**Automated Lane Simulation**

**A Major Project Synopsis Submitted to**



**Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal Towards Partial Fulfillment for the Award of**

**Bachelor of Engineering**

**(Information Technology)**

|  |  |
| --- | --- |
| **Under the Supervision of** | **Submitted By** |
| **Prof. Pawan K Gupta** | **Raksha Sankhala(0827IT161086)** |
|  | **Saquib Qureshi (0827IT161098)** |
|  | **Saiel Wadwekar(0827IT161094)** |
|  | **Yashasvi Sharma(0827IT161121)** |
|  | **Yashraj Sharma(0827IT161122)** |
|  | **Yuvraj Panwar(0827It161123)** |



**Department of Information Technology**

**Acropolis Institute of Technology & Research, Indore**

**December 2019**

**Contents**

1. **Abstract …………………………………………………………………….3**
2. **Introduction ………………………………………………………………..3**
3. **Objective …………………………………………………………………...5**
4. **Scope ………………………………………………………………………..5**
5. **Study of Existing Techniques ……………………………………………..6**
6. **Project Description ………………………………………………………..7**
   1. Goal ………………………………………………………………………….………7
   2. Dependencies ………………………………………………………………………...8
   3. Project Structure …………………………………………………………………......8
   4. Pipeline ………………………………………………………………………………8

1. **Specific Requirements ……………………………………………………12**
   1. Functional Requirements ……………………………………………………….…12
   2. Software Requirements ……………………………………………………………12
   3. Hardware Requirements …………………………………………………………..12
2. **Diagrams ………………………………………………………………….14**
   1. Flow Diagram ………………………………………………………………………13
   2. Block Diagram ……………………………………………………………………..13
3. **Limitations ………………………………………………………………..14**
4. **Conclusion ………………………………………………………………...14**
5. **References ………………………………………………………………...14**

**1. Abstract**

Lane detection is a challenging problem. It has attracted the attention of the computer vision community for several decades. Essentially, lane detection is a multi feature detection problem that has become a real challenge for computer vision and machine learning techniques. Although many machine learning methods are used for lane detection, they are mainly used for classification rather than feature design. But modern machine learning methods can be used to identify the features that are rich in recognition and have achieved success in feature detection tests. However, these methods have not been fully implemented in the efficiency and accuracy of lane detection. In this project, we propose a new method to solve it. We introduce a new method of preprocessing and ROI selection. The main goal is to use the HSV color transformation to extract the white features and add preliminary edge feature detection in the preprocessing stage and then select ROI on the basis of the proposed preprocessing. This new preprocessing method is used to detect the lane.

**2. Introduction**

In any driving scenario, lane lines are an essential component of indicating traffic flow and where a vehicle should drive. It's also a good starting point when developing a self-driving car! In this project, I'll be showing you how to build your own lane detection system in OpenCV using Python. Here's the structure of our lane detection pipeline:

* Reading Images
* Color Filtering in HLS
* Region of Interest
* Canny Edge Detection
* Hough Line Detection
* Line Filtering & Averaging
* Overlay detected lane
* Applying to Video

With the rapid development of society, automobiles have become one of the transportation tools for people to travel. In the narrow road, there are more and more vehicles of all kinds. As more and more vehicles are driving on the road, the number of victims of car accidents is increasing every year. How to drive safely under the condition of numerous vehicles and narrow roads has become the focus of attention. Advanced driver assistance systems which include lane departure warning (LDW), Lane Keeping Assist, and Adaptive Cruise Control (ACC) can help people analyze the current driving environment and provide appropriate feedback for safe driving or alert the driver in dangerous circumstances. This kind of auxiliary driving system is expected to become more and more perfect.

Lane detection is a hot topic in the field of machine learning and computer vision and has been applied in intelligent vehicle systems. The lane detection system comes from lane markers in a complex environment and is used to estimate the vehicle’s position and trajectory relative to the lane reliably. At the same time, lane detection plays an important role in the lane departure warning system. The lane detection task is mainly divided into two steps: edge detection and line detection.

Qing proposed the extended edge linking algorithm with directional edge gap closing. The new edge could be obtained with the proposed method. Mu and Ma proposed Sobel edge operator which can be applied to adaptive area of interest (ROI). However, there are still some false edges after edge detection. These errors will affect the subsequent lane detection. Wang et al. proposed a Canny edge detection algorithm for feature extraction. The algorithm provides an accurate fit to lane lines and could be adaptive to complicated road environment. In 2014, Srivastava proposed that the improvements to the Canny edge detection can effectively deal with various noises in the road environment. Sobel and Canny edge operator are the most commonly used and effective methods for edge detection.

Line detection is as important as edge detection in lane detection. With regard to line detection, we usually have two methods which include feather-based method and model-based methods. Niu used a modified Hough transform to extract segments of the lane profile and used DBSCAN (density based spatial application noise clustering) clustering algorithm for clustering. In 2016, Mammeri used progressive probabilistic Hough transform combined with maximum stable extreme area (MSER) technology to identify and detect lane lines and utilized Kalman filter to achieve continuous tracking. However, the algorithm does not work well at night.

In this project, we propose a lane detection method that is suitable for all kinds of vehicles. First, we preprocessed each frame image and then selected the area of interest (ROI) of the processed images. Finally, we only needed edge detection vehicle and line detection for the ROI area. In this study, we introduced a new preprocessing method and ROI selection method. First, in the preprocessing stage, we converted the RGB color model to the HSV color space model and extracted white features on the HSV model. At the same time, the preliminary edge feature detection is added in the preprocessing stage, and then the part below the image is selected as the ROI area based on the proposed preprocessing.

**3. Objective**

The objective of this project is to use traditional Computer Vision techniques to develop an advanced and robust algorithm that can detect and track lane boundaries in a video. The pipeline highlighted below was designed to operate under the following scenarios:

* To detect *exactly* two lane lines, i.e. the left and right lane boundaries of the lane the vehicle is currently driving in.
* To carry load and transfer load between two divisions of any Industry/Factory.
* To use radium tapes to make lane which will be detected by machine as it will reduce the electricity cost of the factory with manual labour as well.
* To detect both the lanes because if only one of two lane lines have been successfully detected, then the detection is considered invalid and will be discarded. In this case, the pipeline will instead output a lane line fit (for both left and right) based on the moving average of the previous detections. This is due to the lack of an implementation of the lane approximation function (which is considered as future work).

**4. Scope**

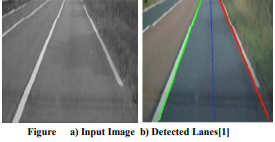
The lane detection problem, at least in its basic setting, does not look like a hard one. In this basic setting, one has to detect only the host lane, and only for a short distance ahead. A relatively simple Hough transform-based algorithm, which does not employ any tracking or image-to-world reasoning, solves the problem in roughly 90% of the highway cases. In spite of that, the impression that the problem is easy is misleading, and building a useful system is a large-scale R&D effort. The main reasons for that are significant gaps in research, high reliability demands, and large diversity in case conditions.

The reliable intelligent driver assistance systems and safety warning systems is still a long way to go. However, as computing power, sensing capacity, and wireless connectivity for vehicles rapidly increase, the concept of assisted driving and proactive safety warning is speeding towards reality. As technology improves, a vehicle will become just a computer with tires. Driving on roads will be just like surfing the Web: there will be traffic congestion but no injuries or fatalities. Advanced driver assistant systems and new sensing technologies can be highly beneficial, along with large body of work on automated vehicles.

**5. Related Work**

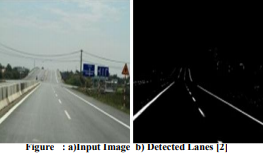
5.1 Results using Hough Transform

Mariut et al in his research paper proposed a simple algorithm that detected the lane marks, lane mark’s characteristics and had the ability to determine the travelling direction. It used the well known Hough transform to detect the potential lines in the images. To ensure the right detection of the lane mark, they had developed a technique that extracts the inner margin of the lane. The margins are highlighted by generating the magnitude image.



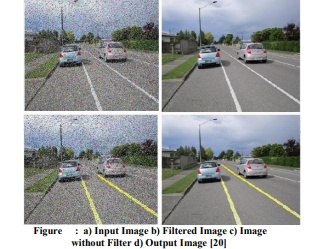
5.2 Results using Hough Transformation and Filters

T.T Tran et al in his research paper proposed an adaptive method based on HSI color model to detect lane marking. First, they converted RGB-based image to its HSI-based image. However, HSI color model was improved by the change in the way to calculate the intensity (I) component from RGB color images. From observing the color images of the road scene in HIS color space, they utilized the limited range of color. Hence, H, S and I component were used in this method. The proposed method can label the location of lane marking accurately.



5.3 Results of Lane Detection based on HSI model

S. Srivastava et al in his research paper proposed an efficient ways of noise reduction in the images by using different filtering techniques in this paper. The main objective was to design, develop, implement and subsequently simulate an efficient lane detection algorithm which will provide high quality results in the case when noise is present in the signal. Various filters used for comparison were median, wiener, and hybrid median filters.



**6. Project Description**

6.1 Goal

The goal of this project is to build up a simple image pipeline (take a frame from video as an input, do something, return a modified version of the frame), which allows detecting lane lines in simple conditions: sunny weather, good visibility, no cars in sight, only straight lanes. One more thing*:*our lane line detector should be linear*.*

Our main contribution in this project is to do a lot of work in the preprocessing stage. We proposed to perform color transform of HSV in the preprocessing stage, then extract white, and then perform conventional preprocessing operations in sequence. Moreover, we selected an improved method proposed in the area of interest (ROI). In this project, based on the proposed preprocessing method (after HSV color transform, white feature extraction, and basic preprocessing), one-half part of the processed image is selected as the area of interest (ROI). In addition, we performed twice edge detection. The first is in the preprocessing stage, and the second is in the lane detection stage after the ROI is selected. The purpose of performing twice edge detection is to enhance the lane recognition rate.

### 6.2 Dependencies

* Python 3.7
* NumPy
* Matplotlib (for charting and visualising images)
* OpenCV 4.1
* MoviePy (to process video files)

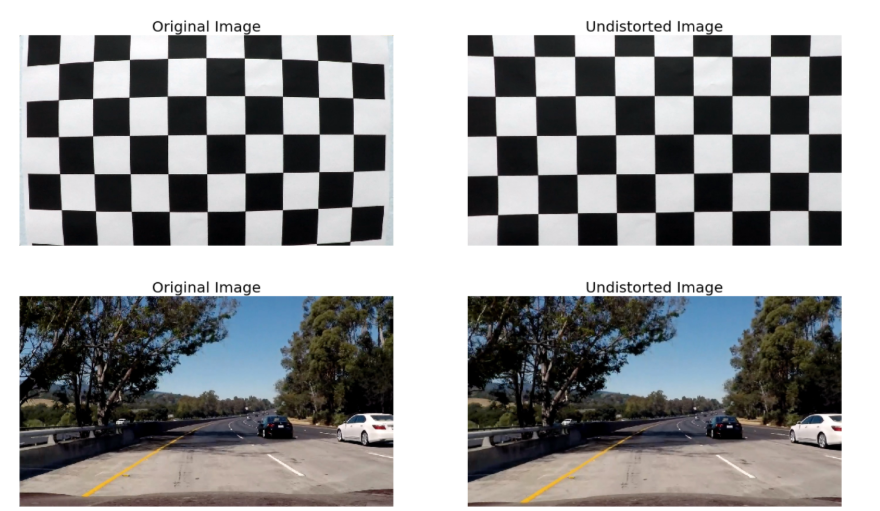
### 6.3 Project structure

* **lane\_tracker.ipynb**: Jupyter notebook with a step-by-step walkthrough of the different components of the pipeline
* **test\_images/**: Folder containing a set of images for test purposes
* **readme\_images**: Directory to store images used within this README.md
* **challenge\_video.mp4**: Video containing uneven road surfaces and non-uniform lighting conditions
* **challenge\_video\_output.mp4**: Resulting output on passing the challenge\_video through the pipeline
* **project\_video.mp4**: Video with dark road surfaces and non-uniform lighting conditions
* **project\_video\_output.mp4**: Resulting output on passing the project\_video through the pipeline

6.4 Pipeline

The various steps invovled in the pipeline are as follows, each of these has also been discussed in more detail in the sub sections below:

* Compute the image/ video calibration matrix and distortion coefficients given a set of chessboard images.
* Apply a distortion correction to raw images.
* Apply a perspective transform to rectify image ("birds-eye view").
* Use color transforms, gradients, etc., to create a thresholded binary image.
* Detect lane pixels and fit to find the lane boundary.
* Determine the curvature of the lane and vehicle position with respect to center.
* Warp the detected lane boundaries back onto the original image.
* Output visual display of the lane boundaries and numerical estimation of lane curvature and vehicle position.



### 6.4.1 Generating a thresholded binary image

### Many techniques such as gradient thresholding, thresholding over individual colour channels of different color spaces and a combination of are to be experimented with over a training set of images with the aim of best filtering the lane line pixels from other pixels. The experimentation yielded the following key insights:

1. The performance of indvidual color channels varied in detecting the two colors (white and yellow) with some transforms significantly outperforming the others in detecting one color but showcasing poor performance when employed for detecting the other. Out of all the channels of RGB, HLS, HSV and LAB color spaces that were experiemented with the below mentioned provided the greatest signal-to-noise ratio and robustness against varying lighting conditions:
   * White pixel detection: R-channel (RGB) and L-channel (HLS)
   * Yellow pixel detection: B-channel (LAB) and S-channel (HLS)
2. Owing to the uneven road surfaces and non-uniform lighting conditions a strong need for Adaptive Thresholding is to be realized.

### 6.4.2 Lane Line detection: Sliding Window technique

A wrapped thresholded binary image where the pixels are either 0 or 1; 0 (black color) constitutes the unfiltered pixels and 1 (white color) represents the filtered pixels. The next step involves mapping out the lane lines and determining explicitly which pixels are part of the lines and which belong to the left line and which belong to the right line.

The first technique employed to do so is: **Peaks in Histogram & Sliding Windows**

1. We first take a histogram along all the columns in the lower half of the image. This involves adding up the pixel values along each column in the image. The two most prominent peaks in this histogram will be good indicators of the x-position of the base of the lane lines. These are used as starting points for our search.
2. From these starting points, we use a sliding window, placed around the line centers, to find and follow the lines up to the top of the frame.

### 6.4.3 Lane Line detection: Adaptive Search

After detecting the two lane lines, for subsequent frames in a video, we will search in a margin around the previous line position instead of performing a blind search.

Although the Peaks in Histogram and Sliding Windows technique does a reasonable job in detecting the lane line, it often fails when subject to non-uniform lighting conditions and discolouration. To combat this, a method that could perform adaptive thresholding over a **smaller** receptive field/window of the image was needed. The reasoning behind this approach was that performing adaptive thresholding over a smaller kernel would more effectively filter out our 'hot' pixels in varied conditions as opposed to trying to optimise a threshold value for the entire image.

## **6.5 Applying Canny Detector**

The Canny Detector is a multi-stage algorithm optimized for fast real-time edge detection. The fundamental goal of the algorithm is to detect sharp changes in luminosity (large gradients), such as a shift from white to black, and defines them as edges, given a set of thresholds. The Canny algorithm has four main stages:

**6.5.1. Noise reduction**

As with all edge detection algorithms, noise is a crucial issue that often leads to false detection. A 5x5 Gaussian filter is applied to convolve (smooth) the image to lower the detector’s sensitivity to noise. This is done by using a kernel (in this case, a 5x5 kernel) of normally distributed numbers to run across the entire image, setting each pixel value equal to the weighted average of its neighboring pixels.

6.5.2. Intensity gradient

The smoothened image is then applied with a Sobel, Roberts, or Prewitt kernel (Sobel is used in OpenCV) along the x-axis and y-axis to detect whether the edges are horizontal, vertical, or diagonal.

6.5.3. Non-mximum suppression

Non-maximum suppression is applied to “thin” and effectively sharpen the edges. For each pixel, the value is checked if it is a local maximum in the direction of the gradient calculated previously.

6.5.4. Hysteresis thresholding

After non-maximum suppression, strong pixels are confirmed to be in the final map of edges. However, weak pixels should be further analyzed to determine whether it constitutes as edge or noise. Applying two pre-defined minVal and maxVal threshold values, we set that any pixel with intensity gradient higher than maxVal are edges and any pixel with intensity gradient lower than minVal are not edges and discarded. Pixels with intensity gradient in between minVal and maxVal are only considered edges if they are connected to a pixel with intensity gradient above maxVal.

## **6.5.5. Segmenting lane area**

We will handcraft a triangular mask to segment the lane area and discard the irrelevant areas in the frame to increase the effectiveness of our later stages.  
  
**6.5.6. Hough transform**

In the Cartesian coordinate system, we can represent a straight line as y = mx + b by plotting y against x. However, we can also represent this line as a single point in Hough space by plotting b against m. For example, a line with the equation y = 2x + 1 may be represented as (2, 1) in Hough space.

## **6.5.7. Visualization**

The lane is visualized as two light green, linearly fitted polynomials which will be overlayed on our input frame.

**7. Specific Requirements**

Functional Requirements

* The lane lines are mapped from the road which are a result of determination of pixels which would be a part of the right and left lines of the lane.
* A way which would help to reduce traffic problems as lesser changing of lanes would take place.
* A method which would combine for the concept of self-driving cars.

Software Requirements

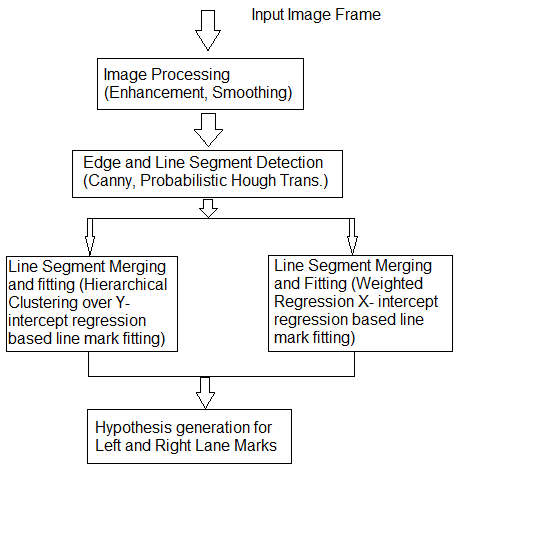
* Jetbrains by Pycharm community edition 2019.2.1
* Python 3.7.4
* OpenCV
* Important packages such as Matplotlib
* Pillow
* Pip
* Numpy
* Opencv-python
* PyBundle
* Kiwisolver and many more library packages.

Hardware Requirements

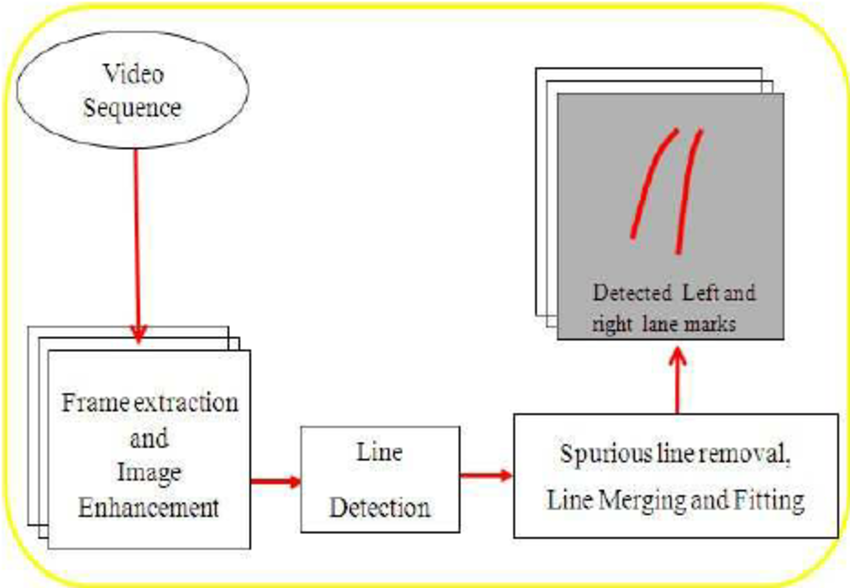
* Intel Core i3 or above
* Windows 10
* RAM 4 GB or above
* Memory 500 GB or above

**8. Diagrams**

Flow Diagram



Block Diagram



**9. Limitations**

* A recorder system to check the utility of the software on road and along with physical conditions is unavailable.
* A running concept of research which is not accepted widely in the current world scenario.

**10. Conclusion**

In this project, we proposed a new lane detection preprocessing and ROI selection methods to design a lane detection system. The main idea is to add white extraction before the conventional basic preprocessing. Edge extraction has also been added during the preprocessing stage to improve lane detection accuracy. We also placed the ROI selection after the proposed preprocessing. Compared with selecting the ROI in the original image, it reduced the nonlane parameters and improved the accuracy of lane detection. Currently, we only use the Hough transform to detect straight lane and track lane and do not develop advanced lane detection methods. In the future, we will exploit a more advanced lane detection approach to improve the performance.

**11. References**

1. D. Pomerleau, “RALPH: rapidly adapting lateral position handler,” in Proceedings of the Intelligent Vehicles '95. Symposium, pp. 506–511, Detroit, MI, USA, 2003.
2. J. Navarro, J. Deniel, E. Yousfi, C. Jallais, M. Bueno, and A. Fort, “Influence of lane departure warnings onset and reliability on car drivers' behaviors,” Applied Ergonomics, vol. 59, pp. 123–131, 2017.
3. P. N. Bhujbal and S. P. Narote, “Lane departure warning system based on Hough transform and Euclidean distance,” in Proceedings of the 3rd International Conference on Image Information Processing, ICIIP 2015, pp. 370–373, India, December 2015.  V. Gaikwad and S. Lokhande, “Lane Departure Identification for Advanced Driver Assistance,” IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 910–918, 2015.
4. Ni Wei and Ma Wanjing “Simulation Based Study on a Study Lane Assignment Approach for Freeway Weaving Section.” 2013.
5. F. Yuan, Z. Fang, S. Wu, Y. Yang, and Y. Fang, “Real-time image smoke detection using staircase searching-based dual threshold AdaBoost and dynamic analysis,” IET Image Processing, vol. 9, no. 10, pp. 849–856, 2015.
6. C.Y Kuo “On the Image Sesor Processing for Lane Detection and Control in Vehicle Lane Keeping Systems.” April 2019
7. M.-C. Chuang, J.-N. Hwang, and K. Williams, “A feature learning and object recognition framework for underwater fish images,” IEEE Transactions on Image Processing, vol. 25, no. 4, pp. 1862–1872, 2016.
8. Y. Saito, M. Itoh, and T. Inagaki, “Driver Assistance System with a Dual Control Scheme: Effectiveness of Identifying Driver Drowsiness and Preventing Lane Departure Accidents,” IEEE Transactions on Human-Machine Systems, vol. 46, no. 5, pp. 660–671, 2016.
9. Q. Lin, Y. Han, and H. Hahn, “Real-Time Lane Departure Detection Based on Extended Edge-Linking Algorithm,” in Proceedings of the 2010 Second International Conference on Computer Research and Development, pp. 725–730, Kuala Lumpur, Malaysia, May 2010.
10. Zu Whan Kim “Robust Lane Detection and Tracking in Challenging Scenarios” March 2008.
11. J.-G. Wang, C.-J. Lin, and S.-M. Chen, “Applying fuzzy method to vision-based lane detection and departure warning system,” Expert Systems with Applications, vol. 37, no. 1, pp. 113–126, 2010.
12. S. Srivastava, M. Lumb, and R. Singal, “Improved lane detection using hybrid median filter and modified hough transform,” International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4, no. 1, pp. 30–37, 2014.
13. Dinesh Kumar and Gurveen Kaur “Lane Detection Techniques: A Review.” Feb 2015
14. J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, “Robust Lane Detection using Two-stage Feature Extraction with Curve Fitting,” Pattern Recognition, vol. 59, pp. 225–233, 2015.
15. Aditya Singh Rathore “Lane Detection for Autonomous Vehicle using OPEN CV Libarary”, January 2019.
16. A. Mammeri, A. Boukerche, and Z. Tang, “A real-time lane marking localization, tracking and communication system,” Computer Communications, vol. 73, pp. 132–143, 2016.
17. C. J. Chen, B. Wu, W. H. Lin, C. C. Kao, and Y. H. Chen, “Mobile lane departure warning system in,” in Proceedings of the 2009 IEEE 13th International Symposium on Consumer Electronics, pp. 1–5, 2009.
18. J. W. Lee, C. D. Kee, and U. K. Yi, “A new approach for lane departure identification,” in Proceedings of the IEEE IV2003 Intelligent Vehicles Symposium, pp. 100–105, 2003.
19. J. W. Lee and U. K. Yi, “A lane-departure identification based on LBPE, Hough transform, and linear regression,” Computer Vision and Image Understanding, vol. 99, no. 3, pp. 359–383, 2005.
20. H. Xu and H. Li, “Study on a robust approach of lane departure warning algorithm,” in Proceedings of the EEE International Conference on Signal Processing System (ICSPS), pp. 201–204, 2010.
21. Ahron Bal Hilel, Ronnen Lener “Recent Progress in Road and Lane Detection: A Survey.”, December 2011.
22. H. Xu and H. Li, “Study on a robust approach of lane departure warning algorithm,” in Proceedings of the IEEE International Conference on Signal Processing System (ICSPS), pp. 201–204, 2010.
23. . Jiang Ryui “Lane detection and tracking Using a New Lane Model and Distance Transform”, January 2011.
24. H. Aung and M. H. Zaw, “Video based lane departure warning system using hough transform,” in Proceedings of the International Conference on Advances in Engineering and Technology, pp. 85–88, Singapore, 2010.
25. Zaier Zaidi, “Evaluating Variable Speed Limits and Dynamic Lane Merging Systems in Work Zones: A Simulation Study” May 2012.
26. S.-C. Huang and B.-H. Chen, “Automatic moving object extraction through a real-world variable-bandwidth network for traffic monitoring systems,” IEEE Transactions on Industrial Electronics, vol. 61, no. 4, pp. 2099–2112, 2014.