ENPM 673 Perception for Autonomous Robots

Project 2

Shelvin Pauly

118426556

Google Drive Link for Audio/Video Outputs: https://drive.google.com/drive/folders/1eOBxEkS5C-bqQfX5p5GbCccO6xbMWQC_?usp=sharing

Answer 1.

Part A. Histogram Equalization

In this question, I try to create a histogram of intensities in the form of a 1D array. Using this histogram, the cumulative sum for each pixel is found. Then, I find the cumulative density function for each pixel which when multiplied by the pixel value gives the final image. This image processing technique helps us to bring back major details that are lost due to over-exposure. Moving a skewed histogram of intensities to more like a normal distribution.

Part B. Adaptive Histogram Equalization

The logic remains the same in this section, but I apply histogram equalization after diving the image into 8 equal parts. This method works better as it does not boost the noises as classical histogram equalization does. However, this technique is computationally expensive.

The difficulty that I faced is after dividing the image and while joining it back, the image does not feel continuous. Another problem I faced is reading all the images one by one and processing it into a video. I still have no clue why the video when played says could not demultiplex stream. I couldn't find answers online as well.

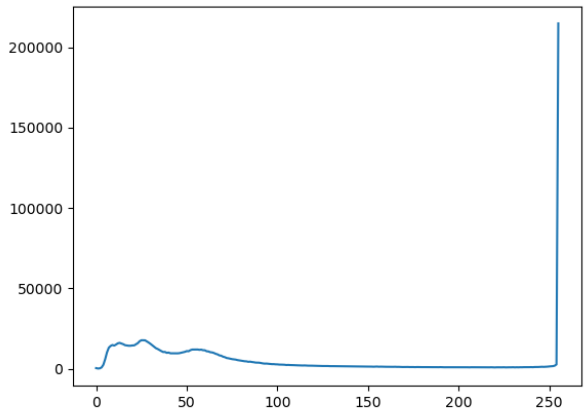

Figure 1. Histogram for frame 1

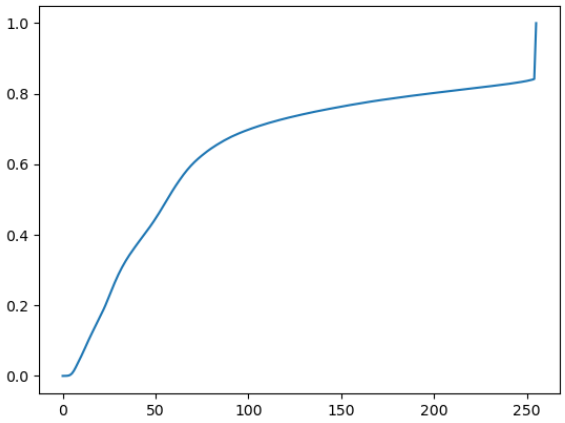

Figure 2. CDF for frame1

Figure 3. Histogram Equalized frame 1

Figure 4. Histogram Equalized frame 1

Answer 2. Straight Line Detection

Here, I get the edges of the lanes using the thresholding and canny edge detection method. Then I transform the perspective to the bird's eye view using getPerspectiveTransform and then warping it with the transformation matrix. The left and the right lines are detected using HoughLinesP. Next, I create a histogram of the first axis of this image. The half with the higher sum will decide whether if it is a dashed line or straight line. Now, I plot the lines using cv2.line. I have used the addWeighted method for blending the lines into the frame. The left and the right lines are detected using HoughLinesP.

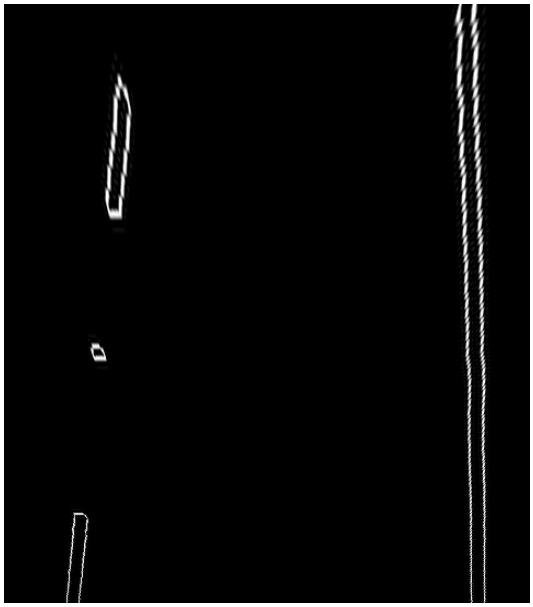

Figure 5. Bird's Eye View after canny for frame 2

Figure 6. Final frame for frame 2

The output video can be found here:
https://drive.google.com/file/d/1toIBYl1clqlh6XVAC_NmHODbLjdq8O4E/view?usp=sharing

Answer 3. Predict Turn

I have divided this question into two following parts: pre-processing, detection and turn prediction,

For pre-processing, I just use basic thresholding and distributing the channels while converting from binary to RGB for the final output. Then I use homography matrix and change our perspective to the bird's eye view with the use of the matrix.

For yellow, I use the HSV space, I find out the pixels within a set range of hues for the frame to segment out yellow. Then I get the equation of the points using polyfit. Using this equation, the line can be plotted. The radius is found using this equation by the method described in the document. Similarly, I found the radius for white line. The coefficient values decide the direction and the radius is calculated by getting the average of the turn in both directions.

Now, the frame is filled with both the white and yellow points and then the frame is transformed back to the original view.

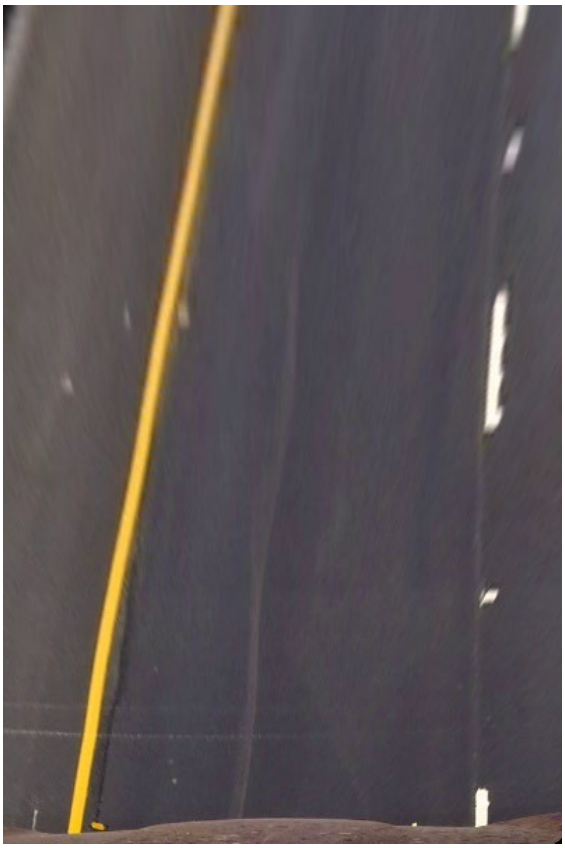

Figure 7 Bird's Eye View for frame 2

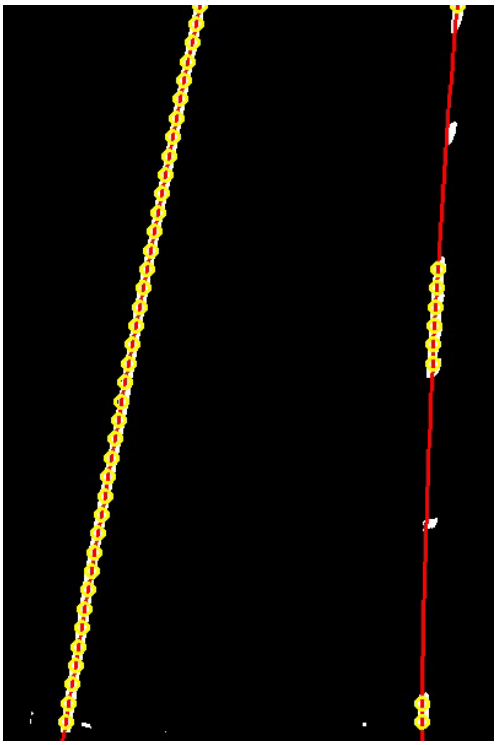

Figure 8. Line fitting for frame 2

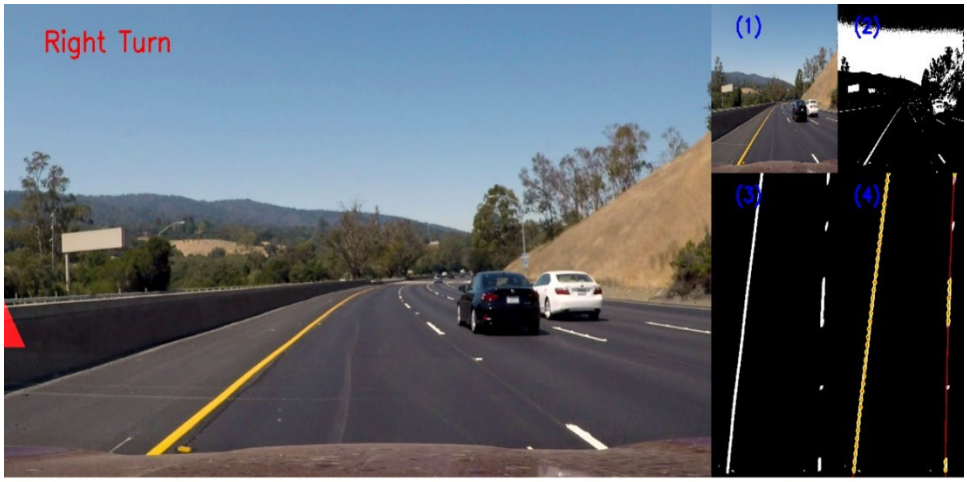

Figure 9. Final image for frame 2

The output video can be found here:
https://drive.google.com/file/d/10PjkH15nIh38jLlz6cvcl733_G425SK2/view?usp=sharing

Homography: I have used homography in my project. Homography technique provides a way or a matrix to transform from one view to another. In this project, I have used it to get the bird's eye view of the road. Important thing to note is that we need at least four points to transform.

Hough Lines: I have used Hough lines to detect the lines and get their position. The Hough space is like a variable space using which we can get accurate values for the position the points in the image space. In the Hough space, the all the possibilities for that specific edge is calculated and the point where these lines coincide the most is treated as the edge point.

Finally, I believe that my piepine will be able to generalize to other similar videos however, we might need to add some more pre processing steps to remove the extra noises and increases the amount of information available to us in every frame.