Journal of Current Research in Engineering and Science
JCRES
Bi-Annual Online Journal (ISSN: 2581 - 611X)

VOLUME:8, ISSUE:1, FEBRUARY 2025, PAPER NUMBER:13

FRUIT HARVESTING ROBOT USING COMPUTER VISION AND ARTIFICIAL INTELLIGENCE

Reesman A
B.tech Artificial Intelligence & Data Science Sri sairam institute of technology

Yazhini R
B.tech Artificial Intelligence & Data Science Sri sairam institute of technology

Amudhini R
B.tech Artificial Intelligence & Data Science Sri sairam institute of technology

Mrs. Sathiya A
Assistant Professor
B.tech Artificial Intelligence & Data Science Sri sairam institute of technology

FRUIT HARVESTING ROBOT USING COMPUTER VISION AND ARTIFICIAL INTELLIGENCE

Abstract

Automatic fruit harvesting robots are Improve crop yields and quality, and provide a more designed to efficiently and effectively harvest fruit without damaging the trees or the fruit. These robots use various technologies such as machine vision, robotics, and AI to locate, identify, and pick the fruit accurately with least amount of damage. The goal of an automatic fruit harvesting robot project is to increase efficiency and reduce labor costs for farmers and also to reducing wastage and improve the quality of the fruit being harvested. Using various sensors as our input source, we first perform object detection to find the best fruit to harvest. Followed by we also try to analyze the fruit location, such that it helps in directing the robotic arm towards its target. The end effector is specially designed in such a way that it handles the fruit in its best way and then harvest the fruit by twisting the hand axis thus causing least damage and maximum productivity.

Keyword: machine vision, robotics, AI, object detection, robotic arm.

I. <u>Introduction</u>

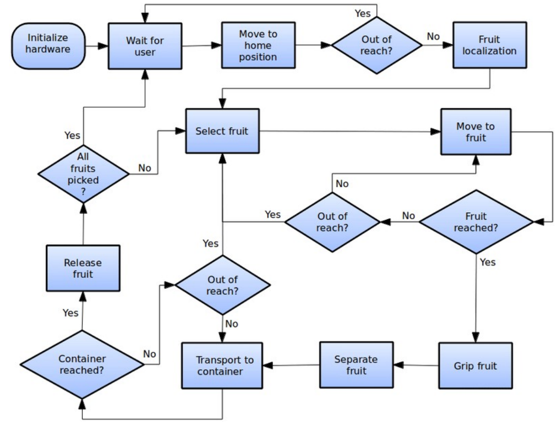
Farming is one of the huge domain sector that act as a huge contributor to our Indian economy. There are a lot of issues faced by these farmers and other commercial farming sectors. One such issue that exist up until today is harvesting with least amount of loss in crops.

Currently, fruit harvesting is a labor intensive process that requires more of time and effort. Manual fruit harvesting can lead to fruit damage, reduces the overall quality of the crop and increases waste. It can also be difficult for workers to reach fruit that is high up in the trees or located in hard-to-reach areas. Despite these challenges, there is significant potential for the use of automatic fruit harvesting robots in agriculture sustainable and efficient alternative to manual fruit harvesting.

By automating the fruit harvesting process, farmers can ensure that their fruit is harvested at the optimal time and with minimal damage, leading to improved crop yields and quality. It can help reduce waste by harvesting fruit that might be left on the tree or discarded due to damage caused by manual harvesting methods. By reducing the need for manual labor, they can also help reduce the carbon footprint of the agriculture industry. The impact of automatic fruit harvesting robots helps to increase efficiency, reduce costs, improve crop quality, and promote sustainability in the agriculture industry.

The ideal state for automatic fruit harvesting robots would be a widespread adoption of this technology by farmers around the world. These robots would be fully integrated into fruit harvesting operations and would be seen as a standard tool for fruit farmers. This would be highly efficient, accurate, and reliable, with the ability to pick fruit quickly and gently without causing damage. They would also be able to adapt to different types of fruit and different growing conditions, allowing them to be used across a wide range of crops and regions.

Possible solutions and innovations: Integrate AI and machine learning into automatic fruit harvesting robots can help them adapt to different types of fruit and environments, learn from experience, and make decisions on the fly. Develop lightweight and flexible robotic arms can improve the reach and mobility of automatic fruit harvesting robots, allowing them to pick fruit from hard-to-reach areas of the tree. To develop cloud-based platforms for data management that can help farmers and operators manage data from remote places using IOT technology. Thus allowing them to track yield, quality, and other metrics in real- time providing them with easy to read insights interface.



a. workflow of fruit harvesting robot

A. <u>Methodology:</u>

We are dividing the whole project into two main phases that is, **phase-1 Detection and Localization phase and Phase-2 Harvesting and Hardware phase.** Detection and localization phase is fully based on computer vision and open cv where the code is written to detect the fruit of interest in real time. Not just detecting the fruit it also classifies them if it is ready to harvest or not seeing its conditions such as, its ripen or damaged. If the robot detects best ready to harvest fruit it immediately detects the distance between the fruit and the robot arm, and give an accurate reading to reach out the fruit.

In harvesting and hardware phase where everything for mobility is connected to Arduino for control and movement and robotic arm is connected to raspberry pi. The camera module is also connected along with the raspberry pi using a pi cam. This does the real-time detection and localization and feeds the information to the arm to help in redirection towards the fruit ignoring all other foliage.

II. Modules

A. Camera and Sensors Module:

The camera and sensor module is pivotal component in the design of our automatic fruit harvesting robot, focusing on the installation and integration of various sensors and cameras to facilitate efficient fruit detection and localization.

i. Cameras:

Cameras are crucial part of the project as this is the one that decides which fruit is ready to harvest. The computer vision technology is applied best when the camera has the right input. We did trials using 3 different camera, web cam, raspberry pi cam, stereo vision cam. Web camera had low resolution and was hard to pass as input. Pi cam was easy integration but distance calculation was not accurate. Stereo vision cam was easy to integrate and gave accurate distance measurement but didn't fit our budget thus we finalized to work with pi cam.

b. Web cam, Stereo Vision cam, Pi cam

ii. Light Source:

To obtain good resolution picture natural light source was very crucial. during day time sun light was more than enough but during night there is no night vision camera is used in the prototype due to cost so we used proper light source to capture the image right.

iii. Distance Sensors:

To find the distance only for mobility module we used ultrasonic sensor to avoid any accidents during the process and to detect any obstacle before the robot.

c. Ultrasonic Sensor

Other criteria we set to detect the fruit is based on its color and texture that reflected through the camera. These listed cam and sensors are the important source used for detection and there are more sensors incorporate for future study as well such as moisture sensor and light sensors.

B. <u>Deep Learning Module:</u>

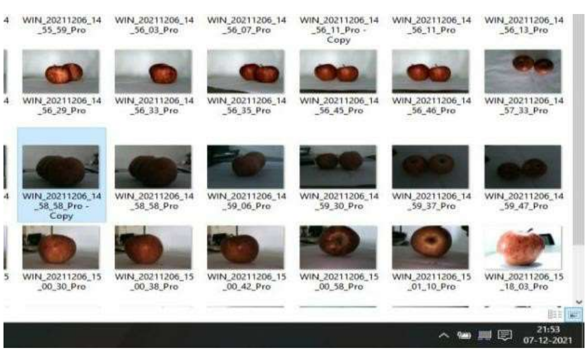
The Deep Learning Module serves as the cognitive powerhouse of our automatic fruit harvesting robot, employing advanced algorithms to recognize and classify fruits. Central to this module is the utilization of a deep learning model, a sophisticated artificial intelligence framework capable of learning intricate patterns and features from large datasets.

In our quest for efficient fruit harvesting, we employed a pre-trained YOLO version 4 model, a real-time object detection framework. To expedite the integration of a custom class for fruit identification, we followed a streamlined process,

1. modifying configuration files
2. creating a custom names file, and
3. updating training data.

i. Dataset Collection:

The dataset acquisition employed two distinct methods. Method 1 involved web scraping images of apples from various online sources, including Google and Kaggle datasets, encompassing diverse representations of both good and defective apples. Method 2 entailed capturing images of apples from different perspectives, ensuring a comprehensive dataset reflecting varied viewpoints. The amalgamation of these datasets resulted in a total of 600 images, comprising 300 depicting good apples and 300 illustrating defective apples, subsequently utilized for the segmentation and classification tasks.

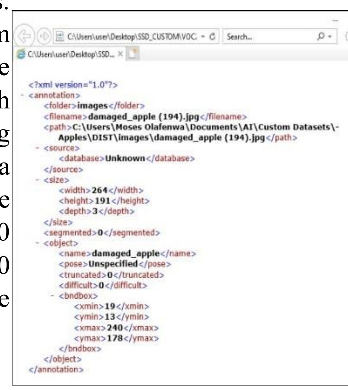


d. Captured Datasets from different Viewpoints

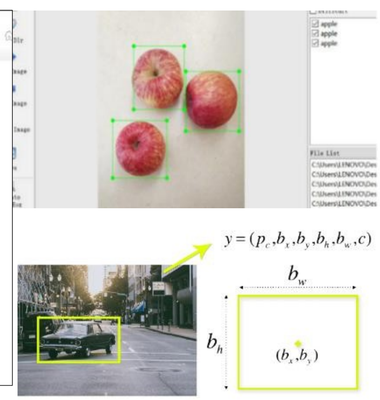
e. Google Datasets from kaggle

ii. Pre-processing the Datasets:

To train the YOLO algorithm, it requires input in the form of XML version annotations for bounding boxes. In our approach, each image in the dataset was meticulously annotated by drawing bounding boxes around the objects of interest, and corresponding XML files were generated to encapsulate this annotation data. These XML files, which include information about the location and class of objects within the images, serve as essential training input for YOLO.



```
<?xml version="1.0"?>
<annotation>
  <folder>images</folder>
  <filename>damaged_apple (194).jpg</filename>
  <path>C:\Users\Moses Olafenwa\Documents\AI\Custom Datasets\Apples\DIST\images\damaged_apple (194).jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>264</width>
    <height>191</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>damaged_apple</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>19</xmin>
      <ymin>13</ymin>
      <xmax>240</xmax>
      <ymax>178</ymax>
    </bndbox>
  </object>
</annotation>
```

f. Xml file of annotated image.

g. Annotation box over image.

iii. Modifying configuration files:

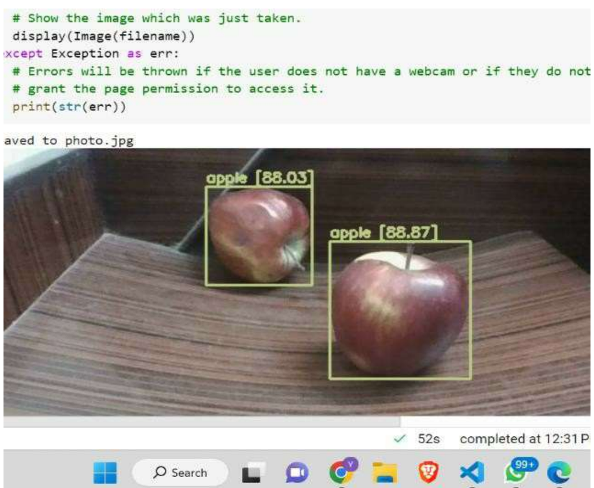
We adjust the YOLO configuration files (e.g., yolov4.cfg) to accommodate changes. Locate the appropriate section (e.g., [yolo]), and update parameters like the number of classes and filters based on the custom class. Ensure alignment with the chosen YOLO version (e.g., YOLOv4) for optimal performance.

iv. Creating custom names files:

Generate a custom names file listing all classes, including the new one, each on a separate line. Save this file in the same directory as the YOLO configuration file. This file serves to label and identify the classes during the training and testing phases.

v. Updating training data:

To pass this input to the YOLO algorithm during training, the path to the directory containing both the annotated images and their corresponding XML files needs to be specified in the YOLO training configuration file. The configuration file typically includes parameters for the location of training data, model architecture, and other training settings. By ensuring the correct path to the annotated data in the configuration file (e.g., train.txt), the YOLO algorithm can effectively utilize the annotated images and XML files for training purposes.



g. Object Detection using YOLO v4 algorithm

C. Robotic Arm Module:

The Robotic Arm Module is an integral component designed to facilitate precise and efficient fruit harvesting. The module features the installation of a robotic arm onto the harvesting robot, strategically positioned to reach and pluck fruits from the trees. The robotic arm's operation is orchestrated by an Arduino microcontroller, serving as the central control unit.

h. Reference robotic arm

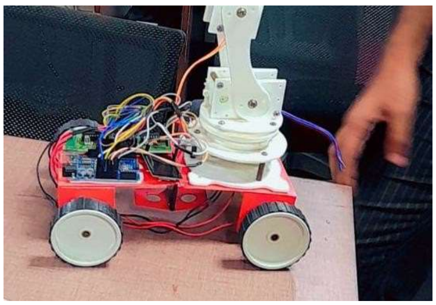
The Arduino executes a set of programmed instructions, dictating the movement, orientation, and gripping actions of the robotic arm. Sensors embedded in the robotic arm provide feedback to the Arduino, enabling real-time adjustments for optimal performance. The structure incorporates servo motors or actuators at each joint, ensuring fluid and accurate movements. This synergy between hardware and control algorithms ensures the robotic arm's dexterity, enhancing the overall precision and productivity of the fruit harvesting process.

i. Robotic Arm Prototype integrated with Arduino

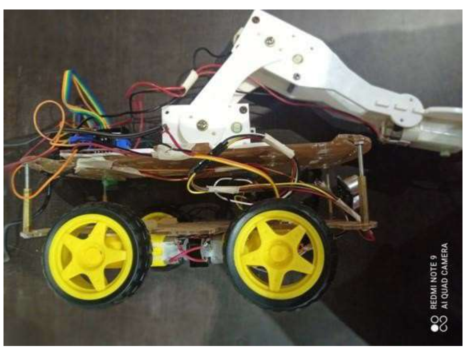
D. Mobility Module:

The Mobility Module is designed for seamless navigation between trees, incorporating an Obstacle Detection System for safe traversal. The system, linked to another Arduino, employs ultrasonic sensors, ensuring effective obstacle detection in farming fields with linear direction. Additionally, durability and weatherproofing measures are implemented to withstand diverse environmental conditions. This module not only facilitates smooth mobility but also integrates seamlessly with the robotic arm, providing a comprehensive solution for efficient and robust fruit harvesting in agricultural settings.

j. Mobility prototype 1



k. Mobility prototype 2



l. Mobility prototype 3(final)

E. Container Module:

The Container Module seamlessly integrates with the harvesting robot, positioned behind the mobility system. Engineered with a thoughtful design, it serves as a receptacle for harvested fruits. The container features a capacity limit indication system, ensuring efficient fruit collection. A sensing system is deployed to relay real-time feedback to the robot, enabling it to navigate to the inventory dump once the limit is reached. This dynamic interplay between the container and harvesting components optimizes the fruit collection process, allowing the robotic arm to seamlessly pluck and deposit fruits while maintaining operational efficiency.

F. Control Module

The Control Module orchestrates the integration of camera, deep learning, robotic arm, and container modules. Driven by a Microcontroller, it processes data from cameras and sensors, making informed decisions. A User Interface facilitates human interaction and monitoring. The module employs a robust Communication System to synchronize operations, enabling seamless coordination among components. This unified control system ensures efficient decision- making, orchestrating the collaborative functionality of the robotic arm and container based on real-time data from the environment.

G. Power Supply Module

The Power Supply Module ensures the robot's energy needs with a focus on safety and reliability. Utilizing Batteries as the primary power source, a sophisticated Charging System maintains optimal charge levels. Voltage Regulators stabilize power output, enhancing system stability. Power Management strategies optimize energy usage, and a Backup Power mechanism ensures continuity. This comprehensive module guarantees a safe, reliable, and sustained power supply for the efficient and uninterrupted operation of the robotic system in various agricultural environments.

III. Distance Measurement Calculation

Precise distance measurement using cameras in fruit harvesting robots is critical for optimal picking efficiency. This technology enables accurate spatial awareness, ensuring the robotic system can discern the distance to fruits, facilitating precise object manipulation. Implementing such distance measurement enhances overall harvesting precision and minimizes potential damage to both the robot and the fruit.

Suppose we have a reference object, a person, with a known width of 16 inches. In our frame, the person's width is measured as 120 pixels. We also know the person is at a distance of 45 inches from the camera.

1. Focal Length Calculation:

- Real width=16 inches
- Width in frame=120 pixels
- Measured distance=45 inches

Using the formula:

- Focal length = (width in frame × measured distance) / real width

Substituting values:

- Focal length = (120 × 45) / 16 ≈ 337.5

2. Distance Calculation:
   - Focal length calculated above
   - Real object width=16 inches
   - **Width in frame=120 pixels**

Using the formula:

- distance = (real object width × focal length) / width in frame

Substituting values:

- distance = (16 × 337.5) / 120 ≈ 45 inches

3. Function definitions used for coding:

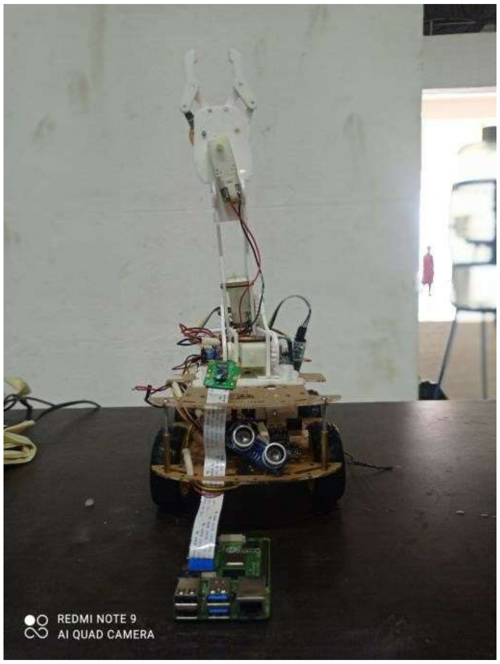
```
Def focal length finder (measured distance, real
width, width in frame):
        Focal length = (width in frame * measured
        distance) / real width
        return focal length

def distance finder(focal length, real object width,
width in frame):
        distance = (real object width * focal length) /
        width in frame
        return distance
```

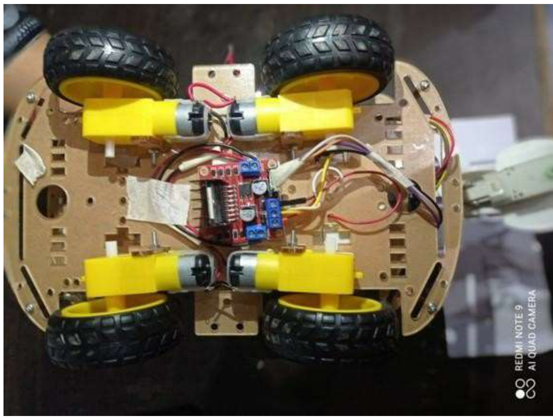
IV. Developed Prototype

Our developed prototype comprises seven meticulously designed modules tailored to precise specifications outlined in the planning phase. These modules, ranging from camera and sensors to power supply and mobility, address diverse functional requirements. The robot's body, crafted from a blend of materials like metal and plastic, seamlessly integrates these modules.

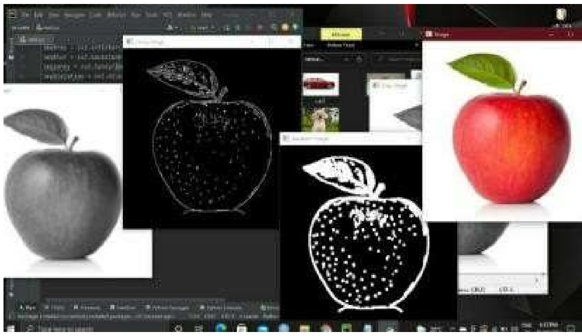
Each component is strategically secured and interconnected, ensuring optimal functionality. This holistic approach in module design and integration underscores the prototype's versatility and robustness, reflecting a comprehensive solution for efficient and precise fruit harvesting in agricultural settings.



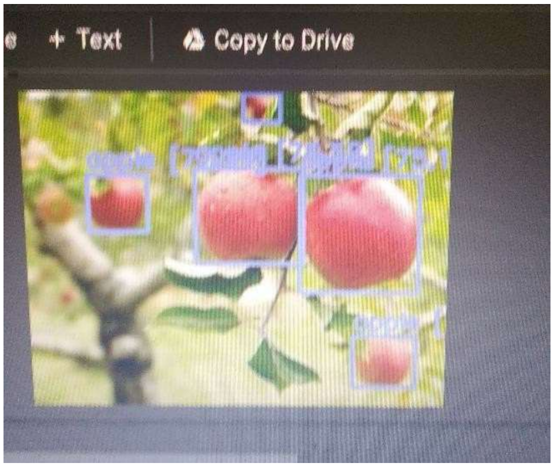
m. Complete hardware Fruit Harvesting Robot (front view)



n. Fruit harvesting robot mobility module



o. Fruit texture analysis training datasets



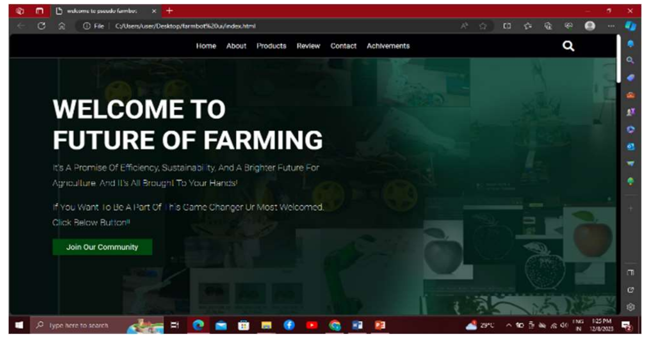
p. Fruit detection among foliage in image (output)

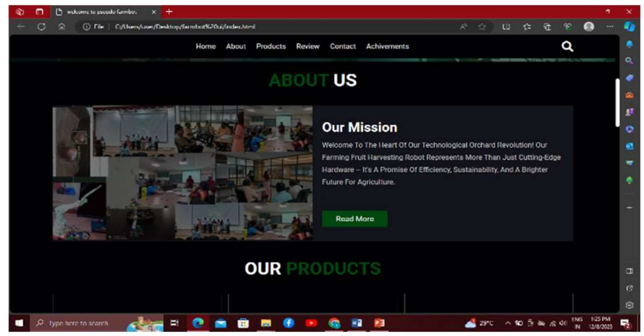
q. Real-time fruit detection among foliage

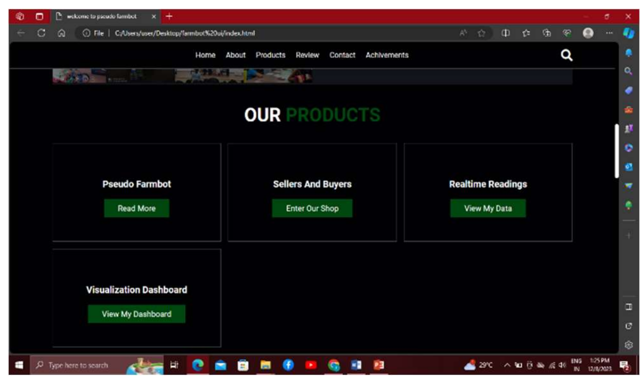
V. Web Interface

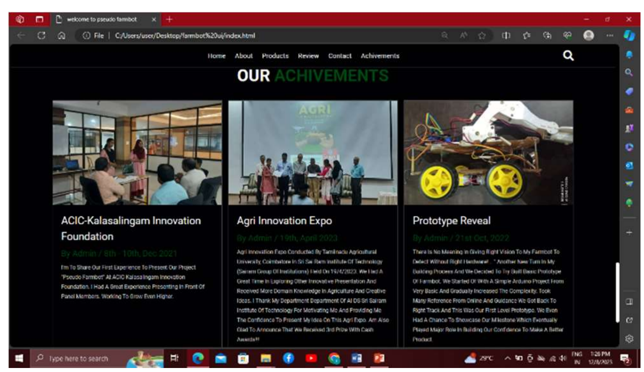
The E-commerce Web Application, developed using Flask, represents a significant advancement in our project. With a functional foundation encompassing login/logout capabilities and a connected database, users navigate seamlessly through a secure environment. Real-time data sourced from Arduino and the serial monitor is intelligently presented on dynamic dashboards, fostering transparency and control.

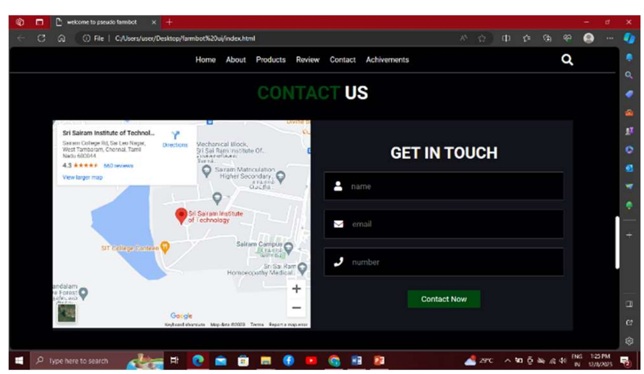
Currently, under active development are the buy and sell functionalities, promising a comprehensive user interface for online fruit transactions. This multifaceted web application boasts a monitoring dashboard for overseeing all system functionalities, an integrated e-commerce platform facilitating the buying and selling of agricultural products, a real-time data display for instant insights.And a user manual section dedicated to simplifying the utilization of the farmbot. Together, these features synergize to create an accessible, user-centric hub that enhances the efficiency and user experience of our automated fruit harvesting system.











VI. Conclusion

In conclusion, our fruit harvesting robot prototype represents a harmonious integration of cutting-edge technologies and thoughtful design. The meticulous development of seven distinct modules, including camera and sensors, deep learning, robotic arm, container, control, power supply, and mobility, underscores our commitment to precision and efficiency in agricultural practices. The fusion of diverse materials in the robot's body, coupled with secure interconnections among modules, ensures a robust and adaptable solution for fruit harvesting.

As we move forward, guided by the mission to revolutionize agricultural automation, our vision is to empower farmers with innovative tools that enhance productivity and sustainability. Our commitment extends to leveraging advancements in artificial intelligence, robotics, and sensor technologies to address evolving challenges in the agricultural landscape.

VII. Future scope

The future scope of our work envisions a seamless fusion of technology and agriculture, fostering a sustainable and efficient ecosystem for crop harvesting and cultivation. With the mantra "Innovation for Cultivation," we aspire to contribute to a future where precision, automation, and sustainability converge to shape the next frontier in agricultural excellence.

VIII. References

[1] Development of an autonomous tomato harvesting robot with rotational plucking gripper publisher:IEEE Journal article., 2016.

[2] Real Time Riped Fruit Detection using Faster R-CNN Deep Neural Network Models Journal article Publisher:IEEE 2022

[3] Object identification and location used by the fruit and vegetable picking robot based on human- decision making Journal article; Publisher:IEEE; 2017

[4] Fresh fruit picked from the tree by Israeli robots The Times of Israeli 2020

[5] Dual Robot Coordination for Apple Harvesting Conference: 2017 ASABE Annual International Meeting 2017

[6] Gongal A, Amatya S, Karkee M, Zhang Q, Lewis K (2015) Sensors and systems for fruit detection and localization: a review. Comput Electron Agric 116:8–19

[7] Okamoto H, Lee WS (2009) Green citrus detection using hyperspectral imaging. Comput Electron Agric 66(2):201–208

[8] Bulanon DM, Kataoka T (2010) Fruit detection system and an end efector for robotic harvesting of fuji apples. Agric Eng Int CIGR J 12(1):203–210

[9] Linker R, Cohen O, Naor A (2012) Determination of the number of green apples in rgb images recorded in orchards. Comput Electron Agric 81:45–57

[10] 3. Stajnko D, Lakota M, Hočevar M (2004) Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. Comput Electron Agric 42(1):31–42