ENPM-808
Independent Study

A. James Clark School of Engineering

Application of Robots in Farming

Abhinav Garg

117553000

Under the mentorship of **Professor Dr Zeid Kootbally**

Acknowledgement

I would like to thanks Professor Dr. Zeid Kootbally to have given me the opportunity to pursue project under his guidance. Professor Zeid helped me with each and every aspect of the project starting from simulation environment to all the way to training of neural networks and handing of robotics arm. I learned a lot throughout the journey of this project and would love to see my project materialize in real world farm applications. I would also like to thank my friend Ishan Tamrakar who helped me a lot with simulation environment and developing CAD models in solidworks.

Introduction

I pursued my independent study in the domain of agricultural robotics, the aim of the study was to learn about robotic manipuators and give them the ability to pick various fruits and vegetables in the field. I particular picked this topic because I had an opportunity during the summer and the fall semester of 2023 to do an internship with a agricultural robotics firm Korechi Robotics and Automation (https://korechi.com/).

Agriculture, being the backbone of human sustenance, has undergone remarkable transformations throughout history to meet the evolving demands of a growing global population. In recent years, the integration of cutting-edge technologies into agricultural practices has given rise to a new era: smart farming. Among the myriad technological advancements within this domain, agricultural robotics stands out as a pioneering field with the potential to revolutionize traditional farming methods.

This report delves into the realm of agricultural robotics, specifically exploring its application in the automated harvesting of fruits. Traditional agricultural practices have long relied on manual labor for harvesting, a process that is not only labor-intensive but also subject to challenges such as labor shortages, rising costs, and the need for increased efficiency. With the advent of robotics in agriculture, there is a paradigm shift towards more sustainable, efficient, and technologically advanced methods of crop management.

One of the key applications gaining traction within agricultural robotics is the development of robotic systems designed to pick fruits. The labor-intensive nature of fruit harvesting, combined with the perishable nature of many fruits, makes it an ideal candidate for automation. Agricultural robots equipped with advanced sensors, machine vision, and robotic arms offer a promising solution to the challenges faced by traditional fruit-picking methods.

This report aims to create a robotic manipulator for agricultural practices with a specific focus on automating fruit harvesting processes. Below are the key objectives achieved during the project.

1. Developed a low level system design framework for harvesting fruits in an agricultural field.
2. Developed a simulation environment mimicking real world agricultural field and environment.
3. Applying inverse kinematics on a robotic manipulator to pick various fruits from the field.
4. Developed a neural network to identify various fruits and vegetables from the field.

Development Stages

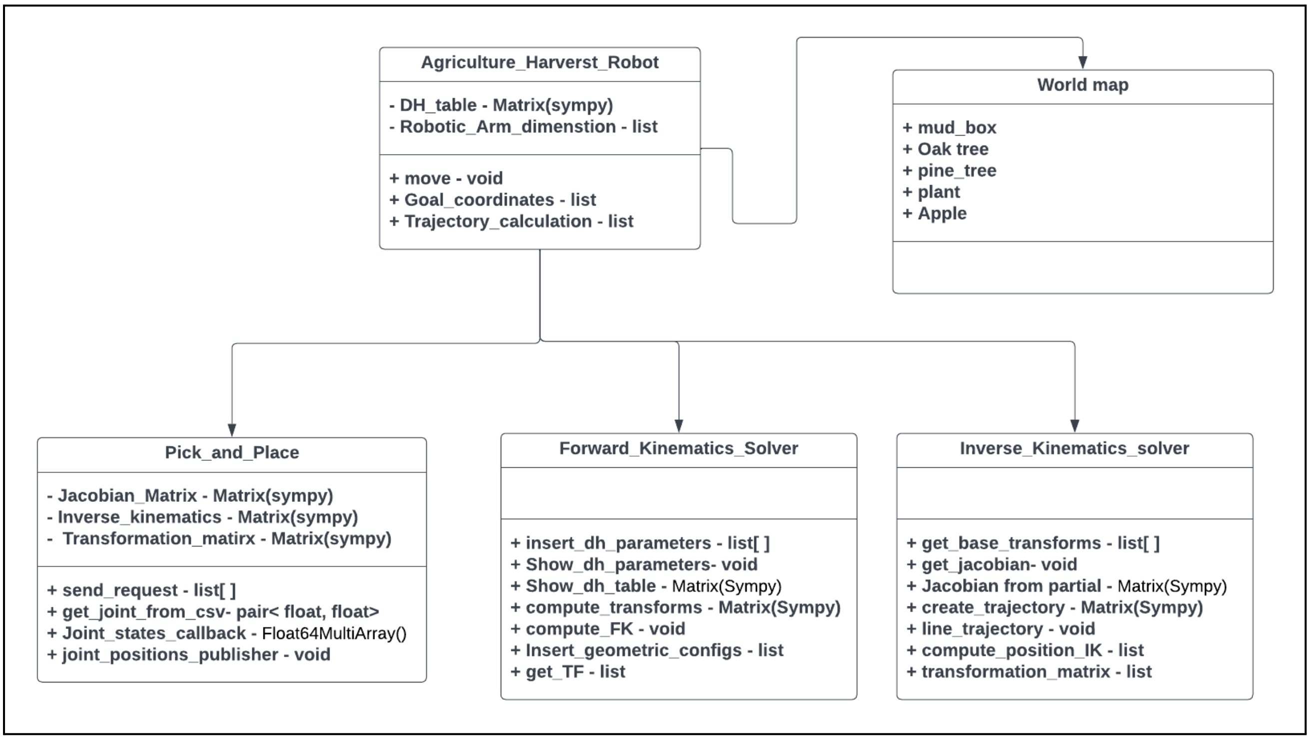
The project was developed and planned in a total of four stages which were first conceived during the start of the project, the work done and tasks completed in each stage has been described below.

First phase: Design Phase

The design phase included conceiving the overall architecture of our harvesting robot and coming up with a UML diagram and low level system design. This laid the groud work of what software design and coding practices will be followed in phases two and three. During this phase, we finalized the software packages and simulation environment.

Softwares used for the development of this project

1. Robot Operating System 2 (Galactic)
2. Gazebo (Ignition)
3. Python 3.10



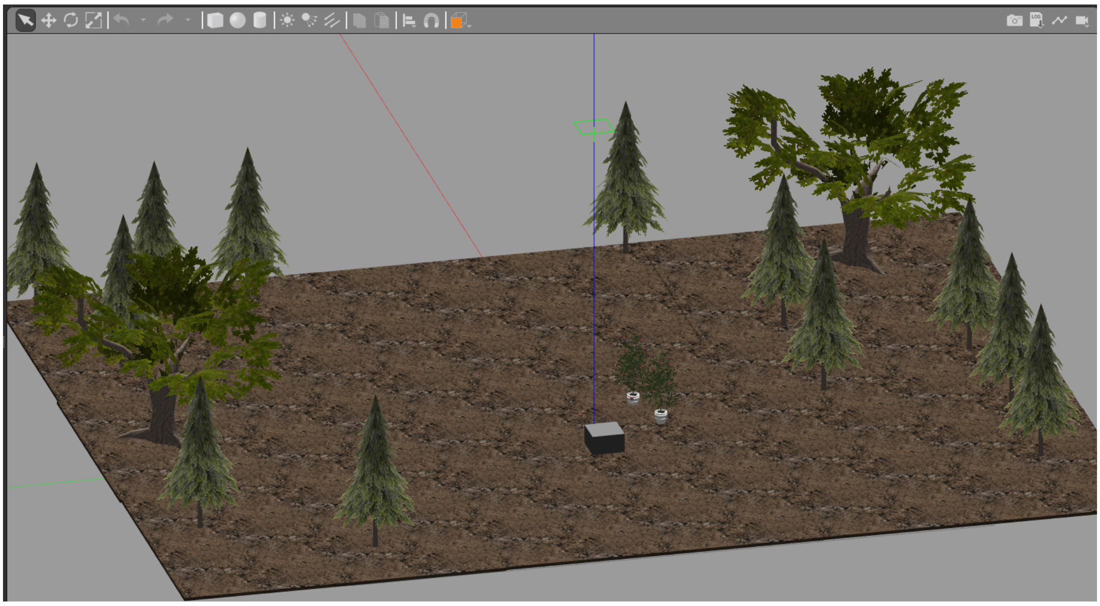
UML Diagram

In total I used 3 classes to plan the trajectory of the arm, all the code related to computer vision part has been put inside a jupyter-notebook to make it more comprehensible.

Second Phase: Simulation Phase

a. Simulation Environment

After finalizing the simulation environment, world map was developed imitating a real-world agricultural environment, this was our testing ground for all our experiments. All the custom objects were created in solidworks.Different trees and plants were constructed inorder to see which which one would best perform for our applications.



The entire world map

Side-view of world map

The fruits were spawned a bit outside of the bushes so that the robotic manipulator's vacuum gripper does not collide with the bushes and instead is able to pickup the fruits smoothly. Also since they didn't had any contact with the bushes and instead were hovering in the air, the gravity was set to 0 for all fruits in the gazebo environment.

Fruit plant constructed

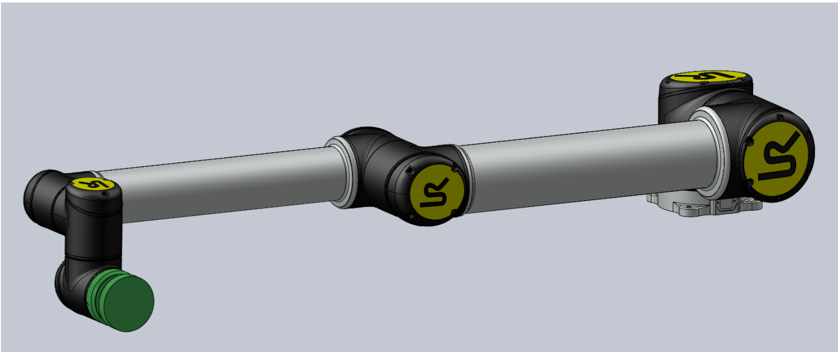
Apple Fruit create in the solidworks

b. Robotic arm manipulator

Following this I designed a robotic arm capable of plucking fruits/vegetables from the tree in solidworks. Furthermore the concept of inverse kinematics was studied and applied to the robotic arm to move it from one place to another.

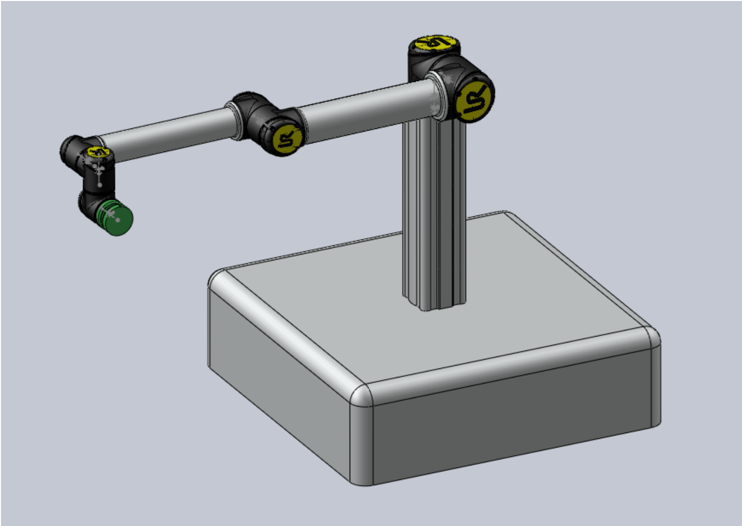
The robot used for the project is UR10e robotic arm developed by Universal Robots. Design files of the arm were fetched from the Universal Robots Website and assembled it in Solidworks by assigning our custom axises and frames.

The stl files exported from solidworks can be found on this link. The design of each link was imported from the Universal Robots website and then assembled in Solidworks using our own frame assignment because our home position is different from the default home position of UR10e Robot.



CAD model of Robotic Arm (UR10e)

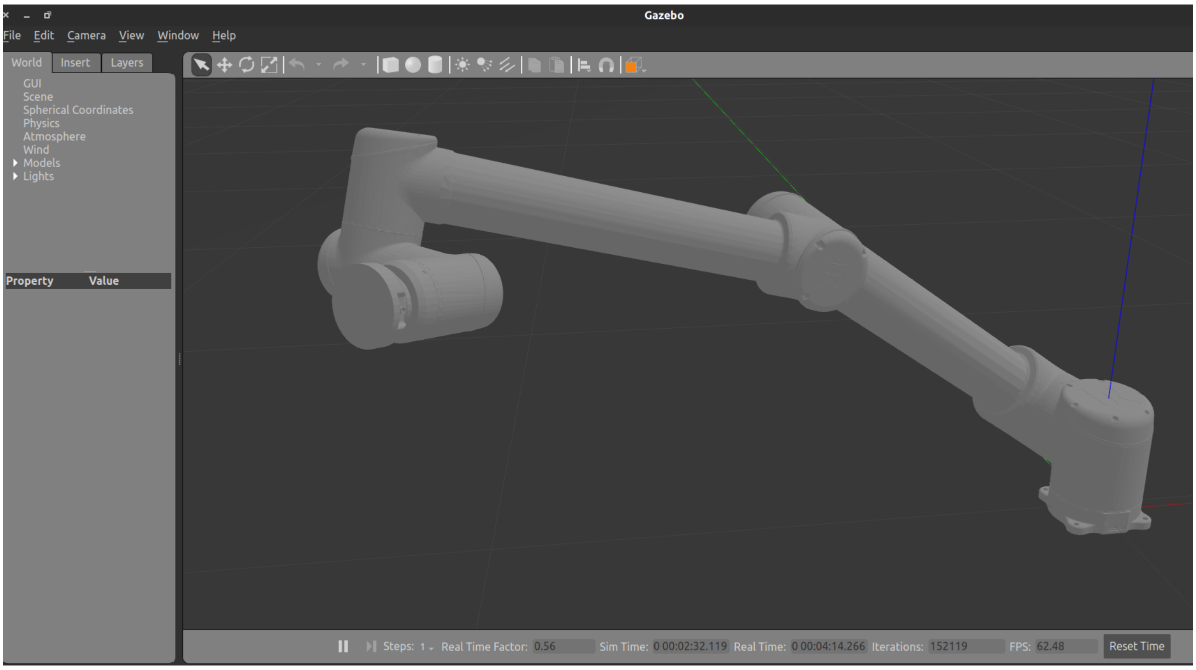
Later the STL format was converted to urdf format and imported into the Gazebo environment. After that we calculated the forward and inverse kinematics. Once the robotic arm was working we placed it in the environment which was than moved using the robotic arm. A vacuum gripper was attached to the end effector in order to facilitate the picking of objects.



Robotic arm mounted on a base

c. Gazebo simulation robotic arm

The stl files of solidworks were than exported to urdf format and spawned into gazebo world as shown below.

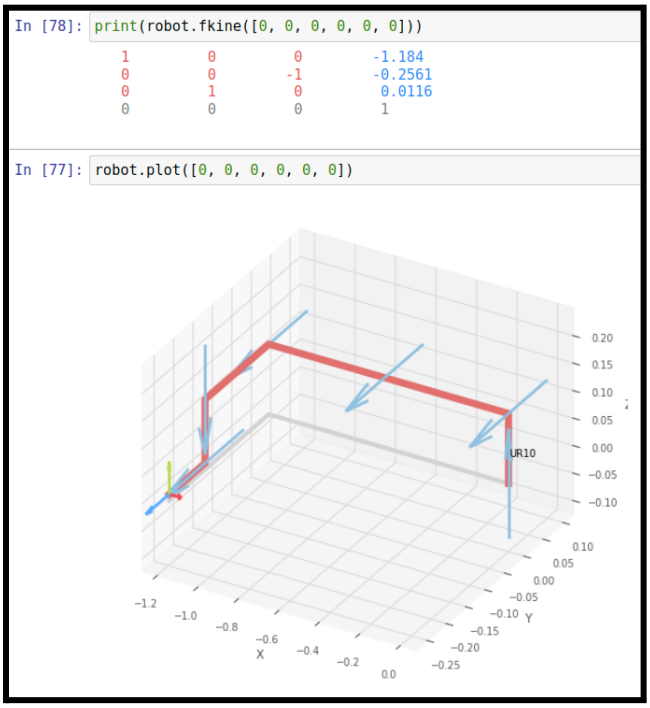


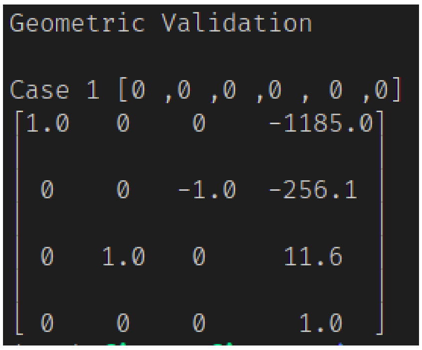
Robotic arm UR10e spawned in Gazebo Environment

D. Geometric validation of Robotic Arm

Inorder to verify the transformation matrix and inverse jacobian matrix of the manipulator, several tests were performed to verify that robotic arm and its joint angles are moving as expected according to the trajectory planned. For verification we used peter corke libraries in python to verify the orientation of the robotic arm.

a. **Validation 1: Joint Positions (0, 0, 0, 0, 0, 0)**



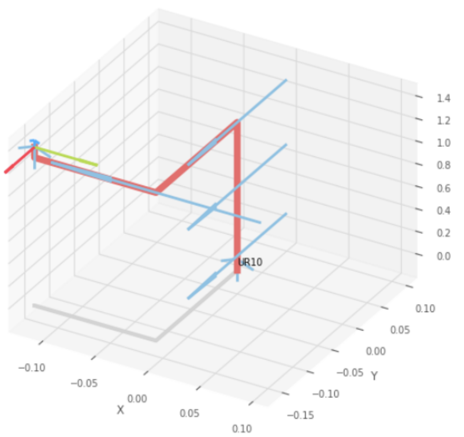


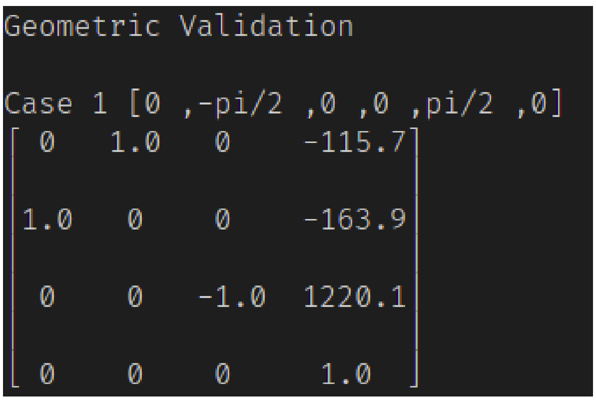
As expected the coordinate at the home position is [-1185.0 , -256.1, 11.6]

b. **Validation 2: Joint Positions (0, -pi/2, 0, 0, pi/2, 0)**

```
In [75]: print(robot.fkine([0, -np.pi/2, 0, 0, -np.pi/2, 0]))
0   1   0   -0.1157
1   0   0   -0.1639
0   0   -1   1.219
0   0   0    1
```

```
In [72]: robot.plot([0, -np.pi/2, 0, 0, np.pi/2, 0])
```

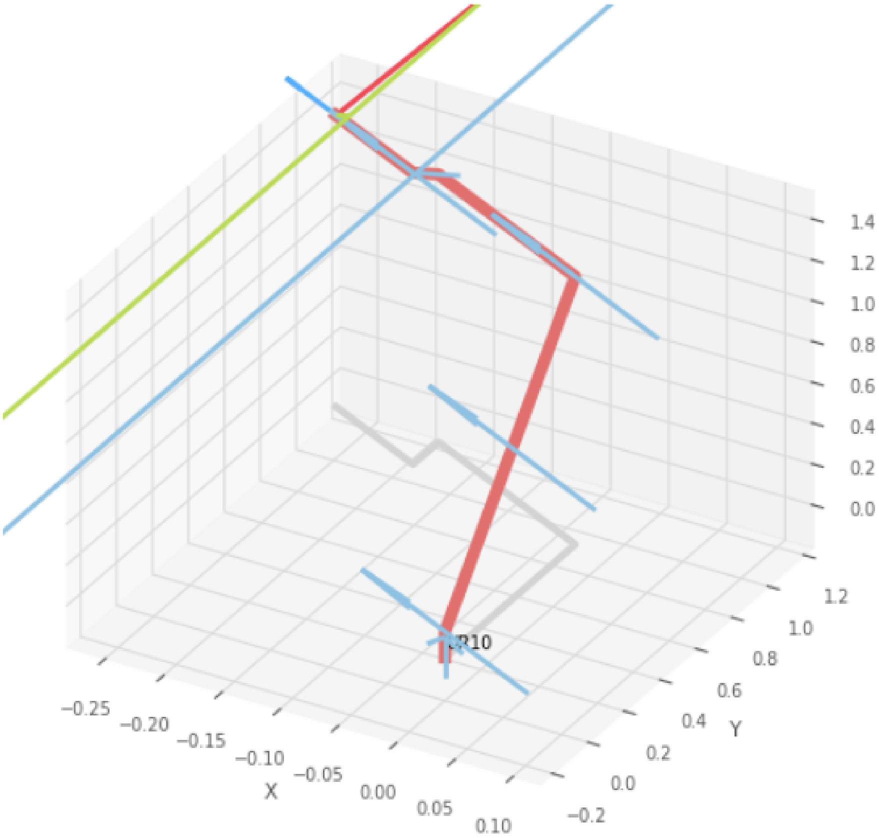




As expected the coordinate at the home position is [-115.7 , -163.9, 1220.1]

c. Validation 3: Joint Positions (3*pi/2, -pi/2, 0, 3*pi/2, 0, 0)

```
In [9]: # robot.figure(figsize=(6, 3))
        robot.plot([3*np.pi/2, -np.pi/2, 0, (3/2)*np.pi, 0, 0],backend='pyplot')
```




```
Out[9]: PyPlot3D backend, t = 0.05, scene:
          robot: Text(0.0, 0.0, 'UR10')

        <Figure size 640x480 with 0 Axes>
```

```
In [10]: print(robot.fkine([3*np.pi/2,-np.pi/2, 0, (3/2)*np.pi, 0, 0]))
```

```
   0         0        -1        -0.2561
   1         0         0         0
   0        -1         0         1.427
   0         0         0         1
```

```
Case 1 [3*pi/2 ,-pi/2 ,0 , 3*np.pi/2 , 0 ,0]
[1.83697019872103e-16  2.24963967399279e-32          -1.0                  -256.1
        1.0           1.22464679914735e-16  1.83697019872103e-16  1.05435966172591e-13
1.22464679914735e-16          -1.0                     0                   1428.0
         0                     0                       0                    1.0     ]
```

Third Phase: Perception

In the third phase of the project, a perception pipeline was developed, as to detect and segment a respective vegetable, since classical perception techniques based on thresholding of colours were not working in this application. I constructed my own neural network with help of pytorch and trained it on dataset with different fruits.

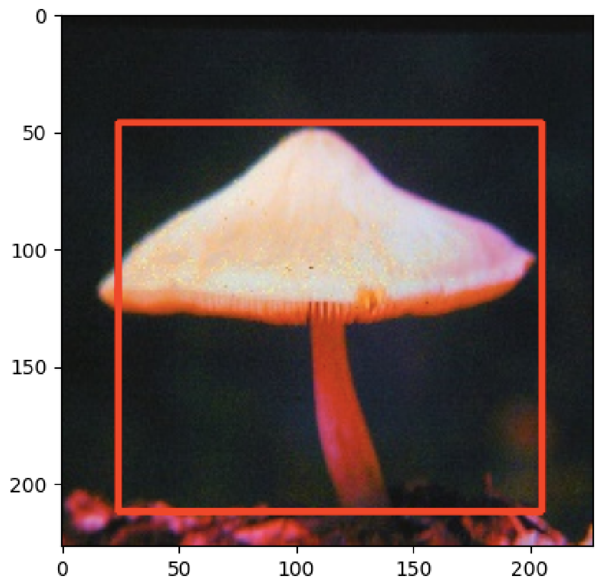
	img_path	xmin	ymin	xmax	ymax	width	height	label
0	train_images/mushroom_51.jpg	24	23	202	183	227	227	mushroom
1	train_images/eggplant_37.jpg	34	34	88	201	227	227	eggplant
2	train_images/mushroom_20.jpg	49	86	183	185	227	227	mushroom
3	train_images/eggplant_51.jpg	51	59	191	164	227	227	eggplant
4	train_images/eggplant_26.jpg	40	70	179	168	227	227	eggplant
...	...	...	...	...	...	...	...	...
181	train_images/eggplant_62.jpg	67	22	177	215	227	227	eggplant
182	train_images/cucumber_45.jpg	11	31	217	208	227	227	cucumber
183	train_images/mushroom_37.jpg	93	13	158	193	227	227	mushroom
184	train_images/eggplant_44.jpg	21	59	192	171	227	227	eggplant
185	train_images/mushroom_16.jpg	43	20	197	182	227	227	mushroom

186 rows × 8 columns

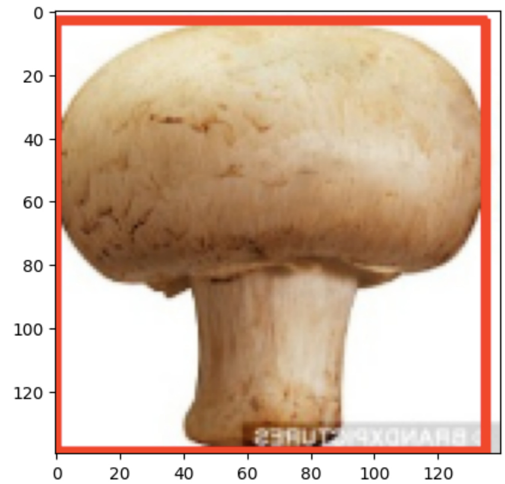
Image classifier (Mushroom, Eggplant and Cucumber)

Since repeated attempts were made to create a dataset of custom images of the virtual environment the models confidence was not very good at detecting the images of apple.

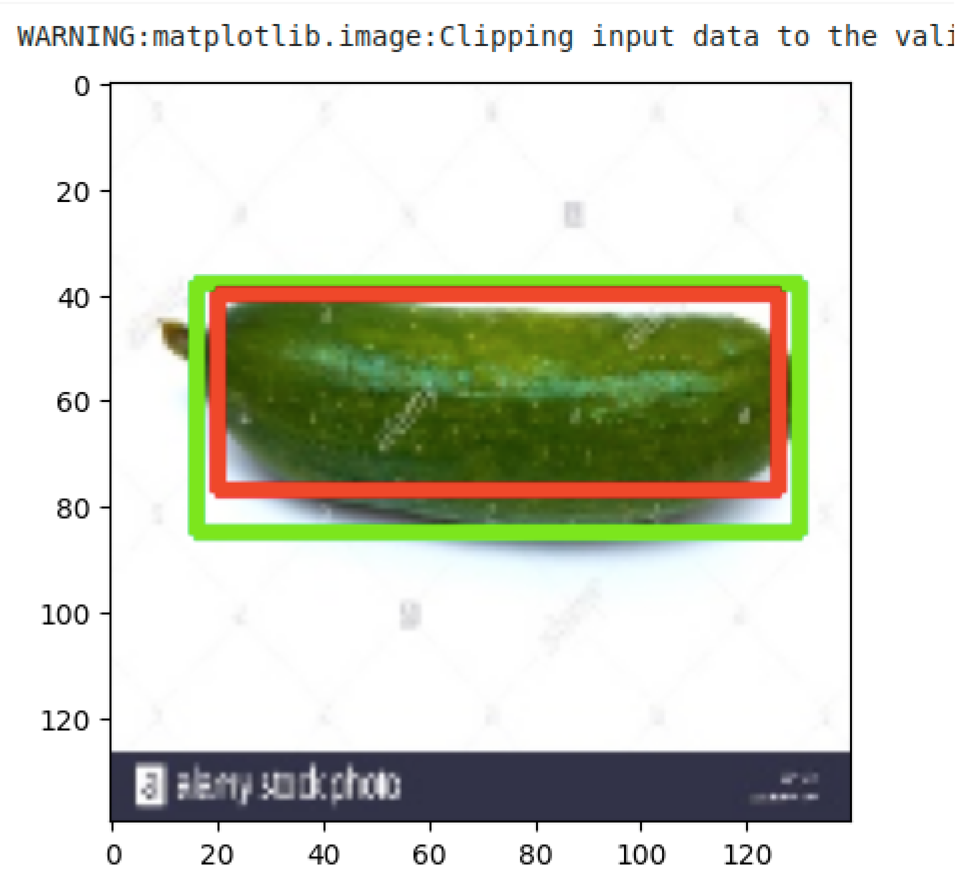
```
<matplotlib.image.AxesImage at 0x7c3711b38430>
```




```
WARNING:matplotlib.image:Clipping input data to the valid range
<matplotlib.image.AxesImage at 0x7c3705ab0220>
```

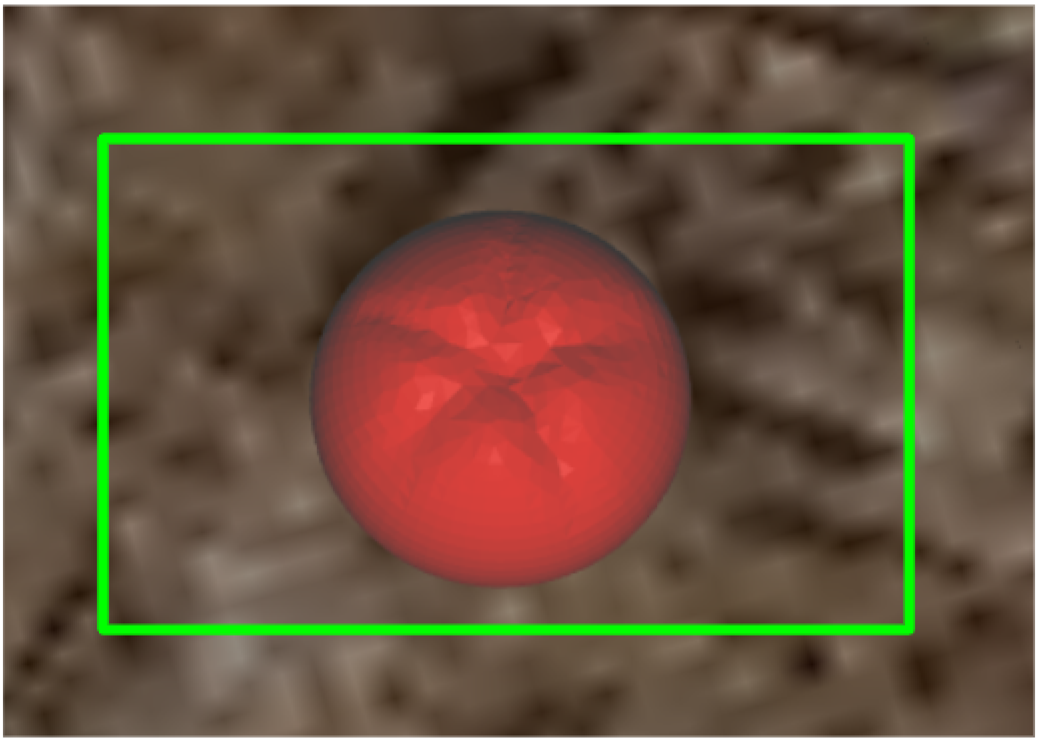


Mushroom Detection



Cucumber Detection

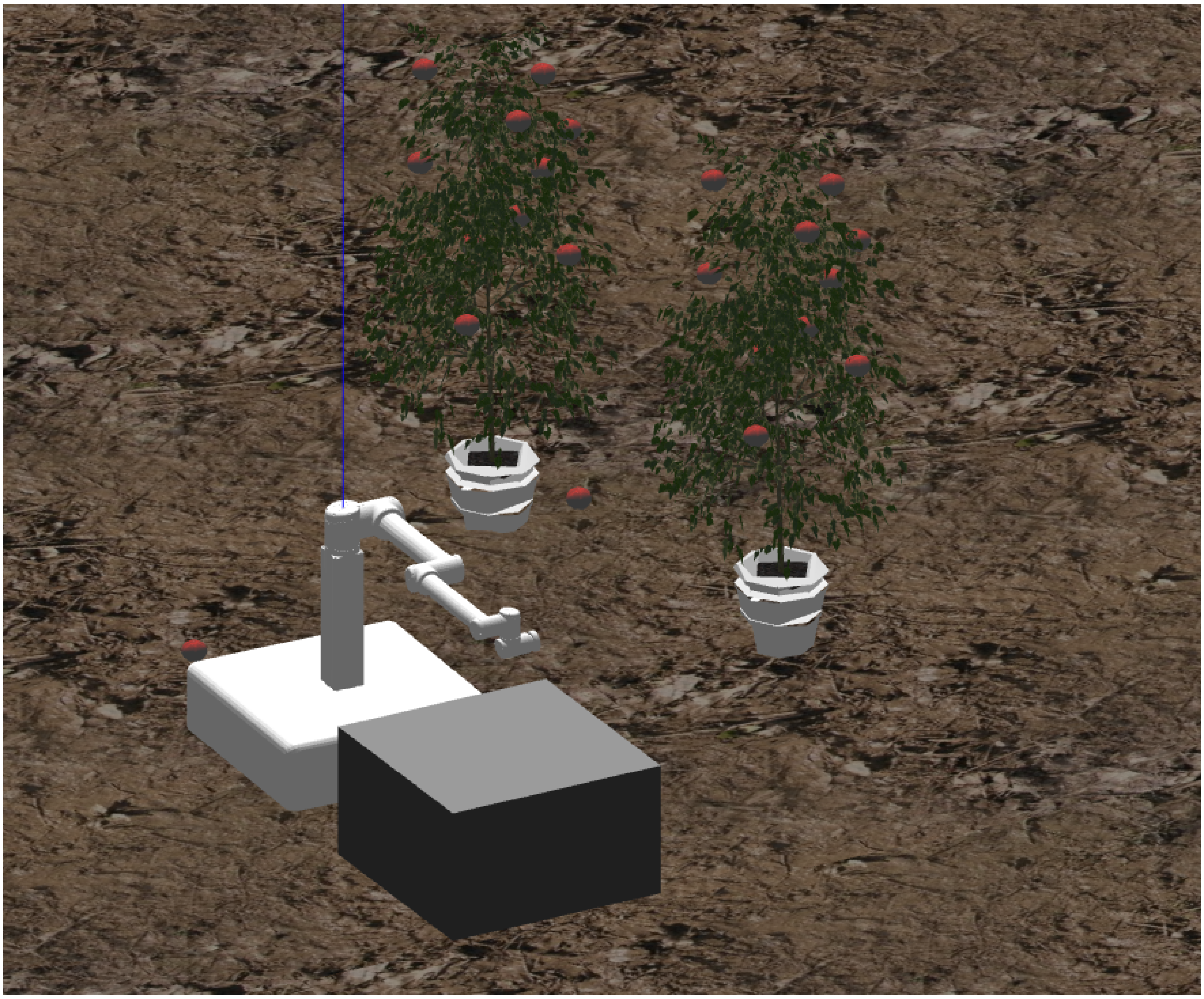
The virtual apple in the environment was also detected but with a very low confidence level.



Apple detection (15% confidence) not good for object tracking and detection

Fourth Phase: Control and Decision Making

Since the confidence level of object detection was very poor, autonomous detection and traversal to the set goal target was not possible on such low detection confidence level. Hence I used inverse kinematics and jacobian matrix to control the robot from set initial point to the location coordinate of the fruit. The final video of the project can be found here. The radius of influence of vacuum gripper was kept to minimum inorder to test the accuracy of the trajectory path taken by the robotic arm.



Final Simulation Environment and Robotic Arm in action

References

1. https://docs.ros.org/en/foxy/index.html
2. https://classic.gazebosim.org/tutorials?tut=ros2_overview&cat=connect_ros
3. https://automaticaddison.com/how-to-simulate-a-robot-using-gazebo-and-ros-2/
4. Course material of ENPM-663 (Introduction to Robot Modelling)
5. Neural Network lectures series of stanford CS231n
6. https://www.solidworks.com/