Christy Jacob
CS 6384.001

Computer Vision Project 1

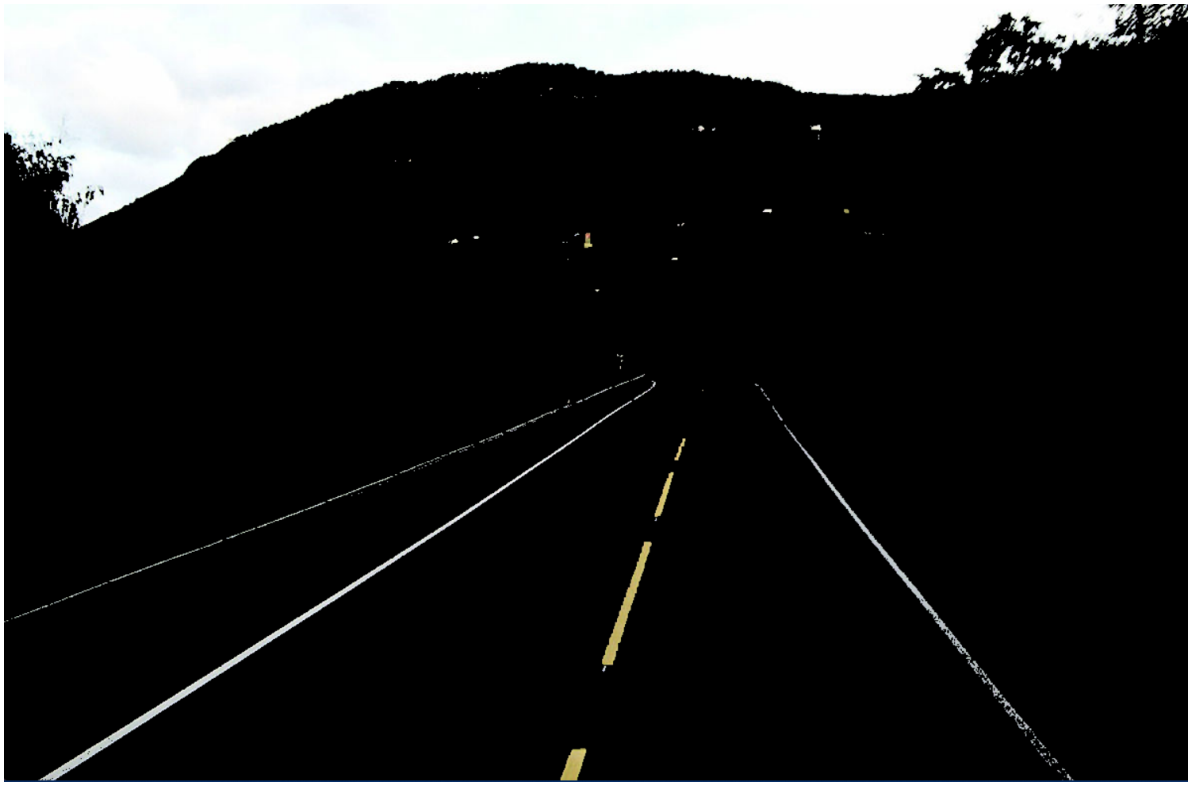
For the computer vision project, we had to take multiple images and return the lanes detected in each image. It was a difficult project because it involved exploring different methods for lane detections, tuning parameters, and putting it all together to display the number of lanes. The approach I took was to first use masks to filter out the color, then remove the noise, then perform edge detection, and finally use Hough Line Transform to get the lanes. We were given 5 images to test our project on. These images didn't always have the similar regions of interest so I was unable to make use of that to eliminate noise.

The first part of the project involved using masks to filter out the colors. The specific colors we were targeting were white, yellow, and red because those are the lane lines we were trying to get. The first step was reading in the images from the folder using Python. Then, I got the height and width of each image for later calculation. One of the sample images was too big to fit the screen so I had to resize the image for testing purposes. I resized the image and maintained the same aspect ratio by dividing the height and width of the image by a constant which was, in my case 4. Then, I had to do some experimenting and exploring to figure out the best way to filter out the colors I was targeting. I started by figuring out whether to convert the image to RGB, GRAY, HLS, or HSV. I tried filtering out each color using RGB, GRAY, HLS, and HSV, and found out that some had better results than the others. I found that using RGB, I could filter out white and using HSV I would have an easier time filtering out yellow. Below is an example image and how it looks in RGB and HSV.

Christy Jacob
CS 6384.001

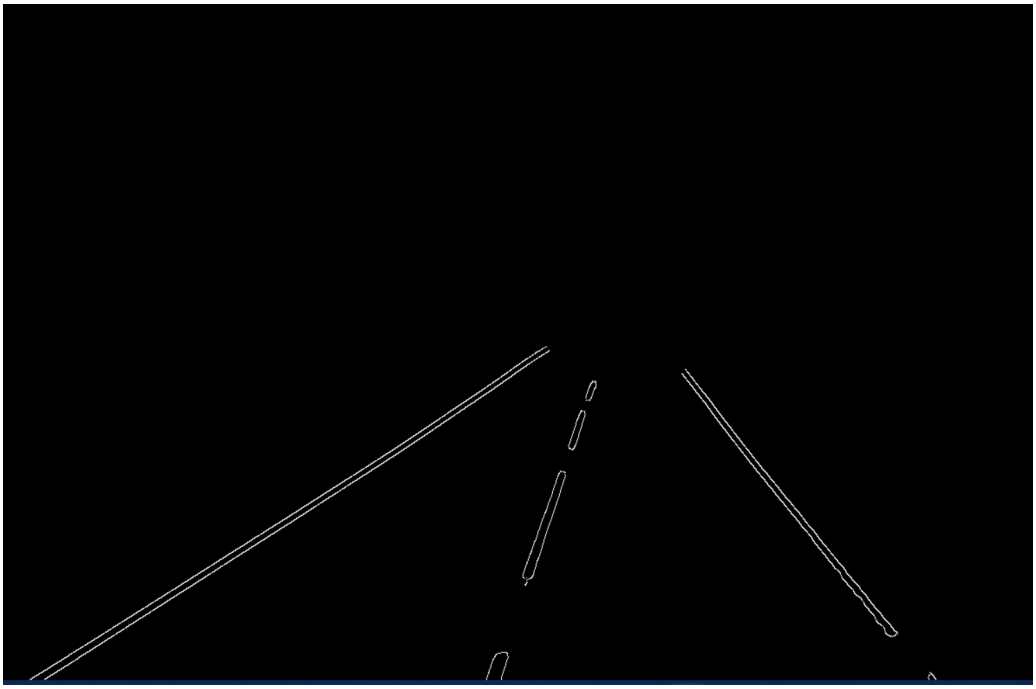
The next part involved finding the lower and upper thresholds to use to get the white and yellow colors. After toying with the thresholds I found the best ones and used them for the masks. Then, I used bitwise-or to get the union of the masks to get all the colors I was going for. Next, I used bitwise-and to apply the mask to the original image. Below is the image with the final mask.

Christy Jacob
CS 6384.001



Then, I converted the image to gray scale and performed binary thresholding to make the image black and white. This is to help with the edge detection process.

I also had to perform blurring multiple times to help get rid of the noise. Then, I used the Canny edge detector to find the edges to use for lane detection. After that, I filtered the image with a region of interest. I used fractions of the image to get the regions of the road you would see if you were looking at a road. Below is an example of the image after applying blurring, edge detection and the region of interest.



Finally, I performed a Hough Transform on the image to get the lane lines. This returns multiple lines based on the parameters you use. I had to filter the lines based on slope as well to only get lines that were not horizontal. Then, I was still getting too many lines, so I had to group the lines together. I decided to do that by grouping the lines that had similar slopes and had similar vertices. Below is the image with the clustering applied. Because of the image processing that I did, especially the region of interest and filtering out the lines, the full lane will not be detected but only the part of it. In a real world application, you could extend the lines if needed by graphing based on the slope.

Christy Jacob
CS 6384.001

The most difficult part of the project was experimenting with the different methods and tuning the parameters to find a combination that would work on all the images. There were combinations that would work perfectly for a few images but fall short on the others, so there was a lot of effort put into that. I also had to struggle to find a way to reduce the noise without eliminating some of the road lanes. There was also a problem where extra lane lines would show up when using Hough Transform Line, so a lot of work went into finding ways to filter out the lines and combine similar lines. I learned a lot about the application of Computer Vision through this project. Lane detection is a real world problem that will be needed in the future for self driving cars so it was fun learning about that. I also got a more in depth knowledge of a few of the methods we had learned in class such as Hough Transform as well as learning a few new ones such as Canny.