Abhijit Vinod Mahalle 117472288
ENPM673 Project 2 Report

Problem 1

1. The task here was to perform Histogram Equalization and Adaptive Histogram Equalization on given set of images.
2. Histogram Equalization is performed on the colored image by separating the channels.
3. In Adaptive Histogram Equalization, the image is divided into 8x8 blocks and histogram equalization is performed on individual blocks.

Histogram Equalization

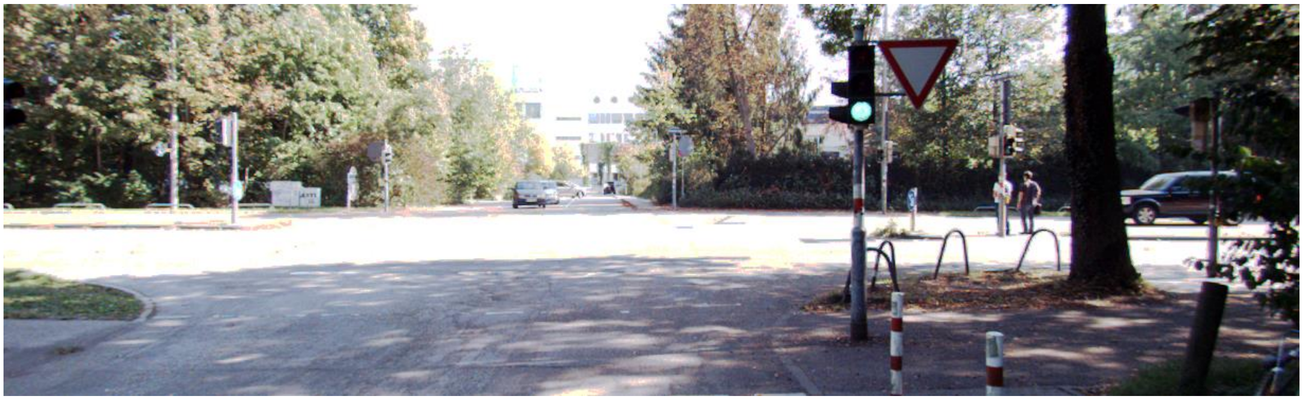


Fig. 1: Histogram Equalization

Adaptive Histogram Equalization

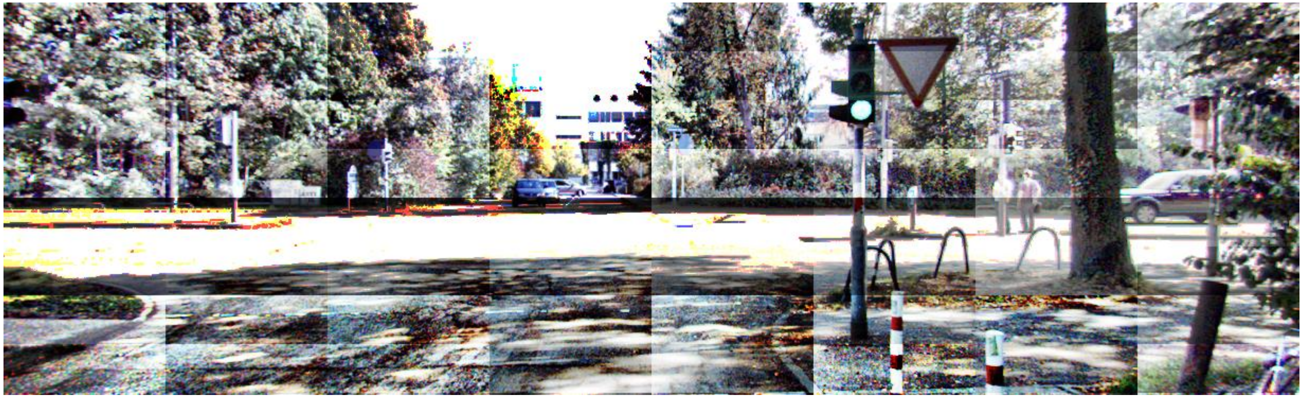


Fig. 2: Adaptive Histogram Equalization

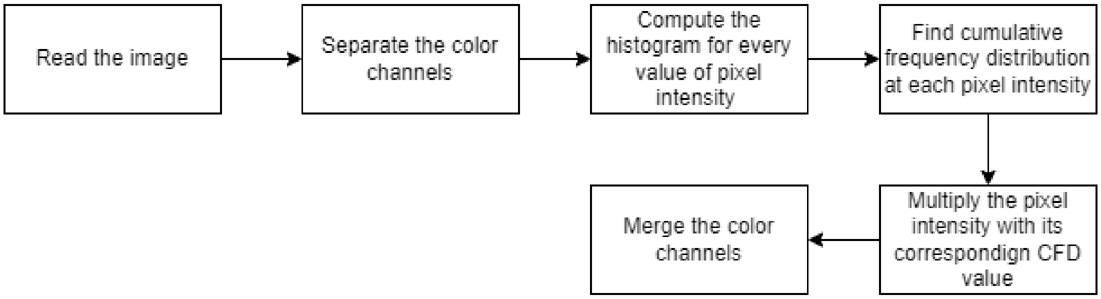

Read the image
Separate the color channels
Compute the histogram for every value of pixel intensity
Find cumulative frequency distribution at each pixel intensity
Multiply the pixel intensity with its correspondign CFD value
Merge the color channels

Fig. 3: Block diagram for Histogram Equalization

Results:

1. Histogram Equalization has increased the brightness of the overall image. Thus, the regions were bright earlier have become brighter. It considers the global contrast of the image.
2. In Adaptive Histogram Equalization, the increase or decrease in brightness value is confined to the given window. This results in an image having localized contrast. However, the image appears choppy due to difference in contrast in each individual blocks.

Problem 2

1. The task was to detect lanes on road and to classify solid and dashed lane lines by marking them green and red respectively.
2. HoughLines function was used to detect lanes in the region of interest of a threshold image.
3. Using the co-ordinates of detected lines, length of each line was calculated and the line with largest length was determined to be the solid line.
4. Lines with slopes close to the solid line were eliminated.
5. Step 3 was performed again to find the dashed line.

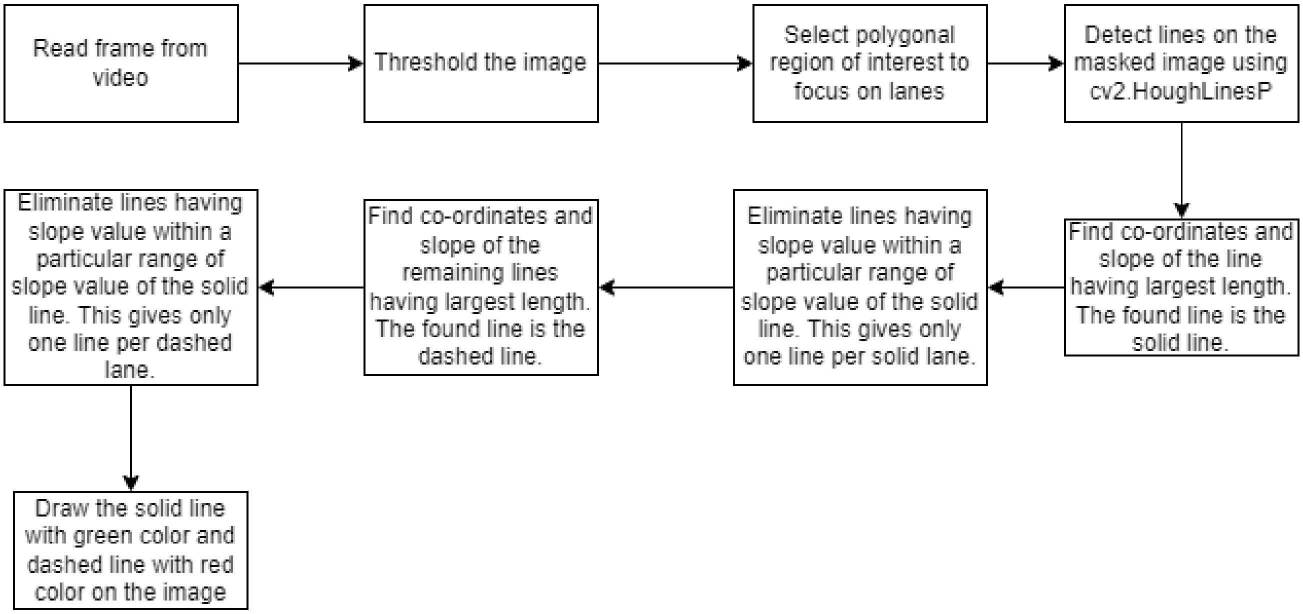

Read frame from video
Threshold the image
Select polygonal region of interest to focus on lanes
Detect lines on the masked image using cv2.HoughLinesP
Find co-ordinates and slope of the line having largest length. The found line is the solid line.
Eliminate lines having slope value within a particular range of slope value of the solid line. This gives only one line per solid lane.
Find co-ordinates and slope of the remaining lines having largest length. The found line is the dashed line.
Eliminate lines having slope value within a particular range of slope value of the solid line. This gives only one line per dashed lane.
Draw the solid line with green color and dashed line with red color on the image

Fig. 4: Block diagram to detect straight lanes

Results:

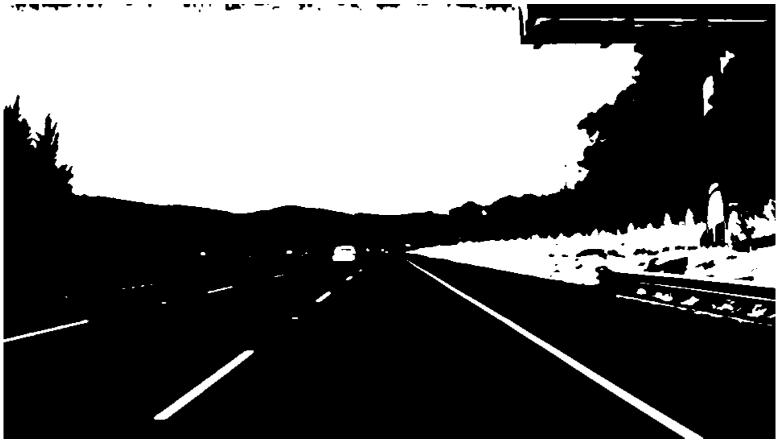


Fig. 5: Thresholded image

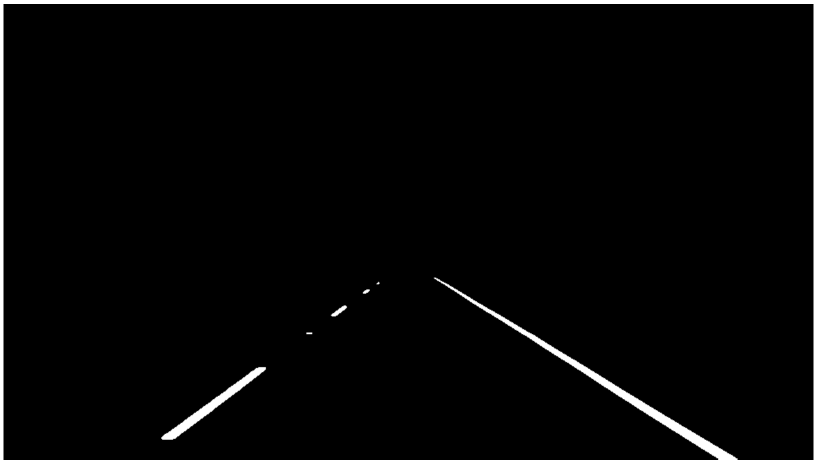


Fig. 6: Masked image

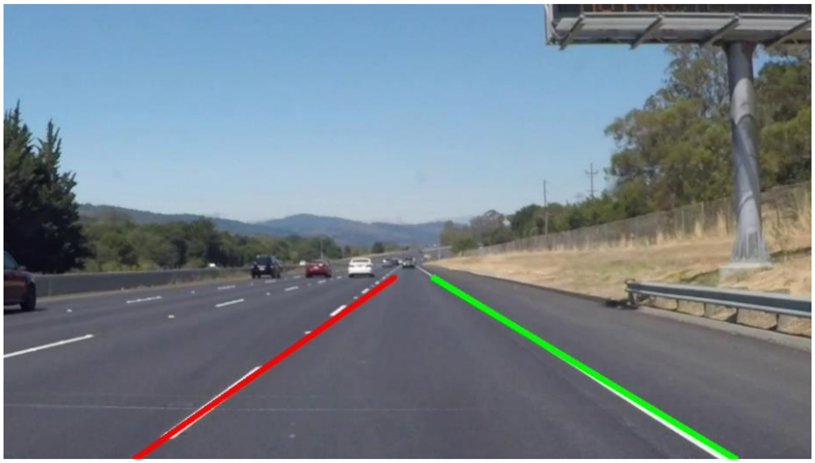


Fig. 7: Image with detected lines

Challenges:

1. Multiple lines were getting detected per lane side. To keep only one detected line per lane side, largest line from the detected lines was found. All the lines with slope values close to the largest line were removed from the list. The remaining list only has the lines found on the dashed lane. Similar process was followed to find only one line per lane side.
2. In some of the frames, not even a single line was getting detected for a dashed lane side. Line information of the previous frame was used to extrapolate and plot the line.

Generalization to other videos:

1. The above pipeline works even if the given video is flipped horizontally.
2. However, the above pipeline will not work when there are dashed lane lines on either sides and they both are to be marked of red color.
3. The above pipeline is suitable only for straight lanes.

Problem 3

1. The task was to detect curved lanes and predict turn and radius of curvature.
2. The video was read frame by frame and below pipeline was applied on all individual frames.
3. The region of interest of the image containing only the lane line were warped to find the bird's eye view of the lane.
4. Equation of the curve was then calculated using np.polyfit.
5. Radius of curvature was found using below equation for the equation of the curve $y = ax^2 + bx + c$:
$$R =[1 + (dy/dx)^2]^{3/2}/|d^2y/dx^2 |$$
6. Curve was unwarped and superimposed over the original image.

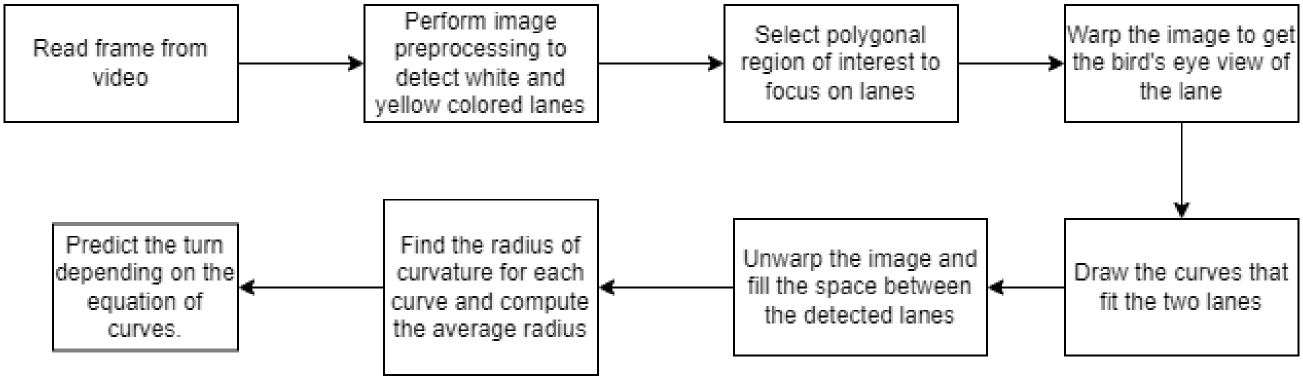

Read frame from video
Perform image preprocessing to detect white and yellow colored lanes
Select polygonal region of interest to focus on lanes
Warp the image to get the bird's eye view of the lane
Draw the curves that fit the two lanes
Unwarp the image and fill the space between the detected lanes
Find the radius of curvature for each curve and compute the average radius
Predict the turn depending on the equation of curves.

Fig. 8: Block diagram to detect curved lanes

Results:

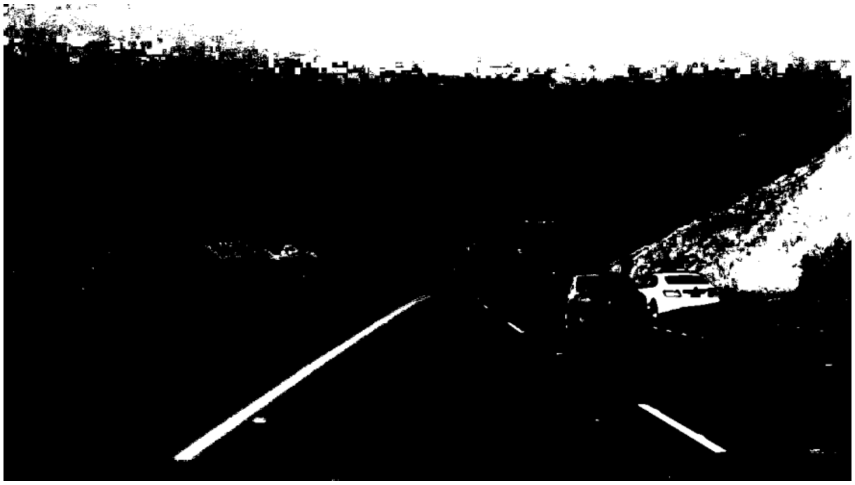


Fig. 9: Processed image

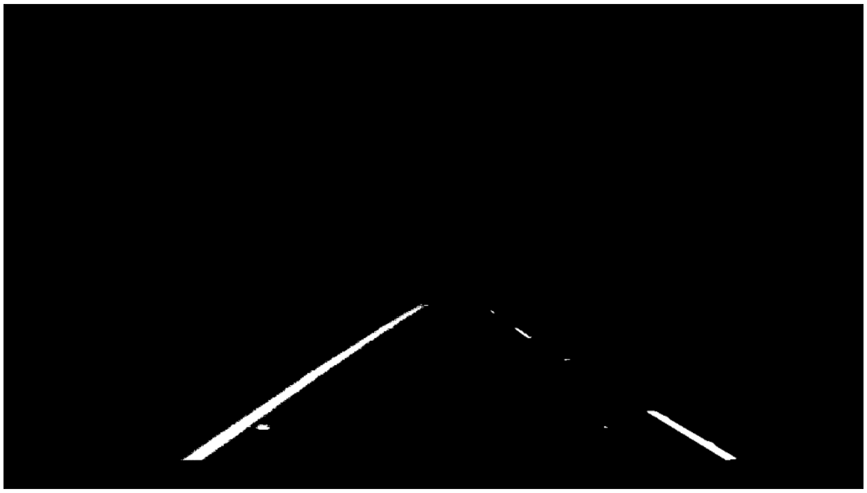


Fig. 10: Masked image

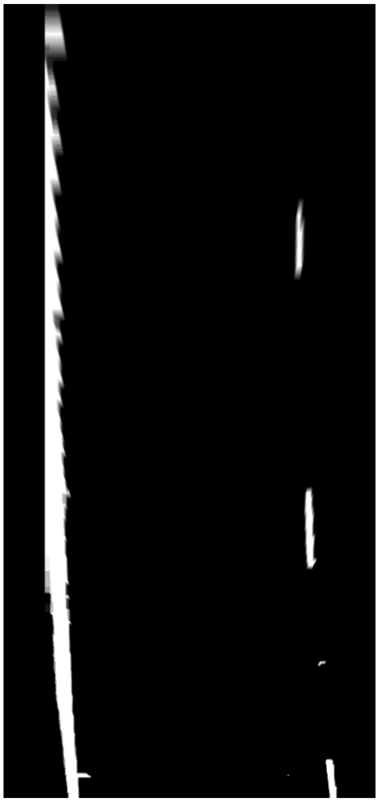


Fig. 11: Bird’s eye view of lanes

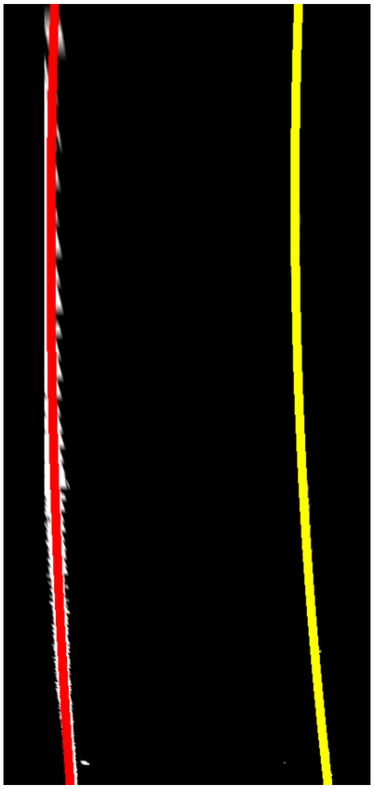


Fig. 12: Curves fitted on lanes

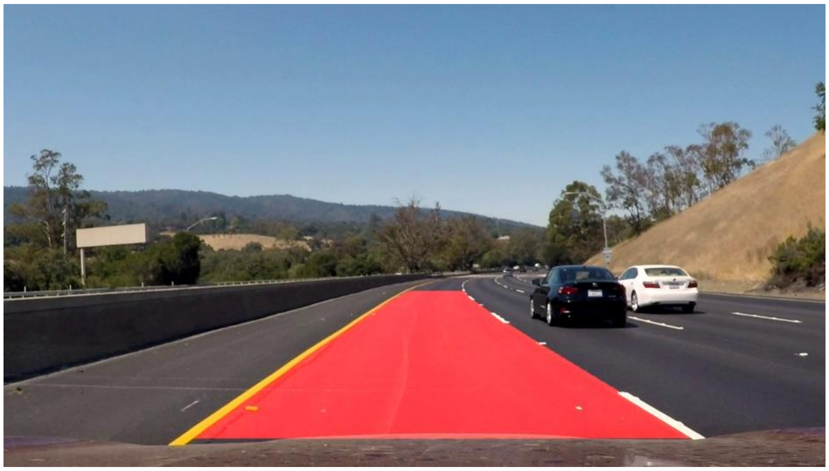
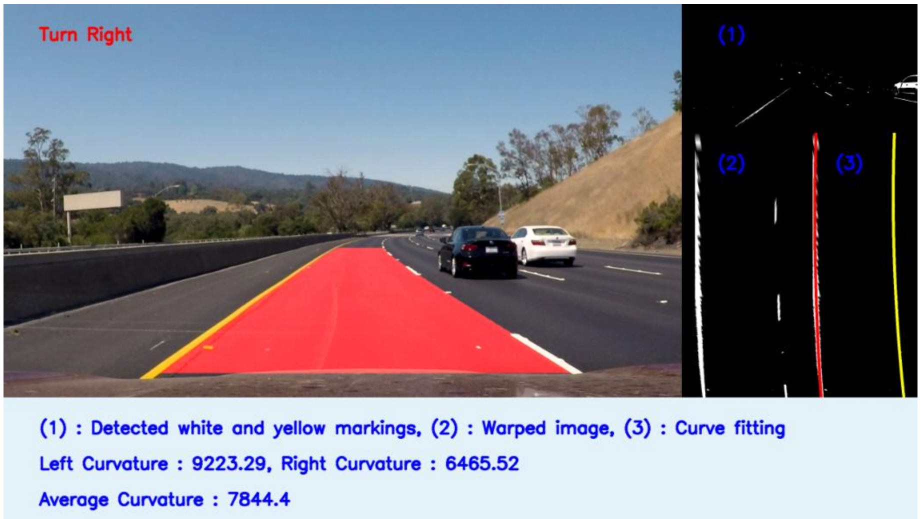


Fig. 13: Detected Lane lines



Turn Right

(1) : Detected white and yellow markings, (2) : Warped image, (3) : Curve fitting

Left Curvature : 9223.29, Right Curvature : 6465.52

Average Curvature : 7844.4

Fig. 14: Concatenated window

Challenges:

1. It was difficult to extract the yellow and white lanes. However, after using HSV filter and appropriate HSV values for yellow and white filter, both colored lanes were detected.
2. It was difficult to fit the curve as there were many stray pixels. I used rectangular windows to find average pixel location and moved the window gradually depending on the previous average.
3. Concatenating various images was also a difficult task.

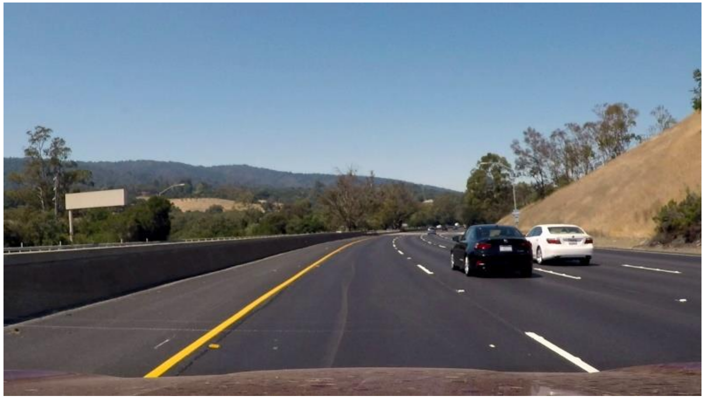
Generalization to other videos:

1. The above pipeline can be applied only to yellow and white color lanes.
2. Pipeline is quite robust to noise, but may not work if the light intensity in the environment decreases.

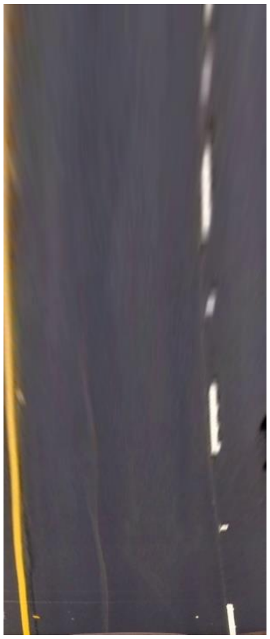
Homography

Homography is a projective transformation from one plane to another plane. A homography matrix gives mapping between points in one plane to corresponding points in another plane. Using homography, an image can be viewed from a different perspective.

For example, below is the image taken from a camera placed on the dashboard of a car.

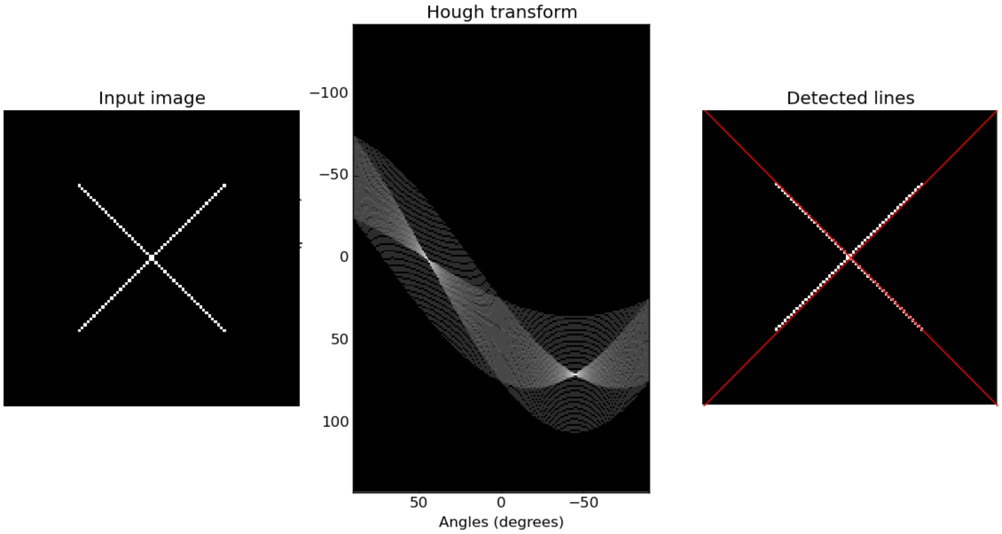


Using Homography transformation, we can convert this image into an image that looks like as if it is taken from a camera placed directly on top of the road to focus on the lanes. Below is the result:



Thus, using homography we can change the perspective of the image without the need of extra.

HoughLines:



Hough transform is a transform from cartesian space to parameter space. A line in a cartesian space is represented as a point in parameter space. A point in a cartesian space is represented as a line in parameter space. The intersection between two lines in parameter space gives the parameters of the line passing through the points in cartesian space corresponding to those lines. Thus, more the number of lines intersecting at a given point in parameter space, more is the number of points lying on its corresponding line in the cartesian space. Thus, a line in cartesian space can be easily detected by counting the number of lines passing through its corresponding point in the parameter space. If the number of lines intersecting in parameter space is above a particular threshold, the edge is definitely a line in cartesian space. In order to detect a line, the edge in a thresholded image is transformed to a parameter space and the line is detected based on the number of intersections in the parameter space. The parameters used are in polar co-ordinates (r, theta) to deal with perpendicular and parallel lines.

Video Output Link:

https://drive.google.com/drive/folders/1KteX_pb8jnk3AgCYdLzgnWaEjlX6dxR6?usp=sharing