# Design and Development of an Automated Robot for collecting a Tennis Ball

Yash Paliwal
*Arizona State University*
Arizona, USA
ypaliwa1@asu.edu

Jonathan Ranjit Thomas
*Arizona State University*
Arizona, USA
jranjitt@asu.edu

Harshit Mandadi
*Arizona State University*
Arizona, USA
hmandad2@asu.edu

***Abstract*—While playing the game of tennis, retrieving the ball is often seen as a tedious task. For this reason, we attempt to design and develop an automated robot that can detect a tennis ball and retrieve it. This could potentially reduce manual labour and make playing the game more enjoyable. We designed an automated robot that detects a tennis ball using machine learning, travels towards it and retrieves it. A Raspberry Pi was used to control the robot, camera and a custom-designed end effector. A video of our results can be found here: https://youtu.be/Ra9vhElyOuQ**

***Index Terms*—Raspberry Pi, machine learning, robot, tennis**

## I. Introduction

A recent area of focus has been the creation of autonomous robots that are capable of performing complex tasks. Robotics, machine learning and artificial intelligence have helped make major strides in technology over recent years. Sports is one sector where automation of certain repetitive processes may help reduce the burden on humans and increase productivity. In this project we focus on retrieving a tennis ball after play

The objective of this project is to provide a solution for automating tennis ball retrieval. We do this by first using machine learning to identify the location of the tennis ball. The robot then travels towards the ball and retrieves it using a custom-designed end effector.

In this project, we focus on solving this problem on a much smaller-scale where the ball is a few metres away from the robot in order to understand the fundamentals of implementing a system like this. From this, we hope to gain knowledge to build an autonomous robot that can be used in more practical scenarios.

In this paper, we go over how we implemented this system. We describe the hardware and software components and evaluate the performance of our system. We also describe the limitations of our approach and future work that can improve the performance of our system

## II. Components

- Adeept 4WD Robot - This robot came with a raspberry pi camera module. It also had a mount for the raspberry pi.
- Cardboard end effector - The end effector drops down once the ball is nearby and surrounds the ball, ensuring that the ball can move along with the robot.
- Raspberry Pi 3B+ - It allows us to remotely control the robot.
- Servomotors - To control the end effector.

The assembled robot, along with the attached end effector, is shown below in Figure 1.

Fig. 1: Adeept 4WD robot

## III. Implementation

The first step was to assemble the robot following instructions from the assembly kit. After this, we configured the Raspberry Pi which allowed us to control the robot remotely. The Raspberry Pi was then mounted onto the robot.

To detect the ball, we build a dataset of tennis ball images taken using the Raspberry Pi camera. We collected and annotated 500 images of the tennis ball at various locations on the campus. To train the model, we used a pre-trained YOLOv8 nano model, due to its fast inference speed. We trained the YOLOv8 model on our dataset for 50 epochs with a batch size of 8 for 4 hours. This gives us a trained model that can detect tennis balls.

The robot can take seven possible actions based on the command it receives.

- *Right* - the robot rotates $10^\circ$ clockwise
- *mRight* - the robot rotates $30^\circ$ clockwise, when ball is not visible in the frame.
- *Left* - the robot rotates $10^\circ$ anti-clockwise
- *lForward* - the robot moves forward 1 cm.
- *mForward* - the robot moves forward 3 cm.

- *Grab* - the robot drops the end effector, to grab the ball.
- *Reverse* - the robot moves backward 3 cm.

We then build a server client setup to communicate between the Raspberry Pi and local machine, where the local machine is the server and Raspberry Pi is the client. The robot captures an image using the Raspberry Pi and sends it to the local machine to predict the location of the ball using the trained machine learning model. Depending on the balls location in the image, the local machine sends a command to the robot. If the ball is not detected initially, the robot rotates clockwise until the ball is visible in the camera's view. The robot keeps rotating until the ball is visible in the frame. The robot then moves right or left based on the location of the ball to ensure that the ball lies in the center of the image.

The robot then moves forward towards the ball. It detects the distance of the ball using the area covered by the ball in the image. If the ball is nearby, the area covered by the ball in the image is greater and this prompts the robot to select the lForward action. If the ball is further away, the area covered by the ball in the image is lesser. This prompts the robot to select the mForward action, since the distance covered by the mForward action is greater than the distance covered by the lForward action. Finally, if the ball is within reach of the end-effector, the robot drops the end-effector and grabs the ball after which takes the ball to the starting location.

We also created a Graphical User Interface (GUI) which allowed us to simultaneously see the image the robot captured as well as the action that it decided to take. This allowed us to get a better understanding of the path that the robot would follow. For example in Figure 2, the robot turns right to try to get the ball to lie at the center of the image.

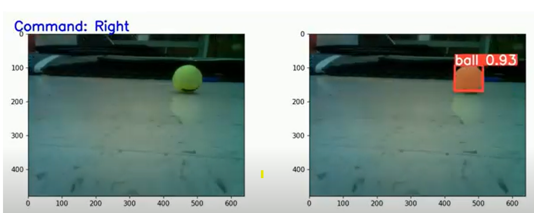



Fig. 2: Graphical User Interface of the robot

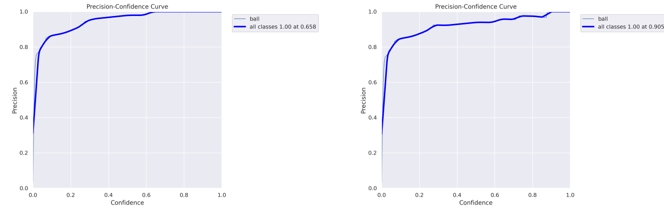
## IV. Results

The results of training on our custom dataset are shown in Figure 3. Figure 3 shows the precision vs confidence score graph for the bounding box and the segmentation mask detection of the tennis ball. The graph indicates that the model is able to perform well as both methods reach a precision score of 1 which shows that the model is able to detect the tennis ball accurately

The output of the model on one of the test image is shown in fig 4. Here, the ball is detected perfectly with a confidence score of 1.

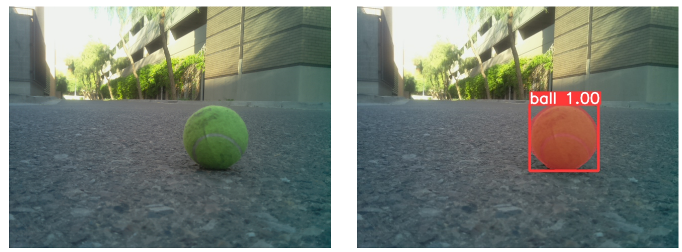
## V. Conclusion

Our robot was successfully able to detect the ball and retrieve it back for the user. This was a challenging experience which helped us apply our skills in robotics, design and programming. We created several programs for detecting the ball and learnt how to control the robotic system using Raspberry Pi. We look forward to applying the skills we have picked up through this project in future endeavours.




(a) Bounding box (b) Segmentation mask

Fig. 3: Precision Vs Confidence graph for bounding box and segmentation mask detection of tennis ball.




(a) Original Image (b) Segmented Image

Fig. 4: The result from YOLOv8 nano model trained on our custom dataset.

## VI. Future Work

Our current approach takes time to implement and requires various components to be connected together using wires. Our end effector and robot is also not robust to retrieve balls of a bigger size. In the future, we would like to build an autonomous robot that can work remotely, is much faster and is also more robust which would be more applicable in real-life sports settings.