ENPM 673 Project 2

Arjun Srinivasan Ambalam, Praveen Menaka Sekar, Arun Kumar Dhandayuthabani

March 12, 2020

1 Introduction

The project focuses on enhancing the contrast and improve the visual appearance of the night video sequence and perform simple Lane Detection to mimic Lane Departure Warning systems used in Self Driving Cars.Our task will be to design an algorithm to detect lanes on the road, as well as estimate the road curvature to predict car turns.

Prior to the implementation of image processing on the image sequence, the video is split into its image frames using cv2.VideoCapture, and once the operations are performed on each of the frames, it is appended into an array. This image array is then used to get the video back using cv2.VideoWriter

2 Problem 1

We aim to improve the quality of the video sequence provided above. This is a video recording of a highway during night. Most of the Computer Vision pipelines for lane detection or other self-driving tasks require good lighting conditions and color information for detecting good features. A lot of pre-processing is required in such scenarios where lighting conditions are poor. Now, using the techniques taught in class your aim is to enhance the contrast and improve the visual appearance of the video sequence. You can use any in-built functions for the same.

Figure 1: Original video frame

2.1 Approach to enhance the night video

The following algorithms were employed:

1. Gamma Correction. This applies a nonlinear scale to the pixel intensities to enhance the darker regions.
2. Adaptive Histogram Equalization using the CLAHE algorithm: Normal Histogram Equalization is a bad choice for this image, since the scaling of the Cumulative Distribution Function (CDF) doesn't handle the bright and very dark high interest regions. To account for both the bright and dark interest regions we implemented the CLAHE algorithm, which essentially applies a Histogram Equalization to each subsection of the image (while filtering out outlying regions).

3. Gaussian Blurring: This helps to reduce the amount of noise in the image.

Figure 2: Original video frame and contrast enhanced video frame

3 Problem 2

3.1 Lane Detection System

1. When processing the camera's road images, we undistort it first using the given camera calibration matrix (K) and distortion coefficients, with the help of cv2.undistort().

2. Once the road images are undistorted, We chose to find the lanes using combined color and gradient thresholding.

3. As pre-processing step to color thresholding, Contrast Limited Adaptive Histogram Equalization is used to normalize visibility of lane lines in different lighting conditions and under shadows.

4. For the above step, we converted the image from RGB to LAB color space, and applied Contrast Limited Adaptive Histogram Equalization to the Lightness channel(L)

5. In order to apply gradient thresholding effectively so that it does not find any unwanted features, we crop the road image to obtain the Region of Interest (ROI) and perform perspective transformation.

6. RGB and LAB color space was used for extracting lanes as the RGB channel helps in isolating white pixels in majority of lighting conditions, whereas LAB channel helps in isolating yellow pixels in majority of lighting conditions.

7. The white and yellow color masks obtained from above thresholding is then combined as a single mask using OR condition.

8. For gradient detection, two methods are used and combined to try to gather more sources of information for robustness.

9. Method 1, Apply gradient threshold along X-axis on warped bird's-eye view road image and apply a denoise filter.

10. Method 2, Apply gradient thresholds by magnitude and direction independently on the original road image, combine them with an AND condition, apply a denoise filter, and warp to bird's-eye view.

11. The results of Method 1 and 2 are then combined with an OR condition to overlap the detected lane lines from both sources.

12. Gradient detection is influenced by noise and can be difficult to extract the lane lines, so using multiple methods may help extract lanes for various conditions.

13. Usage of the color mask and gradient mask depends on detection of lane lines.

14. If there is no previous lanes detected, we use both color and gradient masks for more restrictive selection of lane pixels, thereby minimizing the missed detections from noise.

15. If a lane is detected in previous frame, we use only color mask for more broad selection since the search is already limited to a window around the known lane location.

16. The above lane detection process is carried out by sliding window based histograms and checking the validity of lanes.

    (a) We initially start at bottom $\frac{1}{4}^{\text{th}}$ of the image with the windows centered at initial x positions found by two max peaks in vertical slice histograms.
    (b) We use the average x position of the points within the window to set the center of the next window up.
    (c) If enough points are not found on one side, but the other side has points, set the window center based on the previous detected lane width between left/right windows.
    (d) If enough points are not found on either side, continue to slide the window centers by shifts calculated by the previous window.
    (e) Narrow the window to set the final lane x,y points to reduce noise and apply a 2^{nd} order polyfit to the final lane x,y points.
    (f) Store the detected lane x,y values and polyfits in the lane objects.

17. In case of good lanes being detected in previous frames, the process is even more simple,

    (a) We select any lane x,y points within a fixed margin width of the existing polyfit lane lines.
    (b) Then apply a 2^{nd} order polyfit to the final lane x,y points.
    (c) Store the detected lane x,y values and polyfits in the lane objects.

18. Keeping in mind the effect of different lighting conditions and shadow effects in the video, a lane checking process is also done.

19. To overcome the disadvantages in thresholding process, acceptable lane width conditions were used, and to overcome noise from detection process, acceptable lane pixel conditions were used.

20. Once the lanes and their corresponding points are obtained, we obtain pixels to meters conversions based on our warped image dimensions. Lane width according to US standard = 3.7 m Lane line length according to US standard = 3.0 m Lane Width in pixels = 675 Lane line length in pixels = 83 width per pixel = $\frac{3.7m}{675pixels}$ length per pixel = $\frac{3.0m}{83pixels}$

21. Based on above conversion we then find the radius of curvature and lane center offset.

$$R = \frac{(1 + (2A + B)^2)^{\frac{3}{2}}}{2A} \tag{1}$$

22. The lane center offset is calculated by assuming the car position is at midpoint of the image and find the distance from each lane line to the center of the car.

23. Finally, we the unwarp the lane detected image back to the original road image.

The Intermediate results for dataset 1 and final results for the given 2 datasets following the above procedure is given below.

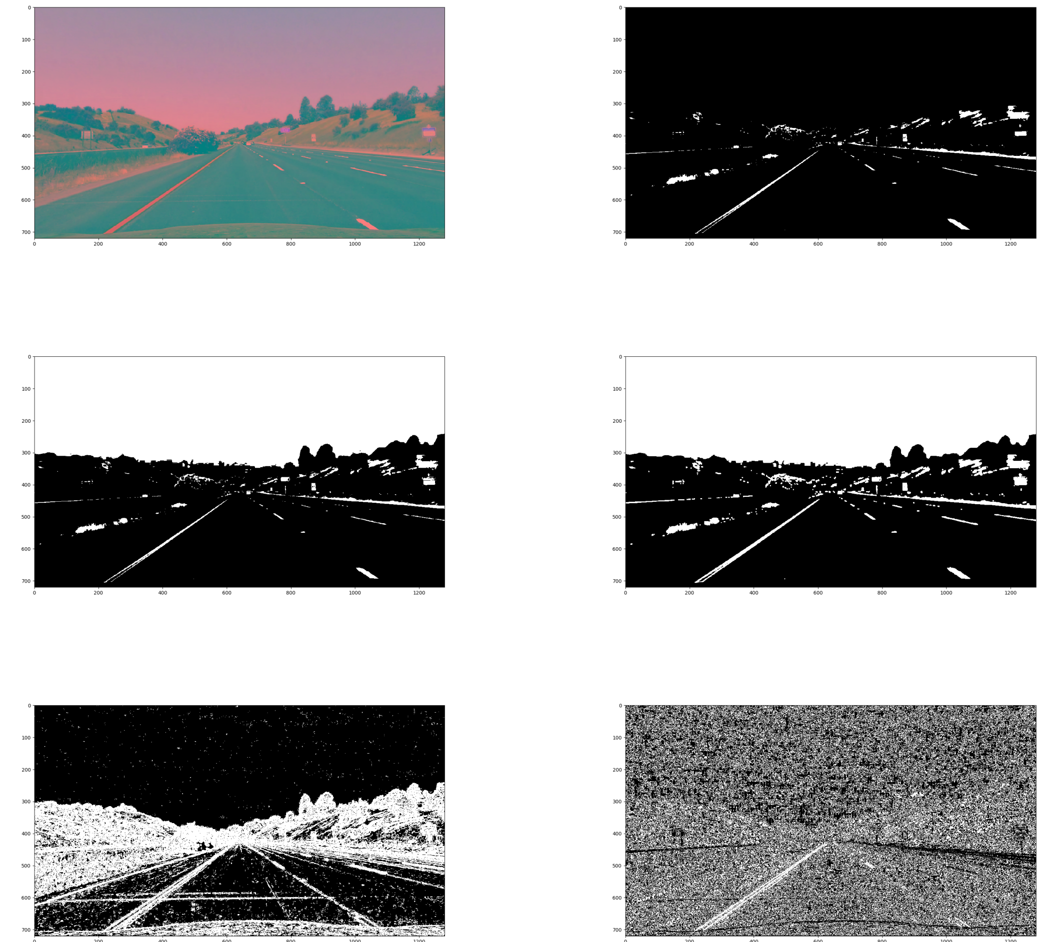

Figure 3: Intermediate Results for dataset 1

Figure 4: Result for two datasets

4 Problems Encountered

1. Upon solving for Problem 1, we implemented a histogram equalization across the frames of the video. The results obtained were pretty bad when compared with the original frame image.

2. Hough lines method was not used for Lane detection system as it was not able to detect curves along the lane (white and yellow) pixels. Extending the Hough Lines along the straight lanes gives proper results. Hence we went with Histogram of pixels approach to include the curves present in the pixels across the frame.

3. Our lane detection system calculates homography and perspective transformation by taking 4 points from a test image pertaining to the video. Hence, this detection pipeline is not universal as different videos require different ROI selection.