Project Overview

AI/ML Tennis Ball Retrieval Robot

RoboQuEST
Jacob, Owen, Daniyal & Sean

Introduction

Founded last year in 2022, we are the second group of students involved in Queen Elizabeth's student-led robotics and engineering society. Our main project for this year involves using computer vision to recognise, drive to and pick up tennis balls on a court. This will combine a significant amount of software with robot hardware. Our focus is investigating the use of AI/ML enabled robots in areas where humans are inefficient.

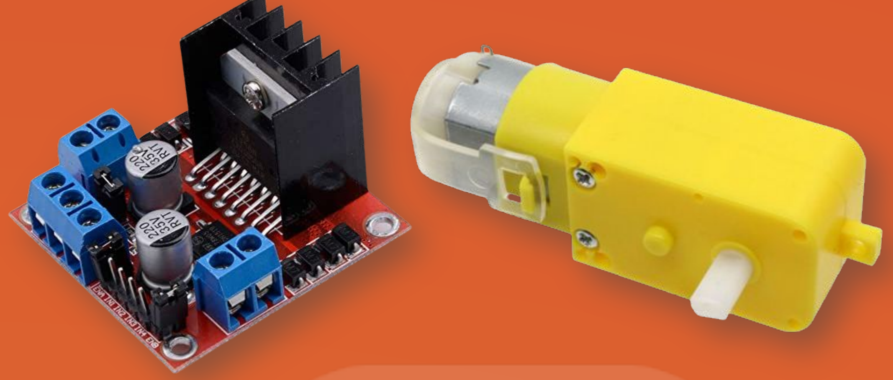
The Hardware

Camera Module

We are using the Raspberry Pi Camera V2.1 module to capture video to send to the Raspberry Pi. This is then downscaled to a suitable resolution for efficient processing, alongside accuracy.

The Raspberry Pi

This is the brain of our robot: the Raspberry Pi Model 4B. It processes the capture from the camera using our trained image detection models, HSV colour calibration and colour masks. This provides an accurate way to search for tennis balls while not being to intensive on the processor. Depending on what it interprets, it sends out signals to multiple motor control boards.



Controlling Movement - Motors and Motor Boards

The project uses L298N motor drivers, each of which control two 3-6V DC motors. The motor driver board takes signals outputted by the Raspberry Pi and moves the respective motor. These boards and motors are powered by rechargeable batteries, for easy maintenance and to be more environmentally friendly.

The Software

We have made a few different iterations of our software. We begun by developing an algorithm based on computer vision and machine learning, which was trained on positive data of annotated tennis ball images. We also used negative training data with no tennis balls or with items that closely resemble them. This allows the model to have a better understanding of what is and what isn't a tennis ball. Once we attempted to transfer this program to the Raspberry Pi, the limitations of its hardware lead to a noticeable delay in the output of the program.

Initial Software - Computer Vision and ML

As mentioned, our computer vision model didn't run very well on the hardware of the Raspberry Pi.

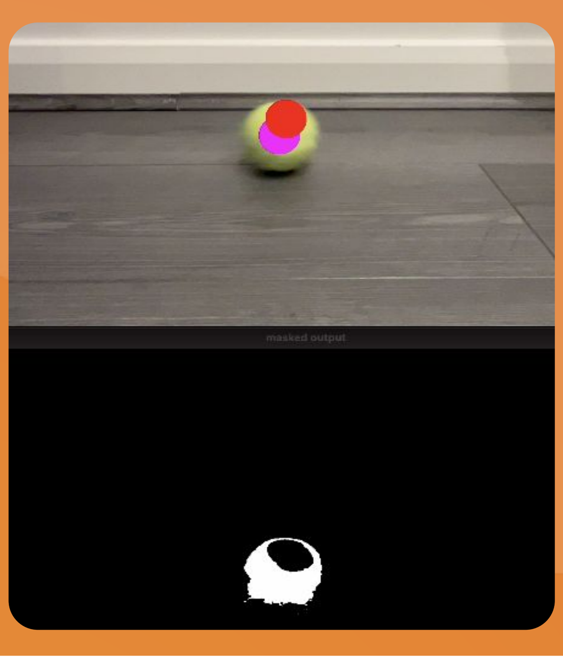
Hardware Causing Lag (ball moving left)

Actual position of the ball | **Believed position of the ball (RPi)**

Above you can see the output of the software running on a laptop compared to the RPi There is a large delay, and this is due to the processing. This would cause the Pi to execute incorrect movements, often just infinitely spinning. Therefore, we needed a new approach.

New Software - Calibration and HSV masking

To reduce dependence on our trained model during runtime, we implemented a solution using HSV (Hue, Saturation, Value) color space, which is highly resilient to lighting variations. By leveraging this, we achieved accurate ball detection with the need for calibration on each use. We developed a laptop-based terminal interface to remotely connect to the robot via its hosted network. Through this interface, users can initiate the calibration mode, which utilizes our object detection model to locate the tennis ball and separate it from the background. By analyzing the color values of the ball, we determine the upper and lower thresholds for its color, considering multiple calibration iterations to account for lighting and environmental conditions. The averaged calibration results are then applied as a mask to the camera input, representing the tennis ball as white and the background as black. By identifying the largest area of white pixels, we place a colored dot at the center of this region in the video feed. The dot's position relative to the robot serves as the basis for determining the ball's location, triggering movement commands accordingly.



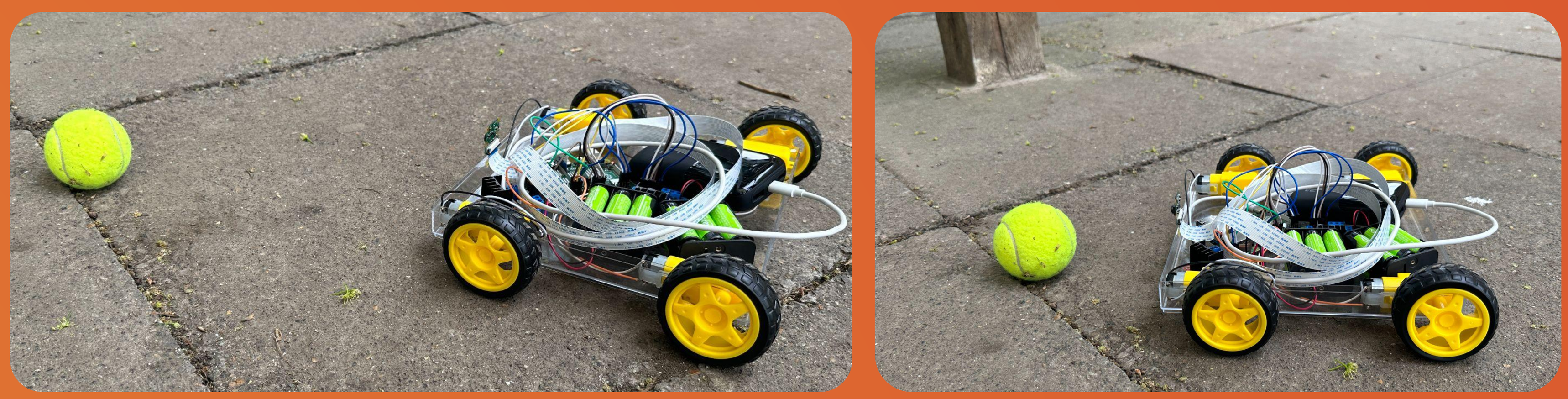
Coordinating movement

Using our mask, we look for large clusters of white pixels - we can assume that these are tennis balls. The larger the clump (closer to the bot), the more pixels, so the higher the confidence. As we are more confident that larger clumps are tennis balls, we should move to pick them up first.

We use image moments to do so and also get the center coordinate of the clump. We define two lines on the screen, one vertical in the center for left and right movement, as well as one horizontal towards the bottom of the screen for movement forwards and backwards. We can check what side of the lines the center is on, allowing accurate movement. We also check how far the center is from the lines, and the further away it is, the greater the speed of the motors.

The Build

Most recently, we've been working on a prototype robot, that successfully scans for, and drives to tennis balls.



The prototype robot turning and moving towards a tennis ball

We adapted our software to run headlessly, meaning that no display is required for it to run and still function correctly. As the robot is now able to move, we adapted the calibration mode slightly. Previously, the user would have to move the bot and tennis ball to account for different lighting conditions. However, the bot can now move itself around to calibrate using different angles of the ball as well as surveying the lighting conditions of the environment.

This solution works well in outdoor and indoor environments, with level ground - which is perfect for tennis courts.

QuEST at Queen Elizabeth's Grammar School

IRIS The Institute for Research in Schools

Want to learn more?

robo-quest.com
quest.molecular.page
jacob@robo-quest.com