SAFS: A Railed Automobile Strawberry Harvester using Dual-Arm Manipulator via YOLOv4 Algorithm for Detection and Navigation with Cloud-based Webpage for Monitoring

A Project Proposal Presented to the Faculty of

Electronics Engineering Department

College of Engineering

Technological University of the Philippines

In Partial Fulfillment of the Subject Requirements for the Degree of

Bachelor of Science in Electronics Engineering

Submitted by:

AREGON, Henna Jane N.

BIHAG, Kurt James B.

CALAMBAS, Kenneth V.

CIRERA, Ma. Erika Rose M.

SERRANO, Fiona Apple H.

VILLARAMA, Chloe V.

Adviser:

Engr. Mark P. Melegrito

2023

APPROVAL SHEET

This Project Study entitled **"SAFS: A RAILED AUTOMOBILE STRAWBERRY HARVESTER USING DUAL-ARM MANIPULATOR VIA YOLOV4 ALGORITHM FOR DETECTION AND NAVIGATION WITH CLOUD-BASED WEBPAGE FOR MONITORING",** has been prepared and submitted by the following proponents:

Aregon, Henna Jane N.,	Cirera, Ma. Erika Rose M.
Bihag, Kurt James B.,	Serrano, Fiona Apple H.
Calambas, Kenneth V.,	Villarama, Chloe V.

In partial fulfillment of the requirements of the Degree of **Bachelor of Science in Electronics Engineering** is hereby recommended for approval:

ENGR. MARK P. MELEGRITO Thesis Adviser	**ENGR. JESSICA S. VELASCO** Panel Member
ENGR. AUGUST C. THIO-AC Panel Member	**ENGR. JOMER V. CATIPON** Panel Member

Accepted and approved in partial fulfilment of the requirements for **The Degree of Bachelor of Science in Electronics Engineering.**

ENGR. TIMOTHY M. AMADO Head, ECE Department	**ENGR. NILO M. ARAGO** Dean, College of Engineering

ABSTRACT

The Philippine agriculture sector has always required intensive labor and monitoring, which is mostly relying on the ever-decreasing number of farmers. To make up for the declining labor force in the sector there is a need to develop and improve on agricultural automation. This study proposes a prototype of an automated method of harvesting strawberries using a mobile dual robotic arm. At the same time, the design has the ability to detect and identify ripe strawberry fruit using YOLOv4 as its object detection algorithm. The process of harvesting starts with the Raspberry Pi camera detecting ready to harvest fruits wherein the processing unit will extract the fruits' coordinate and relay it to the system's micro-controller (Arduino Mega 2560) which utilizes G-code for the arm's movement. In order to cut the fruits' peduncle, a modified 2-finger design of an end-effector was also developed. It can only reach two feet above and below but has five degrees of freedom to make up for it. Vision and movement tests for the system were conducted to verify the efficiency of the harvester in terms of class recognition and movement speed. Through analyzation of testing it was known that the current state of the machine can achieve an average detection time of the vision system is 3 seconds, mean average precision is 75.91%, and has an average harvesting speed of 7 seconds per fruit detected.

ACKNOWLEDGEMENT

This project research would not be possible for the proponents without the help of the following:

Almighty God, for imparting His continuous blessing and mercy throughout the whole process of producing this study.

Engr. Glenn C. Virrey and Engr. Mark Melegrito, for their invaluable guidance and support. Their insightful feedbacks have been of great help for the device's development and improvement, as well as the time imparted on taking online meets and conducting mock defenses have severely impacted the outcome of this research.

Panel members, Engr. August C. Thio-Ac, Engr. Jessica S. Velasco, Engr. Catipon, for being objective, providing constructive criticism, and suggestions that was carefully taken into consideration by the proponents, and have thoroughly refined this study. Their diverse perspectives have pointed out what was lacking and has pinpointed the areas to focus on, playing a pivotal role in this research's form.

Mr. Crispulo Crizar and family, for their willingness to lend their place as the proponents' testing site. Their assistance and daily monitoring of the device is deeply appreciated and that has led to a successful deployment.

Mr. and Mrs. Villarama, Bihag, and Serrano, for their unwavering support, and encouragement all throughout the entire completion of this project. The shelter accommodated, as well as the sustenance supplied by each families' homes have completely provided the proponents the energy and will to finish this research. The proponents appreciate the sacrifices made and the efforts committed into the completion of this study.

The proponents whole-heartedly impart their utmost appreciation and sincere gratitude to these distinguished individuals, recognition is to be applauded to these people, and that their efforts and kind-heartedness will forever be engraved into the making of this research.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER 1

THE PROBLEM AND ITS BACKGROUND

This chapter will present the brief introduction, background of the study, research gap, research objective, significance of the study, scope and limitation, and lay the definition of terms. An overview of the entire project will also be presented.

1.1 Introduction

Traditional farming in the Philippines is a back-breaking labor and requires meticulous monitoring. Aeroponics is a rising alternative solution for this problem. Aeroponic systems exclusively utilize a nutrient-rich mist to provide plants with nourishment. This method removes the necessity for a conventional growing medium, allowing the roots to hang freely in the air and receive nutrients through specialized misting systems. This research aims to achieve a hundred percent automated aeroponic system including the harvesting.

The agricultural sub-sectors of farming and fishing have held significant importance in the Philippines, largely due to its terrain and tropical climate. Sugarcane, rice (palay), coconut, and banana cultivation have been among the most prominent agricultural activities in the country, and these crops have also been major contributors to the country's export market. In recent years, fruits, nuts, as well as animal and vegetable fats and oils, have played a significant role in the country's overall agricultural exports (Statista, 2021).

The agricultural industry confronts various challenges, including a lack of field personnel and escalating expenses related to fruit harvesting. To tackle these issues, the farming sector must prioritize labor-saving techniques and expand its scale of operations. Recent years have witnessed notable progress in agricultural automation, which has led to labor efficiencies and the ability to undertake large-scale farming. Nevertheless, when it comes to fruit harvesting, a significant portion of the work still relies on manual labor.

Manual harvesting involves a ton of physical labor while under the scorching sun, and a time-consuming task. Right now, people are living in the middle of a technological boom, wherein most of the things are innovatively advanced. To make the said task labor-free and entirely automatic, with the help of image processing and a robotics system, an automated harvester can be created to be its solution.

Farmers' earnings have risen steadily over the previous five years, despite the fact that fewer people are entering the agriculture sector and its proportion to overall employment is declining. According to the Agricultural Indicators System of the Philippines Statistic Authority there is a 3 percent decrease in the employment in agriculture from 2019 to 2018. The Federation of Free Farmers clarifies that there are various causes for the decline, this includes the age of the farmers and old farmers that are not replaced by new ones as lack of incentives go into farming (Simeon, 2020).

Concerns about agriculture's future originate from the sector's shrinking workforce, Filipino farmers' increasing age, and low enrolment in agricultural-related higher education programs. The perception that young Filipinos are seen as beneficiaries of agricultural programs rather than leaders has resulted in a significant oversight regarding their participation in policy discussions. The issue of youth access to land and agricultural markets is often neglected, even though it holds the potential to stimulate active involvement and unlock a wide range of opportunities in the field of agriculture, extending beyond traditional farming roles (Asis, 2020).

In agricultural applications, such as the mechanized harvesting of fruit and vegetables, autonomous robotic harvesting is becoming more popular. Farmers are continually searching for ways to enhance the quality of their crops while simultaneously reducing expenses and the need for a large workforce. Harvesting robots prove to be valuable in such circumstances. Although capturing objects has been extensively studied with numerous suggested approaches, accurately determining the precise location of items remains a challenging task due to inherent variations in shape and size, obstructed views, and unpredictable lighting conditions.

However, commercial robotic harvesting is difficult since strawberries are readily broken and bruised. Second, strawberry harvesting necessitates extremely careful techniques because strawberries ripen in different ways. As a result, berries vary greatly in color and size at any one moment. Lastly, strawberries grow in clusters, making it difficult to distinguish and harvest individual strawberries.

1.2 Background of the Study

Robot designs in today's competitive world are created to simplify and improve quality wherever possible. People from the unskilled sector have shifted to the skilled sector as technology and modernization have advanced. The agricultural sector's solution for harvesting fruits and vegetables is manual labor and a few other agro bots, which are costly and have various harvesting limitations. Although harvesting robots are available, the cost of such designs may be prohibitive for small and medium-scale producers. An integrated robot system by Raja et al. (2022), was created and tested against several products using software to ensure its validity. In the form of hardware, a prototype of the designed system was implemented and tested. Automatic crop harvesting was performed in this development using the method of position detection and harvesting using a robotic manipulator with a harvesting hand that does not damage the crop. The YOLO V3 algorithm is used in this study to identify and locate fruits. This includes navigation that employs GPS and QGroundControl software to improve accuracy in locating fruits and navigating the guided vehicle. The accuracy of the detection of crops such as carrots was found to be 93% in 4s time to detect the crop and cantaloupes for 93% in 2s.

This study by Feng et al. (2018), introduced a new robot for harvesting cherry tomato bunches in a specific cultivating mode. As a functional model, the robot could move along the rail automatically, identify and locate the mature bunch, hold and separate the target, and even collect the harvested fruit. The fruit was detected using a vision servo algorithm in this study. The field test results showed that the robot could successfully harvest 83% of the mature bunch prepared for testing, despite the fact that each successful harvest required an average of 1.4 attempts. And a single successful harvesting cycle takes 8 seconds, excluding movement time.

Robotic harvesting in a cluttered and unstructured environment is still difficult. Therefore, Xiong et al. (2020), created an autonomous robot that can pick strawberries in polytunnels continuously. This study includes crop navigation using Hokuyo LIDAR to detect occlusions. To allow the harvesting system to pick strawberries in clusters, a novel obstacle separation algorithm was proposed. The gripper is used by the algorithm to push away surrounding leaves, strawberries, and other obstacles. They presented a theoretical method for generating pushing paths based on the environment. In addition to manipulation, an improved vision system is more resilient to lighting variations, which was developed using color-versus-light-intensity modeling. Furthermore, a low-cost dual-arm system with an optimized harvesting sequence that increases efficiency and reduces the risk of collision was developed. The existing gripper was also improved to allow the robot to pick directly into a market punnet, eliminating the need for repacking. During trials on a strawberry farm, the robots' first-attempt success rate for picking partially surrounded or isolated strawberries ranged from 50% to 97.1 percent, depending on the growth conditions. On a second attempt, the pick success rate increased to 75–100%. During field tests, the system was unable to select a target that was completely surrounded by obstacles. This failure was attributed to vision system limitations as well as insufficient gripper dexterity. However, picking speed improved over previous systems, with manipulation operations taking only 6.1 seconds in one-arm mode and 4.6 seconds in two-arm mode.

A reliable and robust harvesting robot is required for fully autonomous harvesting of oyster mushrooms in the greenhouse. In this study by Rong et al. (2021), propose an oyster-mushroom harvesting robot with a mobile platform on a four-wheel rail that moves in three directions and can harvest oysters and mushrooms throughout the greenhouse. The perception module and the end-effector are two critical components of the harvesting robot. This study employs a 3 DOF robotic arm with a Cartesian coordinate system. Finally, the current soft gripper is modified to grab oyster mushrooms after an enhanced SSD algorithm is suggested to recognize mushrooms. Field experiments demonstrate the feasibility and accuracy of the proposed robot system, with the mushroom recognition success rate reaching 95%, the harvesting success rate reaching 86.8% (without taking into account mushroom damage), and the harvesting time for a single mushroom being 8.85 s.

As labor requirements in horticulture become more difficult, automated solutions are becoming a viable option for maintaining productivity and quality. This paper describes the design and performance assessment of a unique multi-arm kiwifruit harvesting robot developed by Williams et al. (2019) to operate autonomously in pergola-style orchards. The harvester is made up of four robotic arms that have been specifically designed for kiwifruit harvesting, each with a novel end-effector developed to ensure safe kiwifruit harvesting. The vision system takes advantage of recent advances in deep neural networks and stereo matching to detect and locate kiwifruit in real-world lighting conditions using a Fully Conventional Network algorithm. This has a 76.1% success rate in detecting the fruit. In addition, a novel dynamic fruit scheduling system that has been developed to coordinate the four arms throughout the harvesting process is presented. The harvester's performance was evaluated in a comprehensive and realistic field trial in a commercial orchard setting. The results show that the presented harvester can harvest 51% of the total number of kiwi fruit within the orchard in an average cycle time of 5.5 seconds per fruit.

The development of an autonomous tomato harvesting robot by Yaguchi et al. (2016), is presented in this paper. They created a harvesting robot with a rotational plucking gripper, a 6DoF single-arm, a stereo camera capable of taking pictures in direct sunlight, and an omnidirectional mover. They also used the EusLISP inverse kinematics algorithm for harvesting and the RANSAC algorithm for crop detection. In the tomato robot competition, the robot harvested tomatoes from tomato clusters and tomato trees, with a harvesting speed of about 80 seconds per fruit and a success rate of about 60%. On the real farm, they tested the robot with tomato trees in a semi-outdoor setting to demonstrate its effectiveness and robustness in direct sunlight. According to the result of harvesting with real tomatoes, they improved the robot motion, and finally, harvesting speed was up to 23[s/fruit], however, the gripper may grasp multiple fruits in case of a very cluttered cluster and the calyx also may be broken when the stem angle is deep from the rotation axis.

1.3 Research Gap

Through the accumulation of recent technological advancements, it has become possible to explore potential developments in the field of automated crop harvesting. Numerous researches and field studies about mechanized harvesting have been conducted involving varying fruit yielding crops, each having different means of fruit collection and learning algorithms for object detection. However, strawberry harvesting has been deemed as a difficult task for robotic harvesters due to the fruits' clumping tendency, high environmental occlusion and form fragility, making the existing harvesting systems only partially automated caused by their inefficiencies in the said obstacles as stated in the studies above.

Moreover, the researchers have determined that crop localization, detection and harvesting rate are the factors to consider when developing an efficient automated harvesting system. But despite the multitude of academic studies present, a lot of them weren't able to accommodate all of the factors mentioned. In order to fulfill the objective of creating a fully automated strawberry harvesting robot, this study will focus on implementing the culmination of several aspects pertaining to pre-existing studies. First, will be the administration of both crop localization and detection using the YOLOv4 algorithm which is an improvement of YOLOv3, which will also enable us to address the problem regarding occlusions due to environmental factors. In line with improving the software aspect of automated harvesting, this study will also implement a web page that will enable the user to monitor the current status of the harvester, the website's interface will be configured to show the real time state of the harvesting unit: battery percentage, charging time, harvesting status, machine run time, harvesting time for each fruit, harvested strawberries, and a comparison counter for the number of strawberries detected compared to the ones successfully harvested. Then, the system's main harvesting unit will be composed of a dual-arm manipulator with attached modified end-effector to make up for an increased harvesting rate while maintaining the fruit's quality making it suitable for market.

Also, knowing that most of the previous works' field-testing locations were executed in massive areas such as greenhouses and fields, resulting in bigger harvesting prototypes, the researchers intend to build a small-scale prototype to be deployed in a closed area in order to incorporate it in a modified automated aeroponic system. Furthermore, the proponents will also develop the mobility of the harvester by using a railing system as the pathing for the mobile harvesting machine which accommodates the specifications and dimensions of the cultivating area.

1.4 Research Objectives

This research aims to develop a smart strawberry harvesting robot with dual-arm manipulator that has the ability to receive notifications of ready-to-harvest strawberries, localize, detect, and harvest it using machine learning algorithm, moving as well through a modified railway in a smart aeroponic farm, where all harvesting parameters and harvester's information is stored and monitored in a webpage.

Specifically, it aims:

1. To construct a harvesting vehicle with a dual-arm manipulator harvesting robot with an end-effector device that is controlled by the received data and harvesting position via Arduino, moving in a railway developed in the smart aeroponics farm.

2. To develop a vision system that will detect, locate, and harvest the ripe strawberries using YOLOv4 algorithm.

3. To develop and integrate a web page to the system that monitors and displays the harvesting parameters and the harvester's status through cloud database Arduino IoT Cloud.

4. To evaluate the functionality, reliability, and efficiency of the automatic harvesting system through actual field tests.

1.5. Significance of Study

Automated harvesting equipment is essential for farmers in reaping some crops. The findings in the study would be used for both small and large farms, due to the shortage of farmers and lack of machines for harvesting. The system utilizes non-invasive devices, which have an end-effector to harvest and a camera to detect if the strawberries are ready to be harvested. Increased labor in farmers affects the time frame of crop delivery. To reduce this problem, the uses of the automated harvester are being developed to lessen the time being consumed for harvesting.

In regards to DOST research proposal guidelines, under the Harmonized National Research and Development Agenda (HNRDA) this study was under Section III "Agriculture, Aquatic and Natural Resources (AANR)" in the field of temperate fruit crops. Moreover, in Sustainable Development Goals (SDG), this project was concerned in Goal 9 of SDG which is Industry, Innovation and Infrastructure.

The rapid expansion of agricultural crop robotics is the result of the convergence of maturing mechatronics technology and the demand for alternatives to human labor in crop production. The engineering side of agricultural robotics has advanced rapidly (Duckett et al., 2018). However, understanding the economic implications has been slowed.

This study will give rise to new technology. With the help of this study, innovations are awaiting in image processing, surveillance for monitoring, stored data in the cloud and automatic harvesting that may progress for further research.

1.6 Scope and Limitations

The study will be focusing on creating an automated strawberry harvester that utilizes the YOLOv4 algorithm to effectively detect and locate ripe strawberries. The project will be deployed in a four Vertical Aeroponic Farming System at 294 C. Callejo St. Narra Road Las Brisas de Tagaytay, Mendez Crossing West Tagaytay City. A four-wheel cart that is placed in a rail will be the body of the harvester where it will move accordingly on the position of the ripe strawberry. The dual-arm manipulator is responsible for moving the end-effector where it moves based on the position of the ripe strawberry. The two-finger end-effector is designed to cut and hold the peduncle to harvest the ripe strawberry. With the use of YOLOv4 Algorithm, it will determine the location and the ripeness of the strawberry for the end-effector to harvest. With that said algorithm it will still detect the strawberries even with occlusion. The study will also have a web page where the user can check the status of the harvest. The user will be able to monitor the harvesting status, battery percentage, and the number of strawberries harvested. The parameters monitored by the harvesting unit of the Aeroponics system will be collected and stored in the cloud via Arduino IoT Cloud. The website of the Harvesting unit is created using HTML which is then integrated into the Raspberry Pi 4. The robotic arm is run by the Arduino Mega 2560 which is a microcontroller board based on the ATmega2560, while the mobile cart is connected to the Arduino Uno. The two Arduino are connected to the Raspberry Pi I/O ports, while the camera is also linked to the Raspberry Pi. Programming through Arduino IDE will enable movement of the arm which will be based on the detected results' bounding box coordinates via the vision system.

The proposed method has its limitations. The harvester is placed on a rail which means the vehicle's movement is bidirectional only. The machine does not operate on renewable energy, like solar energy, so it runs with a LiFePo4 battery due to the closed area where the project will be deployed and also, to extend the harvesting distance of the robot. The harvesting unit can only be turned off manually as there is no control panel included in the website of the harvesting unit, but can be programmed to automatically stop once an error happens in the system. The observed parameters such as remaining battery, strawberry harvested, and machine status is what you can only see on the website. However, on the machine statues there is what we label as "idle", in which the harvesting unit is inactive.

1.7 Definition of Terms

1. Robotic Arm - A mechanical device that imitates human arm movements to perform controlled tasks.

2. Visual Detection - Using cameras or sensors to capture and interpret visual information from the surroundings.

3. YOLOv4 - A cutting-edge real-time object detection algorithm utilizing deep learning techniques.

4. Grblgru - Open-source CNC software for controlling robotic arms and CNC machines, compatible with Arduino Mega and Raspberry Pi.

5. Arduino Mega - A microcontroller board enabling the construction of various electronic projects, including robotic arms, with multiple input/output pins.

6. Raspberry Pi - A versatile single-board computer used for applications like controlling robotic arms and interfacing with other components.

7. End Effector - The tool or device attached to the robotic arm's end, interacting with the environment for specific tasks.

8. Inverse Kinematics - A mathematical technique to determine the joint angles needed to position the end effector accurately.

9. Forward Kinematics - A mathematical technique to determine the position and orientation of the end effector based on joint angles.

10. Servo Motor - A motor using feedback to control position, velocity, and acceleration, often used for precise control in robotic arms.

11. Stepper Motor - A motor moving in steps, providing accurate positioning and speed control in robotic arms.

12. Object Detection - Locating and classifying objects within images or video streams.

13. Training Data - Labeled images used to train an object detection model like YOLOv4 to recognize specific objects.

14. Neural Network - A computational model inspired by the human brain, capable of learning and performing tasks like image recognition.

15. Convolutional Neural Network (CNN) - A type of neural network commonly used for image processing tasks, including object detection.

16. Real-Time Processing - Performing computations and making prompt decisions with minimal delay for accurate and timely responses in robotic arms.

17. Image Processing - Analyzing and manipulating digital images to extract useful information or enhance visual quality.

18. Object Tracking - Monitoring the movement of a specific object within a sequence of images or video frames.

19. GPIO (General Purpose Input/Output) - Configurable input/output pins on microcontrollers or single-board computers for interfacing with external devices.

20. OpenCV (Open Source Computer Vision Library) - An open-source library with computer vision algorithms and functions, including object detection and image/video processing.

21. Serial Communication - Sending data sequentially, bit by bit, between devices over a serial interface like Arduino Mega and Raspberry Pi communication.

22. ROS (Robot Operating System) - A flexible framework for writing robot software, providing libraries and tools for creating modular and reusable robotic systems.

23. GUI (Graphical User Interface) - A visual interface enabling user interaction through graphical elements like buttons, menus, and windows.

24. HMI (Human-Machine Interface) - The point of interaction between a human operator and a robotic system, involving hardware and software components.

25. Calibration - The process of determining and adjusting parameters or settings in a robotic system to ensure precise and accurate operation, including camera calibration and joint angle calibration.

CHAPTER 2

REVIEW OF RELATED LITERATURE

This chapter provides an overview of previous research on automation of harvesting. It introduces some studies and literature related to the project. Also, the main focus of the research is described in this chapter.

2.1. Smart Farming System

2.1.1. Aeroponic Farming

According to L. Lucero et al. (2020), a contemporary growing technique that may be utilized in place of traditional farming is aeroponics. This culture method increases output while reducing pesticide use and water usage. However, it necessitates a tightly controlled environment, including sufficient compensation for changes in environmental conditions, accuracy in irrigation intervals, fault detection in the aeroponic system that shuts down the nutrient supply, or other circumstances that can lead to long-term plant damage.

To increase the effectiveness of agricultural operations in an aeroponic green leaf lettuce crop, a self-contained supervision and control system has been suggested in the current study. Additionally, a cloud-based IoT application for remote data monitoring has been included, enabling the appropriate system functioning by enabling knowledge of the variable's state. After roughly dropping from 6.9 to 5.9, the pH value stayed within acceptable norms. Due to the greenhouse's climatic circumstances, the temperature and relative humidity were very varied, with values ranging from 10 percent to 94 percent for relative humidity and 8°C to 44°C for temperature.

The automated aeroponic system suggested lowers operational errors resulting from human intentions and can deliver the nutrients solution in the amount required for adequate plant growth, considering its developmental stages and environmental conditions. This improves production by hastening development due to better root oxygenation. Additionally, water resources are being used in a more efficient manner. Finally, regardless of the weather, it is feasible to serve consistent and sanitary veggies all year round.

Table 1. Summary of Aeroponic Farming Related Studies

Author	Year	Title	Relevant Findings	Relationship to SAFS
L. Lucero, D. Lucero, and G. Collaguazo	2020	Automated Aeroponics Vegetable Growing System Case Study Lettuce	To increase the effectiveness of agricultural operations in an aeroponic, a self-contained supervision and control system has been suggested in the current study.	Aeroponic System

2.1.2. Automation and Mechanization in the Agricultural Field

For the past few years, agricultural techniques have drastically changed through the implementation of smart technologies. According to Pitla et al. 2020, both crop and livestock yields can be developed through improvement in a lot of aspects, and one of those include optimal management of inputs which consists of managing how they are cared for. Through automation and mechanization, these parameters can be managed thoroughly, and also cut cost on labor.

Not only does automation create an opportunity for the betterment of agricultural practices through management of crop inputs, it also prepares the field of farming for reverting back to scale neutral technologies and may also progress to Farming-as-a-Service (FaaS). This assumption can be backed by the ever growing demand of mechanization in the field, wherein development of bigger autonomous machines creates the concern probable removal of manned processes and can also cause larger liability during collision accidents. With that in mind, the idea of a smaller scale automation becomes a better option for the future of agricultural mechanization (Pitla, 2020).

Table 2. Summary of Automation and Mechanization in the Agricultural Field Related Studies

Author	**Year**	**Title**	**Relevant Findings**	**Relationship to SAFS**
S. Pitla, S. Bajwa, S. Bhusal, T.	2020	Ground and Aerial Robots for	Mechanization in the field of agriculture is an	The concept of smaller-scale mechanization is

Brumm, T. Brown-Brandl		Agricultural Production: Opportunities and Challenges	ever growing demand, giving opportunity to the integration of newer and smarter technologies to previous innovations	found to be more ideal for agricultural innovation. And SAFS is a small-scale harvester for smart aeroponics systems.

2.2. Autonomous Mobile Robot

Mobile robots are capable of moving independently without the intervention of human operators. A robot is autonomous when it can identify the steps to be taken to complete a task on its own, with the assistance of a perception system. It also requires a cognition unit or a control system to coordinate all of the robot's subsystems. The areas of locomotion, perception, cognition, and navigation are the foundations of mobile robotics (Rubio et al., 2019).

2.2.1. Mobile Robot Motion Control System

Robots are rapidly being employed in a variety of industries, including industry, military, healthcare and allied fields, search and rescue, management, and agriculture, as technology and science grow and productivity improves, allowing humans to do complex tasks. Mobile robots can move from one location to another to perform desired and complex tasks. Software and integrated sensors, such as infrared, ultrasonic, and magnetic sensors, as well as webcams and GPSs, control a mobile robot. Line follower robots can be used in a variety of industrial applications, including the transport of heavy materials.

The researcher of Farkh et al. (2021), presented a mobile-robot platform with a fixed four-wheel configuration chassis and an electronic system built with Arduino Uno interfaces in this study. The findings show that neural networks are ideally suited to mobile robots since they can work with erroneous data. A Raspberry Pi, which is more powerful than any Arduino board, can be used to create and apply more advanced and intelligent control methods based on computer vision and convolutional neural networks. They developed a line follower robot incorporating an infrared sensor connected to an Arduino Uno and DC motor. The study ensures that the robot moves in the desired direction using the infrared sensors that reads line detection. It predicts the direction using the deep learning controller. It controls the speed of the four-dc motor. Functionality, material availability, and mobility were all factors in the design. White lines on a black surface and black lines on a white surface can both be detected by the line tracking sensor that is used in this study. They use Artificial Neural Networks. Image processing, speech recognition, robotic control, and power system protection and control management are among applications where ANNs are most suited.

This study of Durmus et al, they demonstrate that greenhouses can be RGB-D mapped and that this map can be used to navigate the greenhouse autonomously (Durmus et al., 2016). Temperature, moisture, light, and pressure can all be measured locally in the greenhouse using this technique. Because the RGB-D map is updated with each run, changes within the greenhouse environment can be investigated. Precision farming tasks such as spraying remedies and fertilization can be completed autonomously inside the greenhouse using this robot system.

More valuable data can also be gathered from plants by using advanced spectrometers or spectral cameras. RGB-D mapping is sensitive to changes in the area; if the previously mapped area changes between two measurements, the robot will have difficulty localizing itself in the area and will rely solely on visual odometry to do so. More capabilities, such as obstacle avoidance, voice commands, and autonomous charging, are planned to be integrated into the robot software in the future. One of the design stages also includes adding a robot arm to the platform.

Table 3. Summary of Mobile Robot Motion Control System Related Studies

Author	**Year**	**Title**	**Relevant Findings**	**Relationship to SAFS**
R. Farkh, K. Al Jaloud, S. Alhuwaimel, M. T. Quasim, and M. Ksouri	2021	A Deep Learning Approach for the Mobile-Robot Motion Control System	The researchers used an infrared sensor to make a line follower which predicts the direction via deep learning control.	Deep learning control
	2016	Data Acquisition from Greenhouses by Using Autonomous Mobile Robot	This study uses RGB Mapping for the navigation of the robotic vehicle.	The navigation is dependent on image processing.

2.3. Automated Harvesting System

2.3.1. Detection and Localization of Fruit

Various agricultural tasks such as defect detection, food measurement, and field monitoring used imaging techniques. In field monitoring the airborne based remote sensing unmanned aerial vehicle was presented to characterize low-, moderate, and high-vigor zones (Johnson et al., 2001); evaluate crop response to the surrounding (Sindhuja et al., 2018); observe ripeness status (Johnson et al.,2004); approximate the number of the fruit on the trees (Safren et al., 2007), and evaluate the effect of the conditions that are uncontrolled (Chen et al., 2020).

In this study T. Dewi, R. Rusdianasari, R. D. Kusumanto, and S. Siproni mentioned that in every sector of human life, the robot presents automation, including agriculture (2022). Automation in agriculture guarantees a solution for fresh fruit collecting because a robot can keep an eye on the farm for a whole day relentlessly. Digital farming is what this robot invasion is called, and by applying it, it is anticipated that the harvested fruit has less dependency and higher quality. Due to the aging farming and fewer youth participating in the farm, less dependency is necessary.

The most suitable type of robot for harvesting is an arm robot because the end-effector can be constructed accordingly. The eye plays a critical role for collecting the crops, the right method to create a capable eye is visual serving. By using image processing the fruit position which is considered as the target is detected.   Image segmentation, blob analysis, and edge detection are the necessary methods used for image processing. Image segmentation divides a photo captured by the camera into several segments or sets of pixels in order to define the representation of a photo into a set of limitations such as lines and curves in images. Every assigned pixel is tasked to find the pixels with the same characteristics. The edge detection methods work by assuming that the edges occur in a discontinuity in the intensity function that is very steep in the image. Edge is defined as a very contrasting intensity value compared to pixels in the neighborhood. Blob (Binary Large Object) analysis is the technique to isolate a group of pixels representing some properties such as brightness and color and compare to surrounding regions within an image. All the points in a blob are considered similar to each other.

During the convolution algorithm, the process also included noise, therefore the result is still noisy, and shape of the fruit is not clear enough. In Prewitt edge detection, points represent the noise, but the result is better than the former. Canny, which is the last edge detection that is based on a Laplacian algorithm, is not noise sensitive. The output filter image is smaller, but the shape of the fruit is visible.

Overall, digital farming is an efficient solution to harvest higher quality crops because of robots being available all the time for harvesting. Image processing is presented for fruit detection and isolates it from vegetation. The results show that the target is detected and needs fewer time and is simple. Its simplicity and quickness in detecting the target is of importance to ensure the available microcontroller in the market can conduct the processing.

Strawberry harvesting robots require accurate fruit detection and localization**.** Strawberry cluster and maturity determination, on the other hand, remain difficult. Occlusions can also lead to flawed fruit localization. For agricultural robotics, particularly fruit harvesting robots, machine vision is critical for target detection and localization.

This study is conducted to improve the image processing algorithm of the manipulator designed by Ge et al. (2019), on Instance Segmentation and Localization of Strawberries in Farm Conditions for Automatic Harvesting. The researchers used the DCNN model and Mask R-CNN to detect and locate the strawberries. They used three classes of different ripeness of strawberries including deformed strawberries. The study used 310 sets of images captured by smartphones and RGB-D camera. The result shows a high accuracy in ripe strawberries followed by the deformed one. The raw strawberries were detected with a score of 0.64 which in this study was said to be accurate as the data set was the same with the ripe ones. Overall, the average F1 score for other classes aside from the ripe one is 0.88. The researchers used width to height ratio to detect occlusion because of the direction and the position of the occlusions which are the key factors that affect the localization of strawberries.

This study by C. McCool et al. (2016), proposes a crop detection system that is dependent on precise crop segmentation at the pixel level. Crop segmentation is carried out using a conditional random field (CRF) and visual texture characteristics, following Hung et al. study. We suggest a unique approach to crop segmentation using local binary pattern (LBP) features, and empirical results demonstrate that this feature outperforms earlier state-of-the-art features. Finally, our detection method searches for highly likely places and identifies them as observed crops using the probability map from crop segmentation. By not presuming a certain form on the viewable crop, this gives occlusion resilience.

The researchers find it hard to distinguish the crop from the backdrop (leaves) since the crop is harvested even when it is green (the same color as the background). Additionally, we put forth a brand-new assessment metric built on a Bayesian framework that enables us to communicate uncertainty. Using information gathered from three commercial locations obtained both during the day and at night, we do detailed analyses using this and other common criteria. To promote more study in the field, this dataset is made available together with the methods and annotated ground truth photography.

By a study conducted by Y. Yu, K. Zhang, L. Yang, and D. Zhang (2019), due to its wonderful flavor and high nutritional value, the strawberry was one of the first tiny berries to be widely planted worldwide. In strawberry farming, harvesting takes up the most time and effort. Many prototypes of commercial strawberry harvesting robots have been developed because traditional hand harvesting is no longer able to satisfy the industry's fundamental needs (Hayashi et al., 2013, 2014; Yamamoto et al., 2010; Qingchun et al., 2012; Kailiang, 2012). In example, Spain Agrobot created the SW6010, a 14-arm robotic strawberry harvester (AGROBOT, 2017). The fruit detecting system continues to be a weak point despite the constant development of automatic strawberry harvesting prototypes. The target recognition (object identification and placement) of strawberry harvesting robots has significant hurdles due to the complexity of the natural environment of orchards and the unstructured properties of fruits (Bac et al., 2014).

The following are the primary influences of natural factors on target detection: natural light intensity, multi-fruit adhesion overlap, and stem and leaf occlusion. At the moment, machine learning algorithms paired with machine vision are the key techniques for fruit target recognition. The appearance properties of fruits are typically extracted from different color spaces using a variety of image pre-processing techniques, such as threshold segmentation, edge detection, region growth, and gray-scale co-occurrence matrix (Zhao et al., 2016). Features of appearance, such as color, size, form, and texture (Jiang and Zhao, 2012; Rizon et al., 2015; Lu, 2015; Arefi et al., 2011). In order to extract the strawberry target's edge, Ouyang et al. (2012) performed a number of processing operations on strawberry images, including median filtering to remove noise, the OTSU algorithm for image segmentation, mean shift clustering to obtain the most distinct shape features, and morphological operations to extract the strawberry target's edge. Different color expressions create various color spaces, and each color space has its own application area. Using a binocular visual unit, the elevated-trough culture strawberry harvesting robot developed by Qingchun et al. (2012) identified ripe fruits by extracting color attributes of hue and saturation in HSV color space. The methods mentioned above can recognize the fruit target from photos, however when the environment's illumination intensity varies or due to shadowing, the identification accuracy suffers. Additionally, it is challenging to locate a generic machine vision technique that can concurrently detect targets for various strawberry growth types, particularly in unstructured situations like those with several fruits adhering to one another, overlapping, stems, and leaf occlusion.

In this study, a Mask R-CNN model that can recognize ripe and unripe strawberries automatically was trained, and mask images of the fruit were generated by the network. The average detection precision, recall, and MIoU rates for the fruit detection findings of 100 testing photos were 95.78 percent, 95.41 percent, and 89.85 percent, respectively. The trained model was very good at detecting strawberry fruit in photos with variable light intensity, as well as adhesion, overlapping, and other complex growth phases in multiple fruits. The proposed model is more robust and general than the four conventional fruit detection techniques. Although the embedded mobile harvesting robot processes frames at an average rate of 8 per second, its speed is less than this.

According to S. Puttemans, Y. Vanbrabant, L. Tits, and T. Goedemé Autonomous robotic harvesting is a growing trend in agricultural applications such as automated fruit and vegetable harvesting. Farmers are constantly looking for ways to improve their output while lowering operating costs and employing fewer people. Harvesting robots come into play here. While the mechanics of grabbing objects is a well-documented problem with numerous proposed solutions, robustly localizing objects remains difficult due to natural variations in shape and size, occlusion, and uncontrolled lighting conditions.

Fully automated fruit detection and localization in orchards are critical components in developing automated robotic harvesting systems. A lot of research has been done on this topic in recent years, either using basic computer vision techniques like color-based segmentation or by using other sensors like LWIR, hyperspectral, or 3D. Recent advances in computer vision present a diverse set of advanced object detection techniques that have the potential to significantly improve the quality of fruit detection from RGB images. Puttemans et al., 2017, propose using an object categorization framework based on boosted cascades of weak classifiers to integrate and demonstrate a fully automated semi-supervised fruit detector on strawberries and apples. In comparison to existing techniques, we improved fruit detection, particularly in the case of fruit clusters, by employing supervised machine learning rather than handcrafting image filters tailored to the application. Furthermore, we incorporate application-specific color information to ensure that our fully automated detection algorithm produces more stable results. Finally, we make recommendations for effective fruit cluster separation. The developed technique has been validated on both strawberries and apples and has been shown to have significant benefits.

In a study made by J. Chen, Y. Lian, and Y. Li, various agricultural tasks such as defect detection, food measurement, and field monitoring used imaging techniques. In field monitoring the airborne based remote sensing unmanned aerial vehicle was presented to characterize low-, moderate, and high-vigor zones (Johnson et al., 2001); evaluate crop response to the surrounding (Sindhuja et al., 2018); observe ripeness status (Johnson et al.,2004); approximate the number of the fruit on the trees (Safren et al., 2007), and evaluate the effect of the conditions that are uncontrolled.

There has been some research towards calculating the yield of citrus fruits. Annamalai and Suk (2003) devised a method for identifying and counting the said fruits. In order to separate the fruit from the background and the leaves, hue and saturation color planes and histogram information was used. Noise was removed by using erosion and dilation processes. Blob analysis was used to count the amount of fruits.

The original RGB image of the citrus tree's object class yellow mature citrus, light green leaves, dark green leaves, grass, dirt, trees, twigs, and other plants were present. Convert RGB image to HSV, thresholding, orange color detection, noise removal, watershed segmentation, and counting were all part of the counting algorithm. The RGB values for the input image's background (sky, ground, foliage, etc.) were selected to produce the best orange color determination. Binary images have pixels with just two possible intensity values, which are typically shown as black and white. In terms of numbers, the two values are usually 0 for black and 1 or 255 for white. Thresholding a grayscale or color image to isolate and object from the background is a common way to create binary images. Among the various segmentation techniques for objects with ambiguous boundaries, watershed transform is one of the most effective (Tabb et al., 2006). This method provides an effective morphological tool for segmenting textured photos into areas of interest.

An algorithm to detect and numerate citrus fruits was developed by Dori et al., 2017. Under natural lighting circumstances, the prediction of citrus fruits acquired from orchards was computed and compared to human vision-based counting. The simulation results revealed that the new counting algorithm was both appropriate and effective. The R2 correlation coefficient between the citrus counting algorithm and human counting was 0.93. For 21 sample trees, the findings of the citrus fruit recognition and counting algorithm were compared to yield estimations generated for 10-25 percent of the result of the flower counting. The yield estimation before harvesting the crops and the yield estimation obtained using the flower counting algorithm for 25% were highly correlated.

The design of the tomato-harvesting robot heavily relies on a classification system based on maturity levels. Because they are time-consuming and inaccurate, traditional knowledge-based systems are unable to meet the current production management requirements of precision picking. In this study of Zhang et al 2018, they propose an improved deep learning-based classification method for tomato ripeness that improves accuracy and scalability with a small amount of training data. This study looked at the relationship between various dataset augmentation methods and final classification task prediction results.

By training and validating the model on various augmented datasets, we implemented classification systems based on convolutional neural networks (CNN) and attempted to select the best dataset augmentation technique. The experimental results revealed an average accuracy of 91.9 percent with a prediction time of less than 0.01 second. When compared to existing methods, their solution outperformed them in terms of accuracy and time consumption.

The use of advanced computational methods has helped in the accuracy of fruit location in color images, which can be used in agricultural tasks such as automated harvesting, spraying, and counting. The system is mainly developed for robotic fruit harvesting, but can be easily customized for other applications, i.e., Fruit grading, sorting and counting, plant leaf disease detection, variety identification, quality characterization, crops yield prediction, color chart construction, pests and diseases, pesticide reduction, field inspection, yield monitoring, fruit stems characterization and precise localization. Automatic identification and characterization of different types of agricultural products allows for fruit grading and sorting systems, fruit and plant leaf disease detection and stem location systems for robotic or automated harvesting purposes. These are some examples of automated systems that are being developed to help farmers on their daily work. The interpretation of a digital color image of a fruit orchard captured in a field environment is very challenging due to the adverse weather conditions, luminance variability, and the presence of dust, insects, and other unavoidable image noise. This survey is designed to categorize and review the literature on computer analysis of fruit images in agricultural operations. The literature contains more than 60 papers published in the last 10 years. This paper is focused on advanced image processing and analysis techniques used in agricultural applications such as detection and classification of fruits. These techniques have been developed in the last decade, and this paper intends to focus on them specifically. Some evaluation metrics that were achieved in various experiments when using the reviewed techniques are emphasized so that the researchers can make choices and develop new computer vision applications in fruit images.

The study believes that this paper by Preira et. al (2017), reviews the state-of-the-art in computer vision for automatically detecting and classifying fruits in RGB color images, almost all obtained using CCD color cameras saved on a common JPEG file format. They have evaluated a variety of image processing techniques for use on color images for precision agriculture applications, specifically for grading, sorting, and counting fruits, plant leaves, fruit detection, and variety classification. A summary of some of the most commonly used imaging techniques was presented, along with some of the results and evaluation measures that have been used to assess their performance. This will help researchers choose the appropriate techniques for their applications.

Fruit classifications are important research according to the study of Latha et al (2021). In their study they used to classify various fruits such as Apple, Cherry, Blue Berry, Guava etc. The study obtained 97.4% of accuracy with fast 19.5 ms classification. The researchers used CNN layer outputs along with fully connected layers for creating a model to classify fruits. Another study related to this is the study of Mai et al (2020). In the study they used Faster R-CNN in various fruits as the algorithm for the fruit detection. They utilize the features of the R-CNN wherein three different levels for three different classifiers for the fruit classifications.

In the study of Liu et al (2019), they used an improved Mask RCNN to detect cucumber fruits using pixel level. Cucumber fruits are long and large which is quite difficult by traditional object detection. It is also important to know that the leaves of cucumber and cucumber almost look alike. The researchers used the Logical Green Operator to filter out non green colors in the background to limit the range of anchor boxes. The study of Feng-Xu et al (2020), they use the YOLOv3-tiny as an algorithm for tomato detection. The experimental results are 91.92%. They use different time for the detection. They test in Daylight, Evening, and Night.

Table 4. Summary of Detection and Localization of Fruit Related Studies

Author	**Year**	**Title**	**Relevant Findings**	**Relationship to SAFS**
T. Dewi, R. Rusdianasari, R. D. Kusumanto, and S. Siproni,	2022	Image Processing Application on Automatic Fruit Detection for Agriculture Industry	Used a variety of image processing algorithms such as image segmentation, BloB analysis, and edge detection to detect different kinds of fruits.	Relied on image detection in order to automatically detect a desired fruit.
Y. Ge, Y. Xiong, and P. J.	2019	Instance Segmentation and Localization of Strawberries in Farm Conditions for Automatic Fruit Harvesting	Used the DCNN model and Mask R-CNN to detect and locate the strawberries. They used three classes of different ripeness of strawberries	It uses an algorithm that can detect and locate strawberries and identify the ripeness of the fruit.

			including deformed strawberries.	
Christopher McCool, Inkyu Sa, Feras Dayoub, Christopher Lehnert, Tristan Perez, and Ben Upcroft	2016	Visual detection of occluded crop: For automated harvesting	Proposes a crop detection system that is dependent on precise crop segmentation at the pixel level	A crop detection algorithm was used to identify the crop even with the presence of occlusions
Y. Y. Yu, K. Zhang, L. Yang, and D. Zhang	2019	Fruit Detection for Strawberry Harvesting Robot in Non-Structural Environment Based on Mask-RCNN	In this study they were able to detect the ripe and raw strawberries using the RCNN algorithm.	Detecting both ripe strawberries and unripe.
Steven Puttemans, Yasmin Vanbrabant, Laurent Tits and Toon Goedemé	2017	Automated visual fruit detection for harvest estimation and robotic harvesting	Proposed using an object categorization framework based on boosted cascades of weak classifiers to integrate and demonstrate a fully automated semi-	Object categorization to easily classify the object and help to establish a fully automated harvester.

			supervised fruit detector	
J. Chen, Y. Lian, and Y. Li	2020	Real-Time Grain Impurity Sensing For Rice Combine Harvesters Using Image Processing And Decision-Tree Algorithm	Used imaging techniques to observe ripeness status, and approximate the number of fruit on the trees.	Image processing techniques were used to identify the fruit and its ripeness.
U. O. Dorj, M. Lee, and S. seok Yun	2017	An yield estimation in citrus orchards via fruit detection and counting using image processing	Utilized BloB analysis to detect the fruit as well as the amount of fruits.	BloB which is an image processing algorithm was used to estimate the number of fruits.
Zhang, J. Jia, G. Gui, X. Hao, W. Gao, and M. Wang	2018	Deep Learning Based Improved Classification System for Designing Tomato Harvesting Robot	Uses classification method for tomato ripeness that improves accuracy and scalability with a small amount of training data	Classification of method for detecting more accurate ripeness of the fruit.

Carlos S. Pereira, Raul Morais and Manuel J. C. S. Reis	2017	Recent Advances in Image Processing Techniques for Automated Harvesting Purposes: A Review	Automatically detecting and classifying fruits in RGB images.	Used an algorithm that can classify and detect colored images to know the ripeness of the fruit.
Latha, R. S., Sreekanth, G. R., Suganthe, R. C., Geetha, M., Swathi, N., Vaishnavi, S., and Sonasri, P.	2021	Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network	One program can detect various fruits.	
Mai, Xiaochun, Zhang, Hong, Jia, Xiao, and Meng, Max Q.H.	2020	Faster R-CNN with Classifier Fusion for Automatic Detection of Small Fruits	It uses 3 levels of classification for small fruits.	Different classifications for our detection.
Liu, Xiaoyang, Zhao, Dean Jia, Weikuan Ji, Wei Ruan, Chengzhi,	2019	Cucumber fruits detection in greenhouses based on instance segmentation	The utilization of Mask RCNN for detection of cucumber as it looks like its leaves.	The utilization of an algorithm for the peduncle of the strawberry.

and Sun, Yueping				
Xu, Zhi Feng, Jia, Rui Sheng, Liu, Yan Bo, Zhao, Chao Yue, and Sun, Hong Mei	2020	Fast method of detecting tomatoes in a complex scene for picking robots	The researchers detect in different time.	The different time can be important in detection.

2.3.2. Harvesting Robot System

Harvesting robots require reliable and robust systems to detect and harvest fruits and vegetables in unstructured environments. In this paper, Zhang et al., 2019, propose an autonomous system for harvesting most types of peduncle crops. The geometric approach is first used to determine the cutting points of the peduncle based on the fruit bounding box, for which we adapted the Mask Region-based Convolutional Neural Network model from the state-of-the-art object detector (Mask R-CNN). They developed a novel gripper that clamps and cuts crop peduncles without contacting the flesh. They conducted experiments with a robotic manipulator to assess the effectiveness of the proposed harvesting system in efficiently harvesting most crops in real-world laboratory settings. The system performance results demonstrated that the system can detect the cutting point and cut the fruit at the appropriate peduncle position. Because of irregular shapes, the majority of failures occurred during the attachment stage. This issue was avoided by reversing crop directions.

The Development and field evaluation of a strawberry harvesting robot with a cable-driven grip by Xiong et al., 2019, presents the design and testing of a robot for picking strawberries (Fragaria×ananassa) grown on table tops in polytunnels. The robot is made up of a newly designed gripper mounted on an industrial arm that is mounted on a mobile base, as well as an RGB-D camera. The novel cable-driven gripper can "swallow" a target by opening the end effector. It only requires the fruit location for picking because it is designed to target the fruit rather than the stem. Furthermore, because it is equipped with internal sensors, the gripper can detect and correct for positional errors, and it is resistant to the vision module's localization errors. The internal container used to collect berries during picking is another important feature of the gripper. Picking time is reduced significantly because the manipulator does not have to go back and forth between each berry and a separate punnet. The integration allows the robot to continuously harvest by moving the platform with a joystick. Field experiments show that the average cycle time for continuous single strawberry picking is 7.5 seconds and 10.6 seconds when all procedures are considered. Furthermore, the robot can pick isolated strawberries with near-perfect success (96.8 percent). However, in farm settings, the average picking success rate is 53.6 percent, and 59.0 percent when "success with damage" is included, based on testing on the strawberry cultivar "FAVORI." The failure cases are examined, and the majority of failures are discovered when picking strawberries in clusters, where both the detection algorithm and the gripper struggle to separate the berries.

Interest in agricultural automation has grown significantly in recent decades due to benefits such as increased productivity and reduced labor force. However, some current issues associated with unstructured environments make developing a robotic harvester difficult. This paper by Sepulveda et al., 2020 describes a dual-arm aubergine harvesting robot with two robotic arms configured anthropomorphically to optimize the dual workspace. They implemented an algorithm based on a support vector machine (SVM) classifier to detect and locate the aubergines automatically, and they designed a planning algorithm for scheduling efficient fruit harvesting that coordinates the two arms throughout the harvesting process. Finally, they propose a novel algorithm for dealing with occlusions that makes use of the dual-arm robot's coordinate work capabilities. As a result, the main contribution of this study is the development and validation of a dual-arm harvesting robot with planning and control algorithms that, depending on the location of the fruits and the configuration of the arms, enables the following: I simultaneous harvesting of two aubergines; (ii) harvesting of a single aubergine with a single arm; or (iii) collaborative behavior between the arms to solve occlusions. This cooperative operation mimics complex human harvesting motions such as pushing leaves aside with one arm while picking fruit with the other. The proposed harvester's performance is evaluated in laboratory tests that simulate the most common real-world scenarios. It shows that the robotic harvester has a 91.67 percent success rate and an average cycle time of 26 seconds per fruit.

According to H. Wang et al., if the workspace were to be moved to cover the fruit space, it appears from plots of several arm configurations that there would be more apples in the workplace of the robotic arms. For the 5' DOF setup, that would be extremely crucial. The arm's initial position needs to be changed in order to do that. The arm should be placed in the best possible way to get the most overlap between its workspace and the fruit area, increasing the number of apples that can be reached. A new algorithm is created to locate this ideal beginning point.

Think of the intended workspace as the area of the arm that researchers want to overlap with the fruit space. This area would be partitioned into 3D grids with a mesh size of approximately half an apple. Consider the fruit space in which they can locate a point that is the median position of all identified apples at each gator halt as well as the total number of detected apples at each gator stop. It is possible to match each of the grid points (x, y, and z) in the desired workspace to this fruit space's median point (x', y', and z').

There would be one point among these matches that not only maximizes the quantity of apples in the arm's workspace but also reduces the amount of translation the arm must do to reach its ideal starting position. The calibration numbers for all the identified apples at that particular stop are calculated using this point.

Automation and labor savings in agriculture have recently become necessary. Saving labor and increasing agricultural scale are required to solve these problems. Agriculture automation has advanced in recent years for labor savings and large-scale agriculture. However, much of the work in the fruit harvesting industry is done by hand. A viable solution to these issues is the development of an automated fruit harvesting robot.

They performed automatic fruit harvesting using a robot manipulator with a harvesting hand that does not harm the fruit or the tree in this study of Onishi et al 2019. This study suggests using a robot arm to detect fruits and automate harvesting. A Single Shot MultiBox Detector is used in this method to detect the position of fruit, and a stereo camera is used to detect the three-dimensional position. The robot arm is moved to the target fruit's position after calculating the angles of the joints at the detected position using inverse kinematics. The fruit is then harvested by the robot by twisting the hand axis. They demonstrated that the fruit position of 90% or more can be detected in 2 seconds using the SSD. The proposed fruit harvesting algorithm also demonstrated that one fruit can be harvested in about 16 seconds. The fruit harvesting algorithm suggested here is anticipated to work even if the fruit is a close relative of an apple.

A necessary part of any robotic fruit picking system is the tool or end effector. The end point of this robot is to touch and interacts with the crop to provide a desired outcome. When it comes to the design of this tool it is critical to reliable handling and detachment of the crop. This article evaluates intelligent automated harvesting system for sweet pepper, tomato, apple and kiwifruit, to show the recent advances in intelligent robot technology in horticulture. According to this study by Hua Yanbin et. al 2019, in the field evaluations, this sweet pepper harvesting system resulted in a 58% harvesting success rate, 81% grasping rate, and 90% detachment rate. Such high success rates represent a significant improvement over the previous state-of- the-art technologies, demonstrating an encouraging progress towards the possibility of developing a commercially viable autonomous sweet pepper harvester. Different procedures on how to harvest fruit using robots are tackled in this study.

To sum up, four intelligent automated fruit harvesting robots that are designed to harvest sweet pepper, tomato, apple and kiwifruit. Each harvesting robot has its own capabilities for completing its operation. The sweet pepper harvesting robot has a novel end-effector that facilitates an effective vision system for fruit detection, 3-dimensional localization and grasp selection, resulting in a great success rate of grasping, detachment and harvesting. The main feature of the tomato harvesting robot is to use a Kinect sensor, particularly use 3-dimensional position detection and color discrimination together with a fruit pattern template matching, to capture the RGB and IR images, resulting in detecting tomatoes more efficiently. To reduce the workforce and increase of production costs, more research efforts are expected in development for more advanced automatic harvesting robots. Innovative tools will help increase productivity and sustainability in modern agriculture.

Digital farming might be the answer to current challenges in the agriculture industry by employing a robot to provide continuous information about its deployed area and to give right analysis in many aspects of farming. This paper discusses the pilot project of using robots as harvesting robots The fruit to be picked is red and green tomatoes. The purpose of this study is to establish the feasibility of creating a series of robots to be used in agricultural production. This study is unique because the method is easy to use and image processing is kept simple to accommodate processors with limited computing resources.

The robot was designed specifically to harvest tomatoes from a specific size of tree and tomato plant. The tomato location is indicated by a circle that was resulted from image processing and is divided into left, right, and middle positions in the image plane. The image resolution in this study is 320 × 180 pixels. The output angle of the servo motor on the based is 115°, the servo motor joint1 angle is 90°, and the servo motor joint2 angle is 100°. When the robot's arm moves to the left it collects tomato. The average time for harvesting red tomatoes is 4.932 seconds, and green tomatoes are 5.276 seconds. The robot needs 9.676 seconds to detect red tomatoes and return to standby, and 10.586 seconds for harvesting green tomatoes. This time difference is due to the distance between the robot and the tomato, and not the color of the tomato.

The study by Oktarina et. al (2020), found that the robot successfully harvested the tomatoes in the given setting, and this demonstrates the effectiveness of this design as an agriculture robot. This paper presents a pilot project of harvesting tomatoes with a robot. The robot was designed to fit the size of the tomato tree and the tomatoes. The tomato location is indicated by a circle that was results from image processing and divided into left, right, and middle positions in the image plane

Agriculture offers a unique opportunity for the development of robotic systems; robots must be developed that can operate in harsh conditions as well as highly uncertain and unknown environments. Manipulation for autonomous robotic harvesting is one such challenge. This paper of Birrell et al, describes recent and ongoing efforts to automate iceberg lettuce harvesting. Iceberg, unlike many other fruits and vegetables, is difficult to harvest because the crop is easily damaged by handling and is difficult to detect visually. Vegebot, a platform for iterative development and field testing of the solution, which includes a vision system, custom end effector, and software, has been created. To address the harvesting challenges posed by iceberg lettuce, a custom vision and learning system with two integrated convolutional neural networks has been developed. To allow for damage-free harvesting, a custom end effector has been developed. A control method based on force feedback allows detection of the ground, allowing this end effector to achieve repeatable and consistent harvesting. The system has been field-tested, with experimental evidence demonstrating the success of the vision system in localizing and classifying lettuce, as well as the full integrated system in harvesting lettuce. This study demonstrates how existing state-of-the-art vision approaches can be applied to agricultural robotics, and mechanical systems that leverage the environmental constraints imposed in such environments can be developed.

Table 5. Summary of Harvesting Robot System Related Studies

Author	**Year**	**Title**	**Relevant Findings**	**Relationship to SAFS**
Tan Zhang, Zhenhai Huang,Weijie You,Jiatao Lin, Xiaolong Tang, and Hui Huang	2019	An Autonomous Fruit and Vegetable Harvester with a Low-Cost Gripper Using a 3D Sensor	Developed a novel gripper that clamps and cuts crop peduncles without contacting the flesh.	Used a Low-Cost Gripper using a 3D Sensor
YaXiong, ChengPeng, LarsGrimstad, Pål JohanFrom, and VolkanIsler	2019	Development and Field Evaluation of a Strawberry Harvesting Robot with a Cable-Driven Gripper	Development of a cable-driven gripper that can "swallow" a target by opening the end effector. It only requires the fruit location for picking because it is designed to target the fruit.	Both used an end-effector for picking and harvesting fruits.
Delia Sepulveda, , Roemi Fernandez, Eduardo Navas, , Manuel Armada , (Member, IEEE), And Pablo Gonzales-De-Santos	2020	Robotic Aubergine Harvesting Using Dual-Arm Manipulation	Development and validation of a dual-arm harvesting robot with planning and control algorithms.	Both used a dual-arm manipulator.

HengWang, Cameron J.Hohimer, SantoshBhusal, ManojKarkee, ChangkiMo, and John H.Miller	2018	Simulation As A Tool In Designing And Evaluating A Robotic Apple Harvesting System	The researchers concluded that overlap of the arm's workspace and the fruit space is the key in deciding the reachability of the manipulator instead of the volume of arm's workspace.	Both used an end-effector for picking and harvesting fruits.
Onishi, T. Yoshida, H. Kurita, T. Fukao, H. Arihara, and A. Iwai	2019	An automated fruit harvesting robot by using deep learning	Development of A highly fast and accurate method with a Single Shot MultiBox Detector is used herein to detect the position of fruit, and a stereo camera is used to detect the three-dimensional position.	Both used fruit position detection

Yanbin Hua, Nairu Zhang, Xin Yuan , Lichun Quan, Jiangang Yang, Ken Nagasaka, and Xin-Gen Zhou	2019	Recent Advances in Intelligent Automated Fruit Harvesting Robots	Development of Kinect sensor, particularly use 3-dimensional position detection and color discrimination together with a fruit pattern template matching, to capture the RGB and IR images, resulting in detecting tomatoes more efficiently.	a 3D position detector is used to detect the fruits accurately.
Yurni Oktarina, Tresna Dewi, Pola Risma, and Muhammad Nawawi	2020	Tomato Harvesting Arm Robot Manipulator	Development of the end effector consists of a scissor to cut the tomato branch, a webcam as the ”eye,” and a proximity sensor to sense the distance between the end effector and tomatoes.	An end effector used to pick fruits and cut the stems. A webcam for the object detection of fruit.

2.4. End-Effector

2.4.1. Mechanism of End-Effector

Tomatoes are a globally popular vegetable due to their delicious taste, high nutritional value, and ease of preparation, and their output accounts for 10% of total vegetable output. Tomato picking is primarily done by hand, which is a time-consuming and labor-intensive process. The end-effector is the critical component that operates the fruit directly. It should be accurate and flexible due to the complexity of the working environment, irregularity, and flaccidity of the operation object.

This study by Wang et al., 2016, was conducted to improve the automatic harvesting for tomatoes. In this study, the researchers design an automatic tomato harvesting that consists of a vision unit, a jointed manipulator, a railed vehicle, a controller, and end-effector. To harvest a tomato, the vision unit will locate and identify the ripe tomato where this information will be sent to the controller. The controller will send a signal to the manipulator, end-effector, and vehicle to position the end-effector into the ripened tomato. The parameters used in this project is the physical appearance which is the diameter of the tomato horizontally and vertically. The friction coefficient of the fruit or the stalk is also used. The researchers set the maximum pulling strength as 30N. The end-effector clamps the tomato horizontally while the pressure is set to a minimum of 60N. These limits are acquired by trials. The harvester has a pressure sensor and infrared sensor in the end-effector to determine the pressure of clamping the fruit while the latter is to detect whether the fruit is already in the sleeve. The automatic harvester was tested on January 206 in a tomato cultivation greenhouse of a special vegetable garden in Beijing, China. There are 50 Jiali No.14 Tomato that was tested where 86% of it was harvested successfully. The remaining fruit picked are either damage, or unbroken stalk which was deemed as failure. As a result, the key future task for increasing the performance of the end-effector is to properly sense the holding force on fruit and manage the grasping-point on the stem.

Based on the study done by H. Zhou, X. Wang, W. Au, H. Kang, and C. Chen, there are several fruit detaching methods that rely on stem detachment from the branch by applying external force to the stems, which may be classified into five groups based on the force applied: stem cut, stem pull, fruit twist, fruit pull, and vacuum. Fruit twist and fruit pull methods apply their respective actions to the fruit while holding the stem, vacuum methods create a suction force to extract the fruit without direct contact with either the stem or the fruit, and stem cut and stem pull methods apply their respective actions to the stem while holding the stem.

The gathered statistics and analysis about fruit detaching methods, harvest rate and fruit handling methods show that fruit twisting and stem cutting are two common detaching methods, with 75 percent and 65 percent average harvest success rates, respectively. Only two of the four systems with stem pull as the detachment mechanism had a harvest success percentage of 90% or above. However, given the limited sample size, it is speculative to declare one detachment method preferable than another, especially given the scarcity of fruit damage data currently available in the literature.

In fruit handling, gripping is the ideal way for pericarp touch. There are 20 reported methods that do not have any pericarp touch throughout the harvesting process. Contactless harvesting is beneficial because it reduces fruit damage and is commonly used to pick fruits with high fragility, such as strawberries and tomatoes.

Fruits with a relatively flimsy skin or pericarp are more likely to be removed through stem cutting, but fruit twisting can be used on both fragile and moderately tough fruits like strawberries and apples.

Vacuum, pneumatic, and electric actuators are the most common in fruit harvesting grippers. Multiple actuators are also used in grippers to combine the benefits of diverse actuation systems. The majority of the 47 fruit harvesting end-effectors used a sensor-free design, relying on feedforward control through PWM signal to activate intended actions on motors or solenoid valves. The end effector must be able to apply adequate force on the fruit and adjust to diverse shapes and sizes in order to provide a high harvest success rate while minimizing potential fruit damage. To increase gripping flexibility, researchers used compliant mechanisms, soft materials, and sensor-based force control.

This research was presented by A. Gunderman et al., describes a unique tendon-driven gripper that can be used to gather fragile, plant-ripened berries and ADL items. A maximum payload of roughly 2 kg could be handled by this gripper (19 N). A flexible resistive force sensor and an operational amplifier were used to implement force feedback.

Therefore, each finger was capable of applying the necessary force with an average inaccuracy of 0.046 N, ranging from 0.49 N to 1.47 N. Field harvesting experiments show that this gripper can perform just as well as manual harvesting, particularly at lower force feedback levels in relation to RDR, even though RDR remained very low for all force feedback levels. A harvesting pace of less than nine seconds per berry was maintained at all force feedback settings.

In order to lessen the berry damage caused by the plastic cylinder force sensor supports attached to the gripper's fingers, further study will involve mapping joint space force to task space force, which involves detecting fingertip force indirectly by measuring the tension in the tendons. In order to create a more reliable device, it will also be examined how the gripper reacts to outside forces like canes and leaves. The gripper will be mounted to a specially designed mobile robot chassis that combines simultaneous localization and mapping with innovative image processing to autonomously gather blackberries.

According to Kultongkham et al., 2021, it's important to avoid applying too much force while employing robots to pick fruit and vegetables since doing so might harm the crop. Their study presents a three-fingered soft robotic gripper. The finite element approach was used for its design and analysis.

Silicone rubber was used to create each finger. The fingers exert contact force when gripping, also they are fashioned in such a way that it can handle spherical things, like tomatoes or oranges. The fingers are pneumatically operated, and a micro controller also regulates the force used. Between 0 and 95 kPa of pressure exists inside the finger's air chamber. Each finger has a force sensor affixed to the tip of it to provide force feedback. The holding force is then modified and applied to the tomato's surface. With a force smaller than the 2.57 N bio-yield of the tomatoes, the gripper can effectively grasp tomatoes.

Table 6. Summary of Mechanism of End-Effector Related Studies

Author	Year	Title	Relevant Findings	Relationship to SAFS
G. Wang, Y. Yu, and Q. Feng,	2016	Design of End-effector for Tomato Robotic Harvesting	The researchers studied the maximum force the tomato can handle as well as the pulling strength.	This can help the researchers base on how their harvesting technique should be and consider the parameters from this study.
H. Zhou, X. Wang, W. Au, H. Kang, and C. Chen	2022	Intelligent Robots for Fruit Harvesting : Recent Developments and Future Challenges Background	There are several fruit detaching methods that rely on stem detachment from the branch by applying external force to the stems, which may be classified into five groups based on the force applied: stem cut, stem pull, fruit twist, fruit pull, and vacuum.	The researchers were able to narrow down the detaching method to be used which is fruit twist or by rotating the crop.
Anthony L. Gunderman , Jeremy Collins, Andrea Myer and Renee Threlfall, Yue Chen	2021	Tendon-Driven Soft Robotic Gripper for Berry Harvesting	Silicon rubber was used to prevent damage to the fruits.	The fingers are controlled to have enough force in picking the fruits and silicon rubber is put in the end effector.

A. Kultongkham, S. Kumnon, T. Thintawornkul, and T. Chanthsopeephan	2021	The design of a force feedback soft gripper for tomato harvesting	Silicon grippier is used to be a human-like finger for the end-effector	The design for the end-effector finger is much alike like this for the fragility of the strawberry.

2.5. Object Detection

2.5.1. YOLOv4 Algorithm

YOLOv4 is used in real-time object detection, with training occurring on a single CPU. The main objective of YOLOv4 is to make a super-fast object detection with high quality in terms of accuracy. This version is introduced in April 2020 by Bochkovskiy et al., 2020. The algorithm has a significant upgrade in terms of speed and performance compared to its previous and other object detection. One of the 'Deep learning-based approaches' to object detection is YOLO. YOLO is a well-known regression-based algorithm. Because the YOLO detector is very fast, it is used in self-driving cars and other applications that require real-time object detection. YOLOv4 predicts the class of the object, its bounding box, and the probability of the class of the object. The following are the parameters in the bounding box: width of the box, height of the box, class of object, and center position of the bounding box in the imager. To increase the size of their training set, the authors of YOLOv4 use data augmentation.

The value of agricultural output is rapidly changing as a result of "digital" agriculture. Robotic picking of ripe agricultural products allows for precise and quick picking, making agricultural harvesting intelligent. The question of how to increase product output has also emerged as a challenge for digital agriculture. Realizing the rapid and accurate detection of cherry fruits during the cherry growth process is critical to the development of cherry fruits in digital agriculture. Because of inaccurate cherry fruit detection, environmental issues such as shading have become the most difficult challenge for cherry fruit detection.

To detect cherry fruits, in this paper that Gai et al., 2021, authored, they proposed an improved YOLO-V4 deep learning algorithm. It is proposed to expand the network using the YOLO-V4 backbone network CSPDarknet53 network in conjunction with DenseNet. The density between layers, which is the a priori box in the YOLO-V4 model, is replaced with a circular marker box that corresponds to the shape of the cherry fruit. The feature extraction is improved, the network structure is deepened, and the detection speed is increased using the improved YOLO-V4 model.

To validate the efficacy of this method, various deep learning algorithms from YOLO-V3, YOLO-V3-dense, and YOLO-V4 are compared. According to the results, the mAP (average accuracy) value obtained in this paper by using the improved YOLO-V4 model (YOLO-V4-dense) network is 0.15 higher than that of yolov4. In actual orchard applications, cherries with different ripeness in the same area can be detected, and fruits with larger ripeness differences can be artificially intervened, resulting in an increase in cherry fruit yield.

A study by Z. Jiang, L. Zhao, L. I. Shuaiyang, and J. I. A. Yanfei mentioned that deep learning-based object detection techniques typically fall into two categories: regression-based one-stage methods and two-stage methods based on region proposals. The region-based convolution neural network (R-CNN) approach is one of the usual two-stage methods. Fast R-CNN, Faster R-CNN, Region-based Fully Convolutional Networks (R-FCN), Light Head R-CNN, and other improved convolution neural network methods are some examples. Although the one-stage method has a faster detection speed than the two-stage method, the two-stage method has a higher accuracy. The one-stage approach is better suited for use in particular circumstances where a higher real-time requirement exists. The first regression-based one stage method is the You Only Look Once (YOLO) method put forth by Redmon et al. The You Only Look Once version 2 (YOLOv2), which is based on YOLO, was also proposed by Redmon et al. by removing the completely linked layer and the final pooling layer, employing anchor boxes to anticipate bounding boxes, and creating a new fundamental network called DarkNet-19. The final iteration of Redmon, et al., You Only Look Once approach is known as YOLOv3. To increase the detection accuracy and the capability of recognizing tiny objects, it introduces the feature pyramid network, a more robust basic network called darknet-53, and binary cross-entropy loss. Due to the information fusion method used by YOLOv3, low-level information is not fully utilized, a flaw that limits its ability to be used in industry. In order to increase the performance of small object identification, Peng et al. have presented the YOLO-Inception approach, which makes use of the inception structure with a variety of receptive fields. In order to increase the speed of object detection, Yolov4-tiny approach was developed. Using a 1080Ti GPU, Yolov4-tiny can detect objects at a rate of 371 frames per second with an accuracy that can handle the demands of practical applications. It significantly improves the viability of implementing object detection methods on embedded systems or mobile devices. The suggested method outperforms YOLOv3-tiny and YOLOv4-tiny in terms of object identification speed and almost matches YOLOv4-tiny in terms of average precision.

Table 7. Summary of YOLOv4 Algorithm Related Studies

Author	Year	Title	Relevant Findings	Relationship to SAFS
A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao	2020	YOLOv4: Optimal Speed and Accuracy of Object Detection	YOLOv4 is faster than its previous versions which is faster yet more accurate version of YOLOv4.	YOLOv4 is the most latest image processing algorithm which will the researchers use.
R. Gai, N. Chen, and H. Yuan	2021	A detection algorithm for cherry fruits based on the improved YOLO-v4 mode	Improved YOLOv4 in order to detect cherry fruits. The improved model is called YOLOv4-densed	The algorithm improved for this study is the same base algorithm used which is YOLOv4.
Z. Jiang, L. Zhao, L. I. Shuaiyang, and J. I. A. Yanfei	2020	Real-Time Object Detection Method for	Developed an algorithm called YOLOv4-tiny with increased detection	The algorithm used for object detection is a

		Embedded Devices	accuracy and the capability to recognize tiny objects.	variation of YOLOv4.

2.5.2. Image Processing using MatLab Analytics

The primary focus of object detection, mentioned by A. Raghunandan, Mohana, P. Raghav, and H. V. R. Aradhya, is the recognition of actual items, such as people, animals, and objects that are frightening or suspenseful. The required section of an item is extracted using object detection algorithms using a variety of image processing tools. It is frequently utilized in fields like image retrieval, security, the medical industry, and defense. MATLAB 2017b simulates the various object detection techniques with an accuracy of over 95%, including skin detection, color detection, face detection, and target detection. To increase the effectiveness of the algorithms for video surveillance applications, parameters including detection accuracy, RGB Euclidian Threshold "T" in Target Detection, Y, Cb, and Cr in Skin Detection have been modeled and implemented. For video surveillance applications, a single algorithm may be created by considering different detecting parameters like color, face, skin tone, and target of interest.

The study of Ijemaru et al, they examined a novel image processing technique based on MATLAB analytics. It also presented an empirically based model that employs image processing applications with DCT coefficient-derived features. MATLAB was used to evaluate the system. They created color detection algorithms in images and provided updated MATLAB programming codes for cutting-edge image processing operations.

They also provided extensive simulations of several images using different algorithms running on the MATLAB toolbox to evaluate the performance of the proposed approach.

The results show that the proposed technique is a true state-of-the-art approach for digital image processing applications. This research aims to provide readers with various image processing applications running on the MATLAB platform, as well as to provide researchers with a broader understanding of MATLAB-based image processing techniques that can be used for a variety of application specifics. The various steps used for image processing can be properly documented and replicated using MATLAB. In addition, MATLAB-based image processing algorithms are more advanced than those found in other cutting-edge image processing applications.

Table 8. Summary of Image Processing Related Studies

Author	Year	Title	Relevant Findings	Relationship to SAFS
A. Raghunandan, Mohana, P. Raghav, and H. V. R. Aradhya	2018	Object Detection Algorithms for Video Surveillance Applications	Object Detection Algorithms such as face detection, skin detection, color detection, shape detection, target detection are simulated and implemented using MATLAB 2017b to detect various types of objects for video surveillance applications with improved accuracy.	Through this, the researchers were enlightened to use Matlab as the main programming language of this study.

Gerald K. Ijemaru, Augustine O. Nwajana, Emmanuel U. Oleka, Richard I. Otuka, Isibor K. Ihianle, Solomon H. Ebenuwa, and Emenike Raymond Obi	2021	Image Processing System using MATLAB-based Analytics	The Matlab Analytics is used as the image processing technique as it can replicate the image processing steps.	Matlab can support Image Processing and has many other tool in its application.

2.6. Robotic Arm

2.6.1. Robotic Arm using Matlab

In the age of 4.0 technology, robotic arms are becoming increasingly popular in modern industries. As a result, robotic arm research and simulation are very important in improving the efficiency of using this tool in all sectors. This paper of Long et al 2020, focuses on robot arm modeling for industrial polishing applications. They present a simulation of the dynamics and kinematics model of a multiple degrees of freedom robotic arm using the Matlab tool and the Robotics platform.

They examine the multi-step free controller's forward kinematics and calculate the angle of the robot arm's joints using an inverse kinematic model to derive the controller's dynamics solution. The test results demonstrate the effectiveness of the simulation method, providing a potential foundation for industrial applications such as ensuring product quality, improving productivity, and labor efficiency.

This paper investigated a multidimensional robotic arm, and the mechanical arm of this study was modeled using the standard D-H modeling method. The Robotics toolkit is then used to analyze and simulate the robot controller dynamics and dynamics in the Matlab/Simulink environment. They investigate the generation of robotic trajectories and control forces in industrial applications in the future.

It is difficult to develop robot applications without being able to see the robot move. Furthermore, establishing motion paths and adjusting controller parameters on the real robot is time-consuming if no simulation program is available. To address the issues for a low-cost robot, in this study of Lee et al 2020, they created a system that integrated kinematics and motion control simulation using MATLAB and Simulink. The system can then be connected to a real robot using TwinCAT to validate the simulation results. Case studies were conducted to demonstrate that the system worked well and could be applied to robotic arms without simulators.

This research describes a simulator that combines a virtual environment with actual control functions. MATLAB was used to create the simulation environment required for kinematics simulation. The forward and inverse kinematics equations can be used to perform rotation axis simulation, endpoint simulation, and path simulation.

The controller for the robotic arm with its physical parameters can be designed and simulated using the Simulink simulation environment built for task simulation. Using TwinCAT, the simulation results can be used to control a real robotic arm. They demonstrated that the simulation of the robotic arm's motion can be seen on the real robotic arm. The findings of this study can be applied to any similar robotic arm to overcome the challenges of using them without simulators.

The created fixed-site (non-wheeled) robot can connect its arm to an asparagus come from an entry between two edges on developed farmland without contacting non-target stems or requiring ranch conditions to be changed. Moreover, the hand at the tip of the arm handles, cuts, reaps, and tosses the stem into a unique pack intended for get-together horticultural yields. Our initially evolved robot arm has four levels of opportunity and is fueled by engines. It harvests target asparagus stems without coming into contact with different items in a farming setting, and the hand can hold the stem solidly and cut it utilizing the linkage system of a pneumatic chamber driven via gaseous tension. The current study confirmed that the robot arm system could be used to harvest asparagus automatically, and the system was endorsed by several farmers. Furthermore, we conducted experiments on actual outdoor land, harvesting three stems sequentially under the condition that the operator obtained the positional coordinates earlier.

To improve knowledge of the kinematics of any robot arm, parameter design is aimed according to the robot's requirements, and its forward and inverse kinematics are discussed. To form the kinematical equation of the resultant structure, the DR convention Method is used. Furthermore, the Robotics equations are modeled in MATLAB to generate a 3D visual simulation of the robot arm to demonstrate the outcome of the trajectory planning algorithms. The simulation detected each joint of the robot arm's movement and tested the parameters, completing the predetermined goal of drawing a sinusoidal waveform on a writing board.

Table 9. Summary of Robotic Arm Related Studies

Author	**Year**	**Title**	**Relevant Findings**	**Relationship to SAFS**
D. T. Long, T. Van Binh, R. Van Hoa, L. Van Anh, and N. Van Toani	2020	Robotic Arm Simulation by using Matlab and Robotics Toolbox for Industry Application	Using the Matlab tool and the Robotics platform, they demonstrate a simulation of the dynamics and kinematics model of a robotic arm with multiple degrees of freedom.	Algorithms and trajectory planning are important in motion control of the robotic arm's axes.
W. C. Lee and S. A. Kuo	2020	Simulation and Control of a Robotic Arm Using MATLAB, Simulink and TwinCAT	This research created a system that integrates kinematics and motion control simulation using Matlab and Simulink.	Algorithms and trajectory planning are important in motion control of the robotic arm's axes.

Yuki Funami Shinji Kawakura Kotaro Tadano	2020	Development of a Robotic Arm for Automated Harvesting of Asparagus	The research proponents developed an arm robot for the automatic harvesting of asparagus and conducted a verification experiment on actual cultivated farmland with the developed system	The research study can harvest target asparagus stems without coming into contact with other objects and the hand using the linkage mechanism of a pneumatic cylinder driven by air pressure, can hold the stem firmly and cut it.
Alla N. Barakat Khaled A. Gouda Kenz A. Bozed	2017	Kinematics analysis and simulation of a robotic arm using MATLAB	The Robotics equations are modeled in MATLAB to create a 3D visual simulation of the robot arm to show the result of the trajectory planning algorithms.	Algorithms and trajectory planning are important in motion control of the robotic arm's axes.

2.6.2. Robotic Arm using GrblGru

In a study conducted in Indonesia for preparation for robotic skills of vocational students, the proponents have used both Arduino IDE and GrblGru to control a 5-axis robotic arm while testing the programming languages' efficiency and simplicity in terms of the students' learning capabilities. In their progress, they concluded that programming the robot using Arduino IDE proves to be more difficult than using GrblGru software. Adding the motor library in Arduino poses a problem due to probable unavailability, leading to students being frustrated. While with the GrblGru software, with its simplified programming and simulation capabilities in the GUI, allowed for easier control, learning and smoother arm robot movements due to the ability to see the movement of the axes in real time. GrblGru offers a numerical control-based approach and outperforms Arduino IDE in terms of simplicity and usability for the same robotic tasks.

Table 10. Summary of Mobile Robot Motion Control System Related Studies

Author	Year	Title	Relevant Findings	Relationship to SAFS
Marsono, Yoto, A. Suyetno, and R. Nurmalasari	2021	Design and Programming of 5 Axis Manipulator Robot with GrblGru Open Source Software on Preparing Vocational Students' Robotic Skills	The use of GrblGru enables a much more simpler approach in terms of robotic arm movement programming.	GrblGru will be used as the overall software for programming the robotic harvester's trajectory and joint movements.

CHAPTER 3

METHODOLOGY

This chapter presents the research methodologies employed in developing an automated harvesting. It covers the research design, research process flow, hardware and software development, functionality and field test, statistical analysis, and project work plan to complete the development of this study.

3.1. Research Design

The figure below shows the overall flow of action in building the automated harvester.

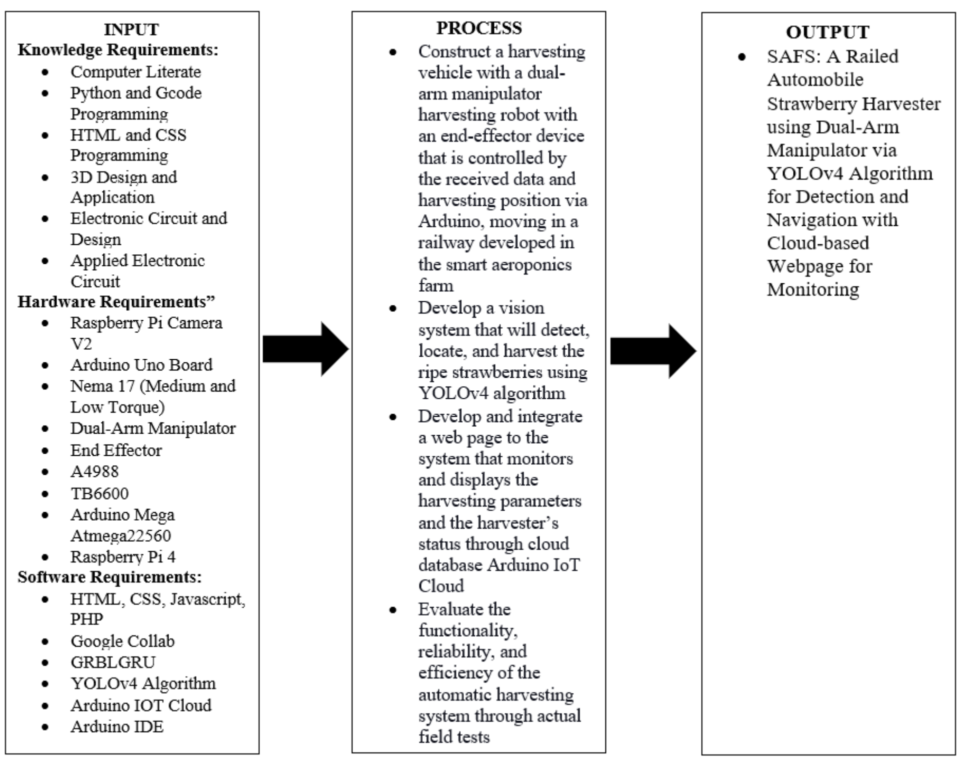


Figure 1. IPO diagram for the research study

In order to fulfill the study's objective to create an automatic harvesting machine, several preparations and steps shall be accomplished. In the IPO diagram above, first comes the Input portion which consists of knowledge requirements, the hardware and software components. The development of the project requires several knowledge in the different aspects of computer literacy, which are the following: programming, electronic circuit designing, applied circuit work, and 3D model creation, hence the listed knowledge above. For the hardware requirement, the three (3) significant components are: Dual-arm manipulator, end-effector and the Raspberry Pi 4. The Dual-arm manipulator will serve as the system's harvesting arm, responsible for moving the grasper to the fruit's location. On the other hand, the end-effector will be the grasper, which will handle the rotation of the fruit to detach it from the peduncle. And finally, the Raspberry Pi 4 will act as the main controller of the harvesting unit, allowing a fully automated harvesting process without much human intervention. Also listed above are the following and their significance to the project: Raspberry Pi Camera V2 will be used as the system's main vision unit, Arduino Uno and Arduino Mega 2560 as the system's microprocessor and controller, A4988 and TB6600 as the stepper motor driver of the robotic arm actuators and mobile platform respectively, Raspberry Pi 4 for minimizing the size of the control system of the harvesting unit, Nema 17 stepper motors as the actuator of the arm, and Nema 23 as the motor for the mobile platform. Moving on to the software part, we will be using the YOLOv4 algorithm for training the strawberry image datasets in order for the system to learn the correct likeness of a ripe strawberry fruit. For the movement and pathing of the robotic arm, a GRBLGRU and GCode will be used. HTML, CSS, and Javascript will be used for the integration of an interactive webpage, wherein the cloud database to be used is Arduino IOT Cloud Platform at the same time it is utilized for data monitoring and analysis.

Secondly, the process portion enumerates the methods which will be done to develop the harvesting machine. The most important process indicated here is to build the robotic arm portion of the system, containing the manipulator, end-effector and the vision unit. Another process to be undertaken is to design and build the railings and paths for the mobile harvester. Assembly will then be done after building the arms and the mobile platform to complete the hardware part of the project. Also, in order for the harvester to function, programming the arms' movement and pathing will be a necessary step. In terms of the fruit classification and localization, it is also needed to program the vision unit and train images using the selected algorithm. As stated above, the prototype will be connected to the web through a webpage that will enable the user to see the harvester's statuses. And finally for the last process, after the culmination of all the methods above, the harvester will undergo multiple trials regarding tests that will determine the machine's efficiency.

After considering the inputs and undergoing the said processes above, this study will be able to produce an automated harvesting machine that will be able to locate and identify ripe strawberries, and harvest them without any human intervention needed. The harvester will also be part of the IoT since it will be connected to the internet where the statuses of its parameters can be viewed.

The study will be using *developmental research* as its main research design, given that the process of making this harvesting system requires developing and designing both the hardware and software.

The goal of the study is to further the development of both the usage of image processing and mechanized automation in the agricultural setting by integrating the two concepts in an automated harvester that is suited to another on-going study on the automation of aeroponics system monitoring and horticulture.

3.2. Research Process Flow

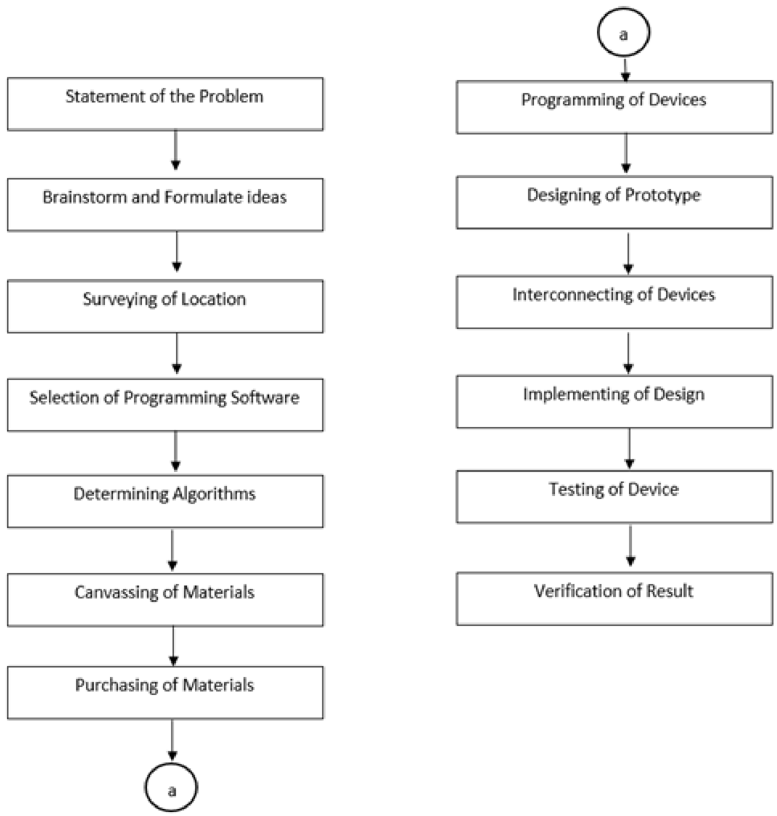


Figure 2. Research process flow of the study

The figure above depicts the flow of the research process of this study. It starts off with what problem will be tackled, from there, the researchers will formulate concepts and ideas on how to resolve the problem stated. Finding a suitable environment where the study would take place would be next followed by the determining of what algorithms would be used to achieve the objective laid out as well as the language. With the software part figured out, the researchers will now start to canvass and purchase the materials needed. Now that the materials, programming software, and algorithms are prepared and ready, the researcher will now proceed to program the devices, design the prototype, interconnect the devices, and implement the design. By then, the study now has a working device, and will now be tested. Lastly, the device's finding will now be verified, ensuring this study's result.

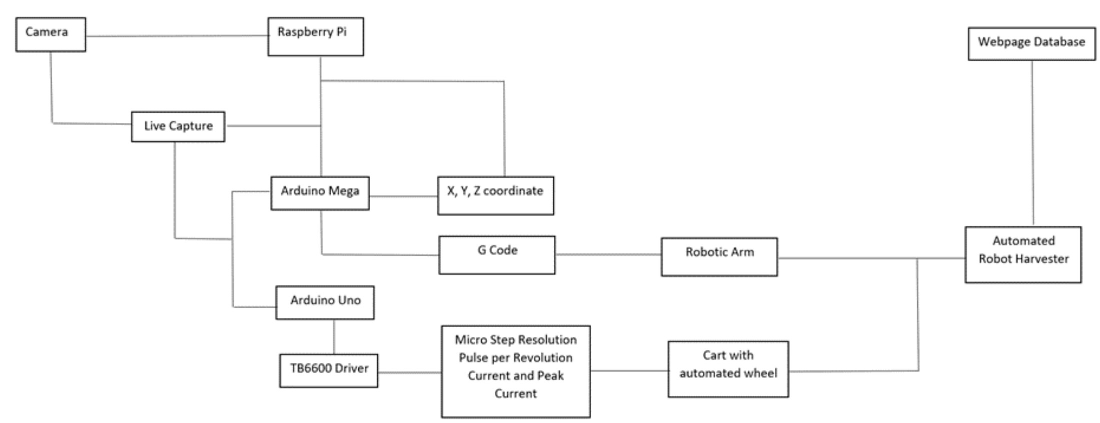


Figure 3. Block Diagram

The figure above illustrates the block diagram of this research. It starts with the Raspberry Pi uploading the algorithm to the camera, the camera's live capture will then be sent to the Arduino Mega, and Arduino Uno. Both will simultaneously carry out a process, the Mega will proceed to formulate the X, Y, Z coordinates, these coordinates will then be sent to the Raspberry Pi to process and will be sent back to the Mega wherein it could now formulate a G Code, and that will dictate the movement of the robotic arm. while the Uno will send its data to the TB6600 Driver. From there, the driver will calibrate the right parameters aforementioned above and the stepper motor within the cart with an automated wheel will carry out the said calibration and will proceed to move. With the cohesion of the robotic arm and the cart's movement, it will fully lead to an automated robot harvester which will lastly send the accumulated data to the webpage's database.

3.3. Development of the Hardware Unit

In conceiving an automated harvesting machine, one's priority should be the construction of the system's hardware unit, for everything else that follows deeply relies on the implementation of the tangible parts. For this study, the harvesting unit will consist of three main parts, namely: the railed vehicle, the harvesting arms, and live cameras. Wherein the harvesting arms will be attached on top of the railed vehicle, and the live cameras are mounted on the arms' wrists therefore be treated as a unit. With these three parts conjoined the harvesting unit is able to move from one place to another by following the path of a railing system accommodated around the cultivating area.

3.3.1. Materials and Equipment

For this study, the main materials needed to construct the automated harvester are listed below:

Table 11. Materials and its functions

Materials/Equipment	Quantity	Function
Arduino Uno Board	1 piece	Open-source microcontroller board that is simple to use and programmable that may be used in a range of electronic applications.

Arduino Mega Atmega2560	1 piece	It is a microcontroller board that executes powerful instructions in a single clock style.
Manipulator	2 pieces	Has the ability to involuntarily move or handle objects dependent on the number of degrees of freedom appointed.
Nema 17 Low Torque Stepper Motor	2 pieces	This kind of motor has a step angle of 1.8 deg and a size of 28 N-cm. This motor works in discrete steps that are powered by a stepper motor driver to shift the rotor with an accurate and established increment of one full rotation. Will be used as an Elbow actuator.
Nema 17 Medium Torque Stepper Motor	2 pieces	This kind of motor has a step angle of 1.8 deg and a size of 48 N-cm. This motor works in discrete steps that are driven by a stepper motor driver to relocate the rotor through a meticulous and permanent increment of one full cycle. Will be used as Shoulder and Base Actuator.
Nema 23 Stepper Motor	1 piece	This kind of motor has a step angle of 1.8 deg and a size of 88 N-cm. This high-torque motor operates in discrete steps controlled by a stepper motor driver, allowing the rotor to move precisely in fixed one full turn increments. Will be used as the mobile platform actuator.
A4988	5 pieces	A motor driver that is designed to operate bipolar stepper motors. This will be used for the stepper motors in the robotic arm.

TB6600	1 pieces	A stepper motor driver that could control a two-phase stepping motor. This will be used for the stepper motor at the mobile platform.
End Effector	2 pieces	The part of the robot that is attached on the end of the robot arm that also interacts with its surroundings.
Raspberry Pi 4	1 piece	A computer that runs Linux, with this the Raspberry Pi Camera was trained using Google Collab, and some specific data present in its system is integrated to a webpage using ThingSpeak
Raspberry Pi Camera V2	2 pieces	A camera module for Raspberry Pi that has 5MP resolution. It is programmed for the detection of strawberries.
ESP3266	2 pieces	A Wi-Fi module used to be a microcontroller of the two sensors integrated in the Arduino IOT website for monitoring the number of strawberries picked.
PIR Sensor	1 piece	This device is used to detect a motion connected to the ESP3266 for the monitoring of data integrated in Arduino IOT to the website.
Ultrasonic Sensor	1 piece	This device is used for detection of an object connected to the ESP3266 integrated in the Arduino IOT and website.
Rail Track Wheels	4 pieces	A wheel specifically made to operate on metal railroad rails.

Rail Tracks	5 meters	Will act as the guide for the pathway of the mobile platform.
LiFePO4	1 piece	Used as the rechargeable batteries of the system

3.3.2. Development of Railed Vehicle

The robot's mobile platform consists of its body which is carrying the hardware – it is made of strong metal to keep all the materials and equipment intact. The mobile platform's wheels are designed to be compatible with rails. The rails are fixed on a platform designed according to the height of the vertical aeroponic system. The carrier of the robot mobilizes on the rail, this transports the robot to the vertical aeroponic system repeatedly and onto the next aeroponic system

The vehicle will consist of 5 wheels, four supporting wheels and one main wheel. The main wheel will be mounted with a stepper motor, making it the main source of movement the cart receives, while the four wheels are just there to help carry some load of the cart. The driver will be connected to the arduino uno board, along with a TB6600 motor driver.

As for the power source of the automatic harvester, the most efficient and durable battery compatible for an automatic railed vehicle is a

12V Lithium Iron Phosphate (LiFePO4) battery pack. This type of battery is known for its stability and safety profile because it allows lower energy density and is naturally non-combustible. It is also designed for high-power applications because of its extended lifespan, low self-discharge rates, and being lightweight. LiFePO4 batteries are also an environmentally friendly option because the solutions used to make them are non-toxic, non-contaminating, and do not include rare earth metals.

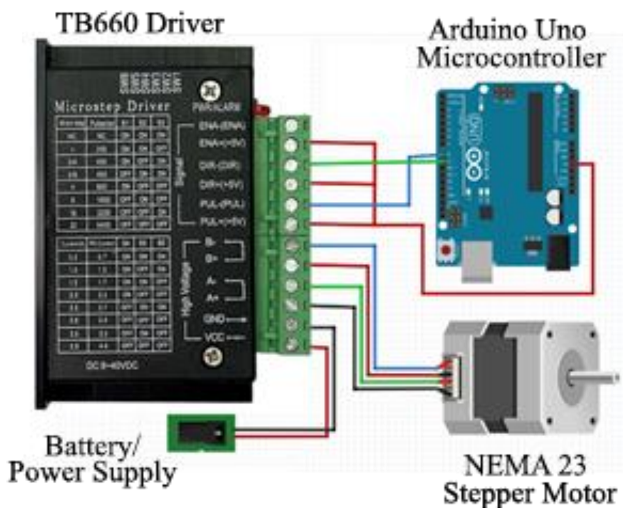


Figure 4. Schematic Diagram of Railed Vehicle

3.3.3 Design of Railed Vehicle

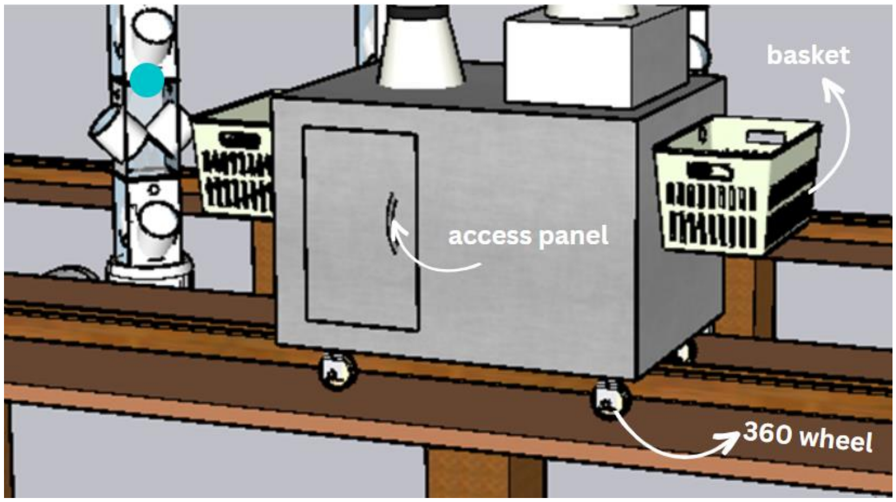


Figure 5. Railed Vehicle Design

3.3.4. Development of Harvesting Arm

3.3.4.1. Configuration of Manipulator

The Manipulator is a reprogrammable and multipurpose mechanical device that performs numerous tasks by moving materials, components, objects, or instruments through programmed motions. It can move or handle things autonomously based on the amount of degrees of freedom it is provided. This will position the end-effector during the harvesting process, serving as the “arm” of the device. The harvesting unit will consist of two arms and in order to achieve that, modification of the manipulator will be done. Two manipulators will be fixed on top of the vehicle, mimicking the human torso. To move, the manipulator will receive the position data from the Arduino Uno.

It contains a Stepper motor that revolves with severe precision. Serves as a helping hand to the manipulator. It aids the manipulator by giving it the freedom to rotate with great precision.

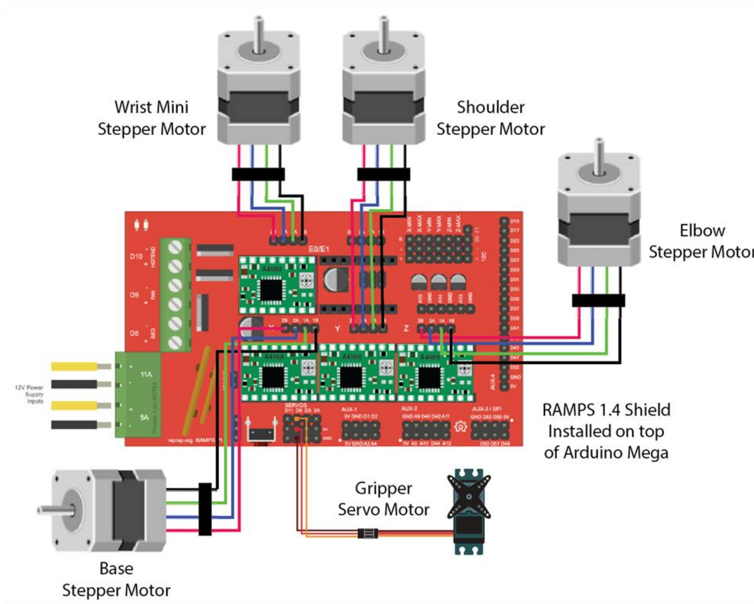


Figure 6. Schematic Diagram of the Arm

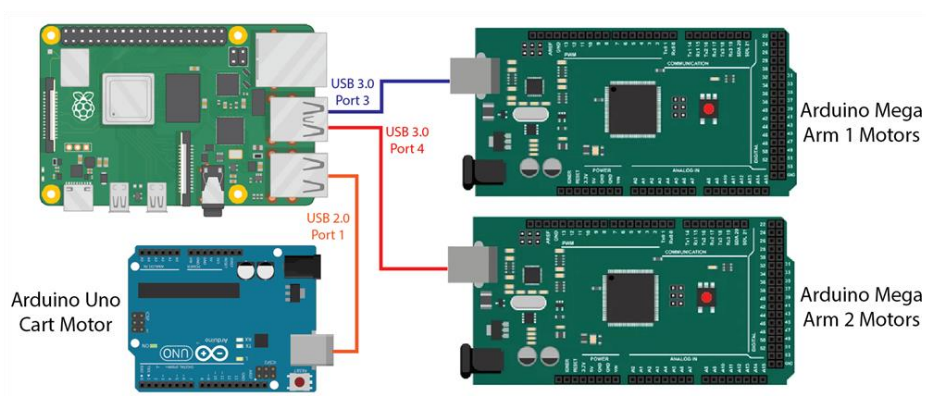


Figure 7. Schematic Diagram Control Unit connections

3.3.4.1.1 Design of Manipulator

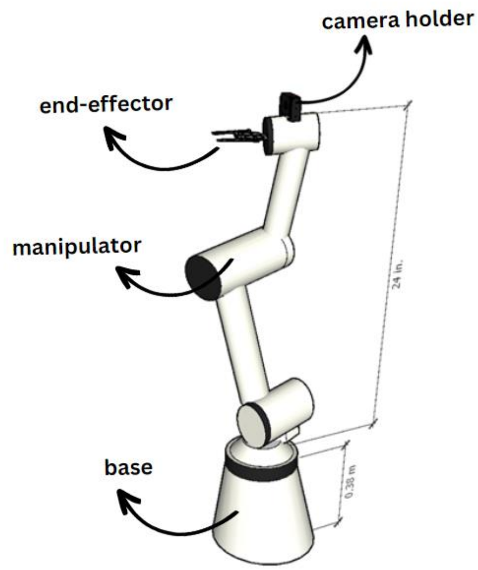


Figure 8. Manipulator Design

3.3.4.2. Configuration of End-Effector

The gripper structure used to effectively harvest the strawberry is a modified design of a two-finger gripper. It is made of polylactic acid and built through the process of 3D printing. It is modified to have a blade placed inside the slots at the top section of the fingers pads. This allows it to act as a cutter while also having the ability to grip with the help of the attached rubber lining in the lower section of the finger pads' surface.

It is established that the process of picking strawberries is delicate, it needs a cautious and meticulous course of action. To prevent harming the fruits, mechanical characteristics of strawberry stalks were assessed, including cutting force and frictional resistance.

The fragile surface of the strawberry is readily harmed by mechanical force, lowering its quality. To carefully pick the strawberries, the two-finger end-effector is designed to cut and hold the peduncle. This design will eliminate the need to directly touch the strawberry, thus preventing the unwanted bruising on the fruit.

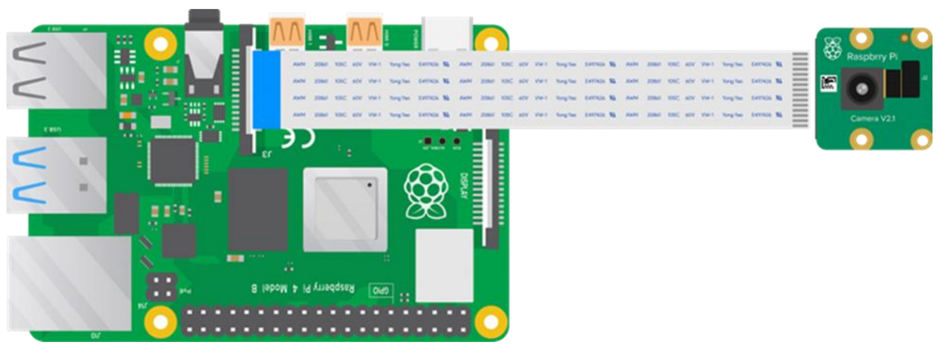


Figure 9. Schematic design of the Camera in the End-effector

3.3.4.2.1. Design of End-effector

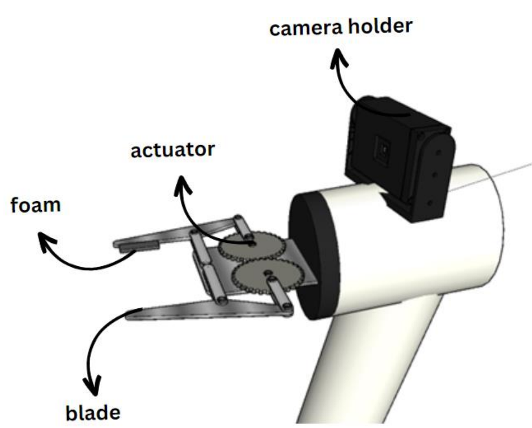


Figure 10. End-effector Design

3.3.5. Design of the Automated Harvesting Unit

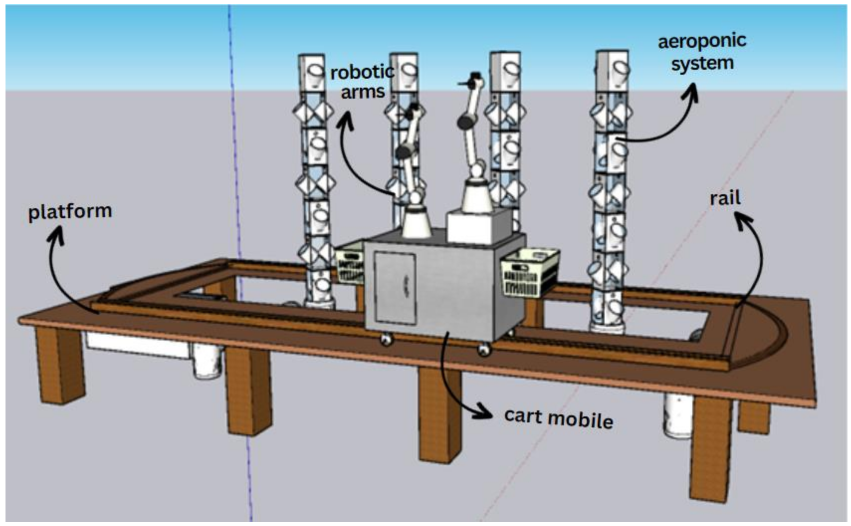


Figure 11. Automated Harvesting Unit Design

3.4. Software Development

Figure 12. Google Collab Logo

An automated harvesting robot heavily depends on its machinery and due to advanced functionalities of the said machinery, the programming software must be able to provide a multitude of utility in order to accommodate to meet each device's requirement. Arduino has one of the most open-source codes readily available on the internet, this makes it easier to find a base syntax and tweak it for the harvesting arm system. Raspberry Pi 4 functions like a computer. It offers a wide-array of functions and one of those functions align with the requirements of the harvesting robot which is the vision system. Through Google Colab, the training of the YOLOv4 algorithm is possible. This is to distinguish and spot the target object which is the strawberry. The vision system directly detects and locates the crop available for harvest, such an incredible feat would need an algorithm deemed suitable to accomplish this goal, Yolov4 (You Only Look Once) would be responsible for making it possible. The trained weights will then be integrated to the Raspberry Pi to make the vision system mobile. And lastly, a cloud database would achieve the storing of the data and monitoring of the parameters of the device such as its battery level with the help of (method of joint partner). These make up for the software of the harvester's system, without it, every programmable device is just a piece of metal.

3.4.1. Programming manipulator and end-effector movement

Figure 13. GrblGru Logo and 6 DOF STL file

Mechanical systems, actuators, electrical systems and environmental models are all part of the development of a robot manipulator. Custom algorithms for robot manipulators can be optimized using Raspberry Pi 4. GrblGru is an open source software application used to control CNC machines and other robotic devices' motions. The software uses G-code, which is the most widely used 3D printing programming language mainly for its capability to provide metric-based computer numerical control which is a much more simplified way of implementing arm movement rather than using inverse kinematics.

G-code implementation can be further compressed into the raspberry pi environment via g-code parser to be interpreted by the Arduino software. Individual joint movements can be easily programmed and recorded through GrblGru's software, which will all be compiled and used for arm movement automation that will respond to the coordinate mapping output of the vision system. As for the end-effector's programming, G-code can also be utilized as the same set of numerical control can be limited to match the gear movement restriction of a servo motor.

3.4.2. Development of the vision system for object detection using YOLOv4.

YOLOv4 being a single staged object detector works more accurately and faster than two stage detectors like RCNN. YOLOv4 comprises three main components: the backbone, neck, and head. The backbone employs CSPDarknet53, which utilizes a CSPNet strategy to divide the feature map of the base layer into two sections and subsequently merges them using a cross-stage hierarchy. The neck component of YOLOv4 utilizes PANet to improve the instance segmentation process by preserving spatial information. It also uses SPP for the pooling layer that removes the fixed size constraint of the network.

For the reference of the ripeness of a strawberry, a set of pictures must be used. The sample data of pictures does not only consist of ripe strawberries but also the unripe ones as well as the peduncle. With this it will not only identify the ripe strawberries but also the unripe ones. To train the YOLOV4 algorithm 1,500 images will be gathered and created to serve as part of the dataset, it will consist of 750 images gathered from the internet and 750 adjusted images to account for data augmentation. The initial training of the algorithm will use 700 images, the gathered images will then be divided into two sets: the training dataset and the testing dataset, a ratio of 80:20 will be used to ensure that there will be enough images for training the detection accuracy of the algorithm as well as still having the right amount of images to test the algorithm unto after the training. Making the initial training use 560 images for training and 140 images are for testing. To further develop the detection capabilities of the algorithm, another training period will be arranged, adding 700 more images. The new images will need to be more diverse in terms of the visibility of the strawberry fruits to ensure that the dataset will properly represent the real world conditioning of a live camera feed. Repeating the implementation of the 80:20 ratio with the increase of the total number of images in the dataset, the training images will then become 1120 images and 280 for the testing images. And a final training will be done using images from the actual fruits from the deployment site, which will mostly consist of augmentation to increase its count to a 100. Overall, 1200 images will be used for training and the remaining 300 are allocated for testing.

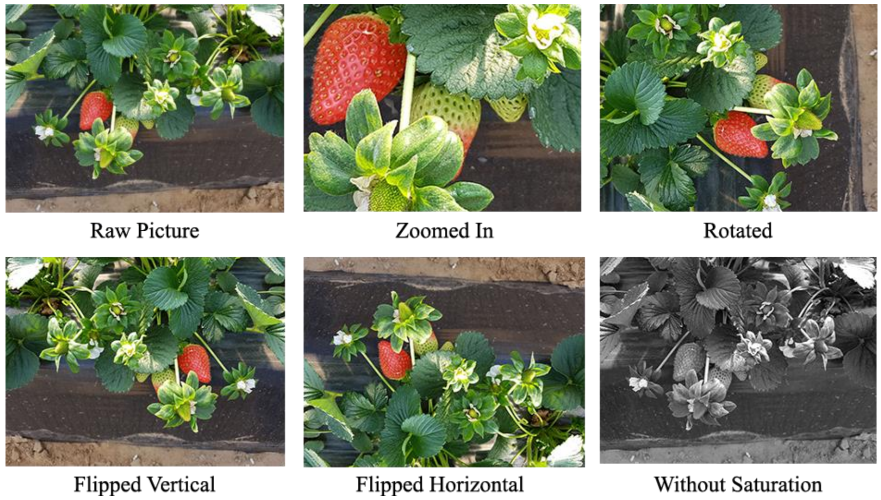


Figure 14. Data Augmentation Examples for the Strawberry Crop Dataset

The Raspberry Pi Camera V2 Module cameras that will be used in detecting ripe strawberries are placed near the end-effectors. A 15 centimeter ribbon cable is used to connect the Raspberry Pi camera to the CSI port. This captures the images of the ripe strawberries where with the help of the YOLOv4, it will determine if the strawberry is ripe and ready to harvest or not. The IP cameras are used to watch over the whole aeroponic system while also mapping the tower locations and also identifying ripe strawberry fruits, this will enable the full automation of the harvester without inefficiently letting the harvester itself circle through the tower setup to check each tower for possible ripe strawberries.

3.4.3 Program Flow of the System

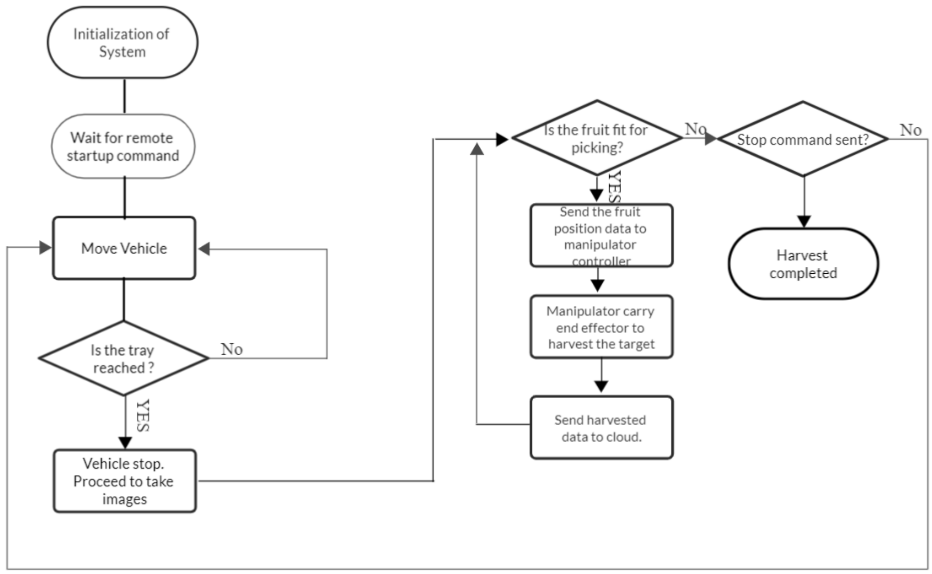


Figure 15. Program Flowchart

The figure above illustrates the task sequence of the harvesting operation. The system will start to initialize first, then will wait for the remote start-up command. The vehicle will proceed on moving and will stop when it reaches the first tower of crops, after stopping the machine will start scanning and capturing images of the crop. If the images were deemed to be available for harvest, the camera used for the detection will now send the data position of the ripe fruit to the system which will be relayed to the system's manipulator. The manipulator will now carry the end-effector to the designated target and the end-effector will proceed on harvesting the crop. Once a crop is harvested, the system will upload the harvesting data to its cloud database. If there are no more available for harvest, the machine will now move on to the next tower and repeat the process until there are no more towers to be scanned and harvested from. The system will automatically stop, indicating that the harvest is completed

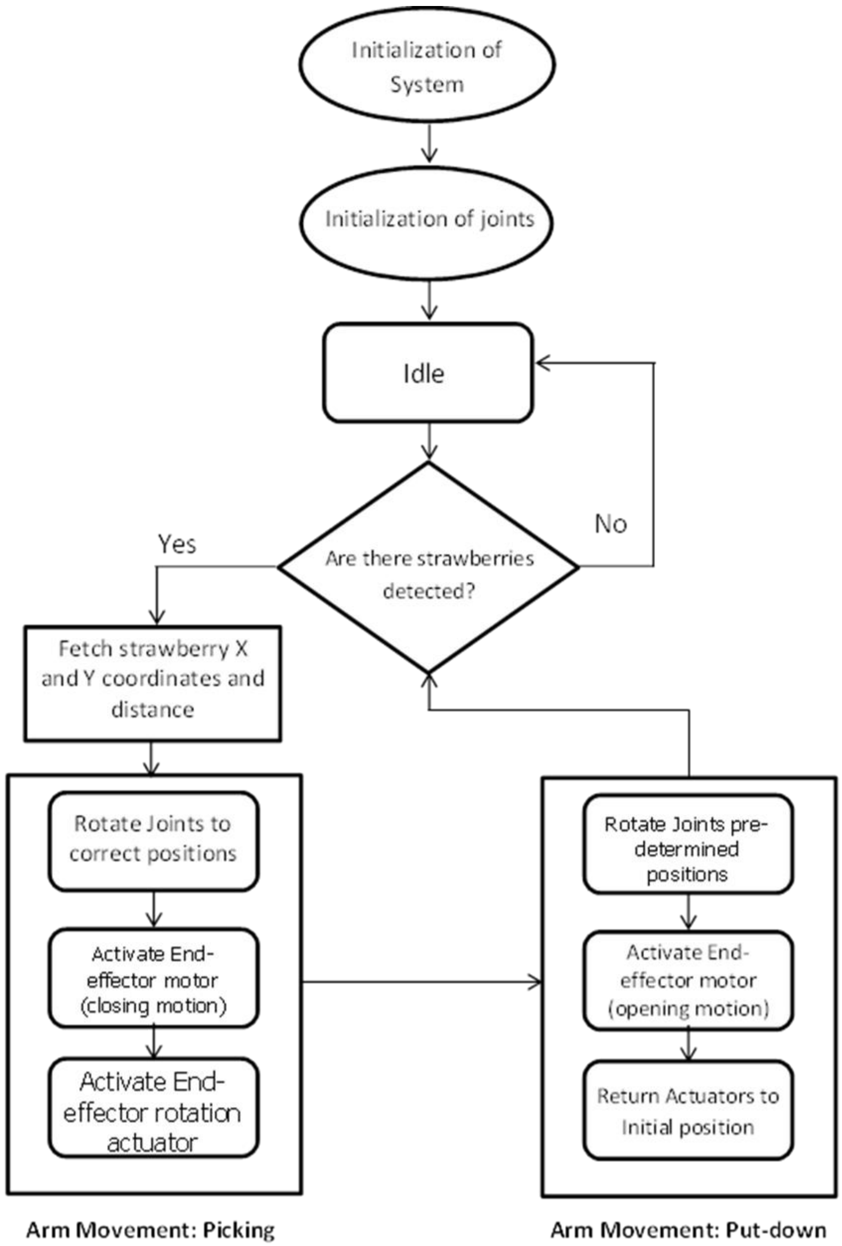


Figure 16. Arm Movement Flowchart

The figure above shows the task sequence of the robotic arm movement during harvesting. The system will start to initialize first, then will wait for the initialization of all actuators in the robotic arm. While the vision system is scanning the cultivation area, the robotic arm and mobile platform will remain idle, and once the system recognizes the presence of ripe strawberries the unit will fetch the coordinates of the bounding box for an individual strawberry in the captured frame of the camera. Provided with the coordinates of the strawberry, the actuators of the unit will then be activated to move the robotic arm according to the location of the fruit. The gripper's actuator will then activate to close its fingers in order to cut the strawberry of its peduncle. After that order of movements, the unit will then proceed to putting down the harvested fruit. The same sequence of tasks will be repeated but instead of using the coordinates of the strawberry's bounding box, a specified movement pattern can be programmed to each arm since the location of the basket remains unchanged. Finally, after putting down the harvested fruit the unit will return to its initial position and repeat the process again beginning from scanning the cultivation area which will continue up until there are no riper strawberries recognized in the captured frame.

3.5. Development of Webpage for data and status monitoring of the harvesting unit.

3.5.1 Parameters to be monitored for harvesting.

The following parameters are to be monitored for the harvesting unit of the aeroponic system:

Table 12. Parameters for monitoring

Parameters	**Units**
Battery Life	Percentage
Machine Run Time	Hour
Harvesting Status	• **On-going** (Fruit detection /Arm Redirection / Fruit Grasping / Harvesting Successful / Harvesting Failed) • **Idle**
Number of Harvested Strawberries	Pieces

3.5.2 Integration of a web page for the harvesting system's parameters and status.

Figure 17. Arduino Iot and HTML Logo

The Arduino IoT Cloud is a user and beginner friendly cloud platform wherein anyone can create IoT projects on its platform. It can be used for the Data monitoring, Variable Synchronization, Scheduler, OTA Uploads, Webhook, Dashboard Sharing, and even Amazon Alexa Support, the charts that will be input in the Arduino IoT for monitoring will be crucial in the assessment of the harvester's efficiency in terms of its fruit detection speed and harvesting rate.

The parameters monitored by the Aeroponics system's harvesting unit will have sensors that will collect the data from the harvesting unit. It is connected to the ESP8266 wherein it will send the collected data to the Arduino IoT Cloud Platform, it will show the collected data real time in the website and at the same time saves its data. A web interface will be created that is made up of several panels that show the monitored system of the aeroponics and the analyzed data parameters of the harvesting unit from the Cloud Platform. This web interface will be accessible via mobile devices for easy monitoring.

To create the webpage, the researchers will use HTML. It will be integrated with the Raspberry Pi 4. The HTML will be uploaded by a free web server and domain that will be efficient as it can be open anywhere as long as there's an internet connection.

3.6 Functionality and Field Test

To test the efficiency of the proposed automated strawberry harvester, the following parameters are to be measured: finding the vertical aeroponic system with mature fruits; vehicle moving to the targeted vertical system; locating the ripe strawberries; manipulator approaching the target; end-effector cutting and holding the fruit; manipulator putting the fruit on to the basket; and manipulator getting back to the initial posture. After evaluating the result of the total time cost the harvester spent in one harvest, bear comparison with the previous studies to test the effectiveness and reliability of the project. And to test its functionality and avoid the collision of the prototype during harvest, three field tests are to take place.

Table 13. Time Cost Composition Table

Time Cost Composition		
No.	Working Procedure	Time cost/s
Total time cost		

Table 14. Harvesting Success Rate Table

Detected Strawberries	Successful Harvesting	Unsuccessful Harvesting
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		

3.6.1. Speed of detecting, locating, and harvesting ripe strawberries

Three trials are needed to take the speed of the harvester in locating, detecting, and harvesting the ripe strawberries. By testing the harvester on how it detects, and locates strawberries, the researchers will measure the different times the harvester detects, locates, and harvest the ripe strawberries. This should be done repeatedly until the researchers obtain their desired results. The mAP test will graph the trend of loss based on the number of epochs or iterations.

Table 15. Harvesting Speed Table

Harvesting Speed				
Attempt Times	Strawberry Quantity	No. of ripe strawberries	Successful harvesting time	Reason of failure
Total attempt times				

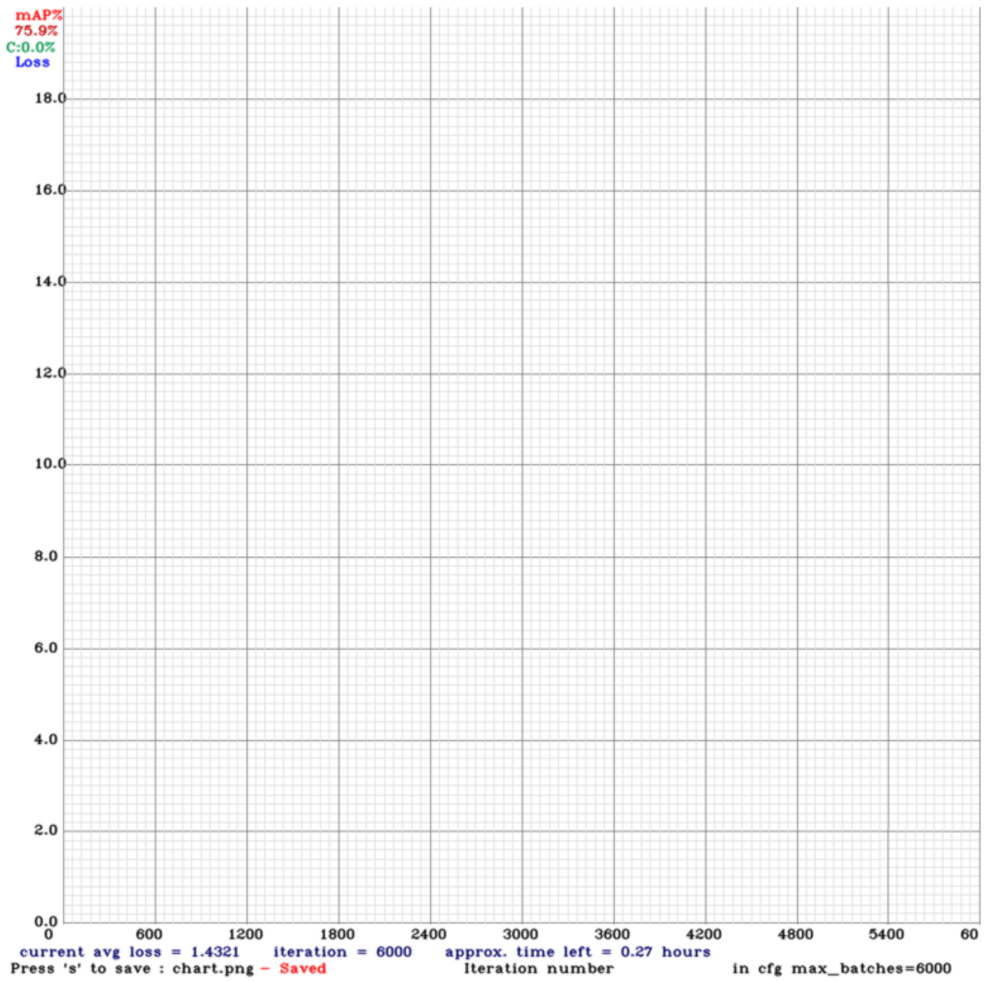


Figure 18. mAP Graph

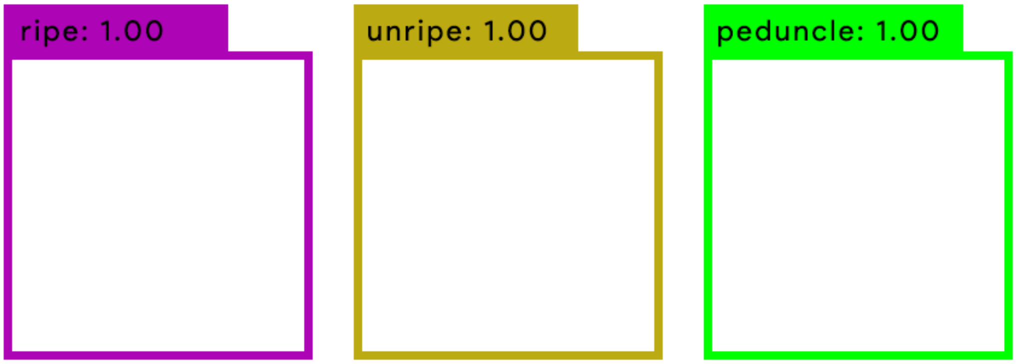


Figure 19. Format for the bounding boxes of each object class

3.6.2 Occlusion Test

In order to establish a reliable reference for the acquisition and detection of accurate fruit specimens, the researchers have devised a methodology involving the simulation and execution of tests specifically targeting cherry tomatoes and kalamansi fruit. This meticulous approach aims to thoroughly evaluate the efficacy and precision of the vision system in accurately identifying and scanning the intended fruits. By subjecting the system to these controlled experiments, the researchers can quantitatively assess its performance, thereby measuring the accuracy and reliability of the vision system's fruit recognition capabilities. The outcomes of this testing will provide valuable insights into the system's ability to discern and successfully scan the designated fruits, thereby contributing to the refinement and optimization of the overall system's accuracy and effectiveness in fruit detection.

Figure 20. Occlusion Testing

3.6.3. Hardware Failure Trials

Table 16. Hardware Mistake Attempt Table

Hardware Mistake Attempt			
No.	**Description of mistake cause**	**Times**	**Calibrations**

In this test, we aim to assess the errors encountered during the production of the hardware, as well as the frequency of their occurrence. Additionally, we will evaluate the effectiveness of the calibration procedures implemented to rectify these mistakes. Our primary focus will be to quantify the extent of these errors and analyze the impact of the calibration measures taken to address them. By meticulously examining these aspects, we aim to gain valuable insights into the overall quality and reliability of the hardware.

3.7 Statistical Analysis

This study implements a developmental research design, with this the researchers are hoping to develop the same if not a better harvesting system. To achieve precise findings, the results of this study should be compared to previous works, and use T-Test to statistically analyze the outcome. The T-test with the formula of $t = (\bar{x} - \mu) / (s/\sqrt{n})$, specifically the confidence interval, measures the degree of either the uncertainty or certainty in a sampling method. It is commonly represented as a percentage whether like 95% or 99% confidence score. The higher the confidence score is the higher the reliability or the results. T-test will help the researchers to assess the likelihood of the data being meaningful difference than just a random occurrence.

The confidence score formula in YOLOv4 for object detection combines the objectness score, which represents the likelihood of an object being present within a bounding box, with the class confidence, which represents the probability of the object belonging to a specific class. By multiplying the objectness score by the class confidence (Confidence Score = Objectness Score * Class Confidence), the confidence score provides a measure of overall confidence in the presence of an object within the bounding box, considering both the objectness and class prediction.

With the resulting data found in the functionality test, it will now be compared to previous similar works. From there, it can be determined whether this study matched the findings from other studies or even surpassed. It will also help in determining whether the study has a significant outcome or difference that other studies do not have.

3.8 Project Work Plan

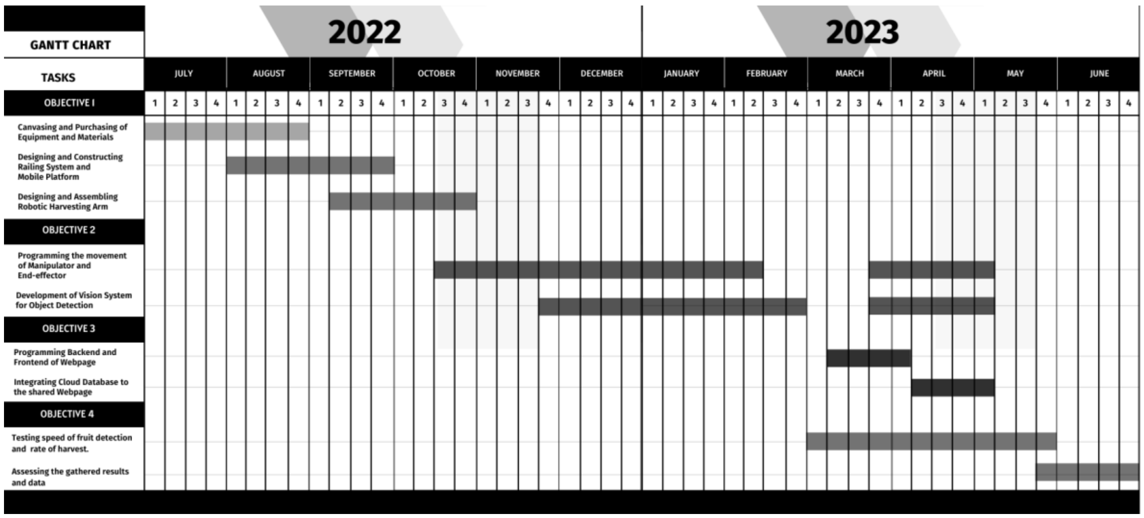


Figure 21: Gantt Chart

Upon the completion of the system, to display the system's overall development schedule, the researchers used a Gantt chart

CHAPTER 4

RESULTS AND DISCUSSION

This chapter presents the presentation, analysis and interpretation of the initial results and discussion.

4.1 Project Technical Description

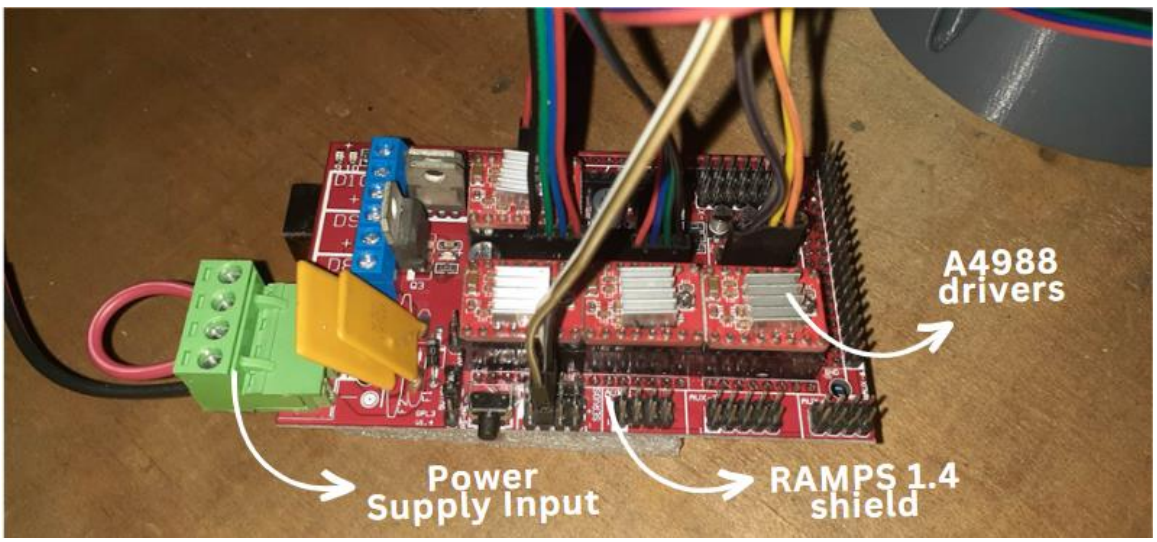


Figure 22. Robotic Arm Connection Configuration

In order to use more than two motors in one Microcontroller, it is necessary to use an extender shield for the Arduino boards. With that need, the proponents used RAMPS 1.4 to be able to use four (4) stepper motors and an additional servo motor using one microcontroller board. It also enabled the board to accommodate several A4988 motor drivers at the same time. Above is the wiring connection of the motors, motor driver placement and orientation, also labeled accordingly based on their intended joint actuator positioning and as end-effector actuator.

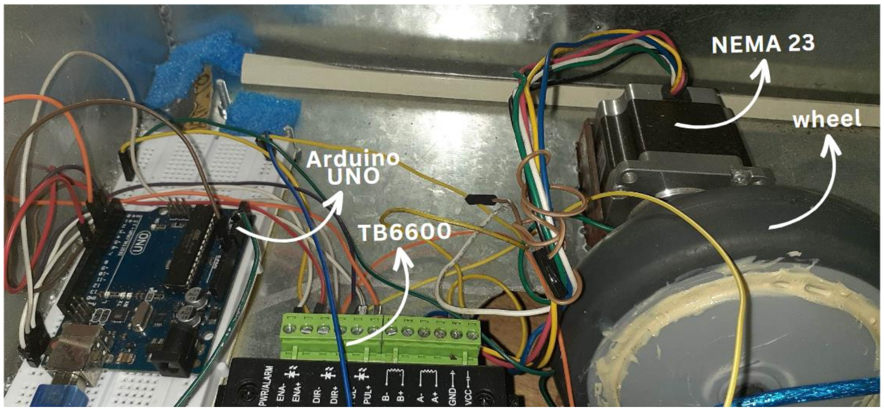


Figure 23. Mobile Cart Connections

The motor for the cart will be using a much heavy duty motor driver since NEMA 23- the chosen high torque stepper motor, need a higher rating of voltage and current supply. With that, the proponents used TB6600 motor driver, which is compatible for higher rated stepper motors such as NEMA 23. Also, a separate microcontroller is used for the cart's specific movements since both the Arduino Mega are already configured for the movement of the robotic arms. For the cart, Arduino Uno is used as the microcontroller. Below is the connection of Arduino Uno to Motor Driver, and Motor Driver to Stepper Motor.

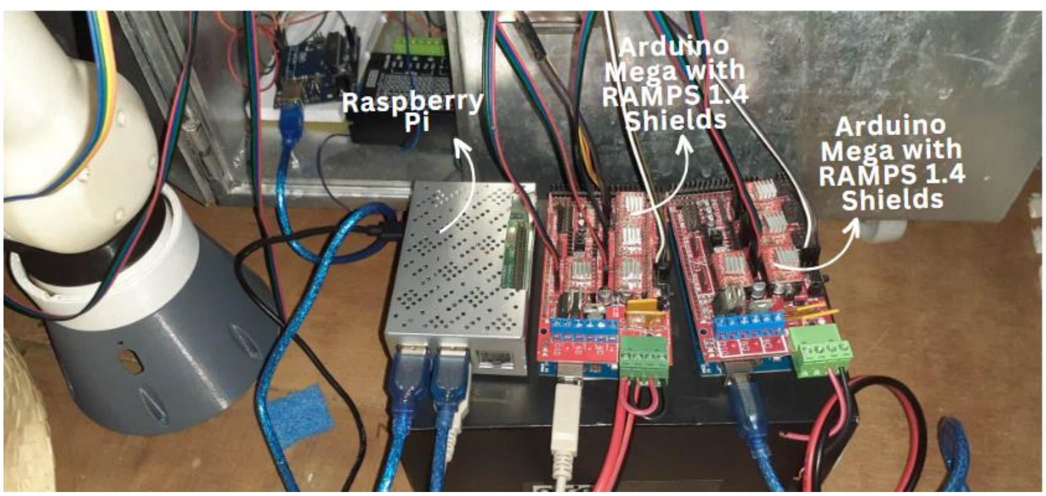


Figure 24. Integration Circuit

In order for the harvesting machine to work and move from one place to another, the use of microcontrollers is essential. Having two robotic arm units and a motorized cart requires the whole device to have more than one microcontroller. And in order to use all of them synchronously, the proponents used Raspberry Pi 4 as the Central Control Unit for three separate microcontrollers: two (2) Arduino Mega for the Arms and one (1) Arduino Uno for the cart's motor. Below is their simple connection.

4.1.1 Project Structural Description

This part of the study showcases the final design of the system's hardware which includes the robotic arm, end-effector, vehicle cart, and platform. This section also incorporates the actual structure of the harvesting system.

Figure 25. Aeroponic Farming

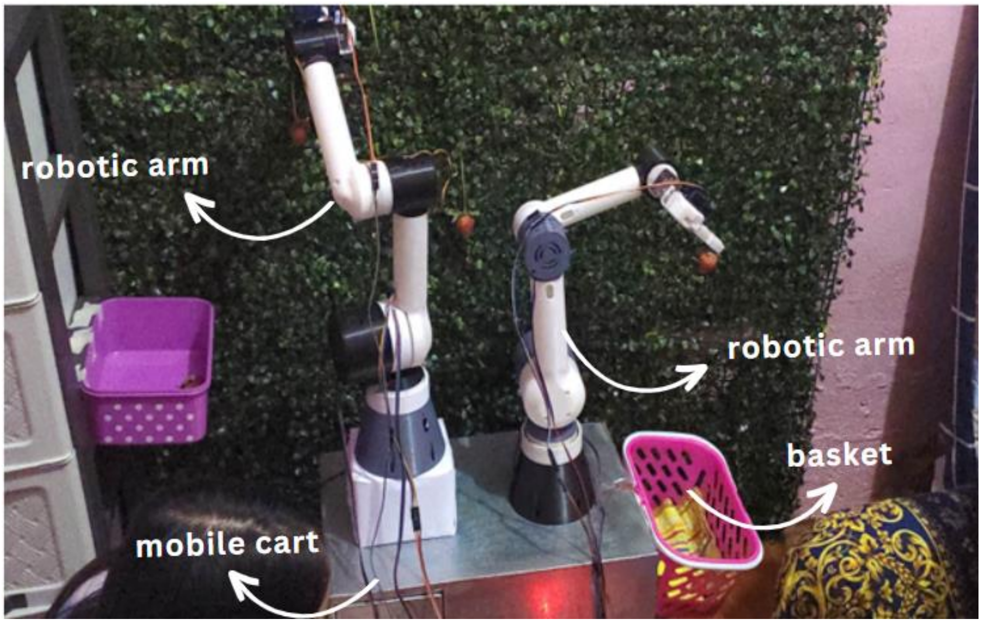


Figure 26. Actual Photo of the Cart

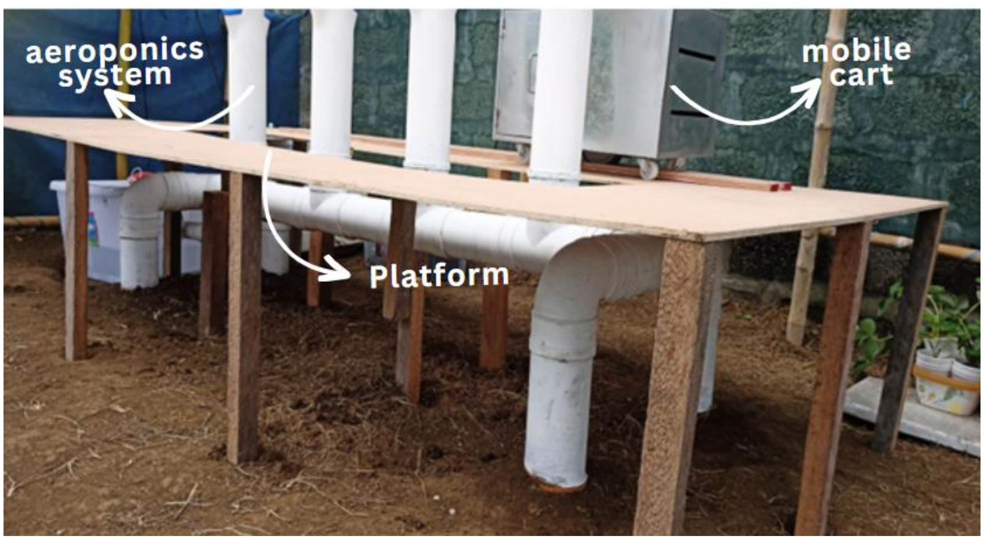


Figure 27. Actual Photo of the Platform

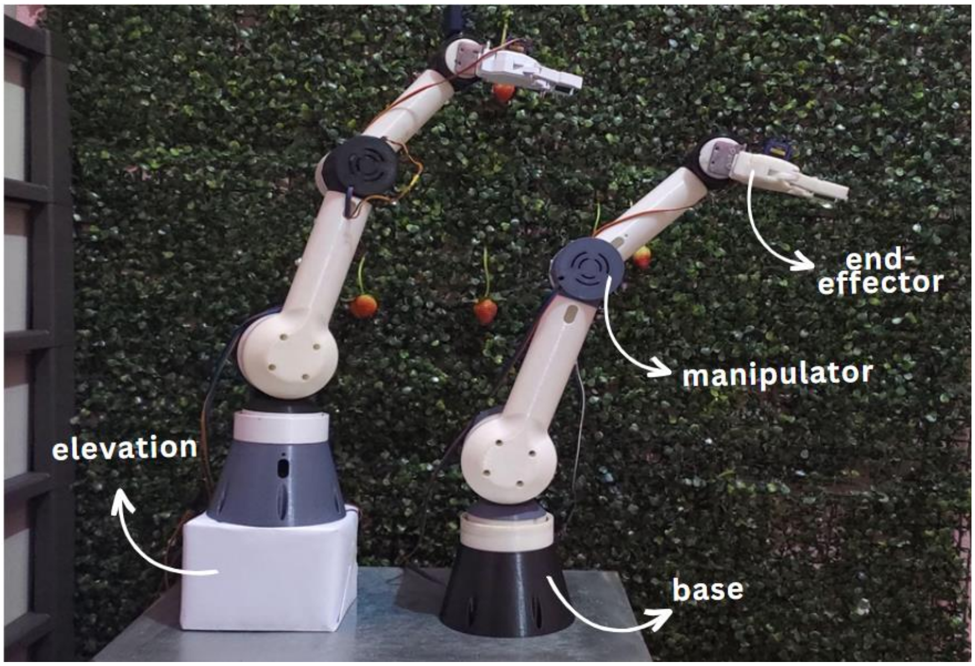


Figure 28. Actual photo of the Arm and End-Effector

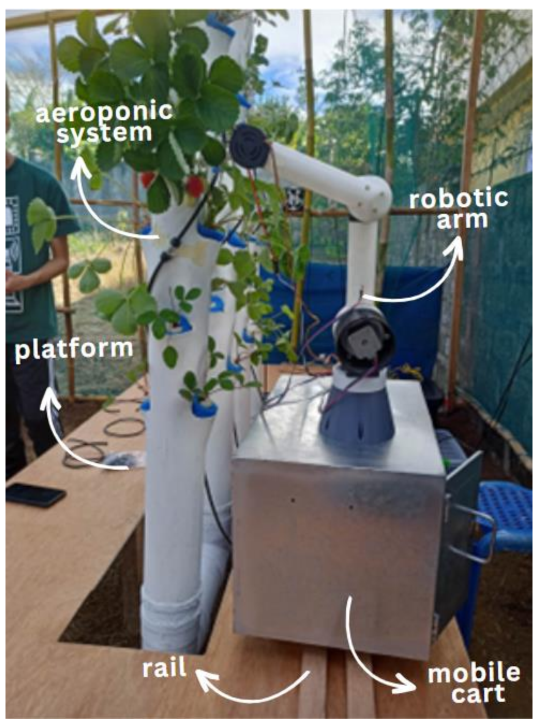


Figure 29. Actual photo of the Harvesting System

4.2 Project Capabilities and Limitations

The Automatic Strawberry Harvester is limited and designed to harvest strawberries inside the Vertical Aeroponic System whereas it is located at Tagaytay. The researchers trained the camera using the YOLOV4 algorithm to determine three classes – ripe and unripe strawberries, and the peduncle of the fruit. On the other hand, the gripper is programmed to cut and hold the peduncle of the ripe strawberries until it reaches the basket.

Moreover, the movement of the robotic arm is programmed using the software GrblGru, a Computer Aided Manufacturing program which has 3D simulations for controlling motions of machines. The robotic arm can move 2-feet above and below from the mid joint. Meanwhile, the cart is composed of one 5-inch wheel which is where the motor is connected to and four 2 inch rotating wheels for support that will move around the platform.

4.3 Project Evaluation

This section of the study presents the analysis and interpretation of the gathered result and initial testing of the prototype.

4.3.1 Hardware Mistake Attempt

The mistakes made within the production of the hardware along with the number of times it occurred, and also the calibrations made to patch up the mistake is presented below.

Table 17. Hardware Mistake Attempt

Hardware Mistake Attempt			
No.	**Description of mistake cause**	**Times**	**Calibrations**
1	Not compatible sizing of gears	4	Redesign of the sun and planetary gears using Fusion 360 and changing the filament used from PLA to PLA+.
2	Unsuitable gripper design and material use	4	Remodeling and revision of the gripper design and its operational mode, transitioning from a grabbing mechanism to a cutting and holding mechanism, employing the software Fusion 360 for the purpose. Furthermore, the investigation entails a material modification, specifically transitioning from PLA, PLA+, and TPU materials to PLAT+ material.
3	Inefficient rotary support for the actuators	3	Transitioning from using ball bearings as rotary support to using O-rings for friction-less rotation.
4	Insufficient power of output of torque	2	Alteration of motor used.

4.3.2 Harvesting Speed

The average time cost of initial testing with ball bearing as rotary support of each task process within 50 seconds is presented below.

Table 18. Time Cost Composition with Ball Bearing as Rotary Support

Time Cost Composition		
No.	Working Procedure	Time cost/s
1	Identify and locate target ripe strawberry.	25
2	Manipulator approach target	9
3	End-effector hold and cut.	1
4	Put fruit harvested onto the basket	15
	Total time cost	50

The time cost of initial testing with ball bearing as rotary support with grease of each task process within 41 seconds is presented below.

Table 19. Time Cost Composition with Ball Bearing as Rotary Support with Grease

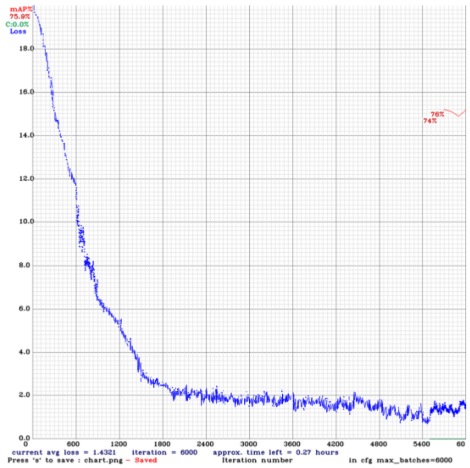
Time Cost Composition		
No.	Working Procedure	Time cost/s
1	Identify and locate target ripe strawberry.	20
2	Manipulator approach target	7
3	End-effector hold and cut.	1
4	Put fruit harvested onto the basket	13
	Total time cost	41

The time cost of the last three trials with O-rings as rotary support with grease of each task process within 15 seconds is presented below.

Table 20. Time cost composition with O-Ring as Rotary Support

Time Cost Composition		
No.	Working Procedure	Time cost/s
1	Identify and locate target ripe strawberry.	5
2	Manipulator approach target	2
3	End-effector hold and cut.	1
4	Put fruit harvested onto the basket	7
	Total time cost	15

4.3.3 Object Detection mAP Graph




```
 calculation mAP (mean average precision)...
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
40
 detections_count = 339, unique_truth_count = 177
class_id = 0, name = ripe, ap = 92.07%          (TP = 63, FP = 4)
class_id = 1, name = unripe, ap = 81.87%        (TP = 21, FP = 7)
class_id = 2, name = peduncle, ap = 53.79%      (TP = 45, FP = 30)

 for conf_thresh = 0.25, precision = 0.76, recall = 0.73, F1-score = 0.74
 for conf_thresh = 0.25, TP = 129, FP = 41, FN = 48, average IoU = 56.37 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.759119, or 75.91 %
Total Detection Time: 3 Seconds
```

Figure 30. (a)Average Loss and mAP Graph   Figure(b) mAP and Detection Time

The image above is the graph of the average loss in line with the number of iterations. In our case, we have 6000 iterations. It can be observed on the graph as the iterations go larger the average loss goes lower. The right figure shows how fast the detection can happen and how many strawberries are detected per each class and its percent. The final average loss came around approximately 1.50 to 1.80, while the final mean average precision @0.50 is 75.91%. In order to improve these numbers, we plan on feeding the algorithm more images from our data set the next time we are able to.

4.3.3.1 Object Detection Confidence Score

This section of the paper presents strawberry images both from the web and deployment site. This section also displays the detection confidence score of determining the three classes: ripe, unripe and the peduncle of the strawberries. It can be seen that the camera is capable of detecting the fruits with high accuracy.


```
[INFO] Initializing YOLO ..
[[572, 407, 699, 524], [354, 287, 498, 350], [497, 476, 578, 566], [923, 620, 1027, 716], [535, 378, 633, 441], [305, 337, 395, 406], [444, 362, 534, 483], [342, 402, 413, 461], [1024, 446, 1080, 503], [1013, 561, 1081, 627], [374, 438, 474, 499], [411, 356, 476, 405]]
['ripe', 'ripe', 'ripe', 'ripe', 'ripe', 'ripe', 'ripe', 'ripe', 'unripe', 'unripe', 'ripe', 'peduncle']
[0.9981639385223389, 0.9981548190116882, 0.9968574047088623, 0.9957260489463806, 0.9951490163803101, 0.9902166128158569, 0.9891964793205261, 0.9024600982666016, 0.9021826982498169, 0.8667629361152649, 0.7990331649780273, 0.7873478531837463]
```

Figure 31. (a) Testing of Actual Strawberries (b) Detection Confidence Score from the Program

The image above displays the initial testing of actual strawberries using the integrated raspberry pi camera. It can be seen that the camera detected all the strawberries in the captured photo, even the ones lying on the table. Provided also above is the Detection Confidence Score of the captured strawberries from the program. It can be seen that unripe strawberries have lower scores than the ripe ones.

Figure 32. Detection Testing of Strawberries from the Deployment Site

The displayed images above are the result of initial testing of the strawberries from the actual deployment site using the YOLOv4. It is evident that the program detected all the three classes with visible detection score.

Figure 33. Added Detection Testing from the Deployment Site 1

The images present above are the newly tested images from the deployment site. It can be seen that the vision system improved from the previous tests.

Figure 34. Added Detection Testing from the Deployment Site 2

The figure above shows that the vision system can even determine very small unripe strawberries. It can also be seen that it failed to detect the other unripe strawberry.

4.3.3.2 Occlusion Test

By testing cherry tomatoes and kalamansi fruits, the researchers have tested the accuracy of the vision system. Below, it can be seen that the system successfully scanned only the strawberries leaving the two other fruits undetected.

Figure 35. Simulated Occlusion Test

4.3.4 Arm Motions

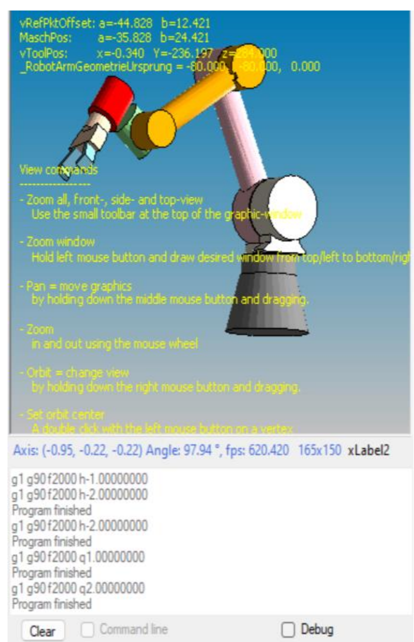


Figure 36. Simulation of the Robotic Arm's Movement using GrblGru

4.3.5 Harvesting Test Result

Presented below is the final harvesting test result, having tried three attempts which consistently comprised seven strawberries per attempt, the first attempt having a number of five ripe strawberries and only being able to harvest two because of the issue of jamming of gears and not being able to detect the other remaining crop. The second attempt successfully obtained four ripe strawberries out of five ones, the last ripe strawberry was not harvested due to the machine failing to cut the peduncle. Lastly, out of seven strawberries, six were ripe and all of those strawberries were harvested. Resulting in a 75% success rate.

Table 21. Harvesting Test Result

Attempt times	Strawberry quantity	No. of ripe strawberries	Successful harvesting times	Reason of Failure
1	7	5	2	Jamming of Gears
2	7	5	4	Failure to cut
3	7	6	6	None
Total attempt times	21	16	12	

Table 22. Harvesting Success Rate

Detected Strawberries	Successful Harvesting	Unsuccessful Harvesting
1		✕
2	✓	
3	✓	
4		✕
5		✕
6	✓	
7		✕
8	✓	
9	✓	
10	✓	
11	✓	
12	✓	
13	✓	
14	✓	
15	✓	
16	✓	

4.3.6 Harvesting Website

The figures below are the results of the integrated web page to the harvester. The results garnered from the PIR and the infrared sensors are save and at the same time have the real time results in the webpage. As the sensors are connected in the Arduino IOT cloud, it takes advantage of its real time data communication between the webpage and the sensors. With this, the researchers can monitor the harvester’s status wherever they are.

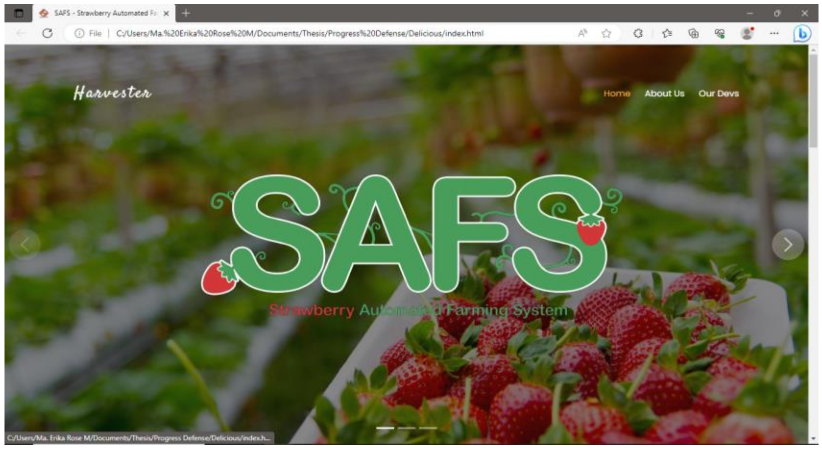


Figure 37. Harvester Webpage

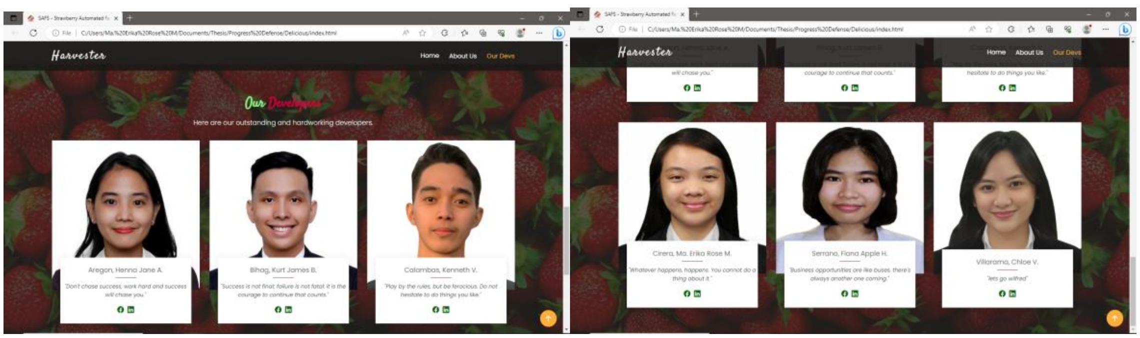


Figure 38. The Proponents posted in the Harvester's Webpage

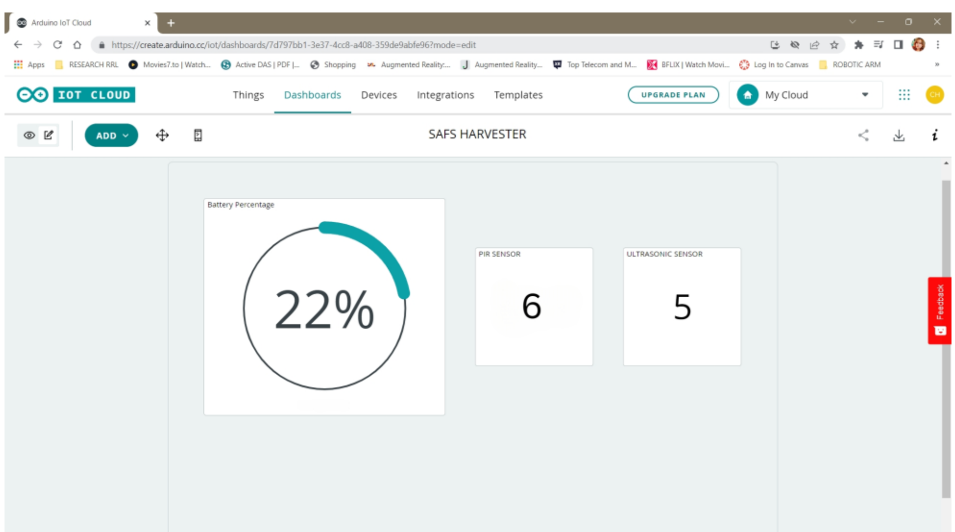


Figure 39. Monitoring page of the Harvester's Webpage

Chapter 5

Summary of Findings, Conclusion, and Recommendations

This chapter presents the comprehensive summary of the findings, elucidates the conclusions derived from the obtained results, and offers recommendations for enhancing the study.

5.1 Summary of Findings

The SAFS project is an advanced and fully automated detection and harvesting system designed specifically for the purpose of efficiently harvesting ripe strawberries. This cutting-edge system employs a precise mechanism that effectively severs the peduncle of the fruit, enabling swift and accurate collection of the fruit. To successfully test the vision system, the proponents used images from the internet, strawberries from the local market, and directly from the deployment site, Tagaytay. The vision system exhibits an average confidence score of 94% and achieves an average detection score of 0.57 milliseconds per object detected when deployed on Google Colab. Conversely, when implemented on the Raspberry Pi, the system demonstrates an average processing time of 15 seconds.

On the other hand, due to the persistent occurrence of the gear jamming attributed to the ball bearings, the researchers opted to use O-Rings as a strategic solution to expedite operations while minimizing or eliminating any potential delays. As a result of this modification, the average duration required for the picking process experienced a notable reduction, decreasing from 60 seconds to 15 seconds.

A single successful harvesting time has an average time of 15 seconds. With the exception of the time cost attributed to robot and vehicle movement, and the visual unit responsible for identifying the ripe strawberries.

5.2 Conclusion

In light of the findings and results obtained from the study, the proponents have derived the following conclusions:

The efficacy of a fully-automated harvesting system, comprising dual manipulator robotic arms utilized by Nema motors, a five-wheeled vehicle driven by a Nema motor, and cutter-type end effectors powered by 360 servo motors, has been substantiated by using Arduino mega, RAMP version 2, A4988 motor drivers, and TB6600 motor driver programmed using GRBL, a highly effective for CNC control due to its simplicity, reliability, compatibility, and strong community support. With a lightweight design and optimized codebase, it offers accurate and reliable control of CNC machines. GRBL is compatible with a wide range of CNC machines and runs on affordable Arduino-based microcontrollers, making it accessible to many users. Despite its simplicity, it provides essential features for CNC control and allows for customization and adaptation to specific machine configurations. This system demonstrates its effectiveness in successfully harvesting delicate vine crops, specifically strawberries, while ensuring the preservation of their fresh quality.

On the other hand, it is proven that the YOLOv4 is considered highly effective for vision systems due to its advancements in object detection accuracy and real-time performance. The use of the CSPDarknet53 backbone network, Feature Pyramid Network (FPN), and optimized object detection techniques improve the model's ability to detect objects of various sizes and shapes accurately. Moreover, the findings obtained from the conducted study underwent rigorous validation via a series of comprehensive trials, revealing that the fully automated harvesting system achieved a functional success rate of approximately 75%.

Moreover, the successful implementation of Arduino IoT Cloud and HTML programming has enabled us to develop a comprehensive website capable of effectively monitoring key parameters. Through this system, we can accurately track and analyze the idle time of the system, providing valuable insights into its operational efficiency. Additionally, the website allows us to monitor the number of strawberries harvested, providing crucial data for yield analysis and production optimization. With these monitoring capabilities in place, we can make informed decisions and drive improvements in the overall performance of the system.

In conclusion, the diligent implementation of various field tests, including the evaluation of time cost composition, harvesting speed, and hardware mistake attempts, has allowed the researchers to effectively quantify the magnitude of errors and assess the impact of the solutions on the entire system. These meticulous analyses have provided valuable insights into the performance and reliability of the system, enabling informed decision-making for further improvements. By thoroughly understanding the errors encountered and their corresponding solutions, future enhancements can be targeted to enhance the overall efficiency and effectiveness of the system, ultimately leading to enhanced productivity and optimized outcomes.

5.3 Recommendations

Based on the conclusions and summary of findings, the following recommendations are proposed to further enhance this research:

Design the arm to prevent locking or jamming, ensuring smoother harvesting through the incorporation of flexible joint systems, advanced control algorithms, and optimized gripping mechanisms.In addition, increase the length of the arms to achieve a wider span and improved access to crops. Enhance the motor used for the wheel into a higher-power alternative. Moreover, utilize a center wheel capable of full 360° rotation, rather than limited forward and backward motion. Implement sensor technology around the cart to enable autonomous operation and eliminate the need for a rail system.

In terms of enhancing the object detection algorithm, feeding the algorithm a greater number of dataset images and wider forms of augmentations might help in the further enhancement of its mean average precision score. Another way of developing the vision systems is through the use of better and higher quality video capturing devices and as well as programming the used vision hardware to enable image filter presets for better vision in all environmental conditions.

To improve the quality of the website, instead of using HTML to create a website, use a PHP programming language to directly create an integration and a connection between the harvester and the webpage. Use only a one page website instead of a scroll like to be more user friendly and to make it more easy to understand. Create another page wherein the user can see the previous harvest. And lastly, develop a mechanism for remote data transmission to a cloud server.

Finally, to create more reliable field tests, search for related studies with a functionality closely similar so that the results would be comparable and make inferences out of. Create another way of testing out the efficiency and effectiveness of the device.

References:

Asis, M. (2020). Sowing hope: Agriculture as an alternative to migration for young Filipinos? In The Future of Work: Labor Migrants in the 21st Century. Retrieved from https://www.icmc.net/future-of-work/report/06-philippines/

Barakat, A. N., Gouda, K. A., & Bozed, K. A. (2017). Kinematics analysis and simulation of a robotic arm using MATLAB. doi: 10.1109/CEIT.2016.7929032

Birrell, S., Hughes, J., Cai, J. Y., & Iida, F. (2020). A field-tested robotic harvesting system for iceberg lettuce. J. F. Robot., 37(2). doi: 10.1002/rob.21888

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. Retrieved from http://arxiv.org/abs/2004.10934 doi: 10.48550/arxiv.2004.10934

Chen, J., Lian, Y., & Li, Y. (2020). Real-time grain impurity sensing for rice combine harvesters using image processing and decision-tree algorithm. Comput. Electron. Agric., 175, June. doi: 10.1016/j.compag.2020.105591

Dewi, T., Rusdianasari, R., Kusumanto, R. D., & Sipronі, S. (2022). Image Processing Application on Automatic Fruit Detection for Agriculture Industry.

Dorj, U. O., Lee, M., & Yun, S. S. (2017). An yield estimation in citrus orchards via fruit detection and counting using image processing. Comput. Electron. Agric., 140, 103-112. doi: 10.1016/j.compag.2017.05.019

Durmus, H., Gunes, E. O., & Kirci, M. (2016, September 26). Data acquisition from greenhouses by using autonomous mobile robot. 2016 5th International Conference on Agro-Geoinformatics, Agro-Geoinformatics 2016. https://doi.org/10.1109/Agro-Geoinformatics.2016.7577696

Farkh, R., Al Jaloud, K., Alhuwaimel, S., Quasim, M. T., &Apologies for the formatting issue. Here are the references in APA format:

Feng, Q., Zou, W., Fan, P., Zhang, C., & Wang, X. (2018). Design and test of robotic harvesting system for cherry tomato. Int. J. Agric. Biol. Eng., 11(1), 96-100. doi: 10.25165/j.ijabe.20181101.2853

Funami, Y., Kawakura, S., & Tadano, K. (2020). Development of a Robotic Arm for Automated Harvesting of Asparagus. Eur. J. Agric. Food Sci., 2(1). doi: 10.24018/ejfood.2020.2.1.18

Gai, R., Chen, N., & Yuan, H. (2021). A detection algorithm for cherry fruits based on the improved YOLO-v4 model. Neural Comput. Appl. doi: 10.1007/s00521-021-06029-z

Ge, Y., Xiong, Y., & From, P. J. (2019). Instance Segmentation and Localization of Strawberries in Farm Conditions for Automatic Fruit Harvesting. IFAC-PapersOnLine, 52(30), 294-299. doi: 10.1016/j.ifacol.2019.12.537

Gunderman, A. L., et al. (2021). Tendon-Driven Soft Robotic Gripper for Berry Harvesting. CoRR, 38(2).

Hua, Y., et al. (2019). Recent Advances in Intelligent Automated Fruit Harvesting Robots. Open Agric. J., 13(1). doi: 10.2174/1874331501913010101

Ijemaru, G. K., et al. (2021). Image processing system using matlab-based analytics. Bull. Electr. Eng. Informatics, 10(5), 2566-2577. doi: 10.11591/eei.v10i5.3160

Jiang, Z., Zhao, L., Shuaiyang, L. I., & Yanfei, J. I. A. (2020). Real-time object detection method for embedded devices. arXiv, 3(October), 1-11.

Kultongkham, A., Kumnon, S., Thintawornkul, T., & Chanthsopeephan, T. (2021). The design of a force feedback soft gripper for tomato harvesting. J. Agric. Eng., 52(1). doi: 10.4081/jae.2021.1090

Latha, R. S., et al. (2021). Automatic Fruit Detection System using Multilayer Deep Convolution Neural Network. 2021 Int. Conf. Comput. Commun. Informatics, ICCCI 2021. doi: 10.1109/ICCCI50826.2021.9402513

Lee, W. C., & Kuo, S. A. (2020). Simulation and control of a robotic arm using MATLAB, Simulink, and TwinCAT. Int. Conf. Adv. Robot. Intell. Syst. ARIS, 2020-August. doi: 10.1109/ARIS50834.2020.9205777

Liu, X., Zhao, D., Jia, W., Ji, W., Ruan, C., & Sun, Y. (2019). Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access, 7, 139635-139642. doi: 10.1109/ACCESS.2019.2942144

Long, D. T., Binh, T. V., Hoa, R. V., Anh, L. V., & Toan, N. V. (2020). Robotic Arm Simulation by using Matlab and Robotics Toolbox for Industry Application. Int. J. Electron. Commun. Eng., 7(10), 1-4. doi: 10.14445/23488549/ijece-v7i10p101

Lucero, L., Lucero, D., Ormeno-Mejia, E., & Collaguazo, G. (2020). Automated aeroponics vegetable growing system. Case study lettuce. doi: 10.1109/ANDESCON50619.2020.9272180

Mai, X., Zhang, H., Jia, X., & Meng, M. Q. H. (2020). Faster R-CNN with Classifier Fusion for Automatic Detection of Small Fruits. IEEE Trans. Autom. Sci. Eng., 17(3), 1555-1569. doi: 10.1109/TASE.2020.2964289

Marsono, Yoto, Suyetno, A., & Nurmalasari, R. (2021). Design and Programming of 5 Axis Manipulator Robot with GrblGru Open Source Software on Preparing Vocational Students' Robotic Skills. Journal of Robotics and Control (JRC), 2(6). doi: 10.18196/jrc.26134. ISSN: 2715-5072

McCool, C., Sa, I., Dayoub, F., Lehnert, C., Perez, T., & Upcroft, B. (2016). Visual detection of occluded crop: For automated harvesting. In Proceedings - IEEE International Conference on Robotics and Automation, 2016, vol. 2016-June. doi: 10.1109/ICRA.2016.7487405

Oktarina, Y., Dewi, T., Risma, P., & Nawawi, M. (2020). Tomato Harvesting Arm Robot Manipulator; A Pilot Project. Journal of Physics: Conference Series, 1500(1). doi: 10.1088/1742-6596/1500/1/012003

Onishi, Y., Yoshida, T., Kurita, H., Fukao, T., Arihara, H., & Iwai, A. (2019). An automated fruit harvesting robot by using deep learning. ROBOMECH J., 6(1), 2-9. doi: 10.1186/s40648-019-0141-2

Pitla, S., Bajwa, S., Bhusal, S., Brumm, T., & Brown-Brandl, T. (2020). Ground and Aerial Robots for Agricultural Production: Opportunities and Challenges. Council for Agricultural Science and Technology (CAST), (70), 4-5.

Preira et al. (2017). Recent Advances in Image Processing Techniques for Automated Harvesting Purposes: A Review. Intelligent Systems Conference 2017, London Uk, doi:10.1109/IntelliSys.2017.8324352

Puttemans, S., Vanbrabant, Y., Tits, L., & Goedemé, T. (2017). Automated visual fruit detection for harvest estimation and robotic harvesting. doi: 10.1109/IPTA.2016.7820996

Raghunandan, A., Mohana, P., Raghav, P., & Aradhya, H. V. R. (2018). Object Detection Algorithms for Video Surveillance Applications. Proc. 2018 IEEE Int. Conf. Commun. Signal Process. ICCSP 2018, 563-568. doi: 10.1109/ICCSP.2018.8524461

Raja, V., Bhaskaran, B., Nagaraj, K. K. G., Sampathkumar, J. G., & Senthilkumar, S. R. (2022). Agricultural harvesting using integrated robot system. Indones. J. Electr. Eng. Comput. Sci., 25(1), 152-158. doi: 10.11591/ijeecs.v25.i1.pp152-158

Rong, J., Wang, P., Yang, Q., & Huang, F. (2021). A field-tested harvesting robot for oyster mushroom in greenhouse. Agronomy, 11(6). doi: 10.3390/agronomy11061210

Rubio, F., Valero, F., & Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. International Journal of Advanced Robotic Systems, 16(2). doi: 10.1177/1729881419839596

Sepulveda, D., Fernandez, R., Navas, E., Armada, M., & Gonzalez-De-Santos, P. (2020). Robotic Aubergine Harvesting Using Dual-Arm Manipulation. IEEE Access, 8. doi: 10.1109/ACCESS.2020.3006919

Simeon, L. M. (2020, November 29). Employment in agriculture declining, but wages are up. Philstar.com. Retrieved June 3, 2022, from https://www.philstar.com/business/2020/11/30/2060309/employment-agriculture-declining-wages-are-up/

Statista. (2021, October 26). Topic: Agriculture in the Philippines. Retrieved June 26, 2022, from https://www.statista.com/topics/5744/agriculture-industry-in-the-philippines/

Wang, G., Yu, Y., & Feng, Q. (2016). Design of End-effector for Tomato Robotic Harvesting. IFAC-PapersOnLine, 49(16), 190-193. doi: 10.1016/j.ifacol.2016.10.035

Wang, H., Hohimer, C. J., Bhusal, S., Karkee, M., Mo, C., & Miller, J. H. (2018). Simulation As A Tool In Designing And Evaluating A Robotic Apple Harvesting System⁎. 51(17). doi: 10.1016/j.ifacol.2018.08.076

Williams, H. A. M., et al. (2019). Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. Biosyst. Eng., 181. doi: 10.1016/j.biosystemseng.2019.03.007

Xiong, Y., Ge, Y., Grimstad, L., & From, P. J. (2020). An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. J. F. Robot., 37(2), 202-224. doi: 10.1002/rob.21889

Xiong, Y., Peng, C., Grimstad, L., From, P. J., & Isler, V. (2019). Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. Comput. Electron. Agric., 157. doi: 10.1016/j.compag.2019.01.009

Xu, Z. F., Jia, R. S., Liu, Y. B., Zhao, C. Y., & Sun, H. M. (2020). Fast method of detecting tomatoes in a complex scene for picking robots. IEEE Access, 8, 55289-55299. doi: 10.1109/ACCESS.2020.2981823

Yaguchi, H., Nagahama, K., Hasegawa, T., & Inaba, M. (2016). Development of an autonomous tomato harvesting robot with rotational plucking gripper. In IEEE International Conference on Intelligent Robots and Systems, 2016, vol. 2016-November. doi: 10.1109/IROS.2016.7759122

Yu, Y., Zhang, K., Yang, L., & Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. Comput. Electron. Agric., 163. doi: 10.1016/j.compag.2019.06.001

Zhang, L., Jia, J., Gui, G., Hao, X., Gao, W., & Wang, M. (2018). Deep Learning Based Improved Classification System for Designing Tomato Harvesting Robot. IEEE Access, 6(c), 67940-67950. doi: 10.1109/ACCESS.2018.2879324

Zhang, T., Huang, Z., You, W., Lin, J., Tang, X., & Huang, H. (2020). An autonomous fruit and vegetable harvester with a low-cost gripper using a 3D sensor. Sensors (Switzerland), 20(1). doi: 10.3390/s20010093

Zhou, H., Wang, X., & Chen, C. (2021). Intelligent Robots for Fruit Harvesting: Recent Developments and Future Challenges Background. Springer US. doi: 10.1007/s11119-022-09913-3

ANNEXES

ANNEX A

Bill of Materials

QUANTITY	COMPONENT	PRICE
2	3D Printed Arm Shell and Actuator Gears	₱ 11,000.00
2	35mm tin can stepper	₱ 230.00
2	SG90 Micro Servo Motor	₱ 240.00
4	Nema 17 (28 N-cm)	₱ 1,880.00
4	Nema 17 (48 N-cm)	₱ 2,600.00
1	Nema 23 (9kg-cm)	₱ 1,000.00
10	A4988 Motor Driver	₱ 1,800.00
1	TB6600 Motor Driver	₱ 350.00
2	Ramps1.4 Motor Driver Shield	₱ 600.00
2	Arduino Mega ATMega260	₱ 1,700.00
1	Arduino Uno	₱ 650.00
1	ESP8266 WIFI MCU	₱ 100.00
1	UltraSonic Sensor	₱ 100.00
1	PIR Motion Sensor	₱ 100.00
1	LiFePo4 12V Battery Pack	₱ 5,500.00

12	O-Rings	₱ 2,510.00
1	Raspberry Pi Camera V2	₱ 470.00
1	USB Camera	₱ 200.00
1	Raspberri Pi 4	₱ 5,000.00
1	32 GB MicroSD	₱ 320.00
1	RPi Metal Enclosure	₱ 200.00
3	RPi Heat Sink	₱ 45.00
1	RPi USB-C Power Supply	₱ 280.00
1	Micro HDMI to HDMI Converter	₱ 100.00
1	Ethernet Cable	₱ 55.00
1	Screw and Nut Kit	₱ 800.00
1	Wire Connector Kit	₱ 300.00
1	Harvester Cart	₱ 1,000.00
1	Harvester Ramp/Elevation	₱ 3,000.00
	Total	₱ 42,130.00

ANNEX B

Incomplete Program Codes

SAFS Web Application Code

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>SAFS - Strawberry Automated Farming System</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="assets/img/icon.png" rel="icon">
  <link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?family=Poppins:300,300i,400,400i,600,600i,700,700i|Satisfy|Comic+Neue:3
00,300i,400,400i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="assets/vendor/animate.css/animate.min.css" rel="stylesheet">
  <link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="assets/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="assets/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="assets/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
```

```html
  <!-- Template Main CSS File -->
  <link href="assets/css/style.css" rel="stylesheet">

</head>

<body>

  <!-- ======= Header ======= -->
  <header id="header" class="fixed-top d-flex align-items-center header-transparent">
    <div class="container-fluid container-xl d-flex align-items-center justify-content-between">

      <div class="logo me-auto">
        <h1><a href="https://create.arduino.cc/iot/dashboards/7d797bb1-3e37-4cc8-a408
359de9abfe6?mode=view">Harvester</a></h1>

      </div>

      <nav id="navbar" class="navbar order-last order-lg-0">
        <ul>
          <li><a class="nav-link scrollto active" href="#hero">Home</a></li>
          <li><a class="nav-link scrollto" href="#why-us">About Us</a></li>
          <li><a class="nav-link scrollto" href="#chefs">Our Devs</a></li>
    </div>
  </header><!-- End Header -->

  <section id="hero">
    <div class="hero-container">
```

```
<div id="heroCarousel" data-bs-interval="5000" class="carousel slide carousel-fade" data-bs-ride="carousel">

 <ol class="carousel-indicators" id="hero-carousel-indicators"></ol>

 <div class="carousel-inner" role="listbox">

  <!-- Slide 1 -->
  <div class="carousel-item active" style="background-image: url(assets/img/slide/slide-1.jpg);">
   <div class="carousel-container">
    <div class="carousel-content">
     <img src="assets/img/safs.png">
    </div>
   </div>
  </div>

  <!-- Slide 2 -->
  <div class="carousel-item" style="background-image: url(assets/img/slide/slide-2.jpg);">
   <div class="carousel-container">
    <div class="carousel-content">
     <img src="assets/img/safs.png">
    </div>
   </div>
  </div>

  <!-- Slide 3 -->
  <div class="carousel-item" style="background-image: url(assets/img/slide/slide-3.jpg);">
   <div class="carousel-container">
    <div class="carousel-content">
```

```
            <img src="assets/img/safs.png">
           </div>
          </div>
         </div>

        </div>

        <a class="carousel-control-prev" href="#heroCarousel" role="button" data-bs-slide="prev">
         <span class="carousel-control-prev-icon bi bi-chevron-left" aria-hidden="true"></span>
        </a>

        <a class="carousel-control-next" href="#heroCarousel" role="button" data-bs-slide="next">
         <span class="carousel-control-next-icon bi bi-chevron-right" aria-hidden="true"></span>
        </a>

       </div>
      </div>
    </section><!-- End Hero -->

    <main id="main">

      <!-- ======= Whu Us Section ======= -->
      <section id="why-us" class="why-us">
       <div class="container">

        <div class="section-title">
       <img src= "assets/img/logozz.png" width="550" height="200"/>
        </div>
```

```
</div>
<center>
<p style="color: white " "margin-left: 200px" "margin-right: 200px">SAFS is an automated harvester designed to be incorporated into AeroPhoenix' V-SAFS. This harvester will be consisting of a dual-armed manipulator mounted on a railed vehicle which will enable for a much higher harvesting speed and can be used for a coordinated fruit picking in situations where the target is occuled. The project will also integrate the use of a webpage for the harvester's condition and harvesting status monitoring.</p>
        </p>
        <ul>
          <li style="color: #90ee90"><i class="bx bx-check-double" style="color: #C70039"></i> Strawberries are the first fruit to ripen each spring.</li>
          <li style="color: #90ee90"><i class="bx bx-check-double" style="color: #C70039"></i> There are 200 seeds on an average strawberry.</li>
          <li style="color: #90ee90"><i class="bx bx-check-double" style="color: #C70039"></i> The seeds can grow into new strawberry plants, but most instead reproduce through runners.</li>
        </ul>
        <p>
          Packed with vitamins, fiber, and particularly high levels of antioxidants known as polyphenols, strawberries are a sodium-free, fat-free, cholesterol-free, low-calorie food. They are among the top 20 fruits in antioxidant capacity and are a good source of manganese and potassium.
        </p>
      </div>

    </div>

   </div>

  </div>
```

```
</section><!-- End About Section -->

<section id="chefs" class="chefs">
  <div class="container">

    <div class="section-title">
      <h2>Our <span>Developers</span></h2>
      <p style="color: white">Here are our outstanding and hardworking developers.</p>
    </div>

    <div class="row">

      <div class="col-lg-4 col-md-6">
        <div class="member">
          <div class="pic"><img src="assets/img/chefs/hens.jpg" class="img-fluid" alt=""></div>
          <div class="member-info">
            <h4>Aregon, Henna Jane A.</h4>
            <span>"Don't chase success, work hard and success will chase you."</span>
            <div class="social">

              <a href="https://www.facebook.com/henna.aregon"><i class="bi bi-facebook"></i></a>

              <a href=""><i class="bi bi-linkedin"></i></a>
            </div>
          </div>
        </div>
      </div>
```

```
<div class="col-lg-4 col-md-6">
  <div class="member">
    <div class="pic"><img src="assets/img/chefs/kurts.jpg" class="img-fluid" alt=""></div>
    <div class="member-info">
      <h4>Bihag, Kurt James B.</h4>
      <span>"Success is not final; failure is not fatal: it is the courage to continue that counts."</span>
      <div class="social">

        <a href="https://www.facebook.com/kurtbihag"><i class="bi bi-facebook"></i></a>

        <a href=""><i class="bi bi-linkedin"></i></a>
      </div>
    </div>
  </div>
</div>

<div class="col-lg-4 col-md-6">
  <div class="member">
    <div class="pic"><img src="assets/img/chefs/calambas.jpg" class="img-fluid" alt=""></div>
    <div class="member-info">
      <h4>Calambas, Kenneth V.</h4>
      <span>"Play by the rules, but be ferocious. Do not hesitate to do things you like."</span>
      <div class="social">

        <a href="https://www.facebook.com/kenneth.calambas"><i class="bi bi-facebook"></i></a>

        <a href=""><i class="bi bi-linkedin"></i></a>
      </div>
```

```
          </div>
        </div>
      </div>

    <h1></h1>
    <br>
    <br>

    <div class="col-lg-4 col-md-6">
          <div class="member">
            <div class="pic"><img src="assets/img/chefs/cirera.jpg" class="img-fluid" alt=""></div>
            <div class="member-info">
              <h4>Cirera, Ma. Erika Rose M.</h4>
              <span>"Whatever happens, happens. You cannot do a thing about it."</span>
              <div class="social">

                <a href="https://www.facebook.com/erika.cirera"><i class="bi bi-facebook"></i></a>

                <a href="https://www.linkedin.com/in/ma-erika-rose-cirera-774b49244/"><i class="bi bi-
linkedin"></i></a>
              </div>
            </div>
          </div>
        </div>

        <div class="col-lg-4 col-md-6">
          <div class="member">
            <div class="pic"><img src="assets/img/chefs/serrano.png" class="img-fluid" alt=""></div>
```

```
              <div class="member-info">
                <h4>Serrano, Fiona Apple H.</h4>
                <span>"Business opportunities are like buses, there's always another one coming."</span>
                <div class="social">

                  <a href="https://www.facebook.com/ionic.lines"><i class="bi bi-facebook"></i></a>

                  <a href=""><i class="bi bi-linkedin"></i></a>
                </div>
              </div>
            </div>
          </div>

          <div class="col-lg-4 col-md-6">

  <!-- Vendor JS Files -->
  <script src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="assets/vendor/glightbox/js/glightbox.min.js"></script>
  <script src="assets/vendor/isotope-layout/isotope.pkgd.min.js"></script>
  <script src="assets/vendor/swiper/swiper-bundle.min.js"></script>
  <script src="assets/vendor/php-email-form/validate.js"></script>

  <!-- Template Main JS File -->
  <script src="assets/js/main.js"></script>

</body>

</html>
```

PIR Sensor

```
int inputPin = 2; // choose the input pin (for PIR sensor)
int pirState = LOW; // we start, assuming no motion detected
int val = 0; // variable for reading the pin status
int counter = 0;
int currentState = 0;
int previousState = 0;
void setup() {
pinMode(inputPin, INPUT); // declare sensor as input
Serial.begin(9600);
}
void loop(){
val = digitalRead(inputPin); // read input value
if (val == HIGH) { // check if the input is HIGH
if (pirState == LOW) {
// we have just turned on
currentState = 1;
// We only want to print on the output change, not state
pirState = HIGH;
delay(1000);
}
} else {
if (pirState == HIGH){
// we have just turned of
currentState = 0;
// We only want to print on the output change, not state
pirState = LOW;
```

```
  delay(1000);
}
}
if(currentState != previousState){
if(currentState == 1){
counter = counter + 1;
Serial.println(counter);
  delay(1000);
}
}
}
```

Ultrasonic Sensor

```
#define trigPin 13
#define echoPin 12
int counter = 0;
int currentState = 0;
int previousState = 0;
void setup() {
 Serial.begin (9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);
}
void loop() {
 long duration, distance;
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
```

```
delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance
```

ANNEX C

Duplication Manual

SAFS: A Railed Automobile Strawberry Harvester using Dual-Arm Manipulator via YOLOv4 Algorithm for Detection and Navigation with Cloud-based Webpage for Monitoring

SAFS

Strawberry Automated Farming System

DUPLICATION MANUAL

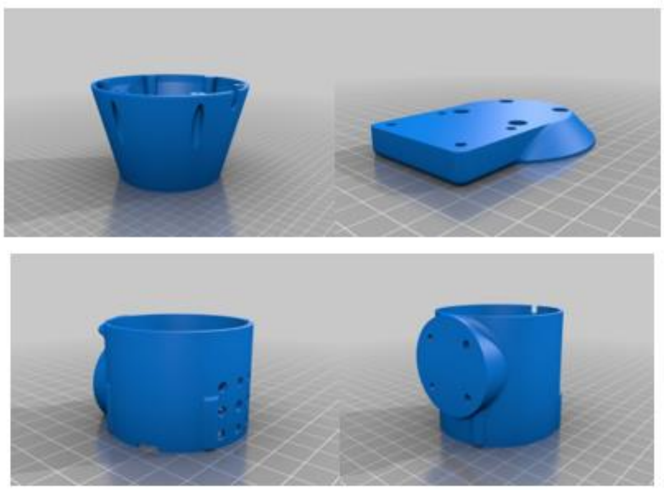
Version 1.0
July 2023

HARDWARE DUPLICATION

A. Design 3D Model for the Robotic Arm

a. Arm Shell Design

The proponents used a cloud-based 3D Modelling and CAD software to design the body of the Robotic Arm. The main component for the robotic mechanism's body is the shell construct, made hollow to reduce the overall weight of the arm in order to have an arm mass that is supportable by the torque of the current available commercial stepper motors.

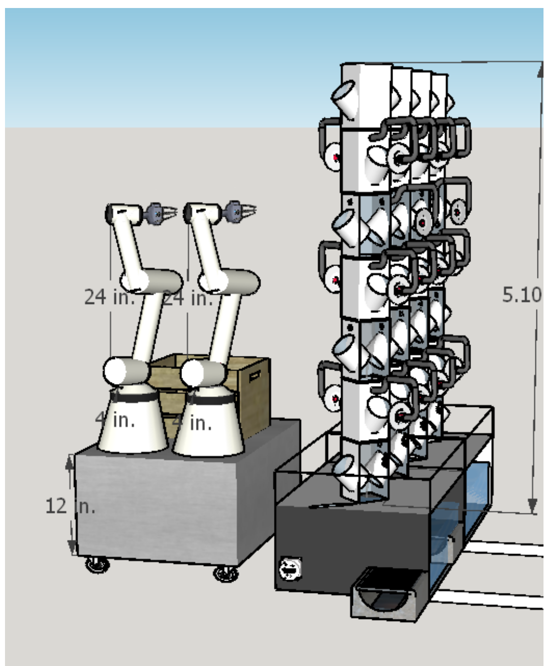


3D Model of Base, Shoulder Support, and Joint Shells

3D Model Upper Arm, Fore Arm, and Joint Shell Lids

b Arm Model Scale

In order to fulfill the project's goal to harvest strawberries from a pre-scaled and pre-designed aeroponic system, the proponents scaled the robotic arms to be able to reach the highest and lowest possible locations that the strawberries can grow on to. The aeroponics tower reaches a maximum height of 6 feet, with 5 feet allocated for the cultivation section; the robotic arm's final dimension is 2 feet for the arm's reach in all directions and has a 1 foot cart for elevation and mobility.

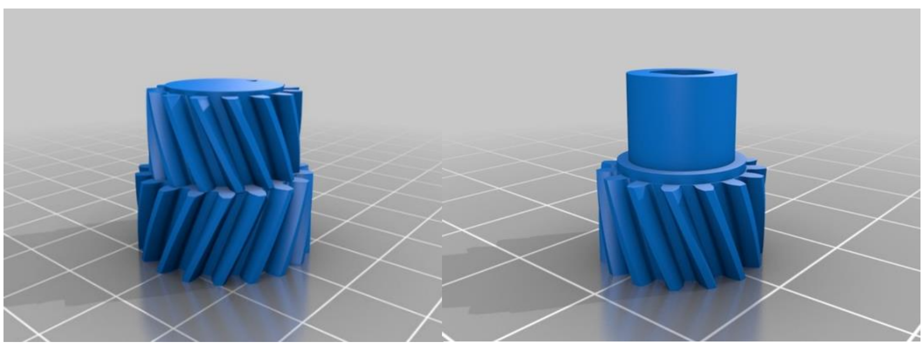


3D depiction of the Harvesting unit next to the Aeroponic Towers

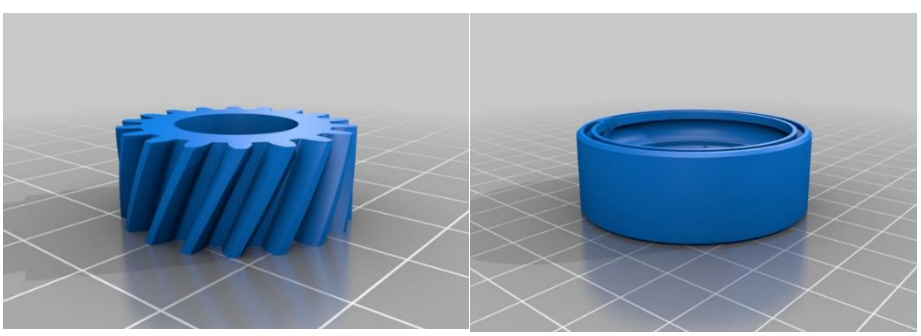
B. Design for Joint Actuators

a. Design of the Gears

With Fusion 360, the proponents were also able to design the gears inside the joint actuators. The gears are designed to work as helical planetary gears, which will reduce backlash and friction from continuous rotation. Each joint will consist of four (4) internal smaller gears: three (3) planetary cluster gears and two (2) interconnecting sun gears, while also being incased by two (2) shell gears: the body and ring gears.



3D Model of Cluster and
Lower Sun gear, respectively



3D Model of Upper Sun and
Body gear, respectively

b. Gear Scale and Ratio

The actuators are made up of three compound planetary gearboxes of different sizes that are perfect for assembling into robot arms. All three sizes utilize 38.4:1 compound planetary gearing and come with an integrated slew bearing. Their compact and pancake-like design, along with convenient mounting surfaces, simplifies the process of creating your own robot arm by stacking these actuators.

The largest actuator, suitable for arm base joints, has a diameter of approximately 3.4 inches and employs 20 pitch helical gears. This variant is specifically engineered to be used with higher torque NEMA 17 stepper motors.

The medium-sized actuator, around 2.4 inches in diameter, is equipped with 30 pitch helical gears and is designed to accommodate smaller NEMA 17 stepper motors.

The smallest actuator, with a diameter of 2 inches, uses 40 pitch helical gears and is intended for 35mm "tin can" style stepper motors.

C. 3D Printing and Parts Assembly

a. 3D Printing

All the 3D Models are then exported as STL files and were sent to a local fabrication shop that has a 3D printing service. The filament used for all the components is PLA as it offers improved toughness and impact resistance compared to standard PLA, making it more durable and less prone to breaking. PLA+ also has a higher heat resistance, allowing it to withstand higher temperatures without deforming.

HARDWARE DUPLICATION

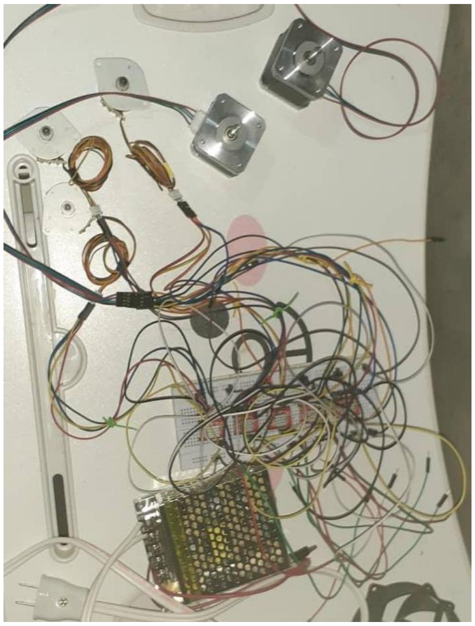
PLA+ Filament used for 3D printing the components of the Robotic Arms

3D Printed Parts

SAFS
Strawberry Automated Farming System

b. Motors

In order to assemble the Robotic Arms, motors with enough torque to lift itself and an additional load will be needed. 35 mm stepper motors are deemed to be enough to drive the wrist actuators of the arms. While NEMA 17 of varying torques are used to drive the base, shoulder and elbow actuators. And finally, a NEMA 23 stepper motor is used to as the cart's main driving motor.



Tin Can and Stepper Motors
for each actuators of the
Robotic Arms

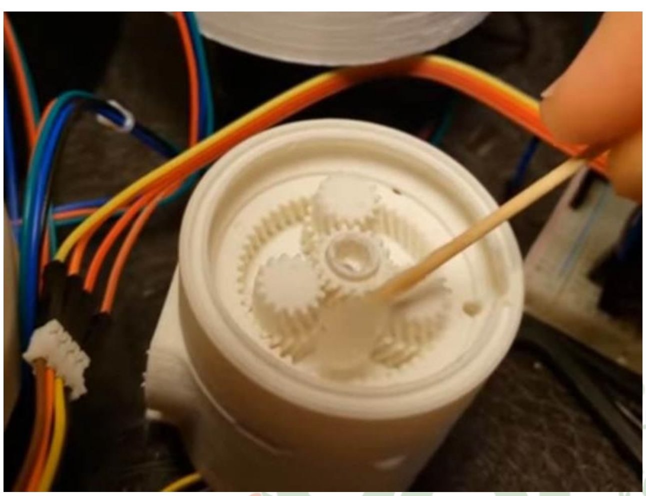
c. Assembly

To assemble the arms, the internal gears are needed to be fit inside the actuators while the motor shafts are fixed into the lower sun gear. 3mm to 5mm flat-screw heads are used for fastening the parts to each other.

HARDWARE DUPLICATION

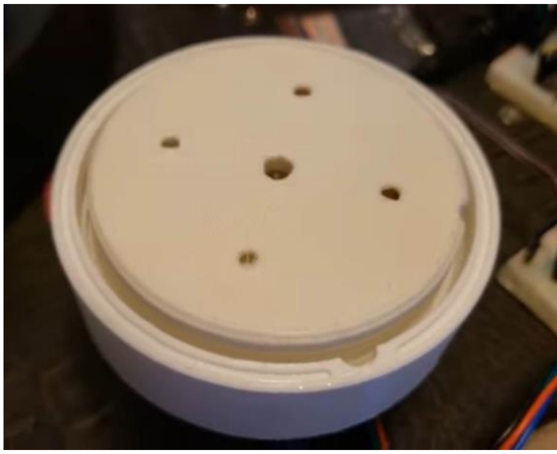
Assembly of Inner gears for arm actuators with NEMA 17 motors

Sun Gear to Motor Shaft assemblt for wrist actuators with 35mm tin can stepper motor

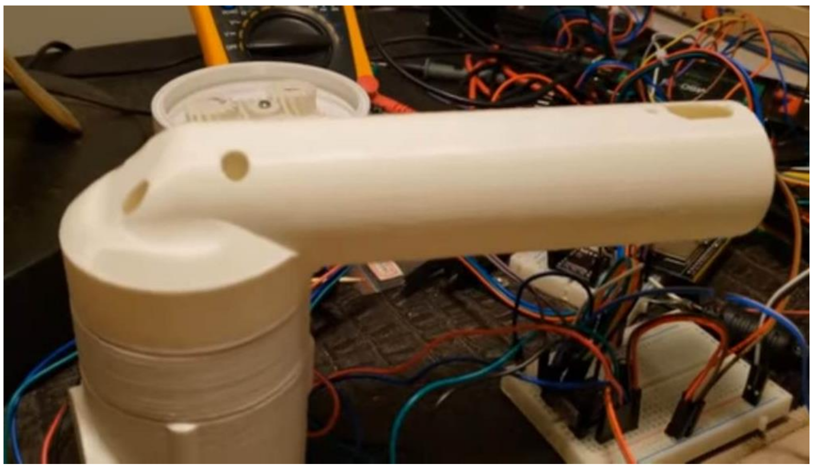


Application of Vaseline for lubricating the gears to reduce friction while rotation

HARDWARE DUPLICATION




Closing the actuator with the
Ring Gear and Attaching to the
Joint Shell



Attaching the Upper Arm Shell
to the Shoulder Actuator

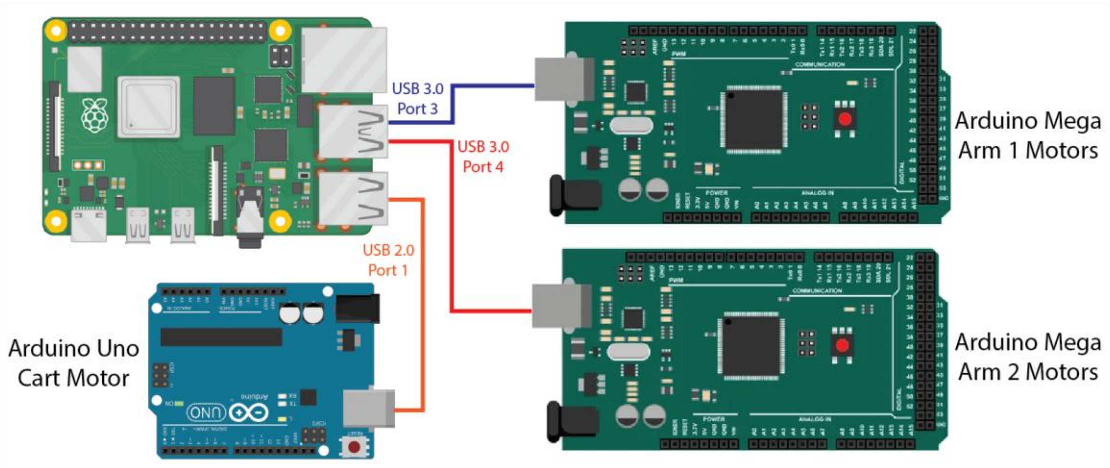
SAFS
Strawberry Automated Farming System

Completed assembly of the
Robotic Arm

D. Wiring Connections

a. Central Control Unit to Microcontroller

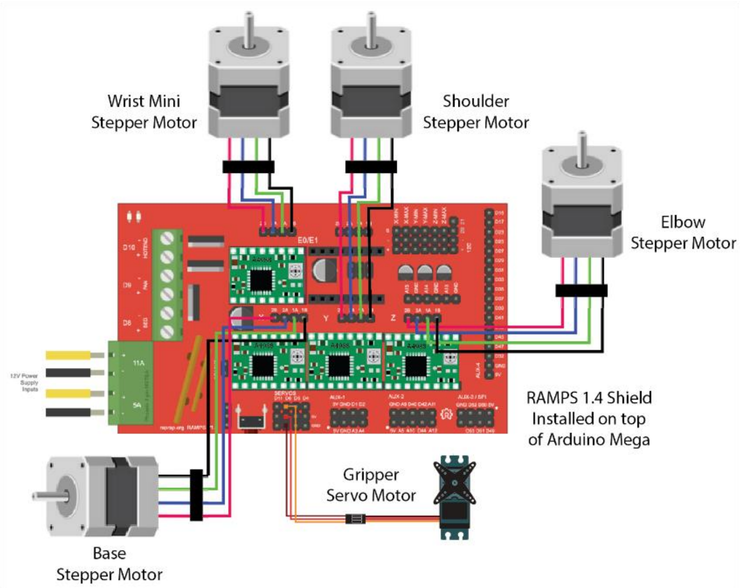
In order for the harvesting machine to work and move from one place to another, the use of microcontrollers is essential. Having two robotic arm units and a motorized cart requires the whole device to have more than one microcontroller. And in order to use all of them synchronously, the proponents used Raspberry Pi 4 as the Central Control Unit for three separate microcontrollers: two (2) Arduino Mega for the Arms and one (1) Arduino Uno for the cart's motor. Below is their simple connection.



Arduino to Raspberry Pi Connection

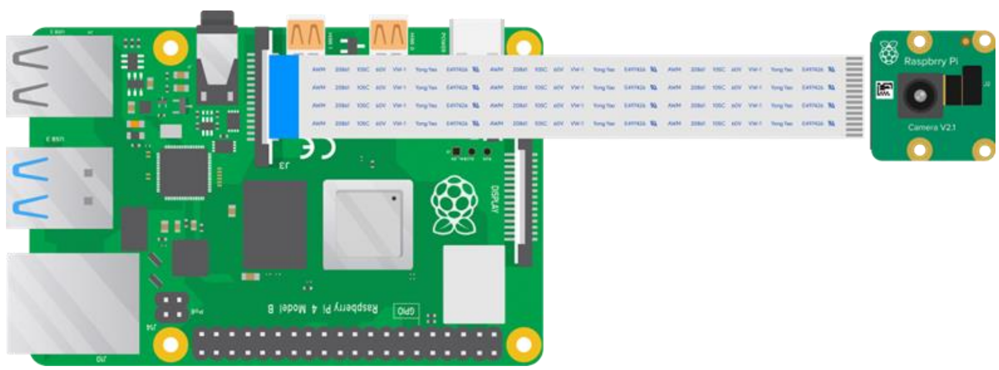
b. RAMPS to Motor Connections

In order to use more than two motors in one Microcontroller, it is necessary to use an extender shield for the Arduino boards. With that need, the proponents used RAMPS 1.4 to be able to use four (4) stepper motors and an additional servo motor using one microcontroller board. It also enabled the board to accommodate several A4988 motor drivers at the same time. Below is the wiring connection of the motors, motor driver placement and orientation, also labeled accordingly based on their intended joint actuator positioning and as end-effector actuator.



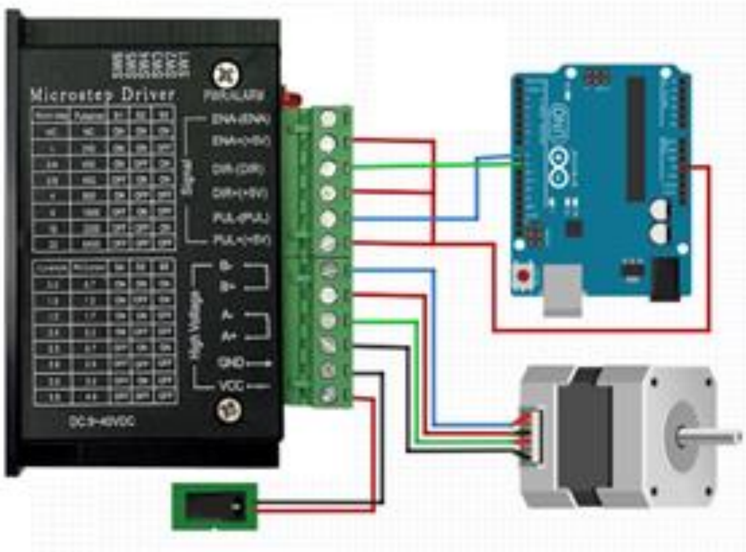
c. Camera Connection

As the project also integrates a vision system for object detection, the proponents have purchased a compatible camera unit for the Raspberry Pi 4. The Raspberry Pi Camera V2 module connects to the Raspberry Pi board via a ribbon cable, providing a compact and integrated solution for capturing images and videos. It is capable of recording high-definition video at resolutions of up to 1080p at 30 frames per second, or 720p at 60 frames per second. Below is the simple connection of the camera ribbon to the Raspberry Pi GPIO ribbon port.



d. Cart Motor to Microcontroller

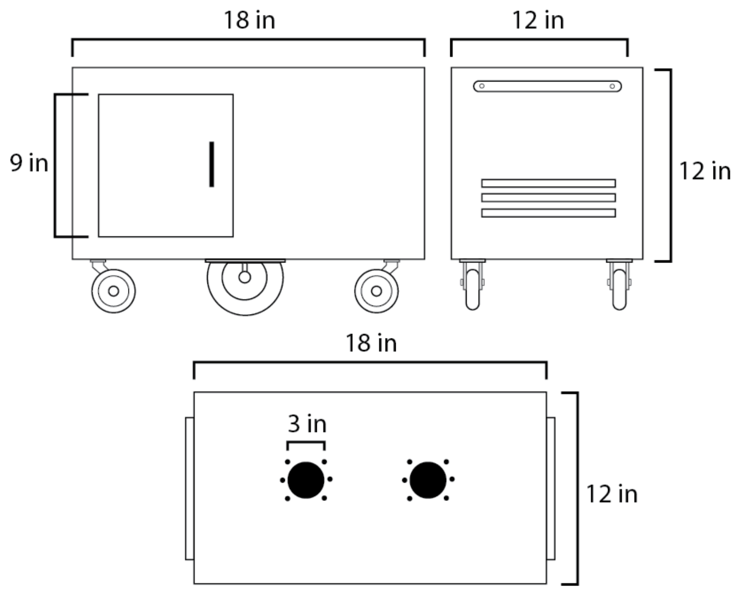
The motor for the cart will be using a much heavy duty motor driver since NEMA 23- the chosen high torque stepper motor, need a higher rating of voltage and current supply. With that, the proponents used TB6600 motor driver, which is compatible for higher rated stepper motors such as NEMA 23. Also, a separate microcontroller is used for the cart’s specific movements since both the Arduino Mega are already configured for the movement of the robotic arms. For the cart, Arduino Uno is used as the microcontroller. Below is the connection of Arduino Uno to Motor Driver, and Motor Driver to Stepper Motor.



E. Mobile Cart

a. Design and Dimension of Cart

The cart is designed to be able to mount the robotic arms to a vehicular-type of elevation, which will allow the arms to have a better reach at the cultivating area's proximity. The cart was planned to be made up of stainless steel to lessen the chance of rust in the cart's surface that can be due to many external factors such as weather, atmospheric condition and humidity from the aeroponic system's water sprinklers.

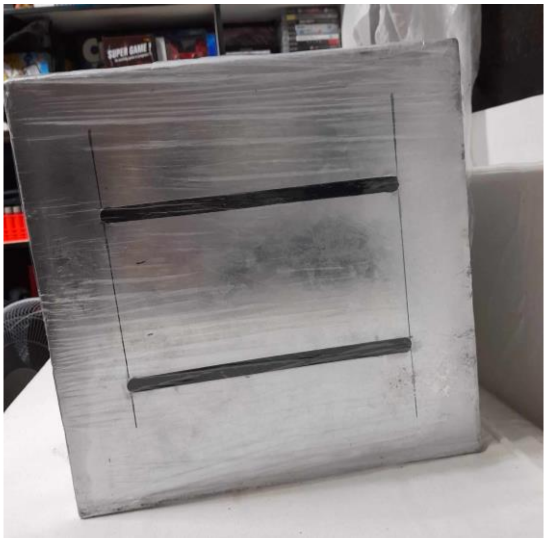


Cart Design and Exact Dimensions

b. Fabrication of the Cart

The cart is fabricated in a metal fabrication shop to ensure the best quality. Additional design changes were also made in collaboration with the person in-charge of fabricating the cart.

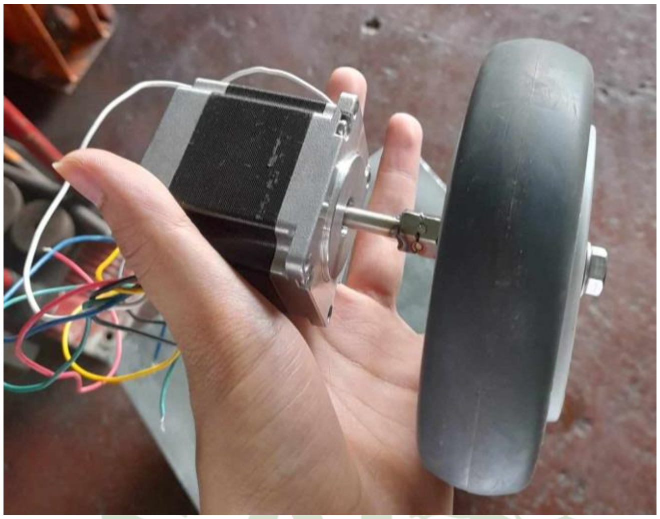
Front of the Cart



Front of the Cart

c. Installing Motor

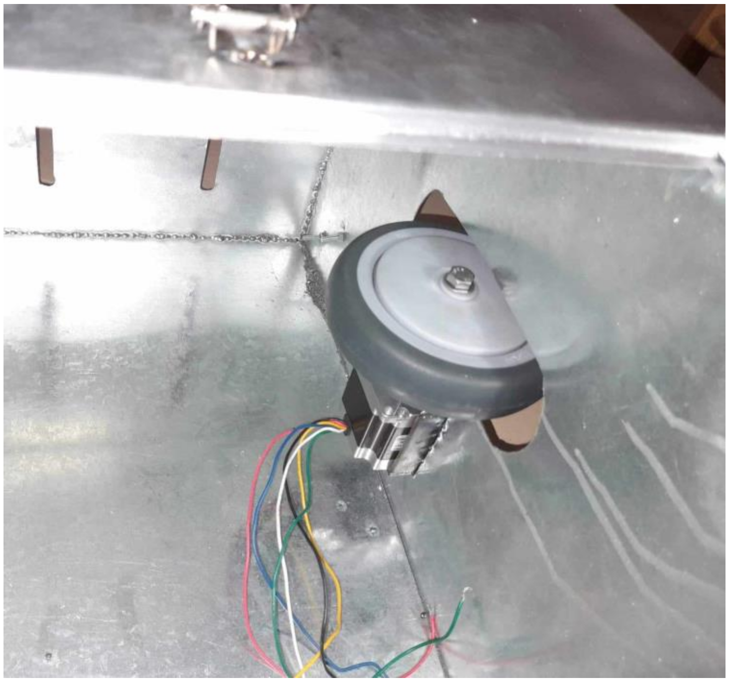
To make the cart be able to move, the NEMA 23 motor is needed to be installed at the center of the cart. This decision is made to ensure that the acceleration force of the central wheel will be well distributed to the corner wheels.



NEMA 23 attached to the central wheel

HARDWARE DUPLICATION

Central Wheel attached viewed under the cart

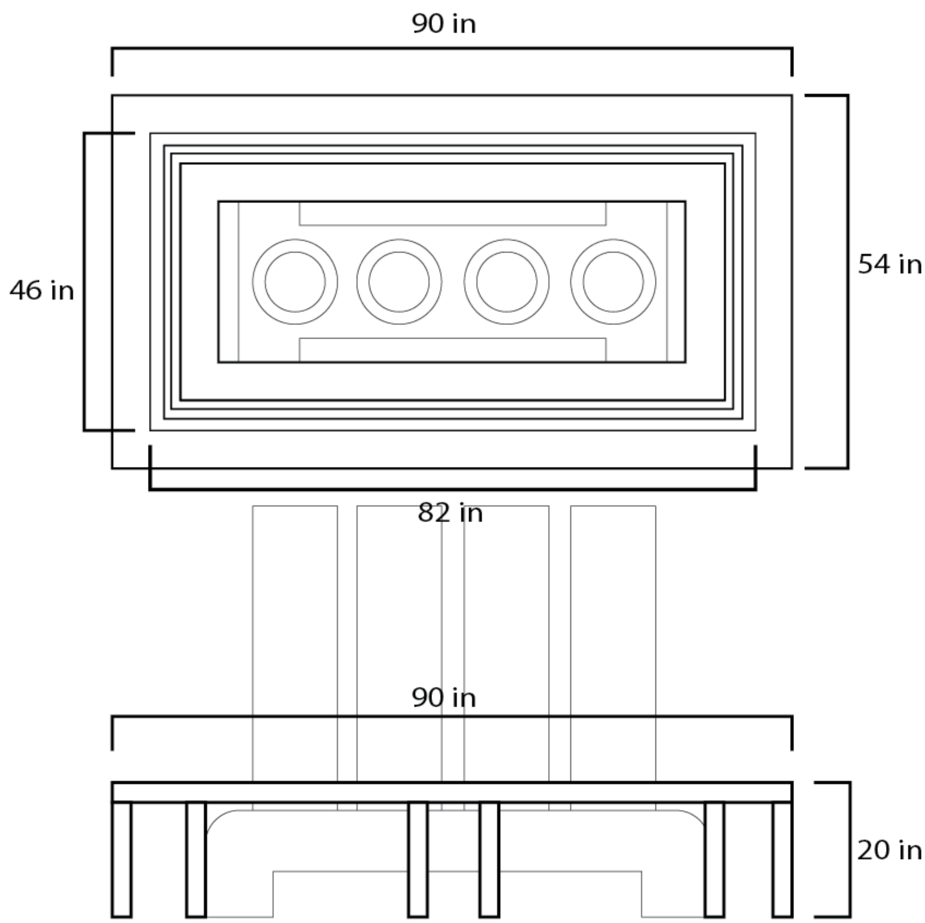


Central Wheel attached viewed inside the cart

F. Ramp Elevation

a. Design and Dimension of Ramp Elevation

In order for the Harvesting unit be able to move the inclusion of an elevated platform is necessary as a means of flat surface for the cart's wheels. After precisely measuring the aeroponic towers and their base pipes, the proponents were able to design and have precise dimensions for the ramp elevation.



Ramp Elevation Design and Exact Dimensions

b. Building of Ramp Elevation

After designing and measuring the ramp elevation, the proponents proceeded to plan for the needed materials in order to build it. The materials used for the ramps are: 2x2 cocolumber for the leg supports, 3 ply Plywood for the surface of the ramp, and nails as fasteners.

SOFTWARE DUPLICATION

1. Google Cloud Platform APIs

The development of the Web Application is largely dependent on the use of different Google Cloud Platform APIs. This includes Google Drive API, Cloud SQL and Google Collab API integration.

2. Front End Development

A. Layout Designing:

8. A wireframe was created to plan the overall structure and layout of the web page.
9. Using editing softwares such as Adobe Photoshop and Illustrator, design elements suitable for the web page’s style were created.

B. HTML Markup:

5. HTML tags were utilized to define the structure and content of the web page.
6. The layout was divided into sections using appropriate tags such as ‘<header>’, ‘<main>’, and ‘<footer>’.
7. With each section, content was arranged using the aforementioned tags.

C. Integration with Cloud SQL:

A. Using appropriate credentials and configurations the Cloud SQL service was set up, and a database was created to store data.
B. SQL queries were executed to retrieve, insert, updatem and delete data from the database.
C. The retrieved data was dynamically displayed on the web page using HTML and JavaScript.

3. Back End Development

D. Integration with Google Drive API:

8. Using appropriate API key and configurations Google Cloud API service was set up, which will be used as a temporary buffer storage where the Cloud SQL can retrieve its data from.

E. Integration with Cloud SQL:

9. Using appropriate credentials and configurations the Cloud SQL service was set up, and a database was created to store data.
10. SQL queries were executed to retrieve, insert, updatem and delete data from the database.
11. The retrieved data was dynamically displayed on the web page using HTML and JavaScript.

4. Camera Vision

A. Training Images

1. Set up Google Colab:

   - GPU acceleration was enabled by going to "Runtime" > "Change runtime type" and selecting "GPU" as the hardware accelerator.

2. Cloned the Darknet repository:

   - Ran the command !git clone https://github.com/AlexeyAB/darknet.git in a code cell to clone the Darknet repository, which contains the YOLO framework.

3. Darknet configuration:

   - The darknet folder in the cloned repository was accessed, and the Makefile was edited to enable GPU and OpenCV support. Darknet was then compiled with !make.

4. Dataset and configuration files:

   - The custom dataset was uploaded, adhering to the required structure with annotated images and labels. A configuration file was created, specifying the dataset paths, model settings, training parameters, and class names.

5. Pre-trained weights download:

   - The pre-trained weights for the YOLOv4 convolutional layers were downloaded using !wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137.

6. Training initiation:

   - YOLOv4 training commenced with the command !./darknet detector train path_to_config_file path_to_weights_file -dont_show, where path_to_config_file and path_to_weights_file were replaced with the actual paths to the configuration file and pre-trained weights.

7. Training progress monitoring:

   - Progress updates, including loss, learning rate, and elapsed time per iteration, were monitored during training. Darknet automatically saved the model weights at specified intervals in the configuration file.

8. Model evaluation:

   - Following training completion, the model's performance was evaluated using validation or test images. The darknet detector test command facilitated object detection on images or videos.

4. Robotic Arm Movement

A. GRBLGRU

1. The software uses G-code, which is the most widely used 3D printing programming language mainly for its capability to provide metric-based computer numerical control which is a much more simplified way of implementing arm movement rather than using inverse kinematics.
2. G-code implementation was further compressed into the raspberry pi environment via g-code parser to be interpreted by the Arduino software. Individual joint movements were programmed and recorded through GrblGru's software, which will all be compiled and used for arm movement automation that will respond to the coordinate mapping output of the vision system.

B. Vision to Arm Movement Integration

1. Using Arduino IDE, the programmed arm movements was paired up with individual coordinate that will represent the location of strawberry fruits in a captured frame by the Camera of the Raspberry Pi.

ANNEX D

Project Documentation

Figure E.1. Topic Defense

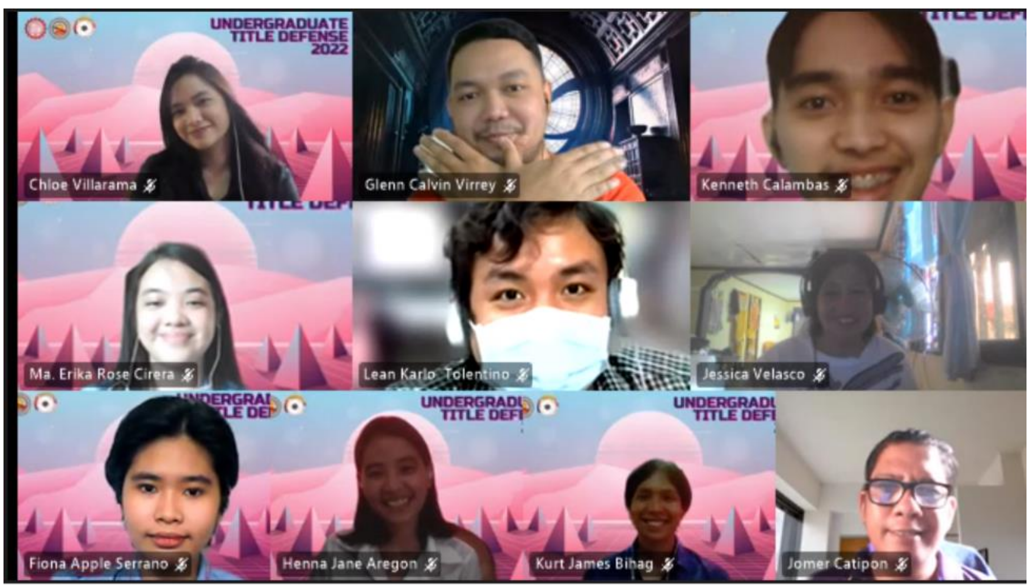


Figure E.2. Title Defense

Figure E.3. Progress Presentation with Thesis Adviser

Figure E.4. Progress Presentation with Panel Members

Figure E.5. Pre-Final Defense with Thesis Adviser

Figure E.6. Pre-Final Defense with the Partnered group and Panel Members

Figure E.7. Final Defense with the Panel Members

Figure E.8. Final Defense with the Partnered group and Panel Members

Figure E.9. APPRECIATE with Thesis Adviser

Figure E.10. APPRECIATE Booth

Figure E.11. 3D Printed Gears and Actuators

Figure E.12. (a)Built Arm and (b)First gripper Design

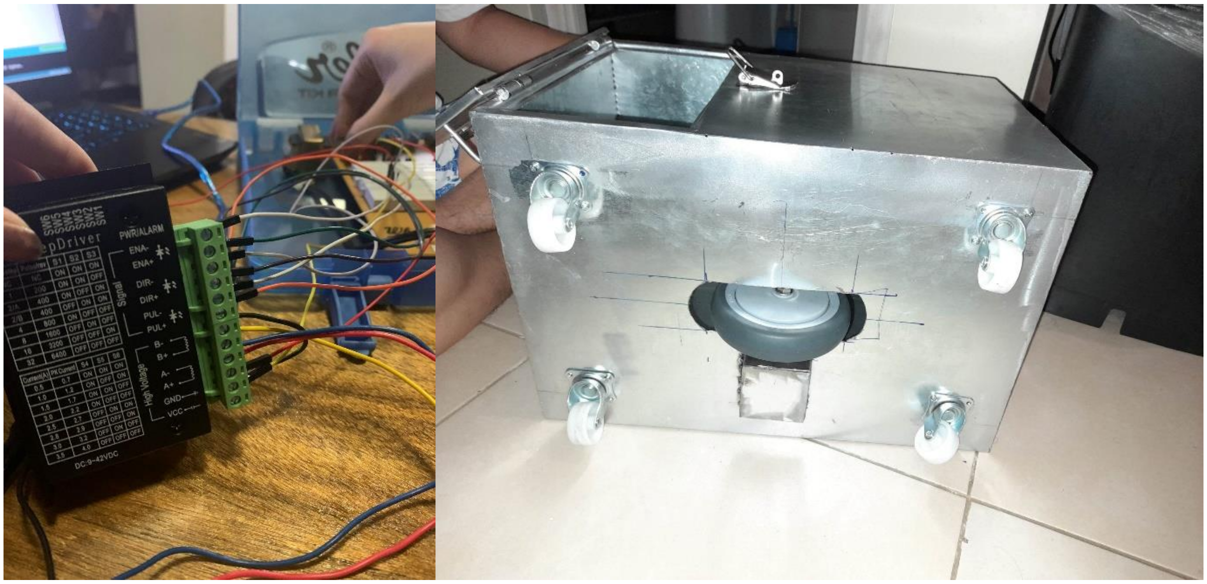


Figure E.13. (a)Wiring Connections of the Cart and (b)Bottom part of the Cart

Figure E.14. (a)Aeroponic Towers with Platform and (b)Aeroponics Tower with the Harvester

Figure E.15. Visit of the Thesis Advisers in the Deployment Place

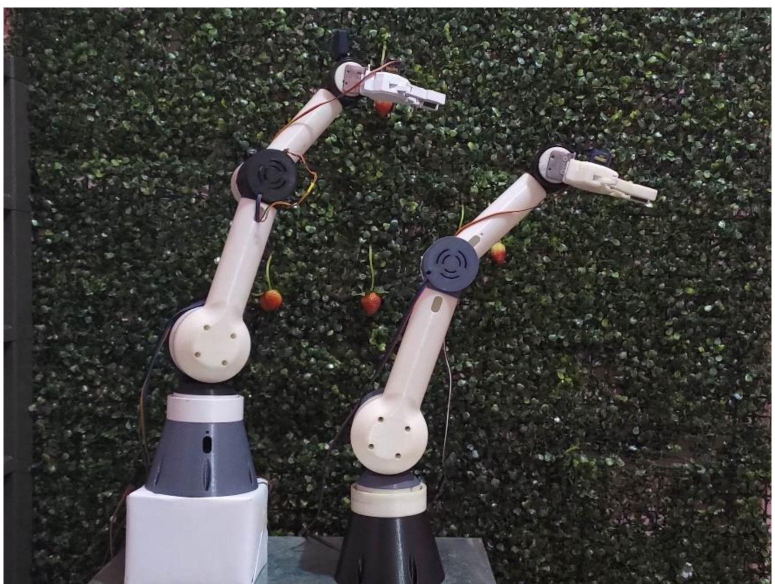


Figure E.16. Simulation

ANNEX E

User Manual

SAFS: A Railed Automobile Strawberry Harvester using Dual-Arm Manipulator via YOLOv4 Algorithm for Detection and Navigation with Cloud-based Webpage for Monitoring

SAFS

Strawberry Automated Farming System

USER MANUAL

Version 1.0

July 2023

INTRODUCTION

This manual provides detailed instructions and information to help you update the unit's central control device, access the web application for monitoring the harvester's status and connecting the harvester to the central control unit.

If you have any questions and concerns not explained in this user manual, please contact the Lead Unit Program Developer at 0927-772-1957or email us at hex0607@gmail.com.

SAFS

Strawberry Automated Farming System

OVERVIEW

The SAFS Web Application is an online website that supports the SAFS Harvester hardware unit. The website consists of a landing page, and monitoring dashboard.

1. **Landing Page** – Shows front page of the website and information about the project. It includes 4 clickable tabs: Home, About Us, Our Devs and Harvester.

2. **Dashboard** – Shows the harvester unit's current statuses. Here the battery percentage, harvested strawberry count, Machine Run time, and Harvesting status is showed in grid form.

SAFS
Strawberry Automated Farming System

Using the SAFS Harvesting Unit:

1. Turn on the unit by locating the power button inside the harvester's cart. Pushing the button will boot up the control unit. If it doesn't turn on, check the power connection on the AC port of the control unit.
2. Make sure to connect the unit to the internet to regularly update the unit's operating system (OS) and libraries. To manually update the system, use your preferred SSH Client to access the control unit's console remotely.
3. Place the Harvesting unit in front of your crops to begin image scan and harvesting. (Remember: The unit will only move linear to your crops' alignment if no rails are present in your set-up.)

Figure 1: Setting up the Harvesting Unit in front of the crops

GETTING STARTED

Using the SAFS Web Application:

1. On your browser, go to www.safs-app.xyz/<your_ip_address>. Make sure that the unit is already connected to your internet connection and you know its IP address. This will bring you to the SAFS Home Landing Page. Here you can click on *About Us* and *Our Devs* for more information about the project.

Figure 2: SAFS Home Landing Page

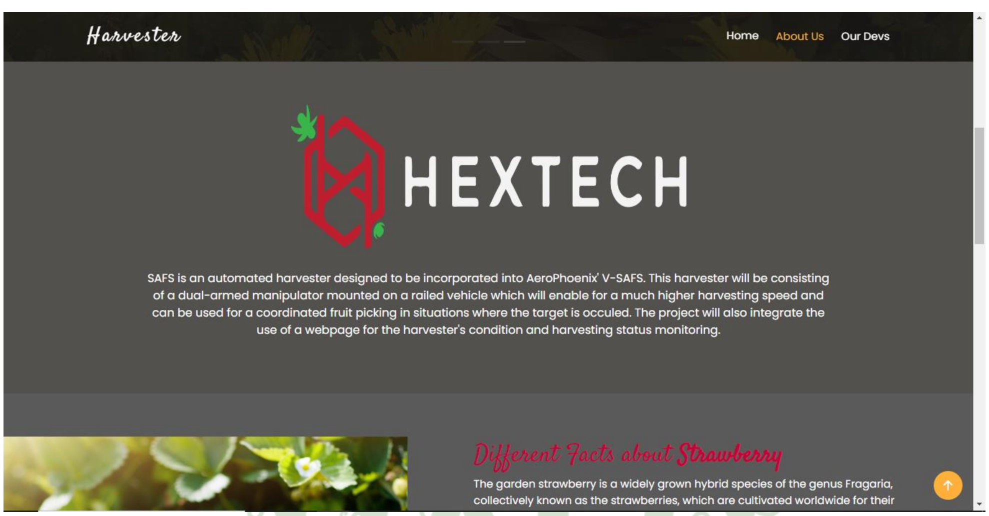


Figure 3: SAFS About Us Page

2. Click on *Harvester* to access your monitoring dashboard. Here you will be able to locate the monitoring grid boxes of you harvesting unit. The first grid indicates your unit's current battery percentage. The second and third box indicates the current count of strawberries harvested, one counted using PIR sensor and the other with ultrasonic sensor.

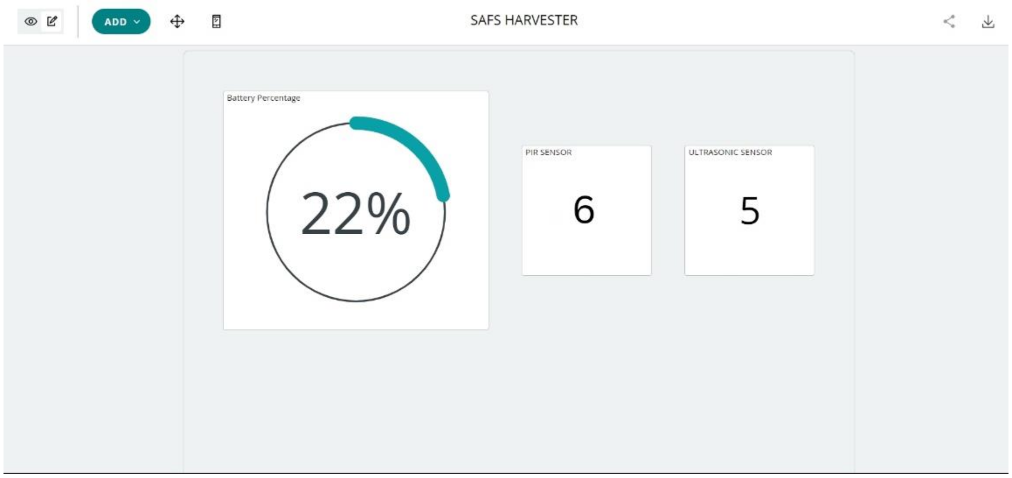


Figure 4: Monitoring Dashboard

SUPPORT

Contact	Email	Role
Fiona Apple H. Serrano	fionaapple.serrano@tup.edu.ph	Project Lead, Vision System Developer, Unit Program Developer
Henna Jane N. Aregon	hennajane.aregon@tup.edu.ph	Lead Hardware Developer
Kurt James B. Bihag	kurtjames.bihag@tup.edu.ph	Lead Hardware Developer
Kenneth V. Calambas	kenneth.calambas@tup.edu.ph	Lead Unit Program Developer, Hardware Developer
Ma.Erika Rose M. Cirera	maerikarose.cirera@tup.edu.ph	UI Developer, Hardware Developer
Chloe V. Villarama	chloe.villarama@tup.edu.ph	Hardware Developer, UI Developer

Table 1: Contact List of Developers

ANNEX F

Student Profile

Henna Jane N. Aregon

531 Purok 2 Sucat, Muntinlupa City

0998-3755-104 l hennajane.aregon@tup.edu.ph

Educational Background

2019-2023 Technological University of the Philippines

2017-2019 Technological Institue of the Philippines

2011-2017 Muntinlupa Business High School Sucat Annex

2007-2011 Sucat Elementary

Personal Background

Age: 22

Gender: Female

Date of Birth: July 18, 2000

Place of Birth: Muntinlupa, NCR

Citizenship: Filipino

Civil Status: Single

Religion: Roman Catholic

Academic Performance

- With Honors (Senior High School)
- With Honors (Junior High School)

Certifications

- **Professional Civil Service Exam Passer**
- **MNET** IP Addressing and Subnetting for CCNA

Skills

- Cadence Virtuoso
- NI MULTISIM
- Packet Tracer
- Microsoft Office
- HTML/CSS Programming
- C++ Programming
- IP addressing and Subnetting
- Basic Python Programming
- Cisco Basic Routing Technique

Kurt James B. Bihag

Blk 13 Lot 32 Camella Colina San Pedro, Laguna

0929-8015-373 l kurtjames.bihag@tup.edu.ph

Educational Background

2019-2023 Technological University of the Philippines

2017-2019 Lyceum of Alabang

2011-2017 San Roque Catholic School

2007-2011 Maquiling School Inc.

Personal Background

Age: 22

Gender: Male

Date of Birth: November 11, 2000

Place of Birth: Muntinlupa, NCR

Citizenship: Filipino

Civil Status: Single

Religion: Roman Catholic

Academic Performance

- With High Honors (Senior High School)
- 8th Honorable Mention (Grade 6)

Certifications

- **Professional Civil Service Exam Passer**
- **MNET** IP Addressing and Subnetting for CCNA

Skills

- Schematic Designing and Layout using Cadence Virtuoso
- NI MULTISIM
- Basic Programming using Quartus and Questa
- Basic Python Programming
- Basic System Verilog Programming
- Packet Tracer
- Microsoft Office
- HTML/CSS Programming
- C++ Programming
- IP addressing and Subnetting
- Cisco Basic Routing Technique

Kenneth V. Calambas

Blk 15 Lot 8 Castle Spring Heights Subd. Camarin Caloocan City

09610125412 l kenneth.calambas@tup.edu.ph

Educational Background

2019-2023 Technological University of the Philippines

2017-2019 South-East Asia Institute of Trade of Technology

2013-2017 Cielito Zamora Memorial High School

2007-2013 Cielito Zamora Memorial Elementary School

Personal Background

Age: 23

Gender: Male

Date of Birth: December 4, 1999

Place of Birth: Caloocan City

Citizenship: Filipino

Civil Status: Single

Religion: Roman Catholic

Academic Performance

- 3rd Honorable mention (Grade 11)
- 3rd Honorable mention (Grade 12)
- With Honors (Junior High School)
- School Varsity (Junior High School & Senior High School)

- 4th Honorable mention (Grade 6)

Certifications

- **Next-Generation Qualification:** The Role of Physics of Failure and Artificial Intelligence in Electronics
- **Unleashing the Power of Data:** SAP Analytics Cloud Workshop
- **STEP UP:** Paradigmatic Career in Today's Semiconductor Industry
- Accessing New Avenues: Exploring the Role of Open RAN in Today's Communication Technology
- **The Data Revolution:** Navigating the Importance of Data Science in Today's World
- **App Dev Insider:** Behind the Scenes of Application Development using Flutter
- **Connecting the Unconnected:** Exploring the Potential of LoRaWAN in Revolutionazing IoT Connectivity for a Smarter World
- **Unlocking the Limitless:** Getting Cloud-Powered with Google Cloud Platform
- **Hack-Proof Your Digital Life:** A Beginner's Guide to Information Security
- **Building Blocks of IC Design:** A Primer for Graduating Electronics Engineering Students

Skills

- Communication Skills: Proficient in Filipino and English, Partial Proficient/Knowledgeable in Japanese, Spanish, and French.
- Computer Literate
- Basic knowledge on
  - PCB Design Layout and Schematic
  - Python
  - MATLAB
  - Arduino
  - AutoCAD
  - NI Multism
  - Soldering
  - Raspberry Pi

Ma.Erika Rose M. Cirera

0789 Purok 5B Brgy. Bagong Pook, Pila, Laguna

0995-2948-757 I maerikarose.cirera@tup.edu.ph

Educational Background

2019-2023 Technological University of the Philippines

2011-2019 San Antonio de Padua College

2007-2011 Maquiling School Inc.

Personal Background

Age: 22

Gender: Female

Date of Birth: February 27, 2001

Place of Birth: Sta. Cruz, Laguna

Citizenship: Filipino

Civil Status: Single

Religion: Roman Catholic

Academic Performance

- With Honors (Senior High School)
- With Honors (Junior High School)
- Salutatorian (Grade 6)

Certifications

- **MNET** IP Addressing and Subnetting for CCNA
- **FORTINET** NSE 3 Network Security Associate 2023
- **FORTINET** NSE 2 Network Security Associate 2023
- **FORTINET** NSE 1 Network Security Associate 2023

Skills

- IP addressing and Subnetting
- Basic Python Programming
- Cisco Basic Routing Technique
- NI MULTISIM
- Packet Tracer
- Microsoft Office
- HTML/CSS Programming
- C++ Programming

Fiona Apple H. Serrano

646 Aranga Street, Sampaloc, Manila, Metro Manila

0929-221-4288 | fionaapple.serrano@tup.edu.ph

Educational Background

2019-2023 Technological University of the Philippines

2017-2019 The National Teachers College

2013-2017 Ramon Magsaysay High School

2006-2013 Juan Luna Elementary School

Personal Background

Age: 22

Gender: Female

Date of Birth: May 15, 2001

Place of Birth: Sampaloc, Manila

Citizenship: Filipino

Civil Status: Single

Religion: Christian

Academic Performance

- Creative Director for Trendsponder TUP-Manila
- Executive Associate for Publications – Institute of Electrical and Electronics Engineers – Manila (2021-2022)

- With Honors (Senior High School)
- Academic Achiever (Elementary)

Certifications

- **MNET** IP Addressing and Subnetting for CCNA
- **FORTINET** NSE 3 Network Security Associate 2023
- **FORTINET** NSE 2 Network Security Associate 2023
- **FORTINET** NSE 1 Network Security Associate 2023

Skills

- Language and Web Development: Basic Programming Techniques in MATLAB, Linux, Python, HTML
- Tools/ Software: Arduino, Adobe Editing Applications, Microsoft Office Applications, NI, Multisim, Proteus, Octave, Autodesk Fusion 360, Autodesk AutoCAD, Cisco Packet Tracer, GNS3, Google Cloud Platform
- Fast Learner
- Good at Multitasking
- Detail-oriented
- Works well under pressure

Chloe V. Villarama

1116 Torres Bugallon St., Tondo, Manila

09062273423 l chloe.villarama@tup.edu.ph

Educational Background

2019-2023 Technological University of the Philippines – Manila (College)

2017-2019 Colegio de San Juan de Letran – Manila (Senior High School)

2013-2017 St. John Academy of Manila Foundation Inc.

Personal Background

Age: 21

Gender: Female

Date of Birth: August 20, 2001

Place of Birth: Manila

Citizenship: Filipino

Civil Status: Single

Religion: Roman Catholic

Academic Performance

- Ambassadress for Sustainable Development Institute of Electronics Engineer of the Philippines (IECEP) Manila Student Chapter
- With Honors (Senior High School)
- Valedictorian (Junior High School)

Certifications

- **MNET** IP Addressing and Subnetting for CCNA
- **FORTINET** NSE 3 Network Security Associate 2023
- **FORTINET** NSE 2 Network Security Associate 2023
- **FORTINET** NSE 1 Network Security Associate 2023

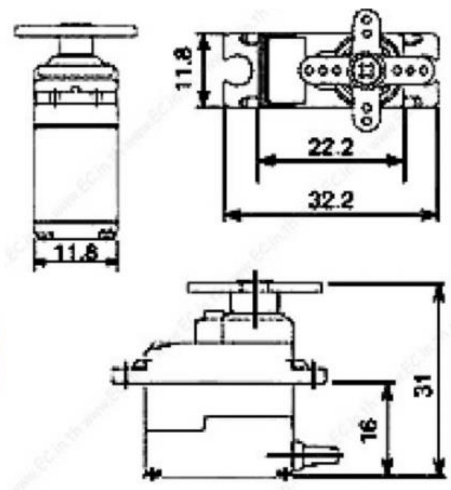
Skills

- Language and Web Development: Basic Programming Techniques in MATLAB, Python, Arduino, HTML, C++ Programming
- Cisco Routing Technique
- Tools/ Software: Sketchup, Adobe Photoshop, Microsoft Excel
- Strong Multitasking Ability
- Strong Communication

ANNEX G

Specifications

SG90 Micro Servo



Specifications
• Weight: 9 g
• Dimension: 22.2 x 11.8 x 31 mm approx.
• Stall torque: 1.8 kgf·cm
• Operating speed: 0.1 s/60 degree
• Operating voltage: 4.8 V (~5V)
• Dead band width: 10 µs
• Temperature range: 0 ºC – 55 ºC
Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is all the way to the left. ms pulse) is all the way to the right, ""-90" (~1ms pulse) is all the way to the left.

Width: 0.48 in (12.2 mm)

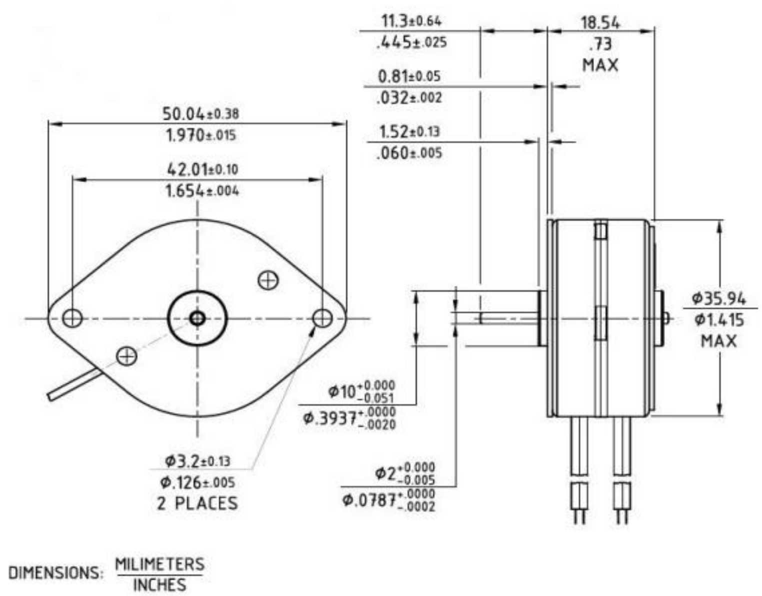
Height:1.14 in (29.0 mm)

Motor Type: 3-pole
Gear Type: Plastic
Rotation/Support: Bushing

Additional Specifications

Rotational Range: 180°
Pulse Cycle: ca. 20 ms
Pulse Width: 500-2400 µs

35mm Micro-Stepper



Electrical Data	35M048B1U Unipolar	35M048B2U Unipolar	35M048B1B Bipolar	35M048B2B Bipolar	
1 Operating Voltage	5	12	5	12	VDC
2 Resistance per Phase, ± 10%	12.5	72.0	12.5	72.0	Ohms
3 Inductance per Phase, typ	7.8	36.0	16.4	86.0	mH
4 Rated Current per Phase *	0.40	0.17	0.40	0.17	A
Coil independent parameters					
5 Holding Torque, MIN *	18.35 (2.6)	18.35 (2.6)	19.76 (2.8)	19.76 (2.8)	mNm (oz-in)
6 Detent Torque, Max	2.12 (0.3)	2.12 (0.3)	2.12 (0.3)	2.12 (0.3)	mNm (oz-in)
7 Rotor inertia	2 (0.011)	2 (0.011)	2 (0.011)	2 (0.011)	(gcm^2) (oz-in-s^2)
8 Step Angle	7.5	7.5	7.5	7.5	Degree
9 Absolute accuracy 2 ph. On, Full step	± .5	± .5	± .5	± .5	Degree
10 Steps per Revolution	48	48	48	48	
11 Ambient Temp Range (operating)	-20 to +70 (-4 to +158)	-20 to +70 (-4 to +158)	-20 to +70 (-4 to +158)	-20 to +70 (-4 to +158)	°C (°F)
12 Maximum Coil Temperature	130 (266)	130 (266)	130 (266)	130 (266)	°C (°F)
13 Bearing Type	Sintered Bronze Sleeve	Sintered Bronze Sleeve	Sintered Bronze Sleeve	Sintered Bronze Sleeve	
14 Insulation Resistance at 500 VDC	100	100	100	100	Mohms
15 Dielectric Withstanding Voltage	650 for 2 seconds	650 for 2 seconds	650 for 2 seconds	650 for 2 seconds	VAC
16 Weight	88 (3.1)	88 (3.1)	88 (3.1)	88 (3.1)	g (oz)
17 Leadwire	AWG 26, UL 1430	AWG 26, UL 1430	AWG 26, UL 1430	AWG 26, UL 1430	

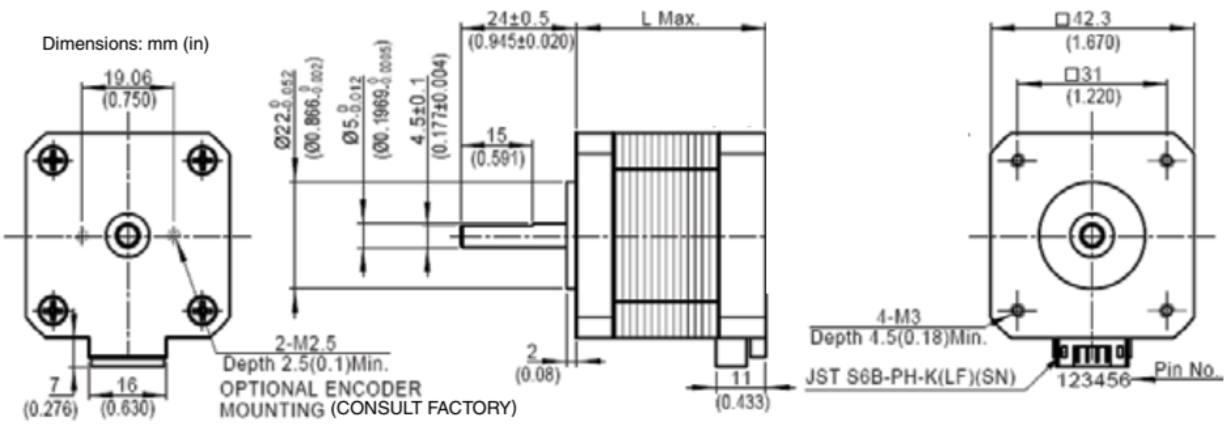
All Motor Data Values at 20°C Unless Otherwise Specified

* Energize at Rated Current, 2 Phase On

Nema17 Stepper Motor

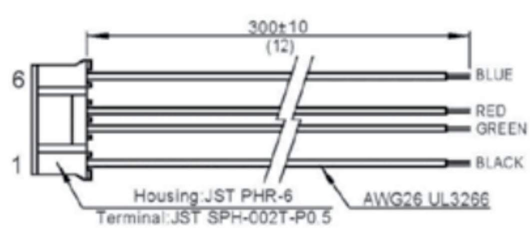
Phases	2
Steps/Revolution	200
Step Accuracy	±5%
Shaft Load	20,000 Hours at 1000 RPM
Axial	25 N (5.6 lbs.) Push
	65 N (15 lbs.) Pull
Radial	29 N (6.5 lbs.) At Flat Center
IP Rating	40
Approvals	RoHS
Operating Temp	-20° C to +40° C
Insulation Class	B, 130° C
Insulation Resistance	100 MegOhms

Description	**Length**	**Mounted Rated Current**	**Mounted Holding Torque**		**Winding Ohms**	**mH**	**Detent Torque**		**Rotor Inertia**		**Motor Weight**	
(Stack)	"L" Max	Amps	Nm Typ.	oz-in Typ.	±10% @ 20°C	Typ.	mNm	oz-in	g cm2	oz-in2	kg	lbs
Single	39.8 mm (1.57 in)	2	0.48	68	1.04	2.2	15	2.1	57	0.31	0.28	0.62
Double	48.3 mm (1.90 in)	2	0.63	89	1.3	2.9	25	3.5	82	0.45	0.36	0.79
Triple	62.8 mm (2.47 in)	2	0.83	120	1.49	3.8	30	4.2	123	0.67	0.6	1.3



Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.

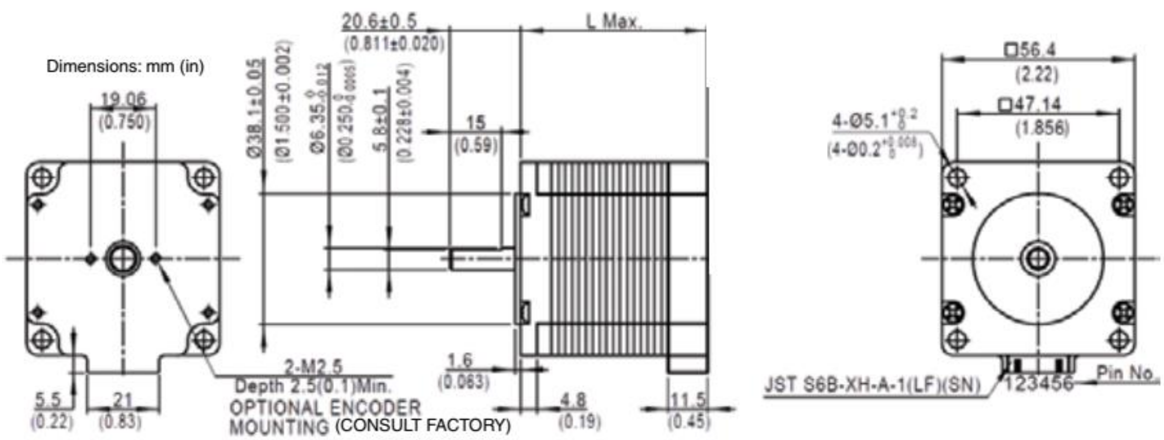
Dimensions: mm (in)
4 Lead Connector, PBC Part#6200490
(Consult factory for optional motor connectors)



Nema23 Stepper Motor

Phases	2
Steps/Revolution	200
Step Accuracy	±5%
Shaft Load	20,000 Hours at 1000 RPM
Axial	40 N (9 lbs.) Push 130 N (30 lbs.) Pull
Radial	70 N (15.5 lbs.) At Flat Center
IP Rating	40
Approvals	RoHS
Operating Temp	-20° C to +40° C
Insulation Class	B, 130° C
Insulation Resistance	100 MegOhms

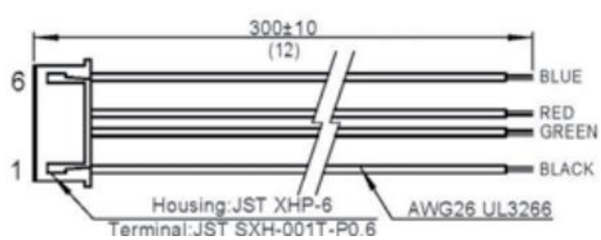
Description	**Length**	**Mounted Rated Current**	**Mounted Holding Torque**		**Winding Ohms mH**		**Detent Torque**		**Rotor Inertia**		**Motor Weight**	
(Stack)	"L" Max	Amps	Nm Typ.	oz-in Typ.	±10% @ 20°C	Typ.	mNm	oz-in	g cm2	oz-in2	kg	lbs
Single	55.0 mm (2.17 in)	2.2	1.50	210	1.6	6.9	45	6.4	220	1.2	0.6	1.3
Double	77.0 mm (3.03 in)	3	2.30	330	1.1	4.5	75	11	390	2.1	1	2.2
Power Plus (Triple)	77.0 mm (3.03 in)	3	3.30	470	1.1	3.7	150	21	390	2.1	1.1	2.4



Motor with leads: Lead wire is 22 AWG UL3266, 300 ±10 (12 ±.5) long

Standard shaft dimensions shown. All other dimensions apply to hollow and extended shaft options.

Dimensions: mm (in)
4 Lead Connector, PBC Part#6200491
(Consult factory for optional motor connectors)



A4988 Stepper Motor Driver

ELECTRICAL CHARACTERIST d)

Characteristics	Symbol	Test Conditions	Min.	Typ.[2]	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R_{DSON}	Source Driver, I_{OUT} = –1.5 A	–	320	430	mΩ
		Sink Driver, I_{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V_F	Source Diode, I_F = –1.5 A	–	–	1.2	V
		Sink Diode, I_F = 1.5 A	–	–	1.2	V
Motor Supply Current	I_{BB}	f_{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I_{DD}	f_{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		V_{DD}×0.7	–	–	V
	$V_{IN(0)}$		–	–	V_{DD}×0.3	V
Logic Input Current	$I_{IN(1)}$	V_{IN} = V_{DD}×0.7	–20	<1.0	20	μA
	$I_{IN(0)}$	V_{IN} = V_{DD}×0.3	–20	<1.0	20	μA
Microstep Select	R_{MS1}	MS1 pin	–	100	–	kΩ
	R_{MS2}	MS2 pin	–	50	–	kΩ
	R_{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	μs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	μs
		R_{OSC} = 25 kΩ	23	30	37	μs

TB6600 Stepper Motor Driver

TECHNICAL DATA

Supply voltage 12 ~ 48VDC

Input current of 1 to 5A depending on the selected

Output current settings.

Stepper Motor output current of 0.2A ~ 5A

Operating Temperature -10 to 45 ℃;

Storage temperature -40 ℃ to 70 ℃

Weight 230 grams

CONTROL SIGNAL INTERFACE

1- Control signals description:

PUL + : step pulse signal positive input;

PUL - : step pulse signal negative input;

DIR + : stepping direction signal positive input;

DIR - : stepping direction signal negative input;

EN + : offline enable signal positive input;

EN - : offline enable signal negative input;

Ramps 1.4 Arduino Extend Shield

Specifications:

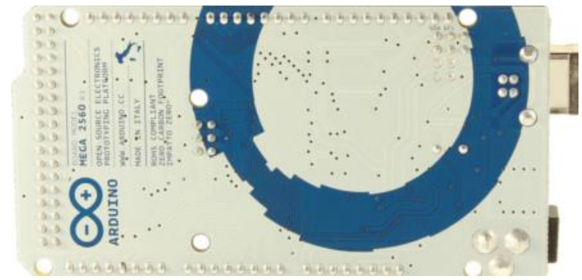
1. Supply Voltage: 12V
2. Standard interfaces (as that of the extruder)
3. Reserved GCI like I2C and RS232
4. 3 MOSFET's are applied to the heater/ fan and thermistor circuit.
5. Another 5A added to protect the component parts.
6. An 11A fuse is added to the hotbed.
7. Support 5 stepper drive board

Arduino Uno

Specifications:

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm

Arduino Mega



Specifications:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

ESP8266 WiFi Module

Specifications:

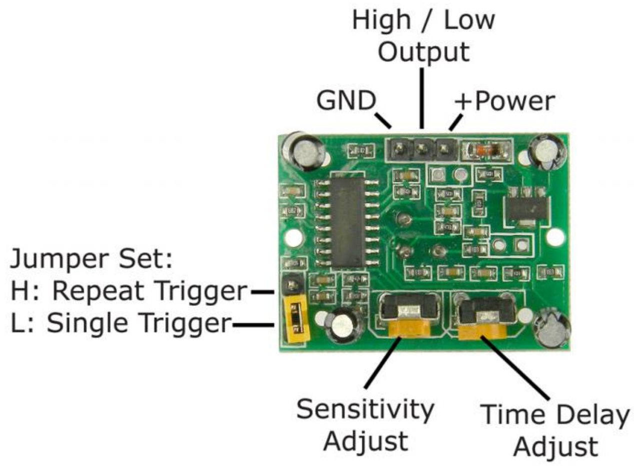
Categories	Items	Parameters
Wi-Fi	Certification	Wi-Fi Alliance
	Protocols	802.11 b/g/n (HT20)
	Frequency Range	2.4 GHz ~ 2.5 GHz (2400 MHz ~ 2483.5 MHz)
	TX Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: –91 dbm (11 Mbps)
		802.11 g: –75 dbm (54 Mbps)
		802.11 n: –72 dbm (MCS7)
	Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit processor
	Peripheral Interface	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/ADC/PWM/LED Light & Button
	Operating Voltage	2.5 V ~ 3.6 V
	Operating Current	Average value: 80 mA
	Operating Temperature Range	–40 °C ~ 125 °C
	Package Size	QFN32-pin (5 mm x 5 mm)
	External Interface	-
Software	Wi-Fi Mode	Station/SoftAP/SoftAP+Station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / Firmware and SDK for fast on-chip programming
	Network Protocols	IPv4, TCP/UDP/HTTP
	User Configuration	AT Instruction Set, Cloud Server, Android/iOS App

Ultrasonic Ranging Module

Specifications:

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
Measuring Angle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

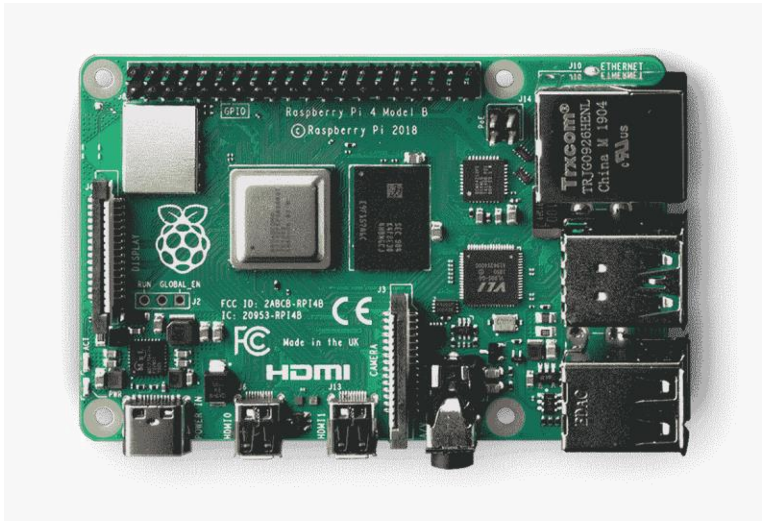
PIR Motion Detector



Specifications:

- Voltage: 5V – 20V
- Power Consumption: 65mA
- TTL output: 3.3V, 0V
- Delay time: Adjustable (.3->5min)
- Lock time: 0.2 sec
- Trigger methods: L – disable repeat trigger, H enable repeat trigger
- Sensing range: less than 120 degree, within 7 meters
- Temperature: – 15 ~ +70
- Dimension: 32*24 mm, distance between screw 28mm, M2, Lens dimension in diameter: 23mm

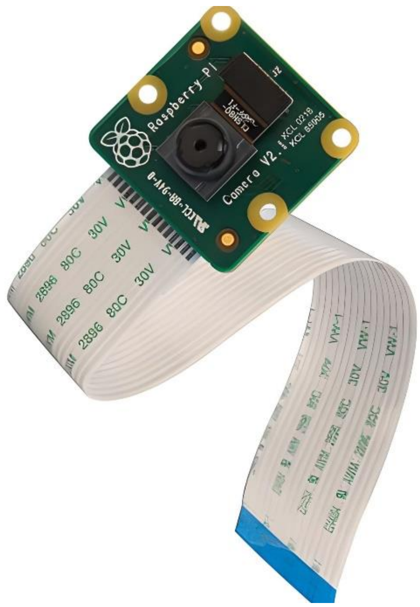
Raspberry Pi 4 Model B



Specifications:

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- 2 × micro-HDMI® ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient

Raspberry Pi 4 Model B



Specifications:

- Official Raspberry Pi Camera Board, supports Raspberry Pi, CM3/3+/4, Jetson Nano, Jetson Xavier NX
- IMX219 8-megapixel sensor
- Camera specifications
    - CCD size : 1/4inch
    - Aperture (F) : 2.0
    - Focal Length : 3.04mm
    - Angle of View (diagonal) : 62.2 degree
- 3280 × 2464 still picture resolution
- Support 1080p30, 720p60 and 640x480p90 video record
- Dimension: 25mm × 24mm × 9mm

ANNEX H

Gantt Chart

Objectives	Activities	2022						2023					
		July	August	Septembe r	October	Novembe r	Decembe r	January	February	March	April	May	June
To construct a harvesting vehicle with a dual-arm manipulator harvesting robot with an end-effector device that is controlled by the received data and harvesting position via Arduino, moving in a railway developed in the smart aeroponics farm.	Canvasing and Purchasing of Equipment and Materials												
	Designing and Constructing Railing System and Mobile Platform												
	Designing and Assembling Robotic Harvesting Arm												
To develop a vision system that will detect, locate, and harvest the ripe strawberries using YOLOv4 algorithm.	Programming the movement of Manipulator and End-effector												
	Development of Vision System for Object Detection												
To develop and integrate a web page to the system that monitors and displays the harvesting parameters and the harvester's status through cloud database Arduino IoT Cloud.	Programming Backend and Frontend of Webpage												
	Integrating Cloud Database to the shared Webpage												
To evaluate the functionality, reliability, and efficiency of the automatic harvesting system through actual field tests.	Testing speed of fruit detection and rate of harvest												
	Assessing the gathered results and data												