Alvin Alaphat and Luke Marushack

Raspberry Pi Car Following Object

Dec 2, 2022

Semester Project: Part 4 Report

1) Our project involves the camera of a Raspberry Pi to detect a tennis ball so it can tell the car's motors to follow after it. We chose the Hough transform detection algorithm because it's most commonly used for circular objects and it excels in detecting when there is a sharp contrast between the pixel colors along the detected edges such as in our application. We ended up using a modified version called HoughCircles which takes a grayscale image and multiple other parameters that we were able to set based on trial and error with our training data of tennis ball pictures and live video capture. The parameters we set for the function are Hough Gradient (detection method), inverse ratio of the accumulator resolution, the minimum distance between the centers of the detected circles (to avoid overlap but this doesn't apply here), accumulator threshold, minimum circle radius, and maximum circle radius. While Hough Transform can have its limitations such as not performing well under noisy images, we don't expect there to be excess noise in our testing environment. Canny edge detection is used implicitly by Hough Transform and Hough transform uses only 3 simple hyper parameters in its voting algorithm to detect circles that are in contrast with the surrounding pixels.

2) The classification accuracy for the model is contingent on a live video stream which doesn't operate the same as a static image for us to take a consistent accurate measurement of. However, as denoted by Figure A below, we can estimate this measurement by taking a still image of the live video capture. We measured this against our ground truth of the actual sphere around the tennis ball and came to .945 overlap accuracy. This is bound to vary as

the ball moves around in the live capture of our robotic car but from multiple tests, the number consistently hovered around .94 accuracy.

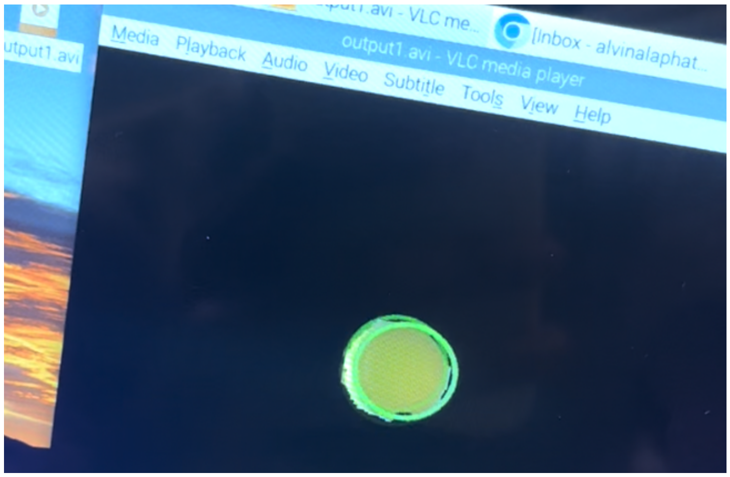

Figure A. Intersection over Union Detection

3) The observed accuracy of the object recognition model is consistent in recognizing the ball regardless of depth and location within the x/y axis. The software is able to control the car and move it forward to go left and right after recognizing the ball through the camera. There are some improvements that we'd definitely like to see for the final iteration which includes smoother turning and quicker recognition of the tennis ball. Sometimes it stalls before it's able to decipher what it should be doing so the process can be more reactive and smoother. Another thing we'd like to improve is the interface of the Raspberry Pi's wheels which sometimes struggles to get turning unless it gets given a push to start. This might be due to poor wiring or poor weight distribution on the chassis so we want to optimize the design of the car so that the electrical circuits work a bit better without sputtering. Finally, we've had to tether our car to an outlet since the power source we were given for the Pi required to be connected to an outlet. This has

restricted our range of motion so we'd like to get a portable power source to allow for more free range of motion. Shown below is the final product in action. With the improvements we have in mind, we expect the car to rival Elon Musk's Tesla.

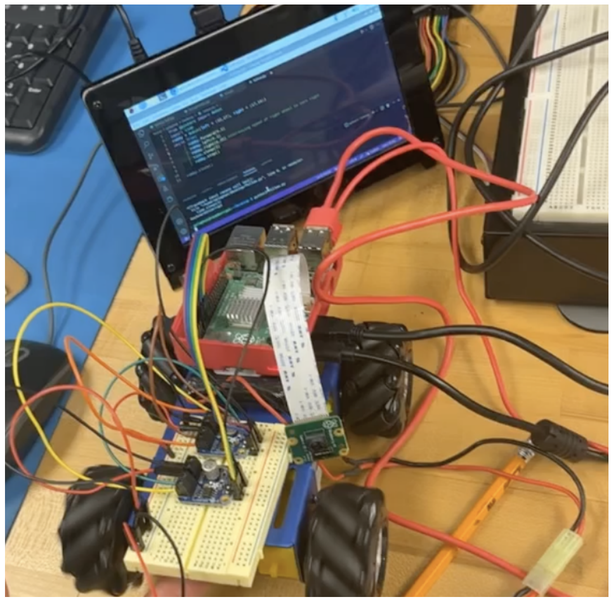

Figure B. Assembled Car Chassis Operating with Code