**What was the goal of your project?**

The main goal of our project was to explore the idea behind autonomous cars and practice our knowledge in computer vision. We wanted our robot to detect traffic signs on the road and react appropriately. We did this through matching keypoints to stored images and detecting color and motion on the road.

**Describe how your system works. Make sure to include the basic components and algorithms that comprise your project.**

We used two algorithms- one for detecting traffic signs and one for detecting motion. For traffic signs, we focused on detecting stop signs and yield signs. For stop signs, we first used match keypoint that Paul created to compare a stored stop sign image with the video input. We minimized background noise, by filtering out everything that wasn’t red. This ensured that everything that was matched was only from the stop sign. Then we counted how many key points were matched and return true if the number of keypoint matches was greater than a threshold. Finally, once the robot receives the message that it “sees” a stop sign, it will try to detect a rectangle in front of it on the floor through shape detection using a Hough Line Transform, and will then stop appropriately.

For the yield sign detection, we used the same algorithm to “see” the yield sign. Then, once the robot confirms that there is a yield sign in front of it, we detect motion on the robot’s side by subtracting differences between consecutive frames and capturing the differences if it is bigger than certain threshold. Once anything appears to be moving on the side of the robot, we return true and the robot will stop for certain time and then keep going straight. If the robot sees the yield sign, but there is no motion detected on the side, it will keep going.

Our powerpoint (https://docs.google.com/presentation/d/1fsRl0Md4HTegYGZjlwrliAitAzLih3Qtsuq5fMPtQvY/edit?usp=sharing) has a step by step explaination and code screenshots.

**Describe a design decision you had to make when working on your project and what you ultimately did (and why)? These design decisions could be particular choices for how you implemented some part of an algorithm or perhaps a decision regarding which of two external packages to use in your project.**

In order to detect the signs, we implemented a filter that removes undesired color after we run match keypoints. We included this in an effort to increase the accuracy of our sign matching by filtering out noise. We thought of three different method to implement this color filter. 1. filter color and then match keypoints, 2. filter color after matching keypoints. 3. pull out just the stop sign from the video feed, and then filter the color. We decided to filter color after match key points, because then the filter would only need to be applied to the keypoints, rather than all the pixels in the image. This option also made the most sense in our code structure because we were able to easily integrate the color filter into our matchKepoints python file.

Another design decision that we had to make was with motion detection. We realized that the stationary background of the frame will always be moving if the robot itself is moving. We were thinking of implementing an algorithm that would solve the problem, but due to time constraints, we instead decided to stop the robot first when it sees the yield sign, detect motion, and then react accordingly.

**How did you structure your code?**

We have one main driver class called receiveImage. This class calls three other files

1. matchKeypoints, which computes similar points between two pictures. This returns a list of keypoints.
2. Hough, which detects a rectangle, returning a boolean
3. MotionDetection, which detects if there is motion in front of the camera, and also returns a boolean

We used these classes in our conditionals that recognized a stop sign and a yield sign.

**What if any challenges did you face along the way?**

Our biggest challenge was simply getting started. We were a bit lost near the beginning about where to begin, but a conversation with Paul helped us figure out some concrete first steps. Paul’s example code helped a lot, especially regarding how to take an image in from the camera. We also had several smaller challenges, such as trouble implementing color detection since the RGB values were the opposite of what we were used to. That bug took us a long time to figure out. We also struggled with integrating our motion detection code with the rest, mainly due to problems with storing images of consecutive frames.

**What would you do to improve your project if you had more time?**

If we had more time, we would like to implement the algorithm to detect motion while the robot itself is in motion. Additionally, we would love to implement object recognition in our algorithm to recognize the difference between human and another robot.

**Did you learn any interesting lessons for future robotic programming projects? These could relate to working on robotics projects in teams, working on more open-ended (and longer term) problems, or any other relevant topic.**

We learned that when working on more open-ended problems, it is important to define our goals first. We had some trouble doing this at the beginning, but once we defined our goals, it was a lot easier to begin implementation because everyone knew exactly what the team wanted. Additionally, we learned when to integrate code that we or others have already implemented vs writing our own code from scratch. In our program to detect signs, we were able to use Paul’s code by importing it, which left us more time to explore other algorithms and expand the scope of our project. It was hard working on an open-ended problems since we had to set our own deadlines, but it was a great experience to design our own problem and implement a solution.