Control Robotics and Machine Learning Laboratory

הטכניון - מכון טכנולוגי לישראל
TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY

הפקולטה להנדסת חשמל
המעבדה לבקרה רובוטיקה ולמידה חישובית

# Autonomous Robot to Collect Table-Tennis Balls

**Yamen Safadi**
**Supervisor: Koby Kohai**

**Winter 2022-23**

# Table of Contents

# 1. Abstract


Advances in robotics and artificial intelligence have the potential to revolutionize the way we live our daily lives. Rather than being a danger to humans, robotics and AI can be used to augment human capabilities, making our lives easier, safer, and more efficient.

The prospect of integrating the ability to utilize the power of AI and machine learning into robotics for understanding incoming data collected from the surrounding environment by providing them with various types of sensors and cameras, as is the case in image and object recognition for example, means that we are heading towards a future of even greater dependency on intelligent connected devices in our daily lives.

The aim of this project is to design and develop an autonomous robot capable of collecting table tennis balls to assist people with disabilities in playing table tennis. This is an excellent example of how robotics and AI can be used to benefit society. By automating the task of collecting balls, the robot reduces the physical burden on players with disabilities and allows them to focus on the game. This project demonstrates the potential of robotics and AI to improve the quality of life for people with disabilities and highlights the importance of responsible development and deployment of these technologies.

The robot, which is based on the iRobot Create 3 platform and is equipped with a mounted camera, is controlled by a Jetson Nano microcontroller. The Jetson Nano processes the input from the camera and utilizes a machine-learning model to identify table tennis balls in the image. Once the ball is detected, the Jetson Nano sends signals to the iRobot Create 3 to move towards the ball. The robot is designed to collect the balls using a servo motor-controlled mechanical arm that picks up the ball and places it in a basket. This project is a step towards making sports more accessible for people with disabilities and has the potential to be extended to other sports with small objects.

# 2. Introduction

All the code of this project is open source and available for the general public on GitHub Yamen-Safadi/iRobot-Table-Tennis-ball-collector repository.

The project was suggested in the CRML department in the Technion. The goal is to automate an iRobot to collect table tennis balls. While similar to other projects done in the past, it had very different architecture and goals.

Similar projects built previously in the faculty:

- Voice Automation of iRobot for the execution of complex commands. Done by johnny Ashkar and Idan Basson.

This project came as a great help since they used the microcontroller (the jetson nano we were able to learn from them how to communicate properly with the iRobot using the jetson nano.

Our project did 4 things differently:

1- Our robot must be fully autonomous and there is no need for humans to give it commands its goal was predetermined.
2- The robot had to interact with the real world differently since it has a mechanical arm and should use it to collect the balls.
3- We supplied the robot with the ability to understand its surroundings using AI-based and non-AI-based algorithms for ball detection and integrated them with ball tracking.
4- we used the robot's built-in sensors to avoid crashing into objects and falling from stairs or any other height.

# 3. Technologies Used

## 3.1. NVIDIA Jetson Nano Developer Kit

The NVIDIA Jetson Nano Developer Kit is a compact single-board computer designed for AI androbotics applications. It features an NVIDIA Maxwell GPU and is capable of running multiple neural networks in parallel. The developer kit is a low-cost platform for developing and testing AIprojects, and is suitable for makers, students and hobbyists. It is based on NVIDIA Jetson Nano System-in-Module and its GPU includes 128 CUDA cores. Also includes a quad-core ARM A57CPU, 4 GB LPDDR4 memory, Gigabit Ethernet, USB 3.0 and HDMI.

In addition to its AI capabilities, the Jetson Nano Developer Kit is also equipped with GPIO and UART interfaces that make it suitable for a wide range of robotic projects.

In the previous picture we can see the Nvidia mini computer (Jetson Nano). The jack on the lowerleft side of the board is the power supply input: 5V DC power with at least 4 amps is needed to smoothly power the device.

## 3.2. Intel dual band wireless-ac 8265 with antennas

Intel Dual Band Wireless-AC is a wireless network adapter produced by Intel Corporation. It supports dual-band Wi-Fi, meaning it can operate on both 2.4GHz and 5GHz frequency bands, providing faster and more reliable wireless connectivity. This technology is commonly used in laptops, desktop computers and other devices for wireless internet access.

In our case such network adapter was necessary since the NVIDIA Jetson Nano does not come with an integrated wireless card. This specific adapter was recommended by everyone who workedwith Jetson Nano and for that reason was our network adapter of choice.

In the previous picture, we can see the Intel Dual Band Wireless-AC. The chip itself goes into theJetson Nano and connects perfectly to it under the block “boxy” cover. This needs to be done manually after purchasing the internet adapter by removing a few screws, connecting the adapter and encasing back the Developer kit.

### 3.3. iRobot Create 3

The iRobot Create 3 is a programmable robot platform designed for students, educators, and developers who want to learn robotics and create their own robotic applications. It is built on the same platform as the Roomba vacuum cleaning robot but without the vacuuming capabilities. The iRobot Create 3 features a variety of sensors, including cliff sensors, bumper sensors, and wheel encoders, which allow it to navigate and interact with its environment. It also includes a number of built-in behaviors, such as driving in straight lines, and turning. Users can program the iRobot Create 3 using a variety of programming languages and environments, including Python, Java, C++, we chose python. It also includes a number of sample programs and tutorials to help users get started. Overall, the iRobot Create 3 is a versatile and powerful platform for learning and experimenting with robotics, and can be used for a wide range of applications, from education to research to hobbyist projects.

### 3.4. iRobot open interface – the iRobot-edu-python-sdk

The iRobot Open Interface is a communication protocol that allows external devices and software to control and interact with iRobot robots, such as the Roomba and the Create. The iRobot-edu-python-sdk is a Python software development kit that provides an interface to the Open Interface, making it easier for developers to write software that can control iRobot robots. The iRobot-edu-python-sdk includes several modules that provide different levels of abstraction for controlling the robot. For example, the low-level module provides direct access to the Open Interface commands, while the high-level module provides a simpler, more user-friendly interface for controlling the robot's movements and sensors. The SDK includes a variety of examples and tutorials that demonstrate how to use the different modules to control the robot. Some of the things you can do with the SDK include:

- Control the robot's movements, such as driving it forward turning, and stopping
- Read sensor data, such as the robot's battery level, distance sensors, and bumper sensors
- Create custom behaviors and programs for the robot to perform
- Interact with other software and devices, such as using a camera to detect objects and commanding the robot to pick them up

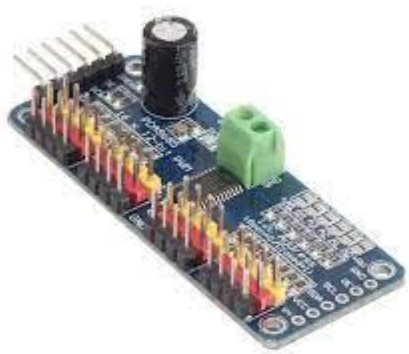
Overall, the iRobot-edu-python-sdk makes it easy for developers to create custom software to control iRobot robots and integrate them into larger systems and projects

### 3.5. Oak-d lite Camera

We used the oak-d Lite camera. The Oak-D Lite camera is a small depth-sensing camera designed for computer vision applications. It is built by the company Luxonis, which specializes in creating computer vision tools and hardware. The Oak-D Lite camera features an Intel Movidius Myriad X VPU (Visual Processing Unit) and a Sony DepthSense IMX556PLR back-illuminated ToF (Time of Flight) depth sensor. These components enable the camera to capture depth information and process it in real-time, making it ideal for applications such as object detection, tracking, and recognition. The Oak-D Lite camera also includes a variety of software tools and libraries to facilitate development, including the Luxonis DepthAI Python API, which allows developers to easily integrate the camera into their projects and applications. Additionally, the camera is designed to be compatible with popular machine learning frameworks, such as TensorFlow and PyTorch, enabling developers to create custom computer vision models and deploy them on the device. Overall, the Oak-D Lite camera is a powerful and versatile tool for computer vision applications, with a compact and lightweight design that makes it suitable for use in a wide range of settings and environments.

## 3.6. The PCA9685 Chip

The PCA9685 is a 16-channel PWM (Pulse Width Modulation) controller that can be used to control a wide range of devices that require precise and accurate timing. It is commonly used in robotics, automation, and other electronic applications. The PCA9685 chip is typically used to control LED lighting, motors, and servos. It communicates with a microcontroller or other device via an I2C (Inter-Integrated Circuit) bus, allowing for simple and efficient control of multiple channels of PWM output. One of the key features of the PCA9685 chip is its ability to generate high-frequency PWM signals (up to 1.6 kHz) with a resolution of 12 bits. This allows for very precise control of the connected devices, making it ideal for applications where accurate timing is critical. Another feature of the PCA9685 chip is its ability to operate in both open-drain and push-pull output modes. This flexibility allows it to be used with a wide range of devices and configurations. Overall, the PCA9685 chip is a versatile and powerful PWM controller that is widely used in a variety of electronic applications. Its high frequency, high-resolution PWM output and flexible output modes make it an ideal choice for controlling LED lighting, motors, and servos in robotics, automation, and other applications.



### 3.7. The MR996R servo motor

The MR996R servo motor is a high-performance, digital servo motor produced by the Chinese company, JX Servo. It is commonly used in remote-controlled vehicles, drones, and other robotic applications that require precise and accurate control of movement. The MR996R servo motor is a metal-gear, coreless, digital servo with a maximum torque of up to 30 kg.cm and a speed of up to 0.12 seconds per 60 degrees of rotation. It also has a wide operating voltage range (4.8V - 8.4V), making it suitable for use with a variety of power sources. One of the key features of the MR996R servo motor is its high-resolution digital control. It uses a 12-bit resolution, providing over 4,000 steps of precision control. This allows for very precise and accurate positioning of the connected device, making it ideal for applications that require fine control. The MR996R servo motor also has a durable metal-gear construction, which helps to ensure long-term reliability and performance. It is also designed to be easy to install, with a standard mounting pattern that is compatible with a wide range of devices and systems. Overall, the MR996R servo motor is a high-quality, high-performance servo motor that is well-suited for use in a variety of robotic and remote-controlled applications. Its high-resolution digital control, strong torque, and durable construction make it a popular choice among hobbyists and professionals alike.

# 4. General Architecture

The project consists mainly of 4 parts:

- NVIDIA Jetson Nano developer kit

  Since the Jetson Nano does not include a Wireless card, we added an Intel Dual Band Wireless-AC to extend its range.

- iRobot Create 3

- A camera we used the oak-d lite camera. We tried using its depth estimation feature but it didn't work out we will elaborate on that in the "Failed Attempts" section.

- and a mechanical arm mounted physically on top of the iRobot.

Let's delve a bit more into the general architecture. Further details about the technologies and implementation will be discussed in later parts of the booklet.

### 4.1. NVIDIA Jetson Nano Developer Kit

Jetson Nano is a powerful small computer used for building embedded applications. In our project, it was used as "the brain" of the iRobot. On this small computer, an object detection model and a controller are running. The object detection model is used to detect where the balls are in the frame that the camera captures and estimate the distance between the ball and the robot. The controller is used to control the iRobot using the iRobot-edu-python-sdk library, and to control the mechanical arm responsible of collecting the balls.

A camera is connected to the Jetson Nano from which the running controller gets further information about the surrounding environment.

### 4.2. iRobot Create 3

The iRobot Create 3 is a programmable robot platform designed for students, educators, and developers who want to learn robotics and create their own robotic applications. It is built on the same platform as the Roomba vacuum cleaning robot but without the vacuuming capabilities. The iRobot Create 3 features a variety of sensors, including cliff sensors, bumper sensors, and wheel encoders, which allow it to navigate and interact with its environment. It also includes a number of built-in behaviors, such as driving in straight lines, and turning. Users can program the iRobot Create 3 using a variety of programming languages and environments, including Python, Java, C++, we chose python. It also includes a number of sample programs and tutorials to help users get started. Overall, the iRobot Create 3 is a versatile and powerful platform for learning and experimenting with robotics, and can be used for a wide range of applications, from education to research to hobbyist projects.

The Create 3 is controlled by the jetson nano via a Bluetooth connection.

## 4.3. Oak-d lite Camera

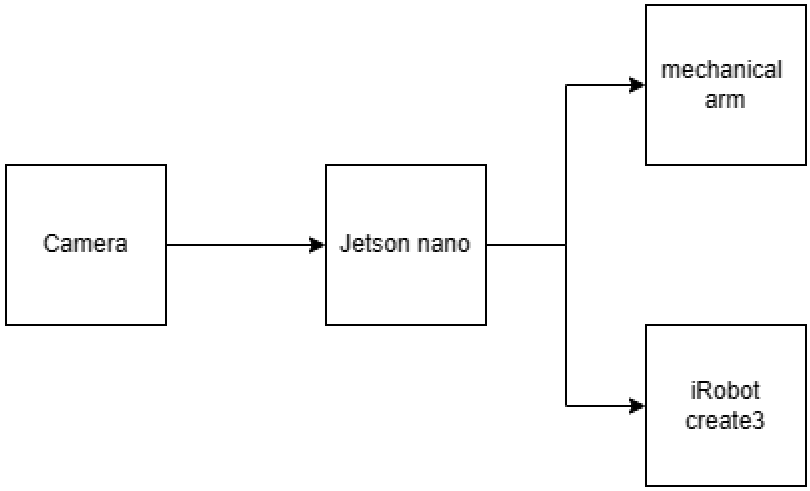
We tried using its depth estimation feature given by the oak-d Lite camera and the library depth-ai, but it didn’t work out we will elaborate on that in the “Failed Attempts” section. So we used the oak-d lite as a camera only witch is connected to the jetson nano via a (USB-A -> USB-C) cable.

## 4.4. Mechanical Arm

We made the mechanical arm as a prototype made of cardboard. The ideas was to build a three joint arm that could reach any ball in 20 cm radius of the robot but since this is only a prototype and its made with relatively cheap materials, the servo motors couldn’t handle the weight of the cardboard and therefore we decided to over cum this problem in the software. And by that we developed that software responsible for controlling the robots movement so that it abroach the ball in a certain angle and by this allowing the mechanical arm to “one dimensional” and only rely on one Servo Motor that is attached to the robot and no weight is being handled by the arm it self.

A better way to do the job is to 3d print the arm so that it holds the motors well and does not get affected by the weight of the arm itself such a design is seen in the photo below:

# Overall:


Camera
Jetson nano
mechanical arm
iRobot create3

# 5. Complete Implementation

All the code of this project is open source and available for the general public on GitHub Yamen-Safadi/iRobot-Table-Tennis-ball-collector repository.

## 5.1. Object detection

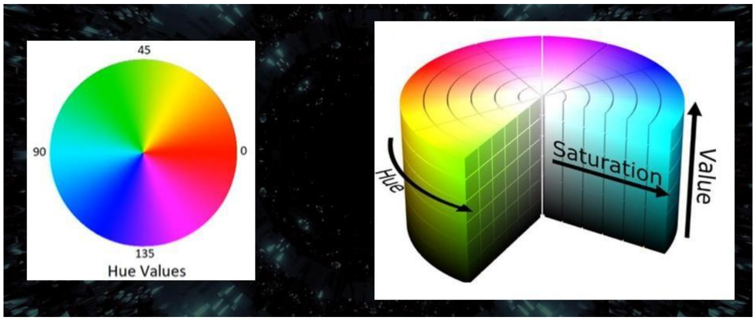
There are two ways to detect objects:

- Using the HSV color space
- Using a Neural network – AI-based algorithm.

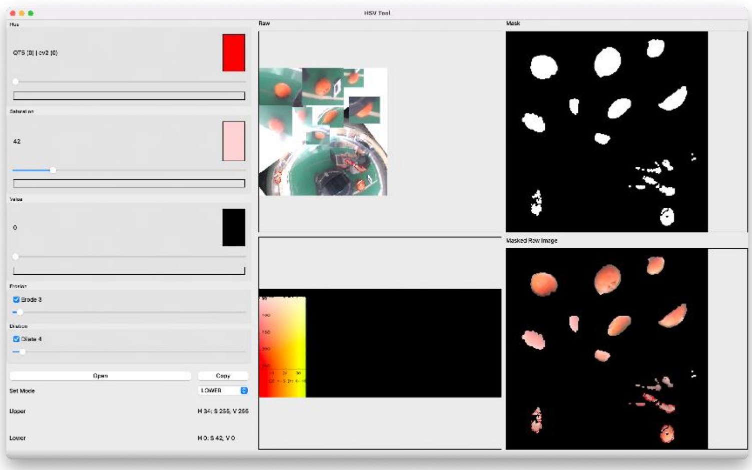
To detect by color, each frame is converted from a typical RGB 3 channel frame to an HSV picture using the three parameters: Hue, Saturation and brightness value to express colors.

The converting is done using OpenCV methods.

OpenCV (Open Source Computer Vision) is an open-source library of programming functions primarily aimed at real-time computer vision applications. It provides a wide range of tools and algorithms for image and video processing, including object detection and recognition, facial recognition, motion tracking, and more.

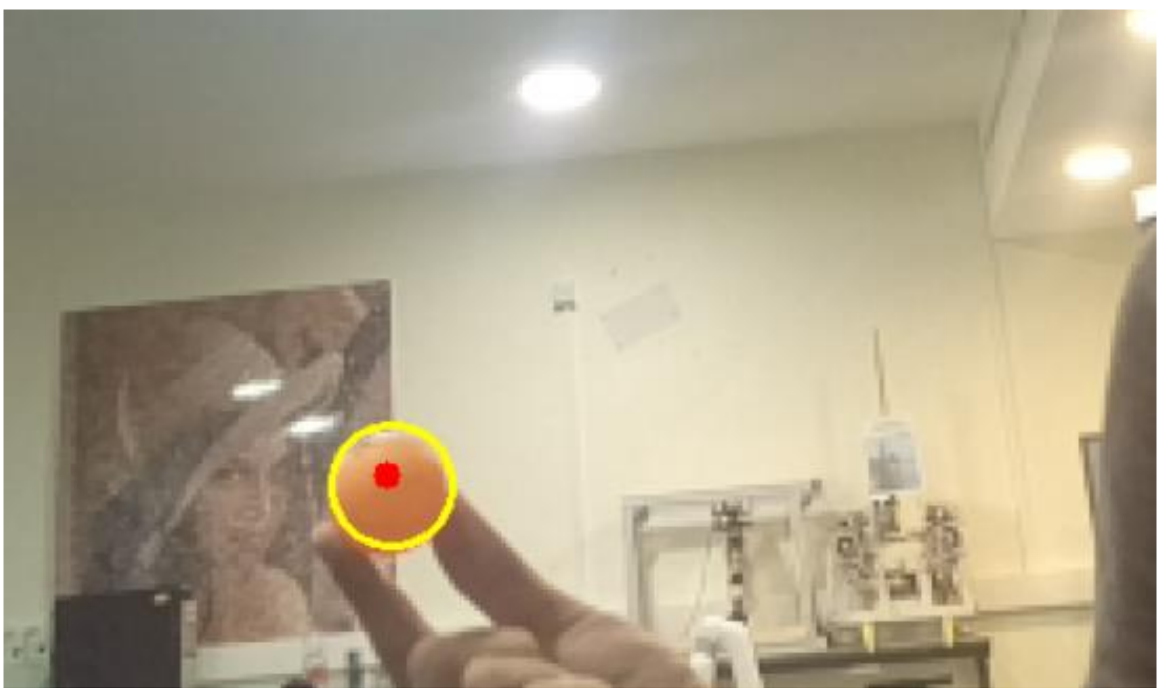



What is done then is filtering by colors. The following picture shows an object and the filter builtby the robot when it sees that object. Once a filter is built the robot adds a contour and a center point to the original object in the original frame. This is now an object that the robot can track.



This GitHub repository explains a lot about the HSV method and gives practical Examples in python

And this article came in as a great help to track the ball around the screen using the HSV method and OpenCV python library.



## 5.2. Controlling the iRobot Create3 with the Jetson Nano

To control the iRobot Create 3 via the Jetson Nano we used the iRobot-edu-python-sdk library and the Bluetooth connection on the Create 3. Since the jetson Nano does NOT have built-in Bluetooth we decided to add the Intel Dual

Band Wireless-AC antenna since everyone who worked with the jetson nano online recommends it. With this added hardware the jetson could connect to the Create3 via Bluetooth:

```
from irobot_edu_sdk.robots import event, hand_over, Color, Robot, Root, Create3
#connect to the robot via Create3 SDK kit
robot = Create3(Bluetooth())
```

After a connection is established between the two the Create3 makes a sound to let the user know that the connection establishment was successful.
The iRobot-edu-python-sdk library uses the “asyncio” library and the commands sent to the robot are non-blocking for example the following loop makes the Create3 makes a rectangle movement:

```
@event(robot.when_play)
async def draw_square(robot):
    await robot.set_marker(Root.MARKER_DOWN)  # Will have no effect on Create 3.

    # The "_" means that we are not using the temporal variable to get any value when iterating.
    # So the purpose of this "for" loop is just to repeat 4 times the actions inside it:
    for _ in range(4):
        await robot.move(6)  # cm
        await robot.turn_left(90)  # deg
    await robot.set_marker(Root.MARKER_UP)  # Will have no effect on Create 3.

robot.play()
```

## 5.3. Controlling the mechanical arm

The arm was made out of cardboard and the joints are controlled by the MR996R Servo Motor. To control the MR996R servo motor using the PCA9685 chip and the Jetson Nano, you can follow these steps:

Connect the PCA9685 chip to the Jetson Nano using the I2C interface. You can refer to the datasheet of the PCA9685 chip and the Jetson Nano pinout diagram for the connections.

Connect the MR996R servo motor to one of the PWM output pins of the PCA9685 chip. You can refer to the datasheet of the MR996R servo motor for the pinout details.

Install the PCA9685 library for Python on your Jetson Nano. You can use the following command to install the library:

```
sudo pip install adafruit-circuitpython-pca9685
```

a Python script to control the servo motor. Here's an example code:

```
import time
import board
import busio
from adafruit_pca9685 import PCA9685

i2c = busio.I2C(board.SCL, board.SDA)
pca = PCA9685(i2c)
pca.frequency = 50  # Set the PWM frequency to 50 Hz

# Define the PWM output pin for the servo motor
servo_pin = 0

# Define the pulse width range for the servo motor
min_pulse = 1000
max_pulse = 2000

# Move the servo motor to the minimum position
pca.channels[servo_pin].duty_cycle = int((min_pulse / 1000000) * 50 * 65535)
time.sleep(1)

# Move the servo motor to the maximum position
pca.channels[servo_pin].duty_cycle = int((max_pulse / 1000000) * 50 * 65535)
time.sleep(1)

# Move the servo motor to the center position
center_pulse = (max_pulse + min_pulse) / 2
pca.channels[servo_pin].duty_cycle = int((center_pulse / 1000000) * 50 * 65535)
time.sleep(1)

# Move the servo motor to the center position
center_pulse = (max_pulse + min_pulse) / 2
pca.channels[servo_pin].duty_cycle = int((center_pulse / 1000000) * 50 * 65535)
time.sleep(1)
```

This code initializes the PCA9685 chip, sets the PWM frequency to 50 Hz, and defines the PWM output pin for the servo motor. It also defines the pulse width range for the servo motor and moves the servo motor to the minimum, maximum, and center positions with a delay of 1 second between each movement.

Note that you may need to adjust the pulse width values for your specific servo motor to achieve the desired movement.

Run the Python script on your Jetson Nano to control the servo motor. You can use the following command to run the script:

```
python servo_control.py
```

This should move the servo motor to the minimum, maximum, and center positions as defined in the script. You can modify the script to control the servo motor to specific angles or positions by adjusting the pulse width values.

## 5.4. Integration

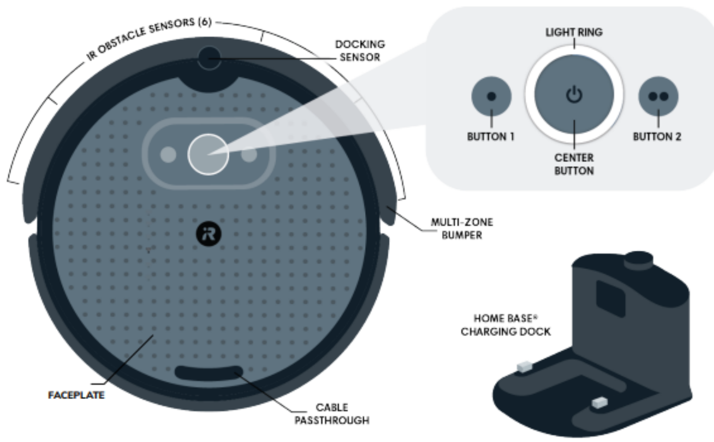
All the tasks must be done in parallel and on time, because of the nature of the problem we are trying to solve. We used the python “threading” library where we created a thread to control each task:

- Thread to analyse the image captured by the camera and locate the ball in the frame and the radius of the given ball.
- A thread to send the controlling commands to the iRobot Create 3
- A thread responsible for the synced communication between the Threads.

After the jetson analyses the frame given to it by the camera feed and it knows where is the ball relative to the robot it sends the appropriate command to the Create3 to move accordingly until the ball is just in the right place to activate the mechanical arm. And the jetson sends a signal to the mechanical arm to collect the ball. After the ball is collected then the jetson sends a signal to the Create3 3 to go and wait in a pre-defined position until a new ball is detected and it repeats this procedure until the button (..) on the Create3 is pressed.

# 6. Operating Instruction

## Hardware Overview




**Home Base™ Charging Dock**
The Create 3 Home Base™ Charging Dock and the Charging Dock cable are used to power on and charge the robot.

**Docking Sensor**
IR sensor to locate Charging Dock.

**Light Ring**
The Light Ring glows different colors and patterns to communicate robot status and/or errors. Users can also program the light.

**Buttons 1 & 2**
Programmable buttons for the user. Button 1 also used to set robot in Standby mode.

**Center Button**
Programmable button that can also be used to power down.

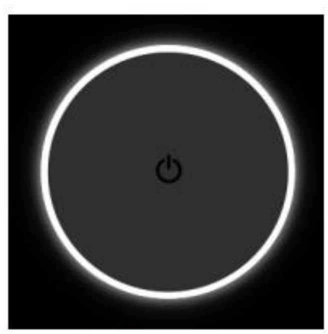
**IR Obstacle Sensors**
Sensors can detect levels of light blocked by obstacles of different material.

**Cable Passthrough**
Thread connectors between hardware on faceplate and bin.

**Faceplate**
Removable top of robot.

**Multi-Zone Bumper**
Bumper contains multiple sensors to detect obstacle collisions.

- Before the start of each table tennis game the Robots have to be on the docking station (charging station) with the camera attached to the jetson nano, and a white light on the light ring.



**Solid White**
Robot is 100% charged

- To Start the collecting process and activate the robot the Stand By button (button 1) a sound will be heard and the robot will undock and the light ring will glow green

- When the game is done and the robot needs to deactivate button 2 must be pressed a sound will be heard and the light ring will glow white

- When The battery of the robot is sub 10% the light ring will glow Red.

**Pulsing Red**
Battery < 10%

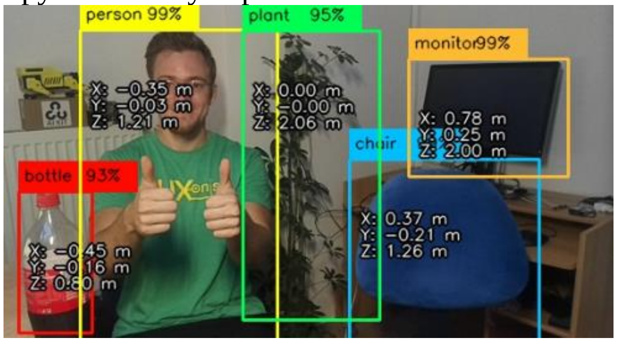
# 7. Failed Attempts

## 7.1. Connecting the Jetson Nano and the Create 3 by a cable

We Tried connecting the Jetson Nano to the iRobot Create 3 By a USB-C to a USB-A cable, but we couldn't make the connection via the Serial Port Work. Tried filling out an Nvidia forum asking for help with this problem and got no response, every year or so the jetson nano gets a system update and it changes all files system, the documentation on the iRobot site mention that the port should be under the "/dev/ttyUSB0" file but we failed to find this port and the lab engineers tried to solve the problem but didn't succeed.
So at the end we decided to mount the Intel dual band wireless-ac 8265 with antennas to connect to the Create 3 wirelessly via Bluetooth.

## 7.2. The Built-in Neural Network in the Oak-d Lite

The Oak-d Lite Camera comes with a built-in neural network model to identify objects, using the python library depth-ai




Although the network works will on its own when running it to try and detects only table tennis balls it ran slowly because it also detects all sort of other things, the code isn't open source so we couldn't modify it to our liking.
In addition to that the camera comes with depth estimation which is achieved using stereo vision, which involves capturing two images of a scene from slightly different perspectives and then using computer vision algorithms to calculate the depth information from the disparity between the two images. This allows the camera to create a 3D representation of the scene. But since the table-tennis balls are so small and the distance to the robot is over 2 meters in some cases the estimation gives inaccurate results.

Therefore we used the HSV method of object detection shown in the previous parts.

# 8. Possible Future Expansions

## 8.1. A more versatile arm

- A more powerful arm so it can collect other kinds of sports balls as well.
- The ability that the arm holds drinks and offers the player refreshing drinks
- Made of stronger materials so that it last longer and the robot could be operated outdoors.

## 8.2. Obstacle avoidance

Obstacle avoidance refers to the ability of a system or device to detect and avoid obstacles in its path. This capability is commonly found in autonomous robots, vehicles, and drones, as well as in various industrial automation systems. There are several ways in which obstacle avoidance can be implemented, including the use of sensors, cameras, and lidar (light detection and ranging) technology. These systems typically work by detecting the presence of an obstacle and calculating its distance and location relative to the system. Once an obstacle is detected, the system can then take action to avoid it. This may involve altering the system's path or speed, or stopping the system altogether until the obstacle has been removed. Obstacle avoidance is an important capability in many applications, as it can help prevent accidents and damage to equipment. In addition, it can enable systems to navigate complex environments more effectively and efficiently.

An example of an Obstacle avoidance algorithm is Tangent Bug.

# 9. References


9.1. AI on the Jetson Nano LESSON 1: Getting Started for Absolute Beginners - YouTube

9.2. Ball Tracking with OpenCV - PyImageSearch

9.3. Create® 3 Docs (iroboteducation.github.io)

9.4. iRobot Education - Python Web Playground

9.5. iRobotEducation/irobot-edu-python-sdk: Python SDK for iRobot Edu robots (Root or Create 3) (github.com)

9.6. dusty-nv/jetson-inference: Hello AI World guide to deploying deep-learning inference networks and deep vision primitives with TensorRT and NVIDIA Jetson. (github.com)

9.7. https://developer.nvidia.com/embedded/jetson-nano-developer-kit

9.8. DepthAI's Documentation — DepthAI documentation | Luxonis

9.9. Color (Image Processing Toolbox) (northwestern.edu)

9.10. Adapting YOLO network for Ball and Player Detection Matija Burić1 , Miran Pobar2 and Marina Ivašić-Kos2

9.11. Tennis Ball Collection Robot Based on MobileNet-SSD Zheqi Zhu, Yingjia Gao, and Shenshen Gu § School of Mechatronic Engineering and Automation Shanghai University Shanghai, China