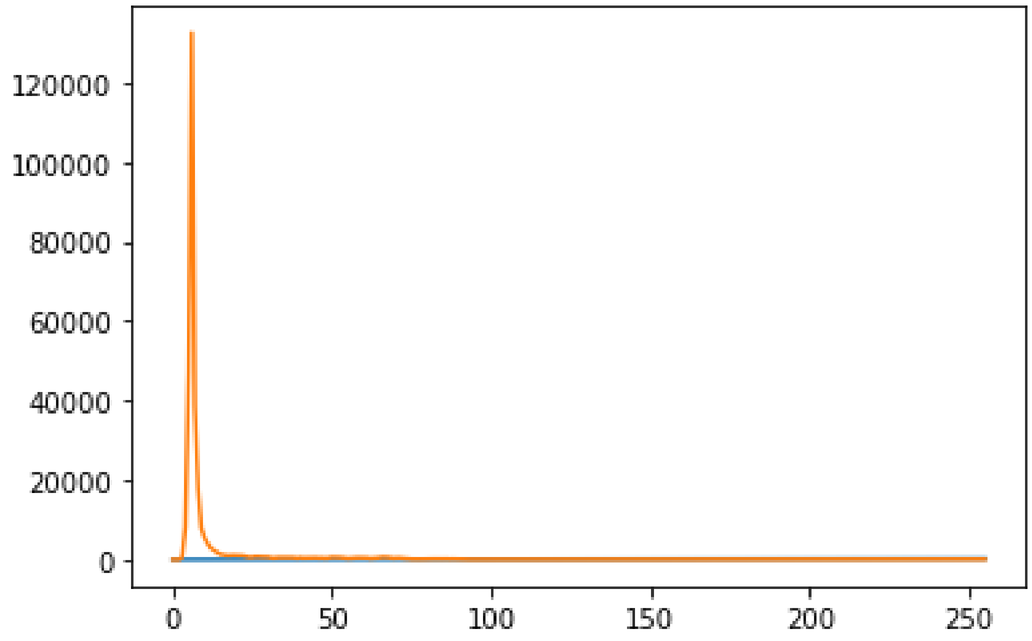
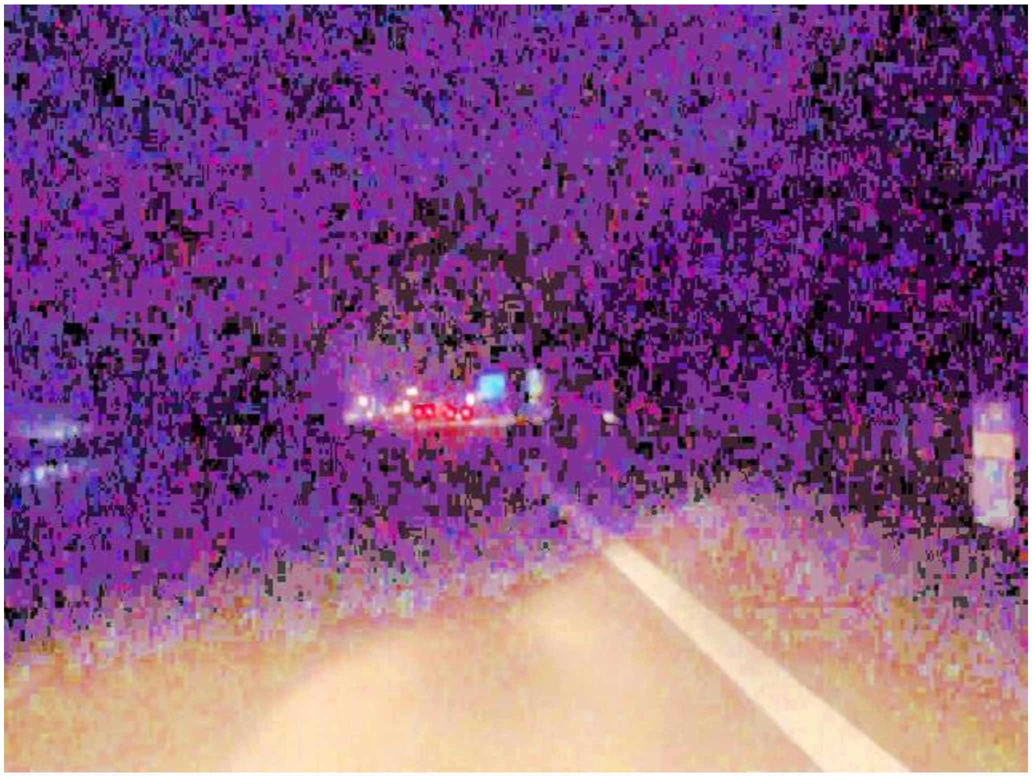
ENPM673
Project 2 Report

➢ **Problem 1**:

- Step 1: The input image is converted into HSV color space
- Step 2: The values of the V-Channel of the HSV image are taken out as a list to calculate the histogram using equation $\boldsymbol{h(i)} = \sum_{x} \sum_{y} [\boldsymbol{I(x,y)} = \boldsymbol{i}$
- Step 3: Cumulative Distribution Function is calculated using the original intensity count using equation $\boldsymbol{cdf(i)} = \sum_{j<=i} \boldsymbol{h}(\mathbf{j})/(\boldsymbol{height} * \boldsymbol{width})$
- Step 4: Calculated the new values as $\boldsymbol{h\_new}(\mathbf{i}) = \mathbf{255} * \mathbf{cdf}(\mathbf{i})$
- Step 5: Replaced the original intensity values of the V-Channel with the new intensity values to get the equalized image



Histogram of original intensities

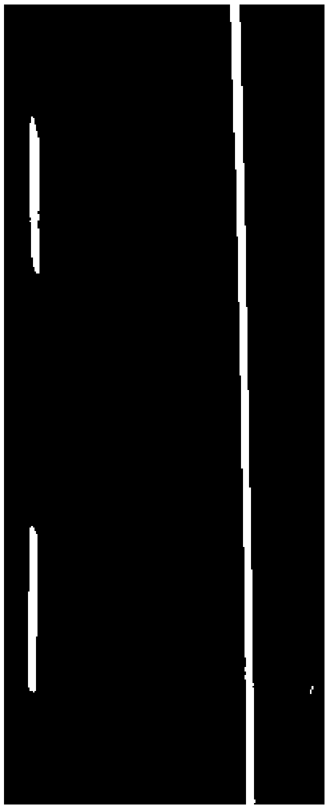

Histogram Equalized Frame

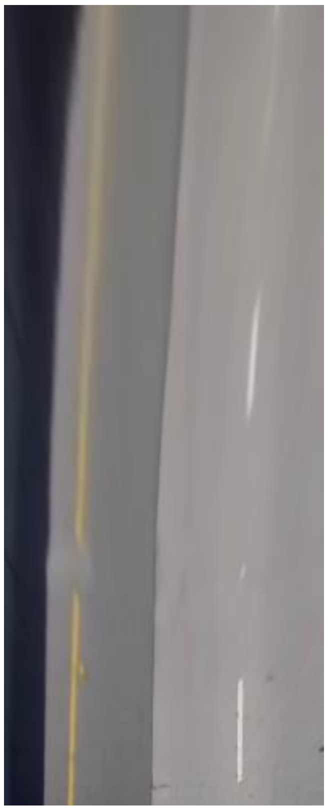
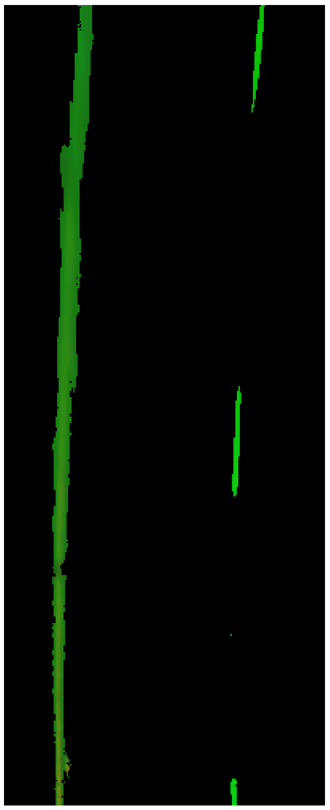
- **Problem 2**:

- Stages:

- **Undistorting and denoising**: In this stage, the input was corrected for all distortions using the Calibration Matrix and Distance Matrix with the cv2.getOptimalNewCameraMatrix() and cv2.undistort() functions, after which the region of interest (ROI) was extracted. This image was corrected for any minute noise using Median Blur filter.

- **Lane Detection**: For the detection of lanes, a Homography function is used to calculate and display the warped image. HSV color masking (for Data Set 1) and HSL color masking (for Data Set 2) is used on the warped image to separate out the white pixels of the left and right lanes.

Warped Image and HSV mask (left to right)


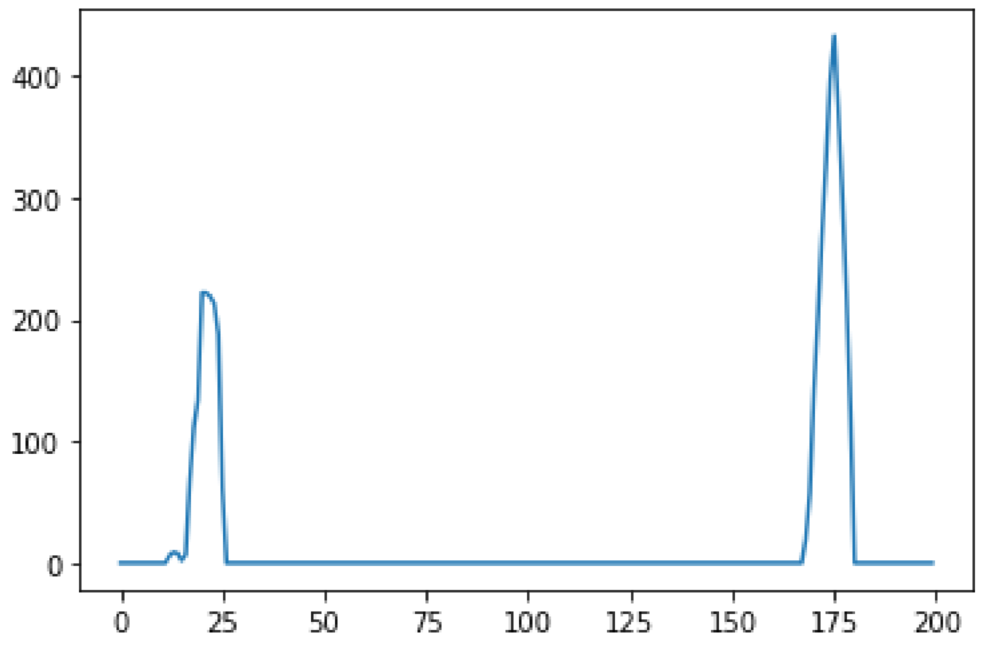

Warped Image and HSL mask (left to right)


- **<u>Homography</u>**: Homography is basically a transformation of points between two frames. It is a mapping of one frame to another in projective space which requires the co-ordinates of the frames to be homogeneous co-ordinates. Homography has 8 degrees of freedom with the element h_{33} as 1.

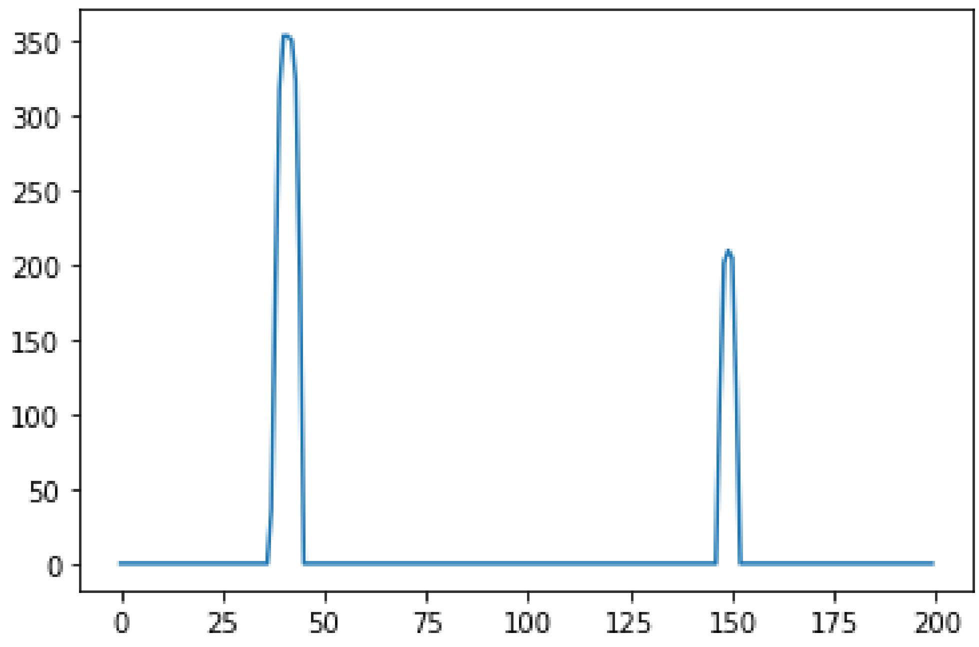
$$\begin{bmatrix} {x_1}^{(w)} & {y_1}^{(w)} & 1 & 0 & 0 & 0 & -{x_1}^{(c)}{x_1}^{(w)} & -{x_1}^{(c)}{y_1}^{(w)} & -{x_1}^{(c)} \\ 0 & 0 & 0 & {x_1}^{(w)} & {y_1}^{(w)} & 1 & -{y_1}^{(c)}{x_1}^{(w)} & -{y_1}^{(c)}{y_1}^{(w)} & -{y_1}^{(c)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ {x_4}^{(w)} & {y_4}^{(w)} & 1 & 0 & 0 & 0 & -{x_4}^{(c)}{x_4}^{(w)} & -{x_4}^{(c)}{y_4}^{(w)} & -{x_4}^{(c)} \\ 0 & 0 & 0 & {x_4}^{(w)} & {y_4}^{(w)} & 1 & -{y_4}^{(c)}{x_4}^{(w)} & -{y_4}^{(c)}{y_4}^{(w)} & -{y_4}^{(c)} \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The general form of homography

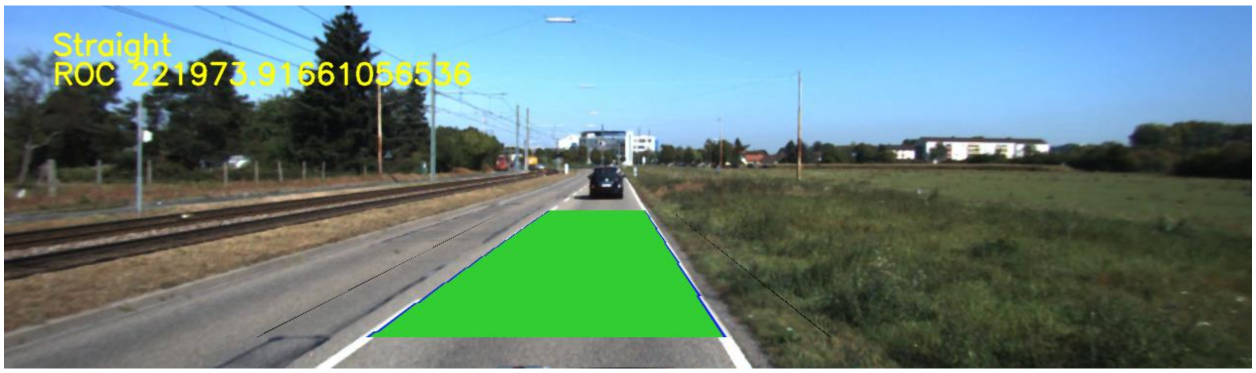
After this, the HSV mask is given to the custom histogram function which returns the indices of the columns with the highest number of white pixels which correspond to the left and right lanes.



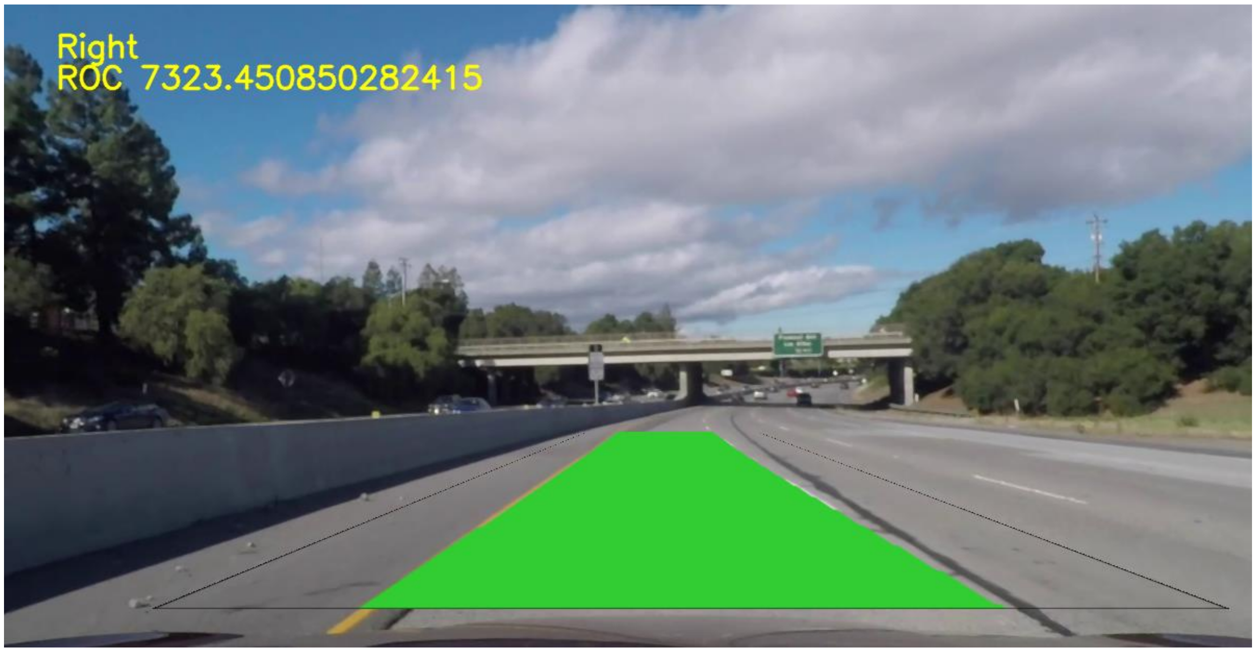
Data Set 1 Histogram with dominant right lane and left lane peaks


Data Set 2 Histogram with dominant right lane and left lane peaks

- **Refined lane detection**: Using the np.polyfit() function, the coefficients of the left and right lane polynomials are calculated and then the calculated Y's of both the lanes are concatenated into a numpy array. This array is then given to functions cv2.pollylines() and cv2.pollyfill() to fill the lane that the vehicle is travelling in. This warped image is then unwarped using inverse homography. A blank mask created using inverse binary thresholding is then used to be added with the unwarped image to get the filled lane onto the road. After this, the upper half of the cropped image is added to the half result to get the entire picture with the lane filled accordingly.



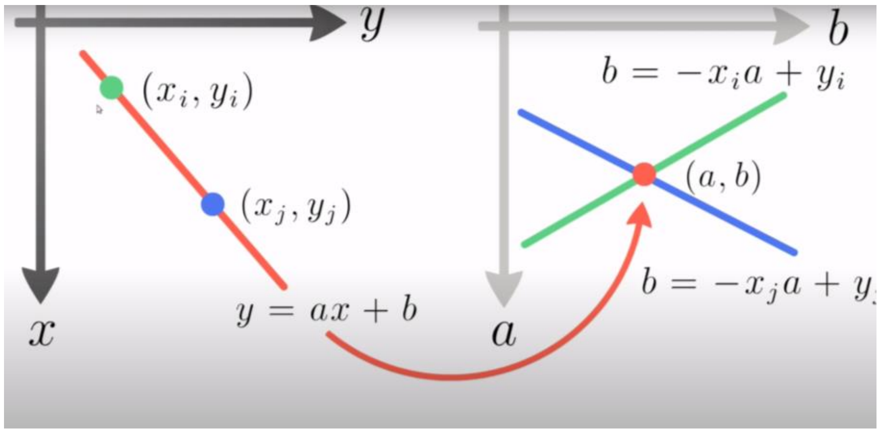
Final Output of the Dataset 1 video



Final Output of the Dataset 2 video

- **Turn Prediction**: The turn of the vehicle is calculated using the difference between the lane center and the image center. According to the threshold, the turn is displayed on the image.

  - **Hough Lines**: Hough transform is primarily a feature extraction technique used to detect features in an image. Hough lines technique is used to detect lines. Hough transform represents the polar co-ordinates (r, ϴ) space, where a line in can be expressed as r = x cos(ϴ) + y sin(ϴ). A point in the image space becomes a line in the hough space.



Every point on a line in cartesian space represents a curve in the polar space. The intersection point of these curves (in polar space) represents the equation of the line from the cartesian space.

- **Likelihood of the code execution for different scenarios**:

The pipeline implemented can be effective and generalize well to other videos if certain parameters are calculated perfectly such as the points for homography, lower and higher masking limits, size of the slices that the warped image is divided in. The pipeline might be considerably weak if there are drastic changes in lighting or in camera tilt.

- **Output video links**:

  - Problem 1 Output: Link
  - Problem 2:
    - Data Set 1 Output: Link
    - Data Set 2 Output: Link