4988 drivers. However, since the CNC shield module wasn't included in the Proteus library, we decided to design it ourselves.

By designing the CNC shield and clearly depicting its linkages and interactions, we were able to ensure optimal component integration. We also designed the A4988 drivers and their internal circuitry in Proteus to completely evaluate the functionality of our device. We were able to detect and address any potential issues using this extensive simulation technique before moving on with the actual implementation, which saved time and resources while ensuring a good result. Before final assembly, all the components and the Arduino code are tested by generating the .hex file of Arduino sketch program. The simulations are as followed:

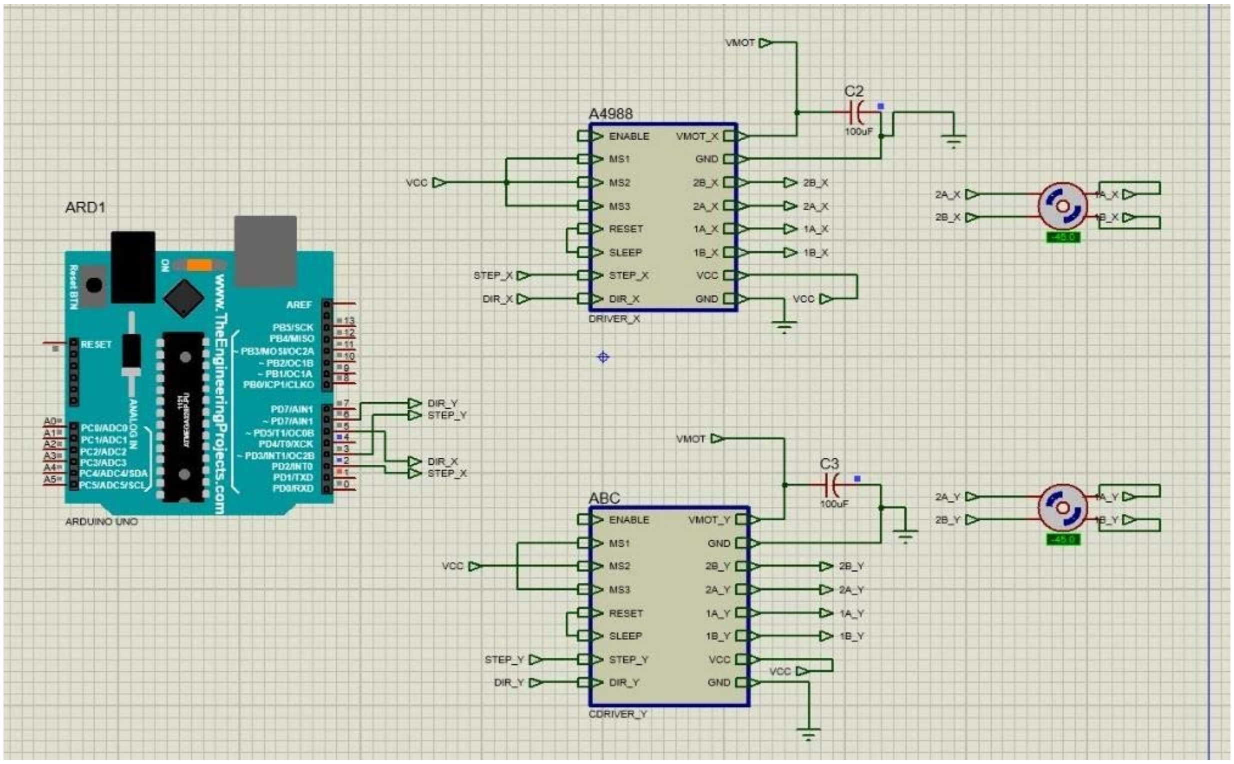


Fig3.8; Circuit simulation

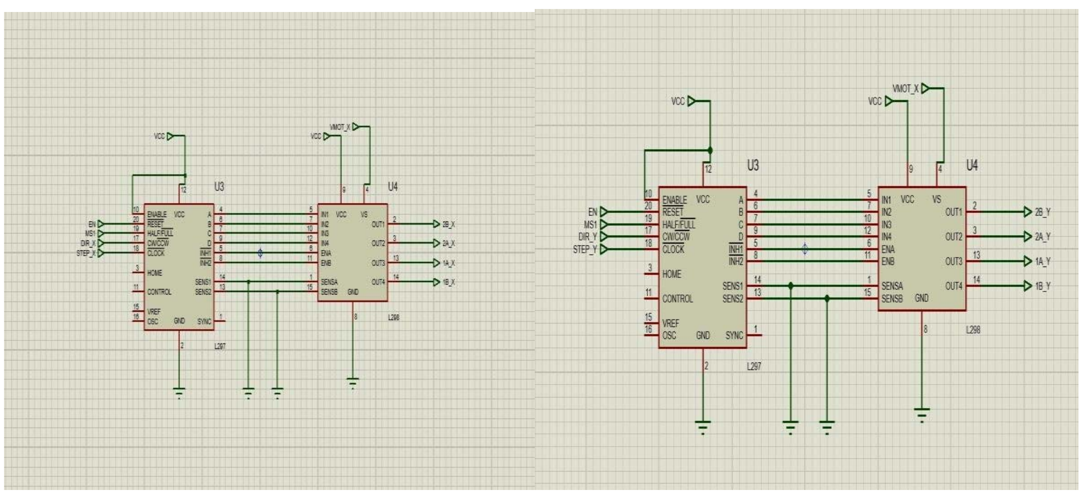


Fig.3.9. A4988 drivers simulation

3.3.2 Prototype Simulation

A prototype of the envisioned structure was designed on a mechanical simulation software for visualization and to check its feasibility considering its required dimensions before practical implementation.

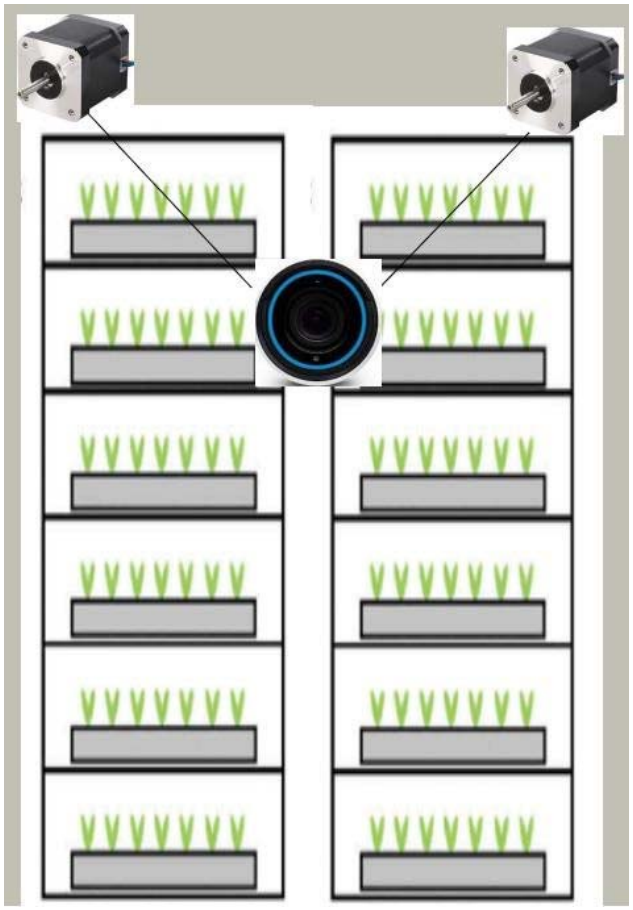


Fig3.9; Prototype designing

3.3.3 Proposed machine learning model

In our project, we deployed a Convolutional Neural Network (CNN) model to identify healthy lettuce leaves in a vertical farming environment. The flatten layer, dense layer, dropout layer, and output layer with softmax activation are added to the CNN design after a number of convolutional and pooling layers. This model can successfully extract features and patterns from images, making it useful for health diagnostics in vertical farming. It can accurately discern between healthy and harmful conditions by observing the minute variations and traits of lettuce leaves. After being trained on a dataset specifically selected for lettuce health evaluation, the programme performs significantly better. We can precisely track the health state of lettuce leaves using our CNN model in real-time, enabling quick process optimisation and intervention in vertical farming.

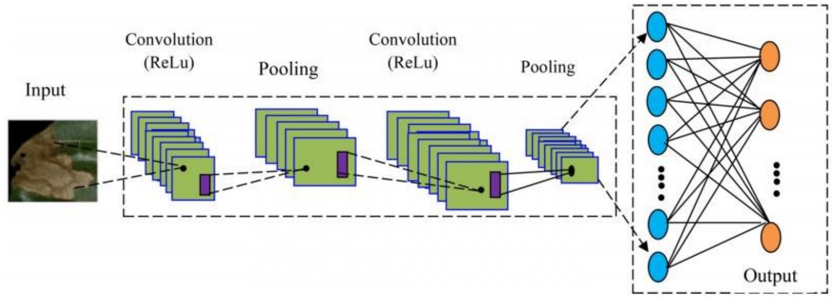


Fig.3.10. Classification of plant disease by CNN

Dataset Preparation:

In the dataset collection phase, We gathered a variety of images showing both healthy and unhealthy lettuce plants. The dataset contained samples of red and green lettuce, representing variations in leaf colour and texture. We photographed the plants in various stages of development, from tiny seedlings to fully grown plants. The dataset was split into two directories, one containing pictures of healthy lettuce and the other including pictures of unhealthy lettuce.

Algorithem

import tensorflow as tf import

matplotlib.pyplot as plt from

tensorflow.keras import layers

_VERSION = '2.6.1'

import numpy as np

import cv2 import

os

from google.colab import drive drive.mount('/content/drive')

Define the path to your dataset

DATASET_PATH = '/content/drive/MyDrive/lettuce datset'

Define the image size for the input

IMG_SIZE = (224, 224)

Define the number of classes

NUM_CLASSES = 2

Define the batch size for training

BATCH_SIZE = 32

```
# Define the number of epochs for training

EPOCHS = 20


# Define the learning rate for training

LEARNING_RATE = 0.0001


# Define the list of classes

CLASSES = ['healthy', 'unhealthy']


# Define the function to load the dataset def

load_dataset():

   data = []     labels =

[]     for cls in

CLASSES:

      path = os.path.join(DATASET_PATH, cls)

      for img in os.listdir(path):

         img_path        =        os.path.join(path,        img)

img_array = cv2.imread(img_path)          img_resized =

cv2.resize(img_array,                              IMG_SIZE)

data.append(img_resized)
```

```
labels.append(CLASSES.index(cls))                    return
np.array(data), np.array(labels)


# Load the dataset data,
labels = load_dataset()


print(data.shape)


print(labels)


# Define the model architecture model
= tf.keras.models.Sequential([
   tf.keras.layers.Conv2D(16, (3,3), activation='relu',
input_shape=(IMG_SIZE[0], IMG_SIZE[1], 3)),
tf.keras.layers.MaxPooling2D((2,2)),
tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D((2,2)),
tf.keras.layers.Conv2D(16, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D((2,2)),
tf.keras.layers.Flatten(),     tf.keras.layers.Dense(128,
activation='relu'),     tf.keras.layers.Dropout(0.5),
```

```
tf.keras.layers.Dense(NUM_CLASSES,
activation='softmax')
])


# Compile the model
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=LEARNING
_RATE),
          loss='sparse_categorical_crossentropy',
metrics=['accuracy'])


# Train the model
history=model.fit(data, labels, batch_size=BATCH_SIZE, epochs=EPOCHS,
validation_split=0.2)


acc = history.history['accuracy'] val_acc
= history.history['val_accuracy']


loss = history.history['loss'] val_loss
= history.history['val_loss']
epochs_range = range(EPOCHS)


plt.figure(figsize=(8, 8))
```

```
plt.subplot(1, 2, 1) plt.plot(epochs_range, acc,

label='Training Accuracy') plt.plot(epochs_range, val_acc,

label='Validation Accuracy') plt.legend(loc='lower right')

plt.title('Training and Validation Accuracy')


plt.subplot(1, 2, 2) plt.plot(epochs_range, loss,

label='Training Loss') plt.plot(epochs_range, val_loss,

label='Validation Loss') plt.legend(loc='upper right')

plt.title('Training and Validation Loss') plt.show()


from keras.utils.vis_utils import plot_model

plot_model(model, to_file='model_plot.png', show_shapes=True,
show_layer_names=True)


model.save('/content/drive/MyDrive/lettuce datset/my_model.h5')


categories = ["healthy", "unhealthy"]


# Set the input size of the images

input_size = 224


# Load the model
```

```
model = tf.keras.models.load_model("/content/drive/MyDrive/lettuce
datset/my_model.h5")


predicted_labels = [] confidences
= []


# Loop through each image in the test dataset for
category in categories:
   folder_path = os.path.join(DATASET_PATH, category)
for img_path in os.listdir(folder_path):
      img = cv2.imread(os.path.join(folder_path, img_path))
img = cv2.resize(img, (input_size, input_size))      img_orig =
img.copy()  # Make a copy of the original image      img = img /
255.0  # Normalize the image      img = np.expand_dims(img,
axis=0)  # Add a batch dimension


      # Make the prediction      prediction
= model.predict(img)[0]      category_idx
= np.argmax(prediction)
category_name = categories[category_idx]
confidence = prediction[category_idx]
```

```
# Add the predicted label and confidence to the lists
predicted_labels.append(category_name)        confidences.append(confidence)


# Append the predicted label and confidence onto the image
text = f"{category_name} ({confidence:.2f})"
cv2.putText(img_orig, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX,
1.0, (0, 0, 255), 2)


# Save the image with the predicted label and confidence
output_path = os.path.join(DATASET_PATH, category_name +
"_predicted")                        os.makedirs(output_path,
exist_ok=True)                              output_path    =
os.path.join(output_path,                          img_path)
cv2.imwrite(output_path, img_orig)


#        Print        the        results
print("Image:",                  img_path)
print("Category:",        category_name)
print("Confidence:", confidence)
```

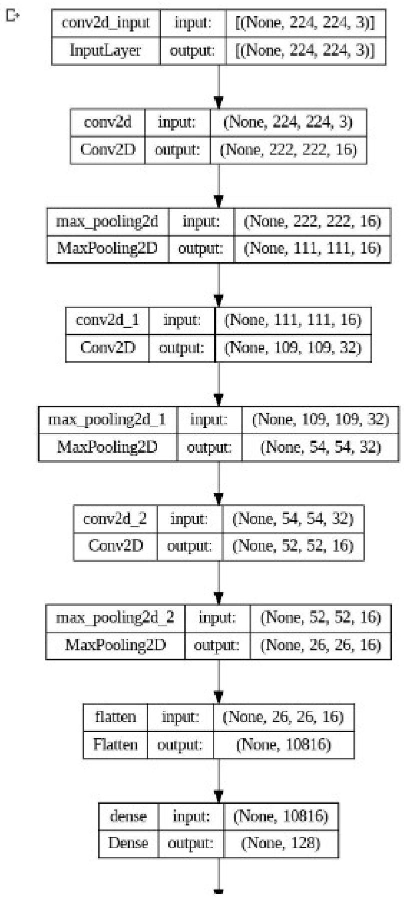
```
# Convert the predicted labels and confidences to numpy arrays

predicted_labels = np.array(predicted_labels) confidences =

np.array(confidences)


# Create a dictionary with the predicted labels and confidences

predictions = {"predicted_labels": predicted_labels, "confidences": confidences}


# Save the dictionary to a file using numpy's savez function

np.savez("predictions.npz", **predictions)


model.summary()


from tensorflow.keras.models import Sequential, save_model, load_model



from IPython.display import HTML, Audio, display

from google.colab.output import eval_js  from

base64 import b64decode  import numpy as np

import io  from PIL import

Image  import tensorflow as tf

import cv2 import

matplotlib.pyplot as plt
```

```
model = tf.keras.models.load_model('/content/drive/MyDrive/lettuce datset/my_model.h5')

VIDEO_HTML = """
<video autoplay  width=%d height=%d style='cursor:
pointer;'></video>
<script>

var video = document.querySelector('video')

navigator.mediaDevices.getUserMedia({ video: true })
  .then(stream=> video.srcObject = stream)

var data = new Promise(resolve=>{  video.onclick =
()=>{   var canvas =
document.createElement('canvas')   var [w,h] =
[video.offsetWidth, video.offsetHeight]
canvas.width = w   canvas.height = h
canvas.getContext('2d')      .drawImage(video, 0,
0, w, h)
video.srcObject.getVideoTracks()[0].stop()
```

```
video.replaceWith(canvas)
resolve(canvas.toDataURL('image/jpeg', %f))
  }
})
</script>
"""


def preprocess_image(image):
   image = Image.fromarray(image).convert('L').resize((224, 224))


   rgb_image = Image.new("RGB", image.size)
rgb_image.paste(image)                    image    =
np.array(rgb_image)  /  255.0            image   =
np.reshape(image,  (1,  224,  224,  3))        return
image


def predict_image(image):
   img = preprocess_image(image)


   pred         =         model.predict(img)
class_idx = np.argmax(pred)
```

```
    if class_idx == 0:
        class_label    =    "healthy"
else:
        class_label = "unhealthy"

    # Appending the class label
    image    =    cv2.putText(image,    class_label,    (10,    30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    return image

def take_photo(filename='photo.jpg', quality=0.8, size=(800,600)):
display(HTML(VIDEO_HTML % (size[0],size[1],quality)))
data = eval_js("data")    binary = b64decode(data.split(',')[1])    f
= io.BytesIO(binary)    return np.asarray(Image.open(f))

img = take_photo() processed_img =
preprocess_image(img) result_img =
predict_image(np.copy(img))

# Predict the class using the model pred
= model.predict(processed_img)
```

```
class_idx = np.argmax(pred)

class_label = chr(class_idx + 65)

plt.figure(figsize=(10,10))

plt.imshow(result_img)

plt.title(f'Predicted class: {class_label}')

plt.axis('off')
```



Explanation:

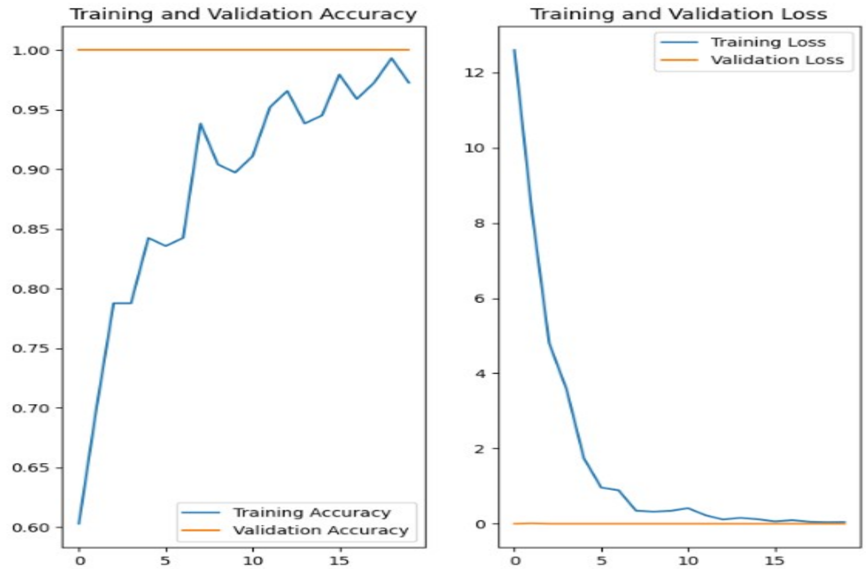
- We begin by importing the necessary libraries, including OpenCV, TensorFlow, Keras, numpy, and Keras.
- The parameters we set next include the dataset directory, image size, batch size, number of classes, number of epochs, learning rate, and class names.

- We created a function called "load_dataset()" with the intention of loading and preparing the photographs from the dataset. The images are resized, converted to numpy arrays, and given the proper names.
- The dataset is loaded with the "load_dataset()" method.
- The model architecture is established using Keras' Sequential API. The first layers in the structure are the convolutional and pooling layers, which are followed by the flatten, dense, dropout, and output layers with softmax activation.

- The sparse categorical cross-entropy loss, and the Adam optimizer accuracy metric are used to build the model.
- Using the fit() method, the model is trained using the input data, labels, batch size, number of epochs, and validation split.
- As variables for accuracy, validation accuracy, loss, and loss during validation, we keep track of the training history.
- Using Matplotlib, we plot training and validation accuracy and loss to show the training process.
- We create a h5 file to store the learned model.
- The model that was previously stored is then loaded for testing.
- We prepare test photos from the healthy and unhealthy categories, load test images from those categories, and then use those test images with the loaded model to produce forecasts. The desired label and confidence are received.
- For an interactive webcam demonstration of image classification, further code is written. It takes a picture using the camera, edits it, and then feeds it to the model for forecasting. Using OpenCV and Matplotlib, the predicted class is shown on the image.

Accuracy:

Initially,the model accuracy was around 80 to 85%, upon fine tuning of the model. The model accuracy has improved significantly.



3.3.4 Arduino code for stepper motors

```
const int stepX = 2;

const int dirX = 5;

const int stepY = 4;

const int dirY = 7;

const int enPin = 8; int

dt = 5000;


// Define the interval in milliseconds for the motors to move (once a week)

const unsigned long movementInterval = 7 * 24 * 60 * 60 * 1000;  // 7 days in
milliseconds


// Define variables to track the last movement time unsigned

long lastMovementTime = 0;
```

```
void setup() {
  // Pin configuration and initialization
pinMode(stepX, OUTPUT);
pinMode(dirX, OUTPUT);
pinMode(stepY, OUTPUT);
pinMode(dirY, OUTPUT);   pinMode(enPin,
OUTPUT);   digitalWrite(enPin, LOW);
digitalWrite(dirX, HIGH);
digitalWrite(dirY, LOW);
}


void loop() {   // Get the current time
unsigned long currentTime = millis();


  // Check if it's time for the motors to move   if (currentTime -
lastMovementTime >= movementInterval) {
   //         Move         the         motors
moveMotors();

   //       Update       the       last       movement       time
lastMovementTime = currentTime;
```

```
  }
}


void moveMotors() {
  // Rotate the X-axis motor
for (int x = 0; x < 300; x++) {
digitalWrite(stepX, HIGH);
delayMicroseconds(dt);
digitalWrite(stepX, LOW);
delayMicroseconds(dt);
  }


  delay(dt); // One second delay


  // Rotate the Y-axis motor
for (int x = 0; x < 300; x++) {
digitalWrite(stepY, HIGH);
delayMicroseconds(dt);
digitalWrite(stepY, LOW);
delayMicroseconds(dt);
  }
```

```
  delay(dt);

}
```

Explanation:

Using the A4988 driver modules, we have created Arduino code to control the motion of stepper motors. This code's objective is to synchronise a motorised camera's movement with the motors in a visual scenario. In a prototype of vertical farming, the camera is made to align with trays. To do this, we must first specify the pin assignments for each motor's step and direction signals as well as the enable pin that turns off the motors. The interval between steps is represented by the "dt" variable.

During setup, the appropriate pins are assigned as outputs, and the enable pin is set to LOW to activate the motors. In addition, we switched the Y-axis motor's direction to LOW and the X-axis motor's direction to HIGH.

In the main loop, the amount of time since the last movement is continuously monitored. Once the specified movement interval—in our case, one per week—has passed, we call the moveMotors() function.

The moveMotors() method manages the motors' rotation. The X-axis motor is rotated 300 times by setting the step pin to HIGH, using the dt variable to add a delay, setting the step pin to LOW, and then adding yet another delay. Repeating this procedure results in the desired number of steps being finished. The Y-axis motor is rotated for 300 steps after a one-second delay using the same logic as the X-axis motor.

By putting this code into use, we ensure that the motors move in accordance with the motorised camera's movements in a specified pattern. In the prototype for vertical farming, the camera is utilised to position itself in respect to the trays. For further analysis and decision-making in our project, the camera's collected data will be added to a machine learning model.

Chapter 4

Testing and Validation/Discussion

4.1 Testing

Motors Testing

First the motors are tested with its code to see if they are moving properly then after the pulleys and the camera is attached the model is trained and tested by implementing the data set including the several amount of images that includes the images of the healthy and non-healthy plants.

After several testing times of testing finally the movement of the motors became stable and the model started to predict the plant's health truly.

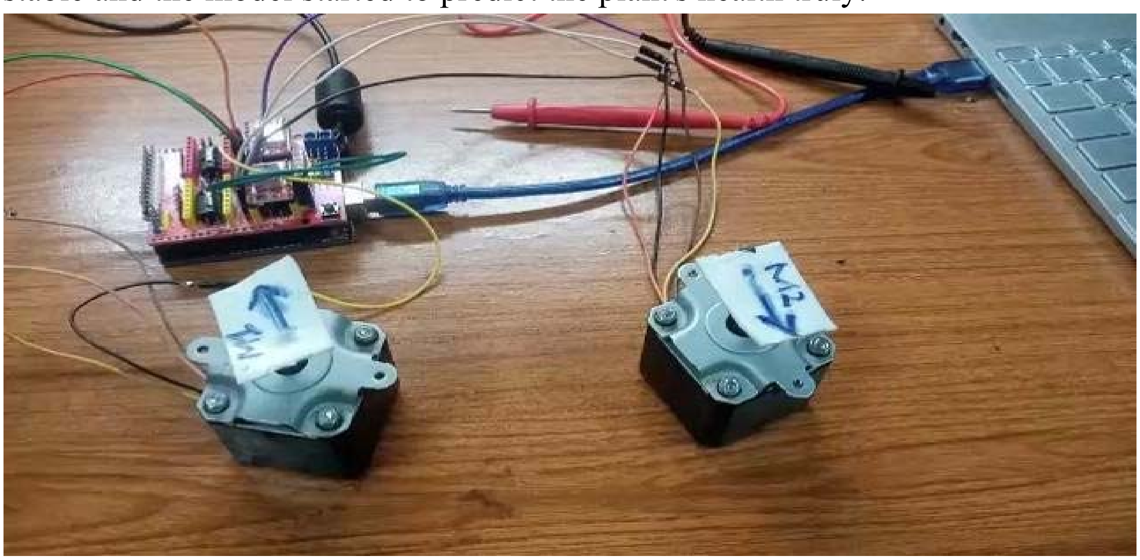
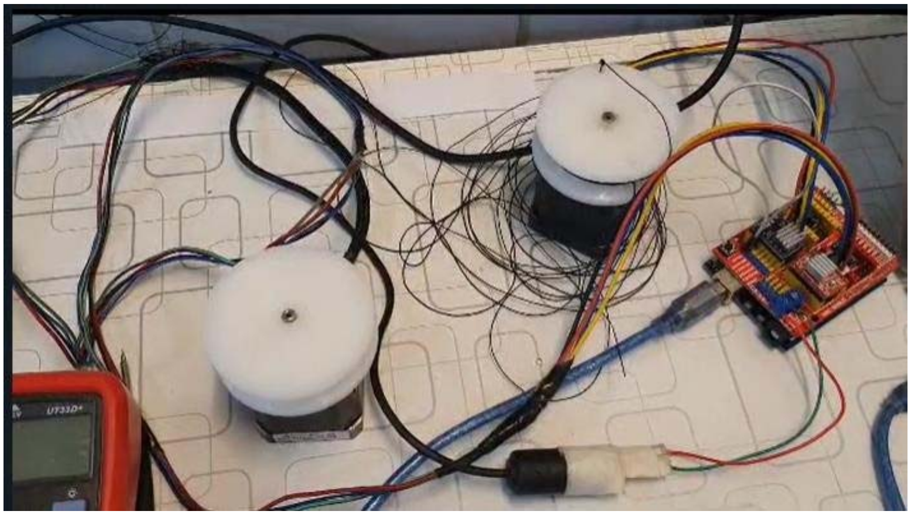


Fig4.1; Testing of working of motors



4.1.1 Prototype

After the testing of motors they are mounted on the prototype structure to test the stability and the movment.in place of camera a small weight is placed to check. The motors moved the weight object from initial to the end one by one in the vertical position.The plants were also set up in the trays to see if they could hold the weight properly.

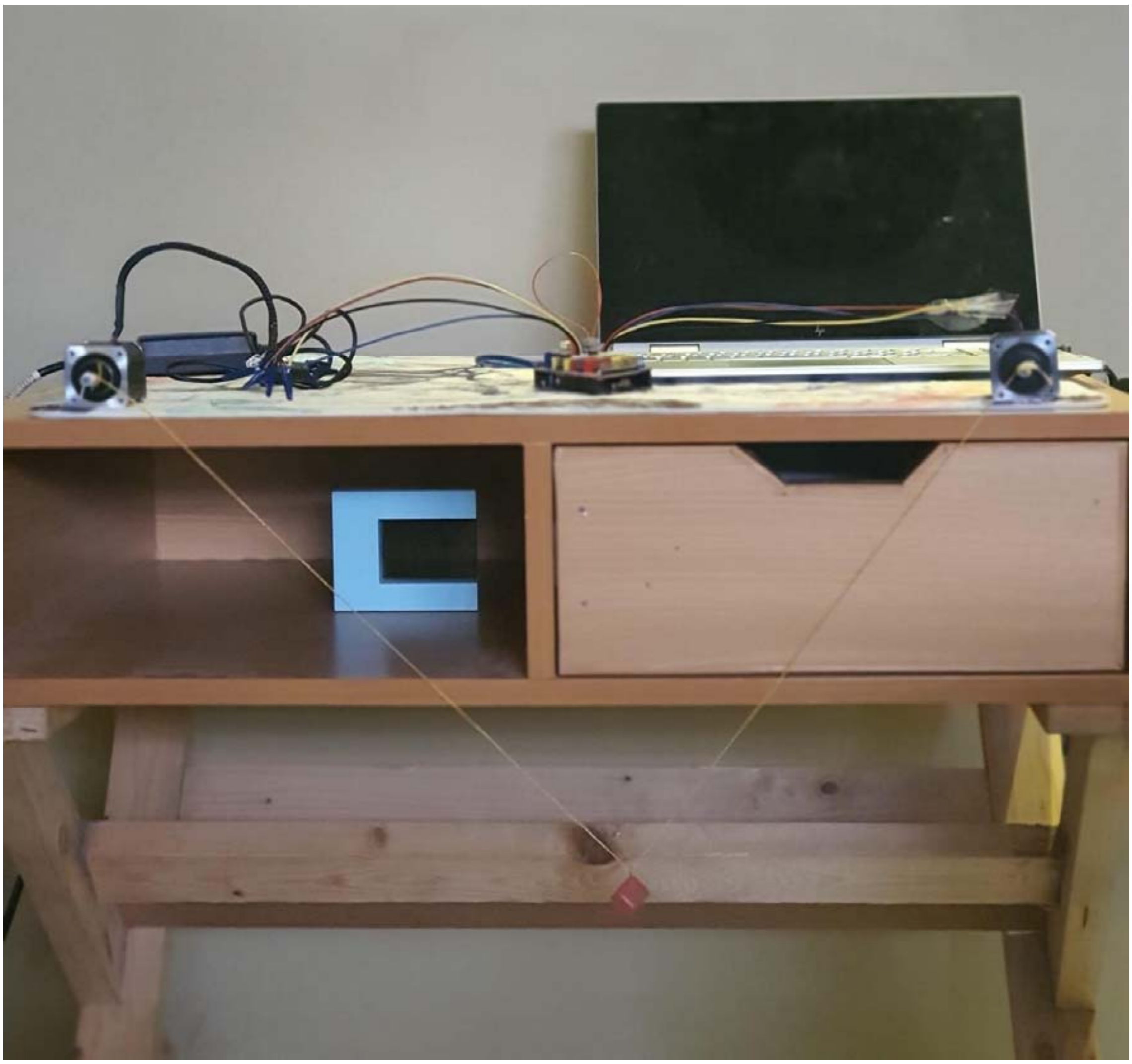
Fig4.2

4.1.2 Test Cases

First the motors are tested and stimulated as they are the most important and the initial part of the model. After stimulation motors the main focus is the stability of the motors and its vertical movement.

The next aim is the detection of the plants and crops diseases and their health assessment. This is done through the machine learning algorithms. The system should monitor each plant and incase of any unusual detection the system send the quick alert to the connected computer.

The various test cases are as followed:



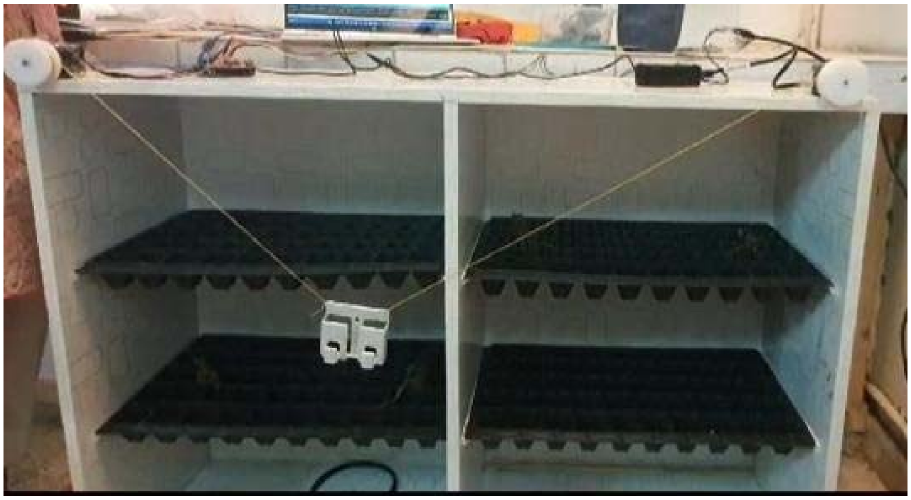
4.2 Results/Output/Statistics

Ml model

After the complete training of the model with the data set and the code for the running of the model. The model successfully predicts healthy and unhealthy lettuce plants with the health category and confidence appended on the predicted images.

Hardware and Mechanical

The model is working properly.it accesses each plant and it detects the unhealthy plants reading the data set accurately. The Arduino is working fine fir the movement of motors and the algorithm for the plants detection provided to the model makes the robot differentiate between the unhealthy and healthy pant easily.



Chapter 5

Conclusion and Future Recommendations

5.1 Conclusion

In conclusion, the objectives set forth for this Final Year Project (FYP) have been successfully achieved.

The main objective was to create an intelligent robotic camera positioning system or precisely integrating 2d cable-driven robot in vertical farming enviroment for monitoring health of plants using computer vision and machine learning techniques. In addition to developing our own machine learning model for plant health detection, we also reviewed and compared a number of other models to determine which was most effective.

One of the significant challenges encountered during the project was stabilizing the motorized pulley system and accurately positioning the camera. Multiple revisions and a significant amount of work were needed for this. Another difficulty was outfitting the machine learning model with computer vision capabilities and properly integrating them. The project's success depended heavily on the selection of the proper computer vision tool and its effective interfacing.

The novelty and success of this project are found in the development of a fully autonomous robotic crop health monitoring system that is especially suited for vertical farming. The system demonstrates cost-effectiveness, robustness, and adaptability. It offers a flexible workspace for effective crop monitoring and management. We have increased accuracy levels beyond what was possible with existing solutions by utilizing computer vision and machine learning.

Overall, we have successfully executed the goals we set at the beginning of the project. The developed robotic system offers a practical and innovative solution for monitoring plant health in vertical farming. The combination of machine learning and computer vision has paved the way for better plant monitoring, planning and management. The results achieved in this project will have a significant impact on the agricultural industry and open the door for further progress in the field of autonomous farming systems.

5.2 Future Recommendations

Based on the achievements and challenges faced during the project, the following future recommendations can be made:

5.2.1 Improved robustness:

The developed robotic system has achieved a certain degree of robustness, but further improvements are possible to increase its durability and stability. This includes improving the mechanical design, optimizing the electric pulley system, and implementing measures to minimize vibration and disturbances.

5.2.2 Advanced sensor integration:

Consider integrating additional sensors to extend the capabilities of your crop health monitoring system. For example, integrating environmental sensors such as temperature, humidity and soil moisture sensors can provide valuable data for comprehensive crop health analysis.

5.2.3 Continuous model improvement:

A machine learning model for detecting plant health has been developed, but the process of improving the model is still ongoing. This includes gathering more diverse and richer training data, exploring advanced machine learning algorithms, and fine-tuning models for improved accuracy and performance.

Collaboration with Agricultural Experts: To further refine the system and ensure its effectiveness in real-world farming scenarios, collaborate with agricultural experts and farmers. Their insights and feedback can provide valuable input for system improvements, as well as validate the system's performance in different agricultural contexts.

References

The project report must be considered as a very standard report, and therefore, it should follow all rules, guidelines, and protocols of gathering and presenting information, and implementing that, and drawing conclusions out of it. All these activities require appropriate and authentic sources of information, and that particular information must be referenced or cited according to the copyrights and other guidelines. Therefore, to make the report original, it should be free from **plagiarism** and must follow standard citations and guidelines of citations to represent the reference names. The appendices of a project report should be written in Times New Roman format of font size 10, and it should contain the information which is appropriate and added to the main text like Embedded C program code, raw data, and so on.

It is highly recommended for students to learn and use a reference management tool like Mendeley, EndNote etc.

[1]. Priti Chichkhedeet.al., "Low Power Globally Asynchronous Locally Synchronous Design", (IJAEST) International Journal of Advanced Engineering Sciences and Technologies Vol No. 9, Issue No. 1, 075 – 081

[2]. *Jonas Carlsson, Kent Palmkvist, and Lars Wanhammar*, GALS Port Implementation in FPGA,Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

[3]. Available online at: http://www.iis.ee.ethz.ch/async/pub/async00.pdf

Annexure 'A'
Title of Annex, if any