MANIPAL ACADEMY of HIGHER EDUCATION
(Deemed to be University under Section 3 of the UGC Act, 1956)

MANIPAL SCHOOL OF INFORMATION SCIENCES
(A Constituent unit of MAHE, Manipal)

Road Classification using Image Processing and Machine Learning

Reg. Number	Name	Branch
221039017	Advait Shirvaikar	Embedded Systems
220139025	Ananth Pai J	Embedded Systems

Under the guidance of

Dr. Mohan Kumar J
Associate Professor,
Manipal School of Information Sciences,
MAHE, MANIPAL

11/11/2022

Contents

LIST OF FIGURES

Chapter 1
INTRODUCTION

An autonomous car is a vehicle capable of sensing its environment and operating without human involvement. A human passenger is not required to take control of the vehicle at any time, nor is a human passenger required to be present in the vehicle at all. An autonomous car can go anywhere a traditional car goes and do everything that an experienced human driver does.

Detection of road is one of the main features that incorporates in a self-driving car. Road recognition is one of the major topics of research now a days because of machine learning and computer vision-based researches. Existing road recognition research majorly focuses on structured road with road lane, and very few recognition models for unstructured roads. In general, structured roads are identified by lanes, or a road bounded by edge line pavement markings. Thus, a structured road maybe a single lane road, or might have multiple lanes, all bounded by these pavement markings. Unstructured roads are the roads which does not have a proper lane markings and boundaries. On the other hand, they might have multiple terrains having sand, gravel, stones, etc.

Figure 1.1 Image of a structured road

Figure 1.2 Image of an unstructured road

Based on this knowledge, we adopted a binary classification problem approach. The methodology that we followed to classify the roads consisted of detecting the edge line pavement markings on the road, and if found, classifying that road as a structured road. Similarly, if no lines were found, we classified them as unstructured roads.

To detect these lanes, and classify the road images into the two categories, the images are required to be in a certain format. This is accomplished using OpenCV and python. We used multiple pre-processing techniques such as thresholding, Sobel filtering, perspective warps, Hough lines, Gaussian filtering, Canny edge detection, etc. For object detection, we used the sliding window algorithm. Further, machine learning models will also be used, to classify roads into the two categories.

Chapter 2
LITERATURE SURVEY

2.1 Analysis of Lane Detection Using OpenCV

Authors have used Canny and Sobel and implemented lane detection techniques. In the lane detection techniques, they have used a camera mounted in front of the vehicle. On the basis of Receiver Operating Characteristic (ROC) curve and Detection Error Trade-off (DET) curve performance of these lane detection techniques has been measured. These curves are the standard parameters for assessing the performance of an algorithm in Computer vision. Confusion matrix has been used for finding FP, FN, TP and TN and on the basis of confusion matrix FP rate, FN rate, TP rate, TN rates have been calculated. Finally, DET and ROC curves have been plotted on the basis of confusion matrix using MATLAB and performance of each method is analysed.

They faced challenges like,

- Different lighting circumstances:

  Changing illumination on road during morning, noon and evening time is a major challenge for lane detection.

- Different weather circumstances:

  Different types of weather include cloudy, windy, rainy, sunny, stormy, foggy and peaceful weather. The weather condition is rarely constant for a given day and varies from time to time.

- Curved roads:

  If some or all part of road gradually diverges in spite of being straight, then it is called curved road. Lane markings also follow the same curve on the road.

- Roads without Lane mark (Unstructured Roads):

  If the road is not having any lane marking, then it is called unstructured road. Lane detection for these roads requires a more robust and effective technique.

- Preceding vehicle, Shadow of tree:

  Due to shadow of the tree on the road and preceding vehicle it is difficult to detect the lane as it changes the illumination and images captured.

- Blockage of visibility:

While travelling, blockage of visibility due to fog, preceding vehicle is also a challenge for lane detection. This blinds the camera sometimes or restricts its view.

- Different colour:

  Due to shadow, day light, evening, colour information of the road is also affected which changes the colour of images getting captured.

- Different texture:

  The variation in texture of the road because of nonuniformity is also a challenge for lane detection.

They concluded by reviewing the theoretical background about lane detection methods and their properties. It has already been stated that DET curves plotted for two methods, the one with greater area under it is better in terms of accuracy (same is with ROC).After analysing the performances of both the methods it is verified that method 1 based on Canny edge detection is better than Sobel operator based lane detection method 2 as it covers greater area.

2.2 Unstructured Road Detection based on Contour Selection

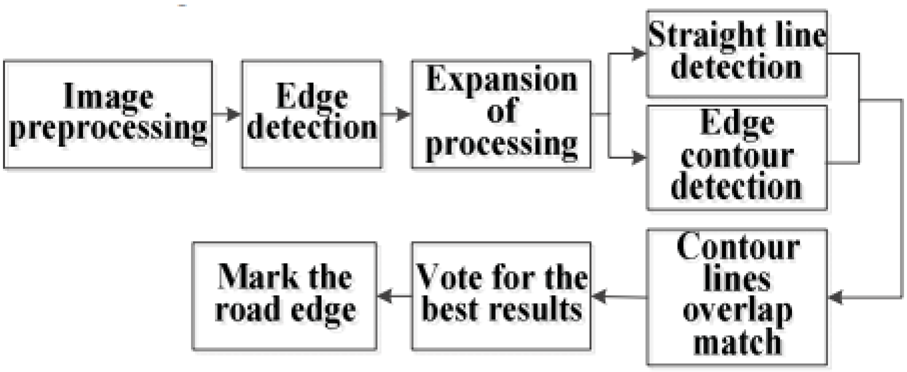

Figure 2.2.3 Flowchart of the model proposed in this paper

The captured image generally contains noise due to a disturbance. Because of randomness of noise and image signals correlation in space and time, the influence of noise on a pixel will make its grey scale significantly different with its neighbour pixel or with the corresponding pixel next frame. The following edge detection algorithm is mainly based on image intensity of the first and second order derivative, and derivative generally is sensitive to the noise. Therefore, it is necessary to use a filter to improve the performance. Gaussian filter is used to reduce noise.

Gaussian is kind of a linear smoothing filter and can eliminate gaussian noise which has a small error of space a good spatial stability, a small error of space position, is widely used in image processing of the noise reduction process. Gaussian filter is the input array of each pixel point convolution operation with the gauss kernel, the convolution of value as output pixel value. The image after Gaussian filter smooth degree, depends on the standard deviation. It is the output of the pixel in the field of weighted average, at the same time the nearer the centre pixel is higher. Compared with the other filter therefore, its softer, smoother and edge retention is better.

Figure 2.2.4 Gaussian filter processing

Canny edge detection operator is a multistage edge detection algorithm which was developed by john canny in 1986. principle of canny edge detection operator is to determine the edge pixels of the image by the maximum value of the image signal function. There are three evaluation parameters for optimal edge detection:

1) Low error rate. All edges should be found, and there should be no spurious response. In other words, the detected edges must be as real as possible edges.

2) Image edges should be well positioned. The edges that have been positioned must be as close as possible to the real edges. In other words, the distance between the point marked by the detector and the centre of the true edge should be minimal.

3) Single edge point response. For the real edge points, the detector should return only a point. In other words, the local maximum around the true edge should be minimal. This means that in only a single edge point position.

In this paper, a kind of unstructured road detection based on contour selection algorithm, is improvements in terms of the conventional linear detection method. Test results of our algorithm for some edge complete roads are good, but there is still limitation that this algorithm relies entirely on the final result of canny edge detection. Because the results of the canny detection algorithm in the cracked road often have a lot of irrelevant edges, which will produce a lot of irrelevant contours for the subsequent contour detection. The final detection effect is poor. Therefore, the pre-processing of original image in our algorithm needs to be improved, but also optimize the line and contour matching algorithm to make the output more accurate.

2.3 Lane-line Detection Algorithm for Complex Road Based on OpenCV

At present, there are much research on lane line detection methods at home and abroad, which are mainly divided into feature-based and model-based methods. Some methods enhance the contrast between lane line and road by adjusting the CCD brightness, gain and exposure time, then, the seed points of image is selected and classified, after this, Hough transform is performed on the seed points. There are also methods use circular curve lane line model and density-based Hough transformation for lane line recognition. Some researchers also adopted the boundary tracking detection algorithm based on fuzzy clustering to realize lane line recognition when identifying the interested area of lane. However, because the collected lane line image is affected by light, wear, vehicle shade and tree shadow, it is still a big challenge to accurately detect the lane line.

Therefore, a fast detection algorithm for lane line pixels based on combined gradient and colour filter of region of interest is proposed. First, the algorithm uses Sobel edge detection operator to detect the edge information of the structured road based on the high contrast between the lane line and the road surface. The second basis is the colour characteristics of the lane line. Generally speaking, there are only two colours of the lane line, white and yellow, so we can filter these two colours in the colour space to extract the pixel of the lane line. Then, in the region of interest model, a relatively stable lane line extraction method can be obtained by combining edge gradient and colour filter.

The authors followed a methodology which proceeded like this,

- Camera calibration
- Edge detection
- Area of interest
- Colour Thresholding
- Sliding window and polynomial fitting
- Restore to original perspective

In this paper, combined edge method and colour filter are used to detect the lane line, sobel operator and HSL geometric structure are also introduced. The lane line is extracted by sliding window and polynomial fitting, and the real lane line is restored by perspective transformation. The test results of Open CV platform show that the algorithm has better

Realtime performance and anti-interference performance, it can well detect the lane line of its dotted line and the solid line, realize the real-time line marking in the video image, this test method effectively avoid the light, wear, car shade and trees shadow effects on the lane line image. Amount of calculation and robustness of algorithm are highly optimized, this method can be also applied to the safety assistant driving system, or autonomous vehicle system.

Chapter 3
OBJECTIVES

- To learn the characteristics of roads in India, and to collect data on the same.
- To implement an algorithm that recognizes structured and unstructured roads, using image processing techniques.
- To create a machine learning model that will classify as well as identify structured and unstructured roads.

Chapter 4
PROPOSED MODELS

4.1 Hough Lines and Threshold masking Model 1

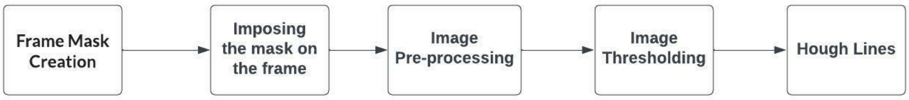

Figure 4.1.5 Flowchart for the proposed model 1

In this proposed model, the lane detection process was divided into 4 main parts:

1. **Frame Mask Creation:**

   The frame mask consisted of a 2-D NumPy array, with each pixel ranging from 0 (black) to 255 (white). But since the region of interest was in the shape of a polygon, the coordinates of the polygon were used to prepare the mask, while setting all the other pixels to 0. The picture below depicts the mask:

Figure 4.1.6 Polygon mask creation

2. **Imposing the mask on the frame:**

   Using OpenCV’s ‘bitwise_and’ function, the mask was imposed on the original frame.

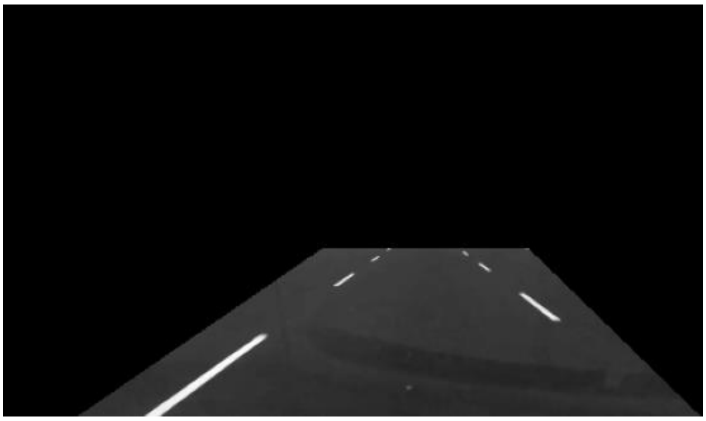

Figure 4.1.7 Mask imposition on the original frame

3. **Image Thresholding:**

   Removing all the mediator grey pixels to create a perfectly binary image. So that the positions of the lines can be seen perfectly for the upcoming processing.

Figure 4.1.8 Thresholded Image

4. **Hough Line transform:**

   Use the HoughLinesP function which is readily available in the OpenCV library. The function generates a continuous line using previously thresholded image as its input, to find the continuous lines. Now the lines and the original image can be superimposed to get a good view on the output of the model which can be seen below in the image.

Figure 4.1.9 Original image imposed with Hough transformed lines

4.2 HSV Filtering and Sliding Window search Model 2

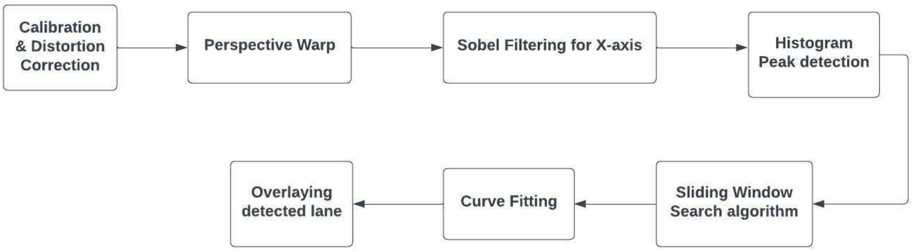

Figure 4.2.10 Flowchart of the proposed model 2

This model was developed using multiple filters and object detection algorithm:

1. **Calibration and Distortion correction:**

   In order to focus incoming light on the camera sensor, camera lenses distort it. Even while it's quite helpful for capturing photographs of our surroundings, they frequently end up slightly misrepresenting light. In computer vision applications, this may lead to inaccurate measurements. Images can provide a distortion model that takes lens distortions into account by being calibrated against a known object. For this, the distortion model was created using 20 images of a checkerboard. The undistorted image was created using OpenCV's calibration and distortion methods. The difference might not be noticeable, but it can have a huge impact on the image processing process.

Figure 4.2.11 Distortion correction applied to a frame

2. **Perspective Warp**

   The perspective warp technique aids in detecting the curved lanes in camera space. By applying a perspective transformation on the image, the birds eye view of the image can be obtained, which makes it easier to process further. By assuming that the lane is on a flat 2D surface, a polynomial can be fir, that can accurately represent the lane in lane space.

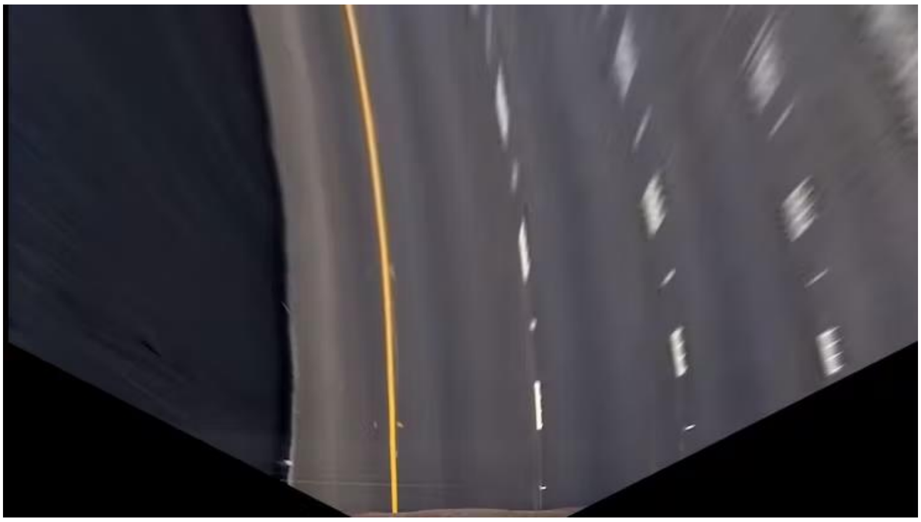

Figure 4.2.12 Perspective warped/birds-eye view image

3. Sobel Filtering

The Sobel filter is used to detect edges, as they perform a 2-D spatial gradient measurement on an image and so emphasize regions of high spatial frequency that correspond to edges. To monitor the saturation and lightness of the image, the HSL (Hue-Saturation-Lightness) colour space is employed. Both channels are subjected to the Sobel operator, and the gradient with respect to the x-axis is extracted. The pixels that pass the gradient threshold are then added to a binary matrix that represents the pixels in the image. As the entire image is not required, only a portion of the image is selected, which is not blurry or noisy.

Figure 4.2.13 Zoomed image of road lines

4. Histogram Peak detection

To determine the starting point of the Sliding Window search algorithm, a lane of pixels is required. To detect this, a histogram was used. Each portion of the histogram below displays how many white pixels are in each column of the image. The highest peaks of each side of the image are then taken, one for each lane line. Here's what the histogram looks like:

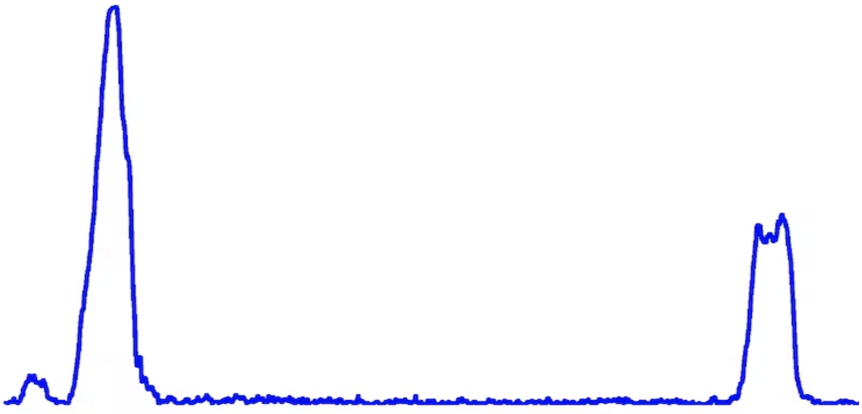

Figure 4.2.14 Identifying the histogram peaks

5. Sliding Window search

Sliding windows play an integral role in object classification, as they allow us to localize exactly “where” in an image an object resides. A sliding window is a rectangular region of fixed width and height that “slides” across an image, where an image classifier can be applied, to determine if the window has an object of interest. Starting from the initial position, the first window measures how many pixels are located inside the window. If the number of pixels reaches a certain threshold, it shifts the next window to the average lateral position of the detected pixels. If enough pixels are not detected, the next window starts in the same lateral position. This continues until the windows reach the other edge of the image.

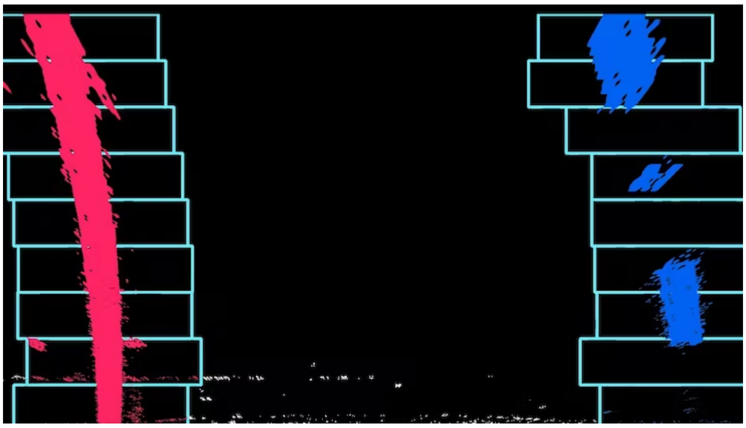

Figure 4.2.15 Sliding window technique on the edges

6. **Curve Fitting**

Finally, in curve fitting, polynomial regression is applied to the red and blue pixels individually. The figure 4.2.8 depicts the combination of sliding window and curve fitting techniques.

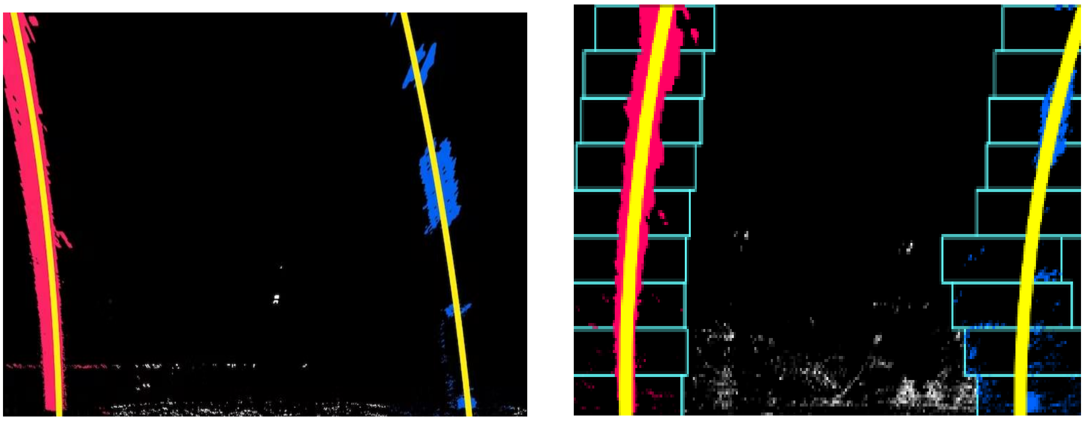

Figure 4.2.16 Curve Fitting, Sliding Window + Curve Fitting

7. **Overlaying detected window**

An overlay was created, which fills in the detected portion of the lane, which can then be applied on the image.

Figure 4.2.17 Overlaying the detected window on the original image

Chapter 5
RESULTS

5.1 Proposed Model 1

Figure 5.1.18 Input image for model 1, Output Image generated by the model 1

As we can see in the images above, the lanes are being detected by the HoughLinesP function. Lines are being drawn on the detected lines on the road. But the output is not as perfect as expected. If vehicles intervene, then the lines on vehicles are also detected as lines on the road.

For unstructured image as input, the lanes are still being detected because of poor pre-processing done on the image. The output for the model 1 is not as satisfactory as expected.

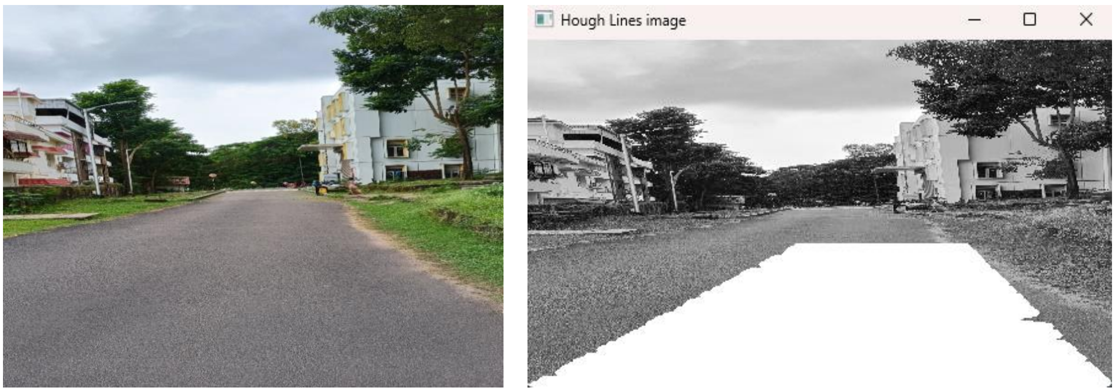

Figure 5.1.19 Unstructured image input and Output for unstructured road

5.2 Proposed Model 2

Figure 5.2.20 Input image Final image where lane is detected using the highlighted path

Figures 5.2.3 is the results for a structured road, i.e., roads having edge lines. The input image was first pre-processed using the image processing techniques described above, and then the sliding window technique was applied, to detect the lines (for object detection). The final image includes a change in the background colour due to the image processing layers used above. They yellow lane indicates the lane/pathway between two pavement markings, which will further be used as the basis to classify structured roads.

Chapter 6
CONCLUSION

The proposed models are giving outputs which are satisfactory. Even though model 2 has a lot of pre-processing and image processing layers, the resulting image produced by it was significantly more promising than that of model 1 for the identification of structured roads. As far as unstructured roads are concerned, they still need more pre- processing because of inconsistency in the terrain and lighting conditions.

The second model with some optimization can be considered for the next stage, where a machine learning model can be used for classification of the roads.

References

[1] W. Xiang, Z. Juan and F. Zhijun, "Unstructured road detection based on contour selection," in *4th International Conference on Smart and Sustainable City (ICSSC 2017)*, Shanghai, 2017.

[2] S. K. Vishwakarma, A. and D. S. Yadav, "Analysis of lane detection techniques using openCV," in *2015 Annual IEEE India Conference (INDICON)*, New Delhi, India, 2016.

[3] A. Sharma, M. Kumar and R. Kumar , "Lane Detection Using Python," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India, 2021.

[4] Z. Wang, Y. Fan and H. Zhang, "Lane-line Detection Algorithm for Complex Road Based on OpenCV," in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, 2019.

[5] P. Joshi, "Analytics Vidhya," 13 May 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/tutorial-real-time-lane-detection-opencv/.

[6] kemfic, "github," github, 22 October 2018. [Online]. Available: https://github.com/kemfic/Curved-Lane-Lines.

[7] kylesf, "github," 22 December 2016. [Online]. Available: https://github.com/kylesf/Advanced-Lane-Detection.

[8] M. Ameen, "github," 24 October 2017. [Online]. Available: https://github.com/mohamedameen93/Lane-lines-detection-using-Python-and-OpenCV/commits/master/Lane-Lines-Detection.ipynb.