Maze Solving Robot

By,

Shantanu Nitin Ghodgaonkar	sng8399	N11344563
Hemant Agrawal	hra5704	N10037896
Mihir Kshirsagar	msk9917	N16144406
Aditya Abishai Pedapati	aap9224	N15101231

New York University, Tandon School of Engineering
Department of Mechanical and Aerospace Engineering

Final Project, Spring 2024

Contents

1 Acknowledgement

We extend our heartfelt gratitude to Professor Vikram Kapila, for his invaluable mentorship and scholarly guidance throughout our academic journey. His expertise and dedication have inspired us, shaping our understanding and passion for the subject.

To our teammates, Shantanu Ghodgaonkar, Mihir Kshirsagar, Hemant Agrawal and Aditya Abishai P we appreciate the collaborative effort and camaraderie that made our academic endeavors are even more rewarding.

This journey would not have been possible without the support and encouragement of these exceptional individuals. Thank you for being constant pillars of knowledge and inspiration in our academic endeavors.

2 Abstract

This project amalgamates the capabilities of Raspberry Pi 4B and Arduino Uno microcontrollers to create an innovative maze-solving robot.Utilizing OpenCV for image processing and implementing breadth-first search algorithms on the Raspberry Pi, the system intelligently navigates the ball towards the maze's exit. By employing two servos controlled by the Arduino, the project orchestrates the movement of a ball through complex mazes.

3 Project objectives

1. Develop a reliable ball detection algorithm using OpenCV to accurately identify the position of the ball within maze and differentiating walls of maze from base.
2. Implement preprocessing techniques to facilitate maze detection.
3. Utilize contour detection to precisely locate the center of the ball in maze.
4. Employ breadth-first search (BFS) algorithms on the Raspberry Pi to plan optimal solution path for navigating the ball through the maze.
5. Integrate Arduino-controlled servos to physically manipulate ball in response to navigation commands from the Raspberry Pi.
6. Setup communication protocol between the Raspberry Pi and Arduino.

4 Introduction

Maze solving has long been a quintessential problem in the realm of robotics and artificial intelligence, serving as a fundamental testbed for developing navigation algorithms and autonomous systems. In this project, we plan to harness the power of computer vision, motion control, path planning as we aim to design a robust system capable of autonomously navigating a ball through a maze. Our approach involves the integration of OpenCV-based image processing techniques with Raspberry Pi-driven path planning algorithms and Arduino-controlled servos for physical manipulation.

5 Key Components

5.1 Raspberry Pi 4

The Raspberry Pi 4 Model B 4GB is a highly versatile single-board computer (SBC) designed to offer exceptional performance and functionality in a compact form factor.

5.1.1 Characteristics

1. Processor: Powered by a quad-core ARM Cortex-A72 processor running at up to 1.5GHz, the Raspberry Pi 4 B 4GB delivers impressive processing power for various computing tasks.
2. Memory: Equipped with 4GB of LPDDR4 RAM

3. Graphics: The VideoCore VI graphics processor enables smooth graphics performance, supporting multimedia playback, gaming, and graphical applications.

4. Display: With dual micro HDMI ports, this model supports dual 4K displays at 60 frames per second, making it suitable for digital signage, media centers, and productivity setups.

5. USB Ports: Two USB 3.0 ports and two USB 2.0 ports provide data transfer and connectivity for peripherals such as keyboards, mice, storage devices, and more.

6. GPIO Pins: The Raspberry Pi 4 B 4GB retains compatibility with the extensive ecosystem of Raspberry Pi HATs (Hardware Attached on Top) and GPIO (General-Purpose Input/Output) accessories, allowing for hardware expansion and customization

7. Storage: MicroSD card slot for booting the operating system and storing data, offering flexibility and expandability in storage options.

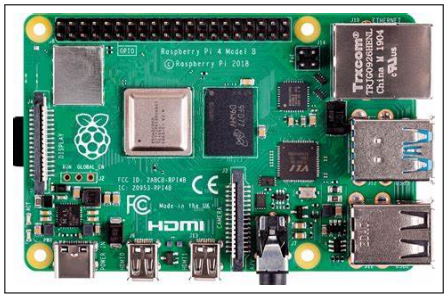

Figure 1: Raspberry Pi 4 Model - B

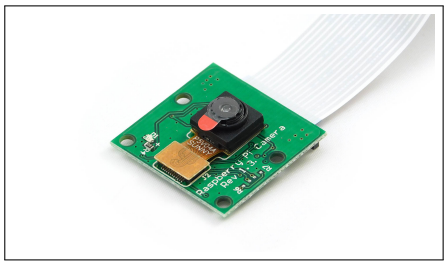

Figure 2: Raspberry Pi Camera

5.2 Raspberry Pi Camera

The Raspberry Pi Camera Module v1.3 is a compact and versatile camera accessory designed specifically for Raspberry Pi single-board computers. With its small form factor and plug-and-play functionality, this camera module offers users an easy way to add imaging capabilities to their Raspberry Pi projects.

5.2.1 Key Features

- Image Sensor: The camera module features a fixed-focus 5-megapixel OV5647 image sensor, capable of capturing high-resolution still images and video footage.

- Lens: Equipped with a standard flat ribbon cable connector, the camera module allows for flexible positioning and mounting options. The lens provides a wide-angle field of view, making it suitable for various applications.

- Resolution: Capable of capturing still images with a maximum resolution of 2592 x 1944 pixels (5 megapixels) and recording video in 1080p Full HD resolution at 30 frames per second (fps), delivering sharp and clear imaging quality.

- Low-Light Performance: The camera module's sensor and lens design provide decent low-light performance, allowing for clear imaging in various lighting conditions

5.3 Arduino Uno

The Arduino Uno is a popular microcontroller board designed for hobbyists, students, and professionals. Launched by Arduino LLC, it serves as an accessible and versatile platform for building a wide range of electronic projects.

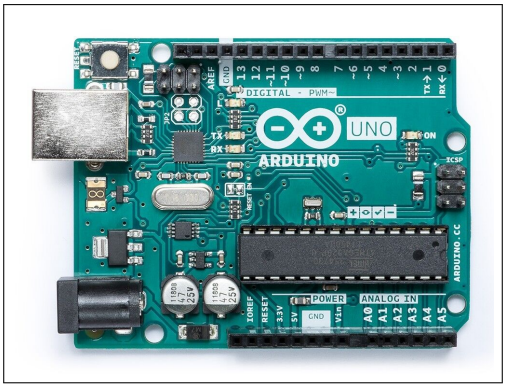

Figure 3: Arduino Uno

5.3.1 Specifications

- Microcontroller: Powered by the ATmega328P microcontroller, the Arduino Uno provides a reliable and capable processing core for executing code and interfacing with external hardware components.
- Clock Speed: Running at a clock speed of 16 MHz, the Arduino Uno offers sufficient computational power for a variety of tasks, including sensing, processing, and actuating.
- Digital and Analog I/O: Equipped with 14 digital input/output (I/O) pins, among which 6 can be used as pulse-width modulation (PWM) outputs, and 6 analog input pins, the Arduino Uno enables interfacing with a wide range of sensors, actuators, and peripherals.
- Power Options: Supporting multiple power options, including USB power, external DC power, and battery power, the Arduino Uno offers flexibility in powering projects based on their intended use cases and environments.
- Ease of Use: With its beginner-friendly design, straightforward programming language (based on Wiring and C/C++), and extensive documentation

5.4 MG996R Servo Motors

Two servo motors are used, each responsible for controlling rotation along one axis. Servo Motors are preferred due to their precision, ease of control.

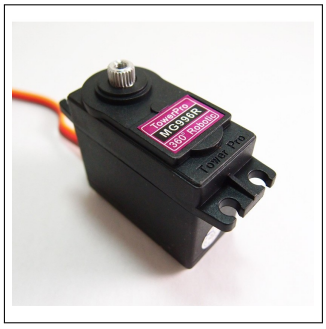

Figure 4: MG996R Servo

5.4.1 Key Features

- Operating Voltage is +5V
- Current: 500 - 900 mA
- Stall Torque: 9.4 kg/cm (at 4.8V)
- Gear Type: Metal
- Rotation : 0° to 180°
- Maximum Stall Torque: 11 kg/cm (6V)

6 Image Processing and Object Detection

In order to accurately localize the position of the ball within the maze, a series of image processing steps are employed. First, the camera sensor's exposure and gain settings are calibrated to ensure optimal image quality and contrast. Next, the captured image undergoes rotation and cropping to focus on the relevant area of interest within the maze. Following this, the image is converted to a suitable color space, to facilitate better detection of the ball. A masking technique is then applied to eliminate glare and enhance the colors, ensuring that the ball stands out against the background of the maze walls and path. This masking process isolates the ball while removing other colors from the image. Finally, the center and boundary of the ball are detected using the Hough Circles algorithm, allowing for precise localization within the maze. This detection process is repeated after every step taken by the ball, ensuring continuous tracking and accurate navigation within the maze environment.

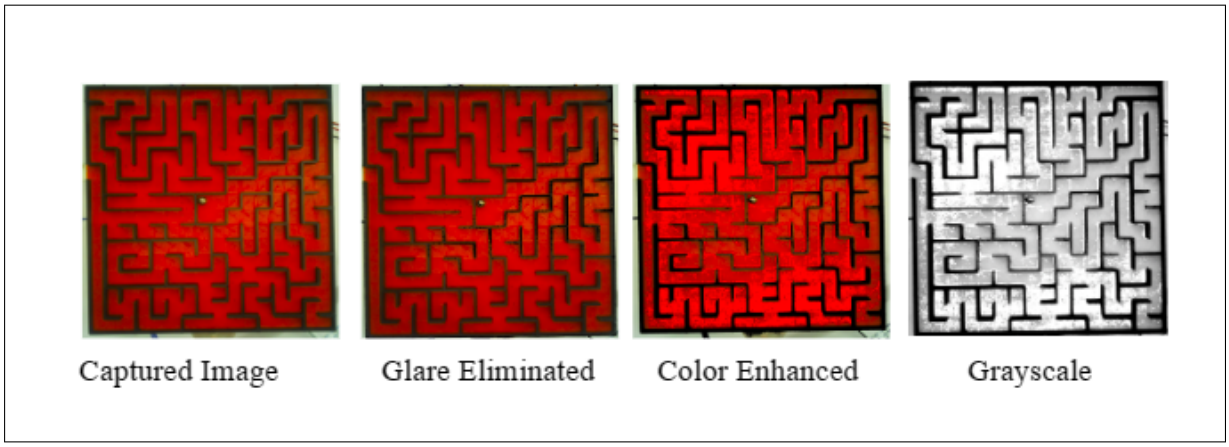

Figure 5:

7 Breadth First Search Algorithm in Maze Solving

In the context of maze solving, BFS is used to find a solution path from a starting point to a destination point (or an exit). The maze will be represented as a graph, where each cell is a vertex, and the passages between cells are edges. By traversing the maze using BFS, we can efficiently find a solution path from the start cell to the end cell.

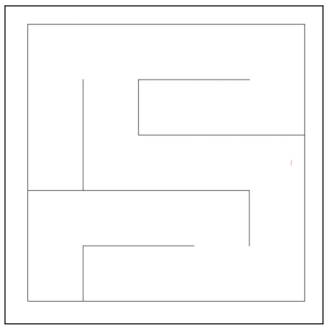

Figure 6: Sample Maze

Ultimately, the maze could be represented as the graph shown with elements explained as below:-

- **Vertices (Nodes): The red circles symbolize specific points or locations within the maze, such as intersections, corners, or individual cells.**
- **Edges: The black lines connecting the vertices represent the pathways or passages between these points. Each edge indicates a connection between two adjacent points in the maze.**

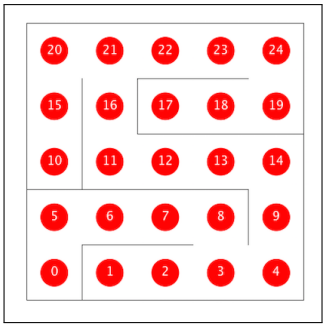

Figure 7: Representation of the maze as Vertices

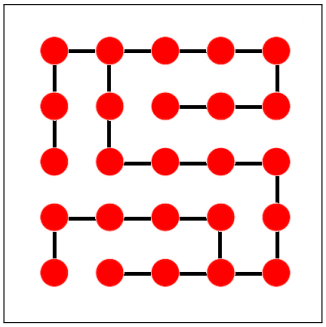

Figure 8: Representation as a Unidirectional Graph

8 Incorporation of BFS Algorithm for Maze Solving

Initialization:

- We begin by initializing a queue, which will store the vertices (cells) to be visited during the maze traversal.
- We also initialize a data structure (such as a matrix or list) to keep track of visited vertices to avoid revisiting the same cell.
- We enqueue the starting vertex (cell) into the queue and mark it as visited.

BFS Traversal:

- While the queue is not empty, we repeatedly dequeue a vertex from the front of the queue.
- For each dequeued vertex, we explore its neighboring vertices (cells) that have not been visited yet.
- For each unvisited neighboring vertex, we enqueue it into the queue and mark it as visited.

9 Integration of Arduino and Raspberry Pi

Combining the Python and Arduino code involves establishing communication between the two platforms and coordinating actions based on the exchanged data

9.1 Raspberry Pi(Python) Code

- Capture an image using the camera.
- Process the image to detect ball and determine a solution path in the maze.
- Send instructions to the Arduino based on the detected solution path.

9.2 Arduino Code

- Receive instructions from the Raspberry Pi over serial communication (UART).
- Control the servos to execute the specified rotations.

9.3 Overall Flow

- The Raspberry Pi captures an image and processes it to determine ball position and a solution path in the maze.
- Based on the relative position of ball with respect to the solution path, the Raspberry Pi sends instructions to the Arduino over serial communication.
- The Arduino receives the instructions, controls the servos accordingly.

10 Circuit Schematic

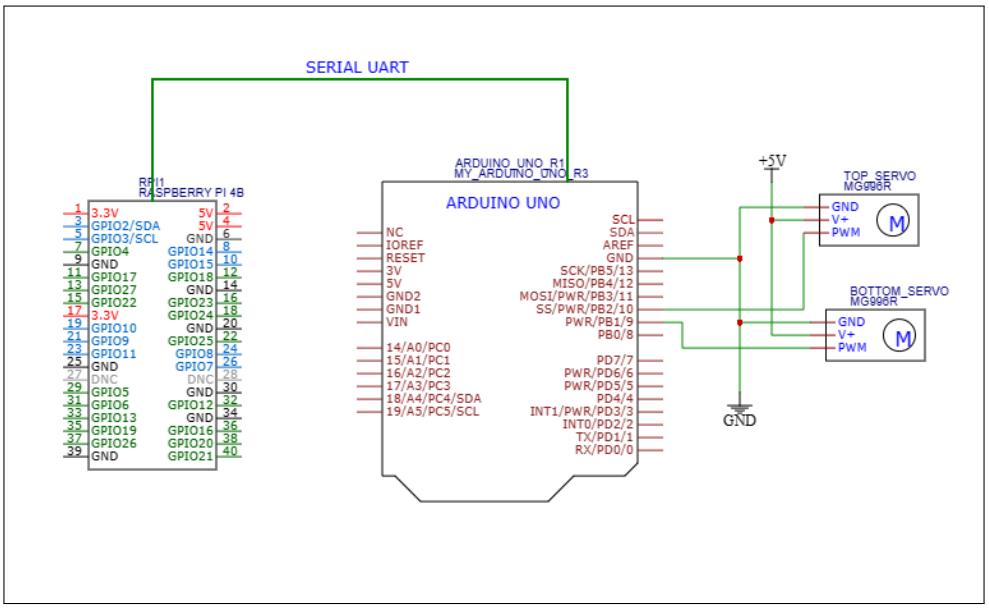

Figure 9: Circuit Schematic

11 Flowcharts

This section illustrates the flowcharts responsible for the control of the ball's navigation through a maze using image processing and Arduino communication.

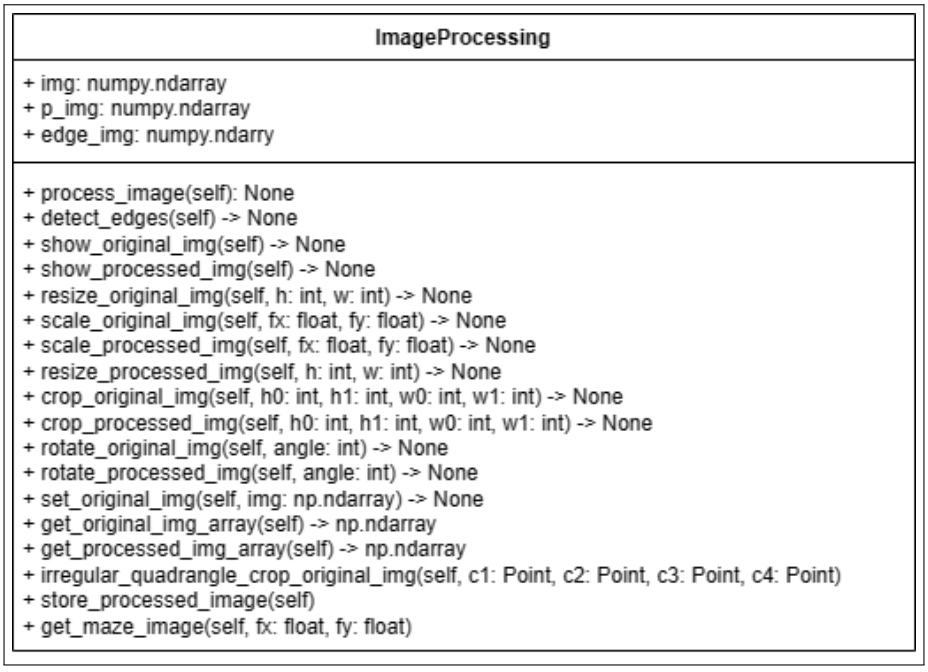

Figure 10: Image Processing Handler Class Structure

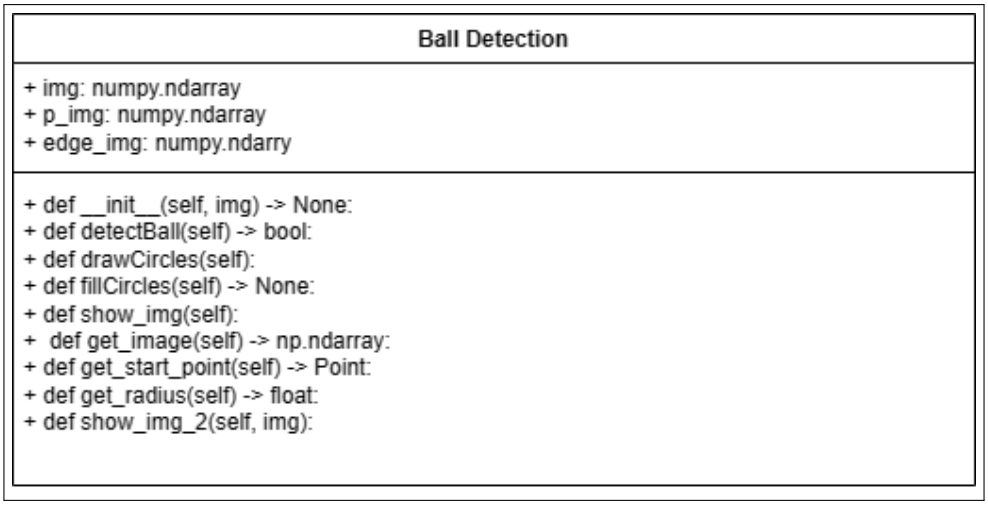

Figure 11: Ball Detection Handler Class Structure

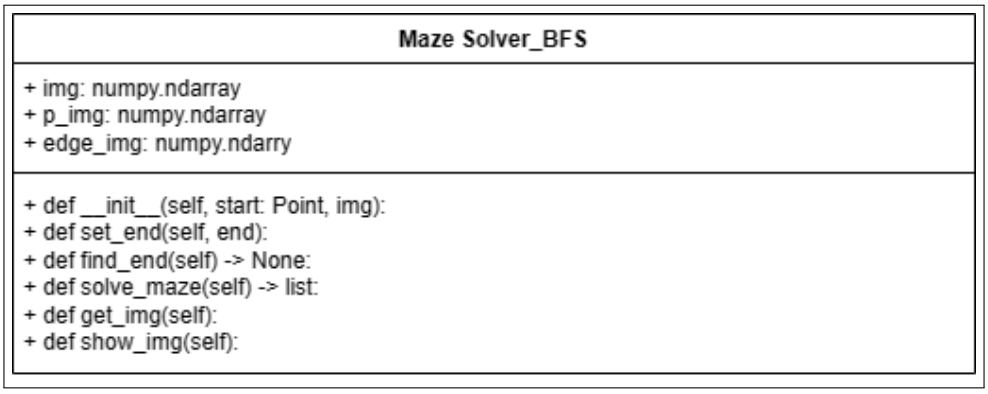

Figure 12: Breadth First Search Algorithm Class

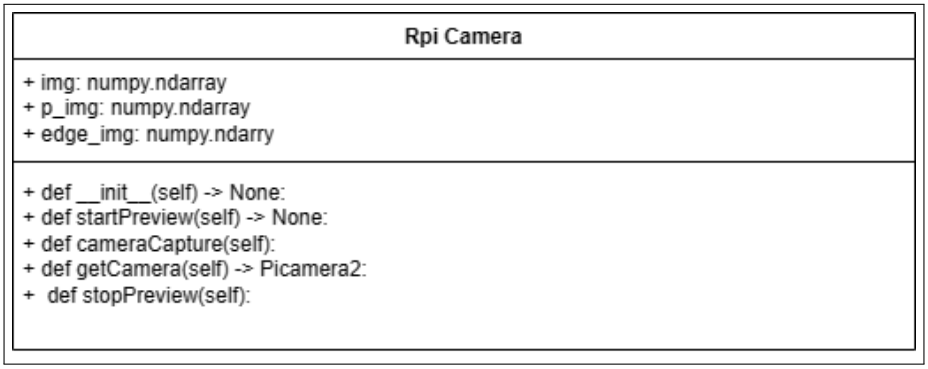

Figure 13: RPi Camera Handler Class Structure

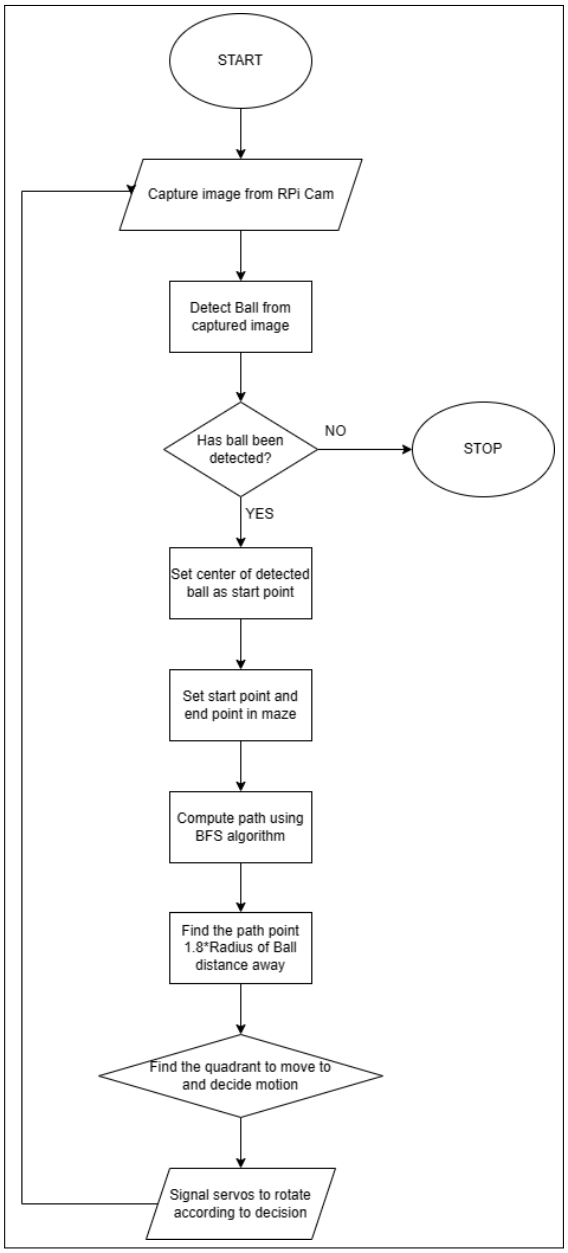

Figure 14: Main Function

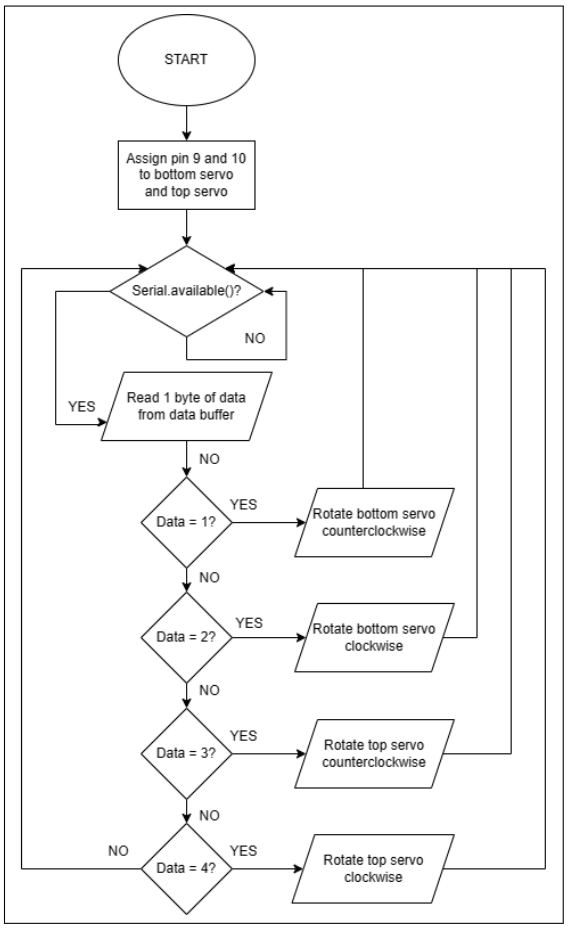

Figure 15: Arduino Firmware Flowchart

12 Results

This section illustrates the results of our project. The following images display the captured and processed images taken from Rpi Camera.

Figure 16: Image captured from Rpi Camera

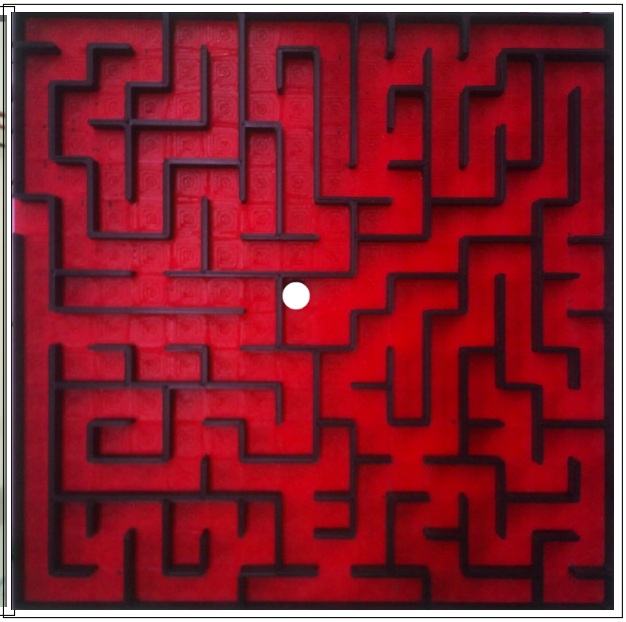

Figure 17: Image highlighting the ball

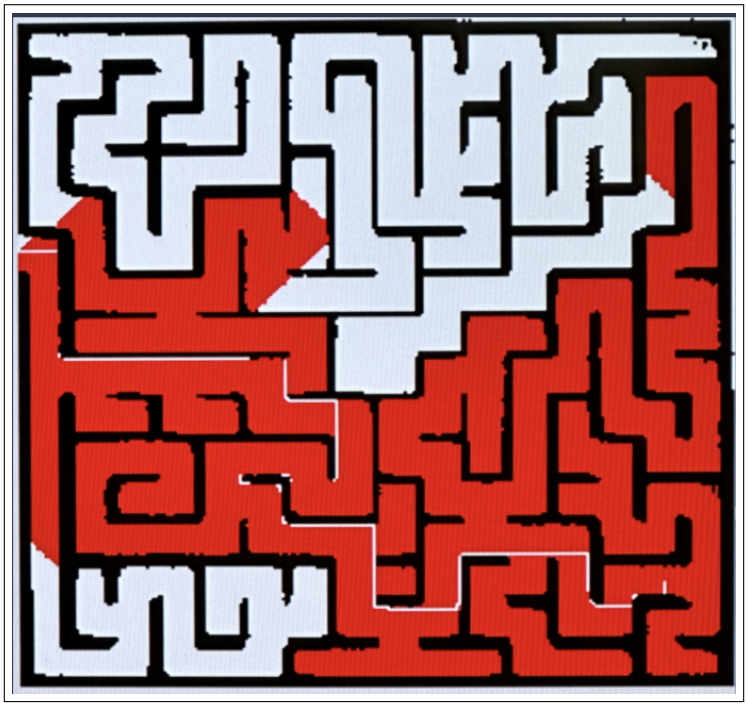

Figure 18: Solution path given by BFS algorithm

13 Future Enhancements

Looking forward, there are two areas to enhance and refine this project.

Design of the System: Integrating this with Stewart platform aka Hexapod and a better camera will significantly improve the results.The platform will help in the dynamic positioning and orientation of the camera.

Optimization of solution path Planning Algorithms: Various algorithms were tested for this project and finally BFS was chosen for it's performance. With continuous refinement and fine-tuning the algorithms, the system can adapt to dynamic maze patterns.

Machine Learning: Implementing machine learning will significantly improve the image processing and ball detection.