A Novel Lightweight Lane Departure Warning System Based on Computer Vision for Improving Road Safety

by

Yue Chen

Thesis submitted to the University of Ottawa
in partial Fulfillment of the requirements for the
Master of Computer Science degree

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

With the rapid improvement of the Advanced Driver Assistant System (ADAS), autonomous driving has become one of the most common hot topics in recent years. While driving, many technologies related to autonomous driving choose to use the sensors installed on the vehicle to collect the information of road status and the environment outside. This aims to warn the driver to perceive the potential danger in the fastest time, which has become the focus of autonomous driving in recent years.

Although autonomous driving brings plenty of conveniences to people, the safety of it is still facing difficulties. During driving, even the experienced driver can not guarantee focus on the status of the road all the time. Thus, lane departure warning system (LDWS) becomes developed. The purpose of LDWS is to determine whether the vehicle is in the safe driving area. If the vehicle is out of this area, LDWS will detect it and alert the driver by the sensors, such as sound and vibration, in order to make the driver back to the safe driving area.

This thesis proposes a novel lightweight LDWS model LEHA, which divides the entire LDWS into three stages: image preprocessing, lane detection, and lane departure recognition. Different from the deep learning methods of LDWS, our LDWS model LEHA can achieve high accuracy and efficiency by relying only on simple hardware.

The image preprocessing stage aims to process the original road image to remove the noise which is irrelevant to the detection result. In this stage, we apply a novel algorithm of grayscale preprocessing to convert the road image to a grayscale image, which removes the color of it. Then, we design a binarization method to greatly extract the lane lines from the background. A newly-designed image smoothing is added to this stage to reduce most of the noise, which improves the accuracy of the following lane detection stage.

After obtaining the processed image, the lane detection stage is applied to detect and mark the lane lines. We use region of interest (ROI) to remove the irrelevant parts of the road image to reduce the detection time. After that, we introduce the Canny edge detection method, which aims to extract the edges of the lane lines. The last step of LDWS in the lane detection stage is a novel Hough transform method, the purpose of it is to detect the position of the lane and mark it.

Finally, the lane departure recognition stage is used to calculate the deviation distance between the vehicle and the centerline of the lane to determine whether the warning needs

to turn on. In the last part of this paper, we present the experiment results which show the comparison results of different lane conditions. We do the statistic of the proposed LDWS accuracy in terms of detection and departure. The detection rate of our proposed LDWS is 98.2% and the departure rate of it is 99.1%. The average processing time of our proposed LDWS is 20.01×10^{-3}s per image.

Acknowledgements

First and foremost, I am grateful to my parents. With their support, I can study the related knowledge in university and devote myself to the research. This is the best gift for me in the first half of my life.

Next, I would like to thank my professor, Dr. Azzedine Boukerche. In the period of my research, he provides full effort to give me much useful guidance and advice to help me find the direction so that I can complete my research and achieve the goal. And I also would like to thank NSERC-CREATE TRANSIT Funds to support my research in these years.

Moreover, I have to appreciate all the help that Paradise Lab members give me. In the last three years, they inspired me and provided all they could to help me finish my research, such as Python programming, and set up the environment to test the proposed model in this thesis. I would like to give special thanks to Dr. Peng Sun, who gives me research direction and a lot of writing advice to complete my master thesis.

Publications

- **Yue Chen** and Azzedine Boukerche, "A Novel Lane Departure Warning System for Improving Road Safety", published by *Proceedings of 2020 IEEE International Conference on Communications (ICC)*, 2020.

- Azzedine Boukerche, Peng Sun and **Yue Chen**, "Analysis and Design of Lane Departure Warning System Methods for Supporting Autonomous Driving," submitted to *IEEE System Journal*, 2020.

- Azzedine Boukerche, **Yue Chen** and Peng Sun, "LEHA: A Novel Lightweight Efficient and Highly Accurate Lane Departure Warning System," is submitted to *Computer Communications*, 2021.

Table of Contents

List of Tables

List of Figures

Nomenclature

ADAS	Advanced Driver Assistant System
BFS	Breadth-First-Search
CED	Canny Edge Detection
FN	False Negative
GLCM	Gray-level Co-occurrence Matrix
IPM	Inverse perspective transformation
ITS	Intelligent Transportation System
LDWS	Lane Departure Warning System
ML	Miss the Line
MPM	Misclassification Penalty Metric
OL	Obtain the Line
PSNR	Peak Signal to Noise Ratio
ROI	Region of Interest
SC	Stochastic Computing
SMC	Stochastic Mean Circuit
TP	True Positive

Chapter 1

Introduction

Recently, the development of vehicles has facilitated people's daily life [13, 14]. However, it does not brings convenience to our travel experience only, one of the most serious disadvantages is that the incidence of car accidents and the fatalities caused by it are also increasing year by year [15].

1.1 Background and Problems

According to [16], the most likely cause of the car accident is that the driver may lose concentrates on the road status when driving. For example, the distance between the position of the vehicle and lane lines or other vehicles around it, especially during long-distance driving, no matter the driver is experienced or not [17]. Another situation is that when the driver driving on the highway, the high speed may cause the driver to have no time to realize the road status, which leads to car accidents. Losing sight of the road is the biggest culprit in car accidents. In 2016, the total fatality of car accidents in the world is over 1,350,000 [18], which is a reminder of the importance of driving safety. Only a few car accidents are caused by bad weather while most by human fault, such as loss concentrate on the road status [19]. Road status contains all the details during driving, such as the position of the vehicles on the road [20]. While driving on the road, especially at high speeds, even a momentary loss of concentration can lead to a deviation of the vehicle, such as make the vehicle approach the lane line and causes a collision [21, 22, 23]. Imagine a situation, while two vehicles driving on the adjacent lanes and both of them are approaching the same lane line, a car accident happens. Another situation is while the vehicle driving at the outermost lane, the outward deviation of the vehicle may hit the objects such as rocks. Therefore, how to improve driving safety is becoming a significant problem and autonomous driving is a potential solution [24, 25, 26].

Accordingly, to avoid such stern questions, various driving assistant systems have been developed [27, 28, 29]. According to [30], there are six levels in general, i.e., from level 0 to level 5, which describes the progressive relationship from no autonomous driving to fully autonomous driving in autonomous driving. Fig. 1.1 shows the relationship between these levels.

Level	**Monitor speed and direction**	**Monitor drivingenvironment**	**Under complicated lane conditions**
Level 0： No Automation	Human	Human	Human
Level 1： Driver Assistant	Human and System	Human	Human
Level 2： Partial Automation	System	Human	Human
Level 3： Conditional Automation	System	System	Human
Level 4： High Automation	System	System	System
Level 5： Full Automation	System	System	System

Figure 1.1: Levels for autonomous vehicles development.

According to Fig. 1.1, from level 0 to level 2, the driver needs to drive the vehicle and focus on the road status to react as soon as possible. In level 3, the system detects the environment to decide whether the driver needs to drive the vehicle, which is called conditional automation. Level 4 and level 5 indicate high automation and full automation respectively, which means the system will fully control the vehicle, and the driver does not need to do any driving operations. LDWS can be used from level 1 to level 5 in the form, which indicates that LDWS is the basic technology in autonomous driving. The purpose of LDWS is to provide a signal for warning drivers when the vehicle he is driving is not in the middle area of the lane and gradually close to one of the lane lines, which is based on the result of lane detection [31, 32, 33].

Among them, a lane departure warning system (LDWS) is considered to be the basic and important component [34, 35, 36]. LDWS aims to alert the driver when the vehicle derives from the centerline of the lane. In LDWS, when the derivation distance between the vehicle and the centerline of the line exceeds the threshold, the alarm will be triggered to attract the driver's attention to take the vehicle back to the safe driving area, which helps reduce the incidence of car accidents [37, 38, 39]. The survey has done by [15] shows that LDWS performs well in safety driving, which reduces the incidence of severities by 18% and the fatalities by 86% in car accidents.

In general, LDWS [40] is the desired functionalities of LDWS is achieved by solving the following three tasks sequentially.

- Image preprocessing: there are several processing steps in the image preprocessing stage, such as grayscale processing, binarization, region of interest (ROI) extraction, and inverse perspective mapping. After these steps, the system can obtain a processed image, which will improve the accuracy of the detection result and reduce the processing time of the subsequent steps.

- Lane detection: in the lane detection stage, the initial step is to extract the lane lines from the background by feature extraction, which creates the new features. After extract the image features, Hough transform is applied to detect and mark the lane lines.

- Lane departure recognition: lane departure recognition stage is used to determine whether the vehicle is out of the safe driving area and needs to alert the driver to approach the middle area of the lane.

1.2 Motivation and Contribution

In our work, we proposed *LEHA (Lightweight Efficient and Highly Accurate LDWS) which only relies on simple CPU and can achieve high accuracy and efficiency*. LEHA consists of the following three modules: image pre-processing module, lane detection module, and lane departure recognition module. More precisely, these three modules work sequentially.

First, the image captured by the dashboard camera will be processed by the image pre-processing module for eliminating the noise in the image. Then, relying on the processed image, the lane detector will identify the boundaries of the lane within the road image. Finally, based on the position of the vehicle, the lane departure recognition module will calculate the distance between the vehicle and the centerline of the lane. After comparing this distance with the preset threshold, the lane departure recognition module determines whether the warning should be triggered or not. The main contributions of our work are summarized as follows:

- In the image pre-processing module, we improve the algorithm of grayscale preprocessing and design the new methods of image smoothing and binarization respectively to eliminate the noise and keep the most details in the original road image.
- In the lane detection module, we design an improved Region of interest (ROI) method and Hough Transform (HT) method to improve the detection accuracy and reduce the processing time.
- In the lane departure recognition module, we use the ratio of the deviation distance and the width of the lane to calculate the actual deviation distance in the real world and show it to the driver. In addition, we present the deviation direction and the deviation distance to remind the driver whether an adjustment of driving direction needs to be applied.

1.3 Thesis Outline

The rest of this paper is organized as follows: In Chapter 2, we compare and discuss both of the classical methods and novel methods in LDWS in recent years, i.e., image pre-processing, lane detection, and lane departure recognition. Accordingly, we also compare different methods in each stage and use forms to make it directly. In Chapter 3, we introduce our proposed methods in the above three stages and briefly discuss the strengths of them. Chapter 4 presents the experimental results and the comparison between our novel LDWS model and other LDWS models which are proposed in these years under different weather and luminance conditions. Finally, in Chapter 5, we conclude this paper and give some directions for future work.

Chapter 2

Related Work

In this chapter, we will simply introduce some basic concepts and methods about image preprocessing, lane detection, and lane departure recognition in LDWS, and present the relationships between these parts. All the comparisons of different methods work on the KITTI dataset [41].

2.1 Preliminaries

In order to explore a better method for lane departure warning, the first important and necessary thing is to review the related researches since the beginning of this area [42, 43, 44]. We conclude the following steps after comparing more than 100 research papers, i.e., image pre-processing, lane detection, and lane departure recognition.

2.1.1 Image preprocessing

In the image preprocessing stage, the purpose of it is to remove the useless image information and maintain the irrelevant information about lanes [45, 46, 47]. Generally, grayscale processing, binarization, ROI extraction, and inverse perspective mapping are considered as the basic techniques for processing the original image [48, 49, 50].

The initial step of image preprocessing is grayscale processing [51, 52]. Grayscale processing is the process of converting color images to grayscale images, which converts each pixel from a coordinate representation in RGB color space to a value representation substantially [53, 54, 55]. In the grayscale image, the value of each pixel is in the range of

[0,255]. The purpose of grayscale processing is that after obtaining the value of each pixel, it is easy to determine the threshold of the image for binarization by a number instead of a coordinate, which saves a lot of time and ensures accuracy as well [56, 57].

The mission of binarization is to ensure the threshold of the image and convert the grayscale image to a black-and-white image [58, 59]. There are two types of widely used methods to find the threshold: global and local [60, 61]. The global threshold is used for the entire image, while the local threshold is applied to different parts of the image depending on different situations, such as lighting. The purpose of binarization is to separate the objects that will be detected from the background in the image, simplify the post-processing and improve the processing speed [62, 63].

To simplify the process of finding the desired objective, the ROI extraction is adopted to remove non-necessary information from the image in the image preprocessing stage [64, 65, 66]. In LDWS, ROI is the area in front of the driver's perspective. According to the method of determining the size of ROI, existing ROI extraction methods can be categorized into two types, i.e., fixed ROI and dynamic ROI [67, 68, 69]. Fixed ROI extraction applies the same shape and position of region to different images, which almost does not take processing time but may reduce the accuracy [70]. Dynamic ROI extraction determines ROI depending on different situations of different images, which improves the processing accuracy, while spends some processing time on it [71].

Inverse perspective mapping is the process of converting the original image to a top-view image, which is also known as bird-view [21, 72]. After obtaining the top-view image, it is easy to know the road status in front of the vehicle visually, which helps improve the accuracy of the following lane detection stage [73, 74, 75].

2.1.2 Lane detection

Lane detection is one of the challenges in computer vision and Intelligent Transportation System (ITS) [76, 77, 78, 79], which is becoming the most basic and significant part of LDWS. Actually, almost all the techniques related to lanes are based on lane detection. The purpose of this stage is to detect lanes on the road and extract the boundaries of the lane from the background of the given image, which provides essential information for further determining the vehicle [80, 81, 82]. Based on the predicted moving, we can determine if the vehicle is driving in the middle area of the lane and if a warning needed to be triggered in lane departure recognition [83, 84, 85, 86].

Feature extraction is the process of mapping an old set of n features to a new one which contains n new features [87, 88, 89, 90, 91, 92]. Fig. 2.1 shows the process of it. In this paper, we divide lane detection into three categories based on different methods: color-based, texture-based, and edge-based.

Color-based feature extraction is the process of recognizing the color distribution in the given image to extract the lane lines [93, 94, 95, 96]. However, color is the global feature of the image and can not precisely describe the local information, which will lose some local details and reduce the processing accuracy [97, 98, 99]. Different from feature extraction based on color, texture-based feature extraction does not focus on the pixels, it needs to perform statistical calculations in an area with pixels, which is the statistical feature. However, both of these two methods share the same disadvantage, which can not process the local details well. Edge-based feature extraction is the most wide-used method in lane detection [100, 101, 102, 103, 104]. In this paper, we introduce three classical methods of it: Sobel edge detection [105, 106], Canny edge detection [107], and Hough transform [108]. Sobel and Canny are two operators who work for edge feature detection, which provides different ways to detect the lane lines and Hough transform works for detecting the lane lines by using coordinate transformation [109, 110].

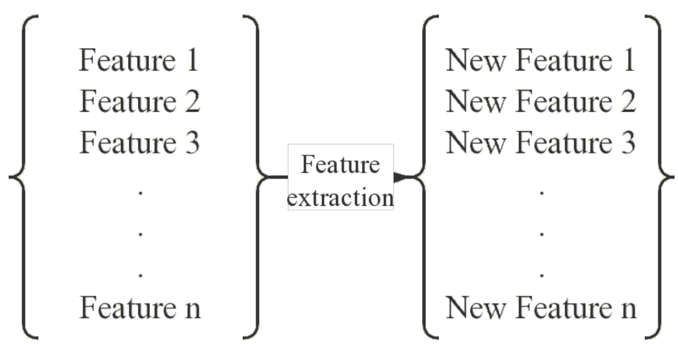

Figure 2.1: The process of feature extraction.

2.1.3 Lane departure recognition

In LDWS, lane departure recognition is the step to determine whether the vehicle is a departure from the safe driving area and needs to be warned or not [111, 112, 113, 114, 115]. In general, there are two ways to determine: lane departure recognition based on vanishing point and lane departure recognition based on the position of origin [116, 117, 118]. In the vanishing point method, the horizontal displacement change of the vanishing point in the continuous frame is used to determine whether the vehicle is a departure or not [119]. The departure distance can be calculated by the ratio of displacement and the width of the lane [120, 121]. In the position of origin method, we set the vehicle and the vehicle's horizontal straight line intersects the lanes on either side of it at two points as three marked points [122]. After setting the position of the origin, we create a coordinate and obtain the coordinate of the above three marked points [123]. Therefore, we can get the distance between the vehicle and the lane lines by calculation and then recognize the lane departure.

2.2 Image preprocessing stage

The image preprocessing stage is the initial stage of LDWS, which aims to simplify and speed up the processing [124, 125, 126]. In this section, we introduce four sequential steps in image preprocessing, i.e., grayscale processing, binarization, ROI extraction, and inverse perspective mapping. The structure of the above methods in this stage is shown in Fig. 2.2.

2.2.1 Grayscale processing

In the RGB color model, each color in the image is obtained by superimposing three colors, i.e., red, green, blue. In this model, each pixel p in the image can be represented by $R(x_p, y_p)$, $G(x_p, y_p)$, $B(x_p, y_p)$. When $R(x_p, y_p) = G(x_p, y_p) = B(x_p, y_p)$, the given image becomes a grayscale image and the process of equalizing these three values is called grayscale processing [127, 128, 129]. However, as the objects in a color image will reflect different colors depending on the lighting conditions, color can not provide enough reliable information [130, 131]. Grayscale images can avoid the uncertainty caused by the changes of colors. Thus, grayscale processing is the initial step in the image preprocessing stage [132]. In general, there are three methods for grayscale processing, i.e., global mapping, local mapping, global and local mapping combination [133, 134, 135].

Global mapping method

Normally, the global mapping method usually applies one fixed transform function to all the pixels that exist in the given image. This method focuses on the global structure

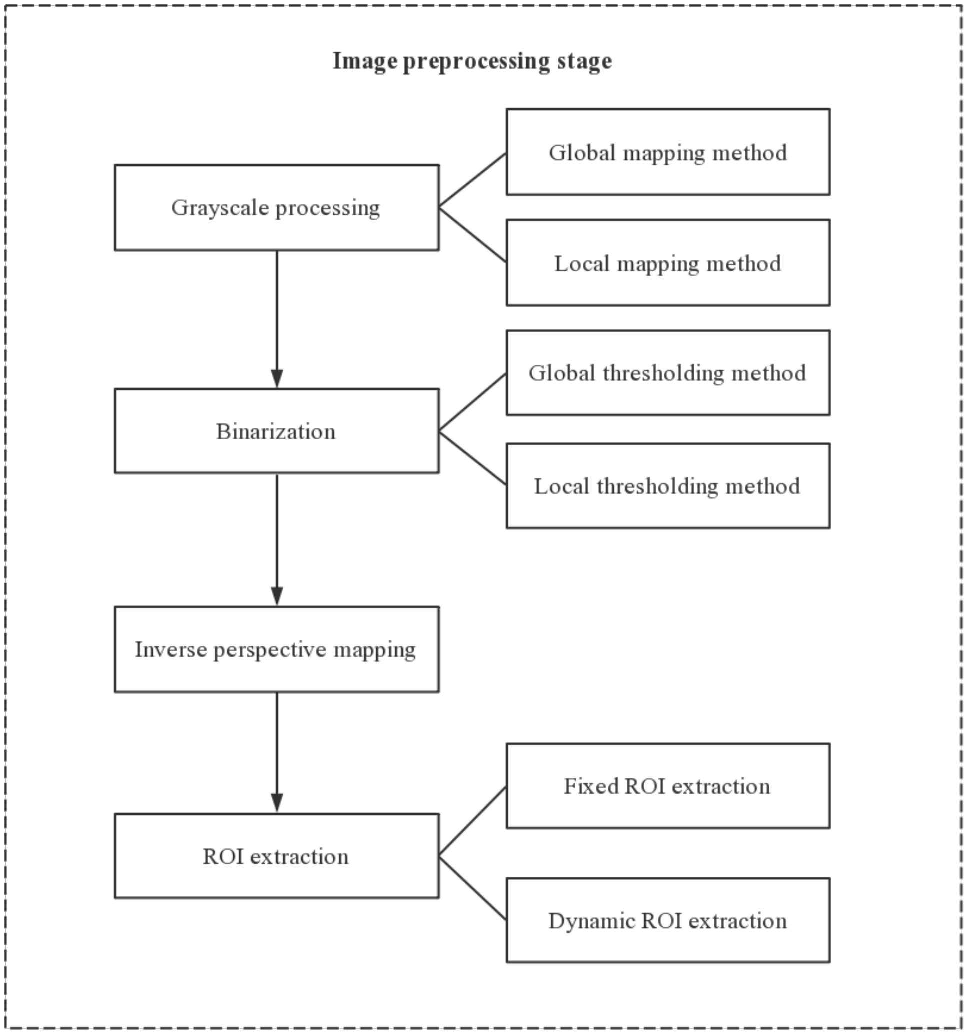

Figure 2.2: Different methods in image preprocessing.

and the gray value of each pixel on the output image is irrelevant with the position of it [136, 137, 138, 139].

In 2005, Gooch et al. [140] proposed a method for converting color image to grayscale image by optimizing local difference information between pixel pairs. The algorithm obtains the target differences between adjacent pixels according to the luminance and chrominance. Finally, the gray image of the original image is obtained by solving the optimization equation, which is constructed through the obtained chrominance and luminance information.

In [141], Grundland and Dodgson presented a real-time color image to grayscale image algorithm by enhancing contrast. As the components of YPQ color space are approximately the same, they are all equal to 1, which makes the calculation simpler and accelerate the image processing speed. There are five steps in the proposed algorithm:

1. Convert the image from RGB color model (R_i, G_i, B_i) to the linear YPQ color space (Y_i, P_i, G_i) by using the following formula:

$$\begin{bmatrix} Y_i \\ P_i \\ Q_i \end{bmatrix} = \begin{bmatrix} 0.2989 & 0.5870 & 0.1140 \\ 0.5000 & 0.5000 & -1.0000 \\ 1.0000 & -1.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix}. \tag{2.1}$$

2. Use Gaussian pairing in image sampling between pixels to analyze the color contrast distribution in the image features;

3. Use predominant component analysis to realize dimension reduction, in order to reduce the complexity of calculation;

4. Combine luminance and chrominance information by projecting the sampled (P_i, Q_i) to obtain the main color contrast information and then introduced into the channel Y_i;

5. The dynamic range of the gray image is normalized by saturation correction.

The advantage of [141] is that it improves the speed and efficiency of calculation, reduces the computational complexity, and shortens the processing time. However, when differences in luminance and chrominance cancel each other out, the features of the image may lose [142, 143].

In [144], a visual contrast model based on salience was proposed, which takes color difference and spatial relationship into consideration. This algorithm transforms the problem of preserving contrast in the grayscale processing algorithm into the problem of minimizing the salience between the gray-scale image and the original image. The author introduced CIE LAB color space [144] in this method, which is used in the contrast model, and RGB color space linear combination is used in the global mapping function by giving the parameters $x = (x_1, x_2, x_3)$ as follows:

$$gray = x_1 r + x_2 g + x_3 b. \tag{2.2}$$

As the global mapping method applies one fixed mapping function to the entire image, it can greatly keep the global structure of the image [145, 146, 147]. However, based on this reason, the processed image may be too smooth to lose some local details, and can not keep enough image features. Table 2.1 shows the comparison of the classic algorithm in the global mapping method.

Table 2.1: Global mapping methods of grayscale processing

Methods	Color space	Pros	Cons
Gooch et al. [140]	CIE LAB	Keep global color consistency of the image.	1. The computation complexity is $O(n^2)$, which is not suitable for practical application. 2. May lose image details for not focusing on the local color consistency of the image.
Grundland and Dodgson [141]	YPQ	Improve the real-time performance of grayscale processing.	May lose some important details of the image because of the proposed formula.
Zhou et al. [144]	CIE LAB RGB	Keep the contrast information for the majority of the images.	When the contrast of each part of the image is low, the advantage of this algorithm is not obvious.
Nafchi et al. [148]	RGB	1. The computation complexity is lower than most of the grayscale processing method. 2. The speed of processing is high.	This method will make the image smooth and lose some details of the it.

Local mapping method

In the local mapping method, the gray-scale value is determined by the location of the pixels. The pixel mapped from the color image to the gray image changes with space [149, 150].

Bala et al. [151] introduced a local mapping grayscale processing method. This method maintains the difference between the adjacent colors to obtain the result of the grayscale image. The advantage of [151] is that it can restore the lost color details in the image. However, this method is also easy to be affected by chromaticity and luminance, which may affect the accuracy of grayscale processing. In [152], the author improved the accuracy of grayscale processing performance by local operators SIFT [153] and avoided using color quantization and destructing the gradient information. This method improved the reliability and efficiency of grayscale processing.

Zhu et al. [154] present a method of grayscaling processing by using Channel Salience, which measures channel differences according to the filtering theory. The result of this algorithm G is obtained by adding luminance channel L_{ZHU} to the constrained contrast map αM_{ZHU}. The mapping function is as follows:

$$G_{Zhu} = L_{Zhu} + \alpha M_{Zhu}. \tag{2.3}$$

The advantage of the local mapping method is to precisely keep the local details of the original image [155]. However, as the mapping function focuses on the local information and can not be used in the entire image, the grayscaling-processed image may lose some global information and make the global structure unclear [156]. Table 2.2 shows the comparison of the above three classic local mapping method of grayscale processing and Fig. 2.3, 2.4, and 2.5 show the comparison of the original image, global mapping method result [148] and local mapping method result [154]respectively.

Table 2.2: Local mapping methods of grayscale processing

Methods	**Color space**	**Pros**	**Cons**
Bala et al. [151]	RGB	Helps restore the color details in the image.	This method is easy to be affected by the change of chromaticity and luminance, which will affect the result of processing.
Ancuti et al. [152]	RGB	Improve the real-time and the speed performance of grayscale processing.	When processing a non-ideal image with various noises, the performance of this method is much less than the expected.
Zhu et al. [154]	IHLS	1. The efficiency of this method is higher than most of the grayscale processing method. 2. This method greatly removes the noise of the image and preserve the contrast.	When the contrast of each part of the image is low, the advantage of this algorithm is not obvious.

Figure 2.3: The original image with lane.

Figure 2.4: The result of global mapping method.

Figure 2.5: The result of local mapping method.

2.2.2 Binarization

Binarization is the process of converting the gray-scale image to a black-and-white image. In a gray-scale image, each pixel has a value from 0 to 255, where 0 and 255 stand black and white respectively, and the value between 0 to 255 represents different brightness of the pixel [157, 158, 159]. Binarization is a method of classifying pixels of an entire image into 0 and 255 by setting a threshold value T. If the pixel value $P_i(x_i, y_i)$ is greater than T, set it as 255, which represent as white. Otherwise, set it as 0, which is black. The basic idea of it is as follows:

$$P_i(x_i, y_i) = \begin{cases} 255 & \text{if } P_i(x_i, y_i) \geq T \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

Finding a suitable threshold value is the main task of image binarization [123, 160, 161]. In general, there are two methods for setting the threshold: the global thresholding method and the local thresholding method. In order to illustrate the efficiency of the methods of binarization, we introduce some evaluation criteria, i.e., F-Measure, Peak Signal to Noise Ratio (PSNR), and Misclassification Penalty Metric (MPM). F-Measure is the harmonic mean of recall and precision. For every binarization method, when the values of MPM are higher and the value of PSNR becomes lower, the performance of this binarization is better [162, 163, 164]. The computation formula of the above evaluation criterion are shown as follows:

$$F - Measure_{Binarization} = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{2.5}$$

$$PSNR = 10 \times log_{10} \left[\frac{N_x N_y 255^2}{\sum_{x=1}^{N_x} \sum_{x=1}^{N_y} (f(x,y) - \hat{f}(x,y))^2} \right], \tag{2.6}$$

$$MPM = \frac{\frac{\sum_i d_{FN}^i}{D} + \frac{\sum_j d_{FP}^j}{D}}{2}, \tag{2.7}$$

where N_x and N_y indicate the number of pixels of the length and width in the image, $\sum_i d_{FP}^i$ and $\sum_j d_{FP}^j$ are the distances between the i^{th} pixel to the boundary and the distance between the j^{th} pixel to the boundary. D presents the sum of the distance between all the pixels in the image to the boundary.

Global thresholding method

The global thresholding method is applying the same threshold to the whole image. Applying different threshold values to the same image will output the different binary images with different brightness [165, 166]. In [167, 168], these image binarization algorithms set the fixed threshold value at first. Actually, one threshold value can not be used for all the images, focusing on different images will use a different threshold value, which is very difficult to determine. Fig. 2.6, 2.7, 2.8, 2.9, and 2.10 show the original image and the comparison of different threshold values as 50, 128, 210 and 240 apply to the same image.

Figure 2.6: The original image with lane.

Figure 2.7: The result of setting threshold as 50.

Kittler [169] is one of the most wide-used image binarization algorithms. The basic idea of it is calculating the weighted average of the gradient grayscale of foreground and background by using Gaussian distribution and use this average value as the threshold. The threshold of the foreground is T_f, the threshold of background is T_b, and the weighted average value $T_{Kittler}$ calculated by combining T_f and T_b is obtained as follows:

$$T_{Kittler} = \theta T_b + (1 - \theta) T_f, \tag{2.8}$$

Figure 2.8: The result of setting threshold as 128.

Figure 2.9: The result of setting threshold as 210.

Figure 2.10: The result of setting threshold as 240.

Where θ and $1-\theta$ are the proportion of background pixels and foreground pixels in the entire image respectively.

As image preprocessing requires higher accuracy in computational precision and shorter computational time in computational complexity, more binarization algorithms were proposed [170, 171, 172]. In 1979, Otsu [173] proposed a classic Otsu binarization algorithm. The algorithm divides the image into two clusters: greater than the threshold value and less than the threshold value, namely the foreground and background. We can calculate the individual class variance of these two parts. The greater the variance between classes, the greater the direct gray difference between the two parts. After trying all the 256 thresholds from 0 to 255, we can find the maximum individual class variance V_{Otsu}, which is usually the best threshold value we are looking for [174, 175, 176, 1]. The equation is as follows:

$$V_{Otsu} = w_f w_b (\zeta_0 - \zeta_1)^2, \tag{2.9}$$

where, the proportions of foreground and background in the entire image are w_f and w_b.The average grayscales of w_f and w_b are ζ_0 and ζ_1 respectively [176]. Table 2.3 shows the comparison of the above-introduced methods in the global thresholding method.

Table 2.3: Global thresholding methods of binarization

Methods	F-Measure	PSNR	MPM	Pros	Cons
Fixed threshold method [167]	0.8126	13.9543	21.3411	The threshold of this method is set by user, which saves the time for calculating the threshold.	This method does not suitable for all the image because different thresholds will get different results.
Kittler method [169]	0.8929	15.3774	20.3543	This method uses the weighted average of the thresholds of the background and foreground to get the global threshold, which improves the accuracy.	The computation complexity is higher.
Otsu method[174]	0.8646	17.2342	19.5523	When the area difference between the target and the background is small, the image can get expected accuracy.	When the area difference between the target and the background is large, the performance of this method can not achieve the expectation.

Local thresholding method

When an image is not suitable for the global thresholding method, such as contains much noise or in the uneven illumination condition, we consider applying the local thresholding method for image binarization [177, 178, 179]. The local thresholding method is that we divide the image into several areas depending on different conditions, such as the lighting condition, and use different thresholds in different areas in the given image [180, 181]. There are four classic and wild-used algorithms in the local thresholding method, i.e., Niblack method, Sauvola method, Bernson method, and the adaptive methodcite [182, 183].

Niblack method. The Niblack method [184] calculates the binarization threshold $T_{Niblack}$ by the mean $m_{Niblack}$ and standard deviation $v_{Niblack}$ of the gray values of pixel points in a certain pixel and its neighborhood, which we name it window. It is significant to determine the size of the window. If the size of the window is too big, it will lose details of the image. If it is too small, it will be affected by the binarization noise in the image. The equation of Niblack algorithm is as follows:

$$T_{Niblack} = m_{Niblack} + k \cdot v_{Niblack}, \tag{2.10}$$

$$v_{Niblack} = \sqrt{\frac{\sum {P_i}^2}{NP_{Niblack}} - m_{Niblack}^2}, \tag{2.11}$$

where $NP_{Niblack}$ is the total number of pixels in the entire image, k equals to -0.2 which is defined by the author.

The advantage of Niblack algorithm is that it is flexible because the value of k and $v_{Niblack}$ determines the threshold $T_{Niblack}$, which can be adjusted according to the actual situation. However, when traversing the entire image, if the pixels in the window are all considered as background, Niblack algorithm will determine part of them as a target, which will produce binarization noise. In recent years, Niblack algorithm is widely used in image binarization, such as [185] and [186].

Sauvola's method. The Sauvola's method [187, 188] is similar to the Niblack algorithm, where the value of threshold $T_Sauvola$ is also determined by the mean $m_{Sauvola}$ and standard deviation $v_{Sauvola}$ of the given pixel and its neighborhood. The calculation method is different from the Niblack algorithm, which is as follows:

$$T_{Sauvola} = m_{Sauvola} \cdot [1 + k \cdot (\frac{v_{Sauvola}}{DR} - 1)], \tag{2.12}$$

where DR is the dynamic range of the standard deviation, which is always 128 in this method, and k is a custom correction parameter, which is in the range of [0,1].

In 2015, Najafi and Salehi [189] proposed a novel architecture for reducing the computing effort of Sauvola's algorithm by using a stochastic computing (SC) approach. The proposed architecture introduces a 9-to-1 and an 81-to-1 stochastic mean circuit (SMC), in order to get the average value of the input bitstream. After obtaining the average value, this architecture generates the standard deviation bitstream and threshold bitstream to get the threshold of the input image. The advantages of this architecture are fast and process in the low-power mode, which saves a lot of time and energy. However, it can only use a big window size because of the technical question, which may lose some details of the image [190].

In [191], the author proposed a novel method that combines Sauvola's algorithm and Chan-Vese active contour model. In the proposed method, Sauvola's algorithm is used to detect the High-Probability Text Pixels and introduce this result to the Chan-Vese model. After getting the initial result, the Chan-Vese model will detect all the targets in the image, even some of them that are left out in Sauvola's algorithm. The proposed model can detect both the large target and small target by using the same size of the window and maintain high accuracy. The disadvantage of it is that this method takes a longer time than the classic Sauvola's algorithm because of the Chan-Vese model and the advantage of high precision can only be realized when the contrast between foreground and background is high.

In order to improve the accuracy of binarization, [192] was proposed by using the WAN method to improve Sauvola's algorithm to obtain a more accurate threshold value. When the contrast of the given image is low, the classic Sauvola's algorithm gets a low threshold value, which results in the non-ideal binarization result. WAN method is proposed to solve this question. In the classic Sauvola's algorithm, we find that the mean value m greatly affects the threshold result. Based on this reason, the WAN method uses m_{WAN}, which is the average of the highest intensity of the entire image $max(x, y)$ and the original value of classic Sauvola's algorithm m to get a better binarization result in the low-contrast image. By increasing the value of mean m_{WAN}, the WAN method can greatly reduce the probability of losing details in the low-contrast image, and maintain the high accuracy of detection of the objects in the image. However, with the value of mean increases, this method can also produce some binarization noise. The equation of the proposed method is as follows:

$$T_{WAN} = \frac{max(x,y) + m_{WAN}}{2} \cdot [1 + k \cdot (\frac{v_{WAN}}{R} - 1)]. \tag{2.13}$$

Bernson method. Bernson method [193] is based on the image segmentation technique. Given a pixel in the image (x, y), the gray value of it is $GB(x, y)$. Consider a sliding window, which the center is the given pixel, and the size of it is $(2w + 1)^2$. The threshold value of it can be calculated by the following equation:

$$T_{Bernson} = \frac{maxGB(x + a, y + b) + minGB(x + a, y + b)}{2}, \tag{2.14}$$

where a and b are both in the range of $[-w, w]$.

In [194], an improved Bernson binarization method was applied. Instead of using the average of the maximum and the minimum gray values, the improved method uses the weighted average of them. The size of the window is $(2w + 1)^2$, and the equation is as follows:

$$T_{Bernson1} = \lambda_1 maxGB(x + k, y + k) + \lambda_2 minGB(x + k_1, y + k_1), \tag{2.15}$$

where k_1 is in the range of $[-w, w]$, $\lambda_1 > \lambda_2 > 0$, and the sum of λ_1 and λ_2 is equal to 1. In this paper, we set λ_1 is 0.7 and λ_2 is 0.3. This enhanced Bernson method is more effective than the classic Bernson method, which spends less time and keeps the similar accuracy as others.

In 2017, EYUPOGLU [2] enhanced the Bernson method by considering both of the gray values and the contrast in the custom sliding window. This algorithm defined the contrast limit l and the size of the neighborhood r^2 by user first. The contrast $C(x, y)$ is calculated by the following equation:

$$C(x, y) = maxGB(x + a, y + b) - minGB(x + a, y + b). \tag{2.16}$$

If $C(x, y) < l$, we can consider that this window contains only one foreground and background. By applying the value of contrast $C(x, y)$, this method can divide the foreground and background directly, which reduces the useless information and keep most of the details in the entire image.

In [3], Yang and Feng combine the Otsu method and Bernson method to obtain the binarization result. By combining these two methods ensures that the image binarization process retains the advantages of both methods while avoiding their disadvantages for improving accuracy. However, this also takes a much longer time than each of the methods. The procedure of this combined binarization is summarized as the following five steps:

1. Get the global threshold T_{Global} by using Otsu method;

2. Set Z to 15, and $T_1 = \frac{T_{Global}+128}{2}$. T_2 is the average value of $maxGB(x,y)$ and $minGB(x,y)$, which is the value of the classic Bernson method;

3. Calculate the average value $A_{Bernson}$ and the mean-square error ME of the pixels in the window by using the following equation:

$$A_{Bernson} = \frac{\sum GB(x,y)}{(2w+1)^2}, \tag{2.17}$$

$$ME = \frac{GB(x,y) - A_{Bernson}}{2w+1}, \tag{2.18}$$

where $(2w+1)^2$ is the size of the window;

4. Set $T' = \pi_1 T_{Global} + \pi_2 T_2$, where $\pi_1 \geq 0, \pi_2 \geq 0$, and $\pi_1 + \pi_2 = 1$;

5. Compare Z and ME. When $ME \geq Z$, if $T_{Global} \geq T'$, set $GB(x,y)$ to 255. Otherwise, set $GB(x,y)$ to 0. When $Z \geq ME$, if $T_2 \geq T_1$, set $GB(x,y)$ to 255. Otherwise, set it to 0.

Table 2.4 shows the comparison of each method in the local thresholding method, and Fig. 2.11, 2.12, 2.13 and 2.14 show the original road image, the results of the above two types of binarization methods, and the result of the combination of both two methods.

Figure 2.11: The original image with lane.

Figure 2.12: The result of global binarization method [1].

Figure 2.13: The result of local binarization method [2].

Figure 2.14: The result of the combined binarization method [3].

Table 2.4: Local thresholding methods of binarization

Categories	Pros	Cons	Methods	F-Measure	PSNR	MPM
Niblack's method	The threshold is flexible because it depends on the thresholding formula.	The accuracy is lower when the detected area is background.	Khurshid et al. [184]	0.8831	15.2341	22.4554
Sauvola's method	This method improves the performance of Niblack method in the uneven light condition of the image.	When window size of this method can not reach a suitable size in binarization, it may lose some details of the image.	Sauvola et al. [187] Najafi et al. [189] Hadjadj et al. [191]	0.8902 0.9131 0.9275	16.3476 19.8324 17.2342	9.2342 10.1284 9.4574
Bernson method	This method can find the threshold depends on the pixel distrubution of the grayscale image, which is more flexible and accurate.	The processing time of this method depends on the image size. When the size is too large, it may take too much time to obtain the result.	Talab et al. [194] EYUPOGLU et al. [2]	0.9196 0.9123	16.9239 20.1259	12.3454 11.9837

2.2.3 ROI extraction

When detecting some of the significant objects in the image, it is unnecessary to detect the whole given image [195, 196]. Based on this reason, we need to consider extracting the ROI. ROI is the area in the image, which only contains the objects that needed to be detected. For instance, in the lane detection area, given an image of a car driving on the road, the significant objects are the lane lines in front of the car. Sky, buildings, trees, and other objects have no way to help with lane detection but may cause the non-ideal detection result. Extraction ROI can reduce the useless part in the image and keep the relevant part for detection, which saves much time and increase the detection accuracy. There are two ways to extract ROI area, i.e., fixed ROI extraction and dynamic ROI extraction.

Fixed ROI extraction

The fixed ROI extraction method ensures a fixed shape, such as a rectangle, and applies it to the image. In general, the shape is selected almost based on the position of lane lines in most of the images. As most of the images are based on the driver's perspective, the bottom of the ROI is also the bottom of the image.

In [5], Deng and Wu proposed a fixed ROI method. This method divides the image into three parts, i.e., near field, far vision field, and sky area. Among them, the sky area is 5/12 in the given image and the sum of the near field part and far vision field part accounts for 7/12. According to the method introduced in [5], only the sky area does not contain the lane lines, which can be ignored, and the authors define that the far vision field part accounts for half of the near part.

Wu et al. [4] proposed another fixed ROI method. This method determines the size of the area S_1which contains the lane lines by using the proportion of width W and height H of the input image. Then this method determines a certain area S_2which is similar to S_1. For instance, in Fig. 2.15, 2.16 and 2.17, S_1 equals to $0.39W \times 0.29H$, so the ROI is determined to be $0.5W \times 0.2H$.

Figure 2.15: The original image with lane.

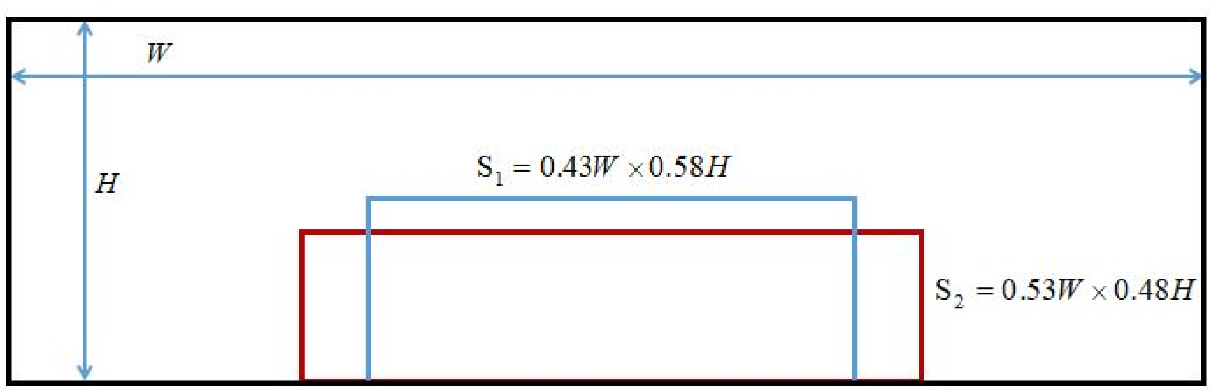

Figure 2.16: Determin ROI by the ratio of W and H.

This method [10] extracts the fixed ROI automatically, which approximately spends no time on it, this is the advantage of it. However, this method will lead to an inaccurate ROI region setting and can not be used to most of the images, which makes the subsequent

Figure 2.17: The result of ROI [4].

detection waste more time and the accuracy is low. In order to avoid these disadvantages, the dynamic ROI method begins to be widely used. Different from the fixed ROI method, dynamic ROI determines the ROI area based on the different input images.

Dynamic ROI extraction

The dynamic ROI extraction method is to obtain the area based on the position of lane lines. In general, the shape of ROI is the quadrilateral formed by the four starting and ending points of the lane lines in front of the driver.

Huo et al. [6] proposed a dynamic vanishing point adjustment method to determine ROI. When the lane lines are straight lines, we define the intersection of the two-lane lines extending out according to the image as the vanishing point. When the lane lines are curved, the vanishing point is the intersection of the two-lane lines according to the tangents in the given figure that extends out. The bottom of the image and the vanishing point form a triangle. ROI in [6] is defined as the area where the triangle overlaps the given image.

In [7], Rathnayake and Ranathunga applied a trapezoid as ROI, which the bottom of it is the bottom of the image. Depending on the angle of view, the value of the distance between both of the lane lines in front of the car becomes smaller. The two peak points of the lane lines and the bottom of the image form a shape of a trapezoid. Fig. 2.18, 2.19, 2.20and 2.21 show the comparison of the original image and the results of ROI extraction by using the methods of [5], [6] and [7] respectively, and Table 2.5 shows the strengths and limits of the fixed ROI extraction and dynamic ROI extraction. The advantages of the dynamic ROI method are as following:

- It can be used to different images with different positions and angles of lane lines;
- It determines the dynamic range of ROI which works for different images;
- It reduces the useless information and keeps most of the details of the image accurately;
- It saves much time for the next detection step.

Figure 2.18: The original image with lane.

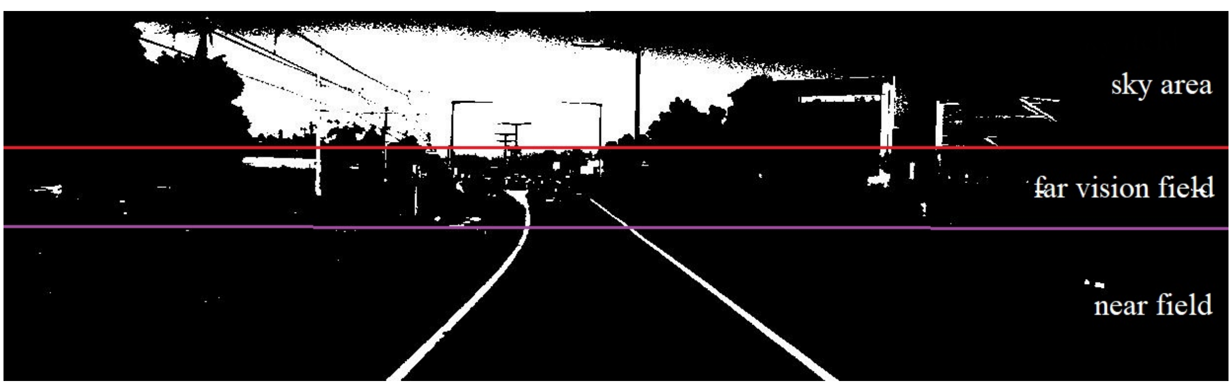

Figure 2.19: The result of ROI extraction in [5].

2.2.4 Inverse perspective mapping

Inverse perspective transformation (IPM) has been widely used in computer vision and road traffic sign detection and recognition in recent years. IPM is the mapping process from the 2D coordinate system of the image to the 3D coordinate system of the real world. The benefits of IPM are that it eliminates the interference and error of the influence of

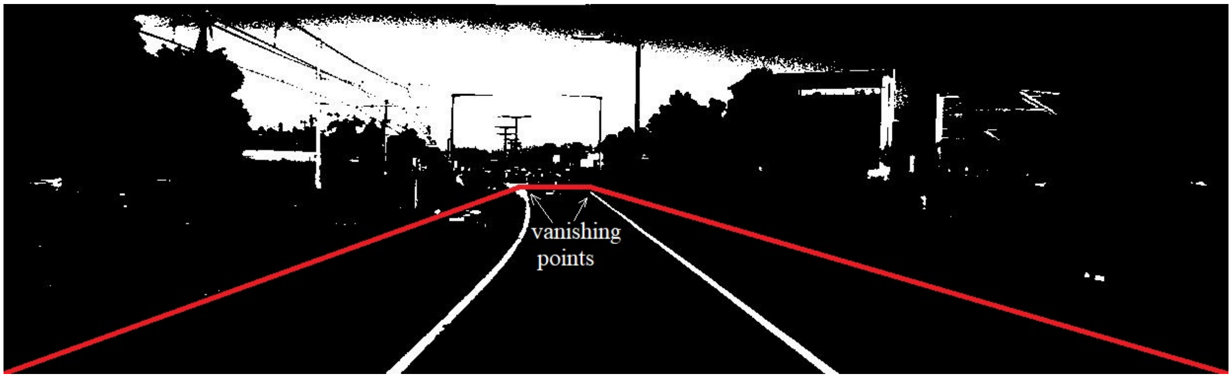

Figure 2.20: The result of ROI extraction in [6].

Figure 2.21: The result of ROI extraction in [7].

Table 2.5: Method of ROI extraction

Category	**Methods**	**Pros**	**Cons**
fixed ROI extraction	Deng and Wu [5] Wu et al. [4] Gamal et al. [10]	Easy to ensure the area of detected objects.	The accuracy is low and the method is not flexible.
Dynamic ROI extraction	Huo et al. [6] Rathnayake and Ranathunga [7]	The position of detected area is flexible, and the accuracy is high.	It takes some calculation, which may reduce the time efficiency.

perspective on image detection and recognition tasks, and helps to correct the distortion of the original image by converting the image of the driver's view of the road to the top-view image.

The method of applicating IPM is introduced in [197, 198, 199]. Suppose there exists a point (F_x, F_y, F_z), the coordinates in the world;s coordinate system is $(F_x, F_y, F_z)_{world}$ and the coordinates in the camera's coordinate system is $(F_x, F_y, F_z)_{camera}$. The relationship between them can be represented by the following equation and Fig. 2.22:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ 1 \end{bmatrix}_{camera} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \\ 1 \end{bmatrix}_{world}, \tag{2.19}$$

where R is a rotation matrix and T is a translation vector.

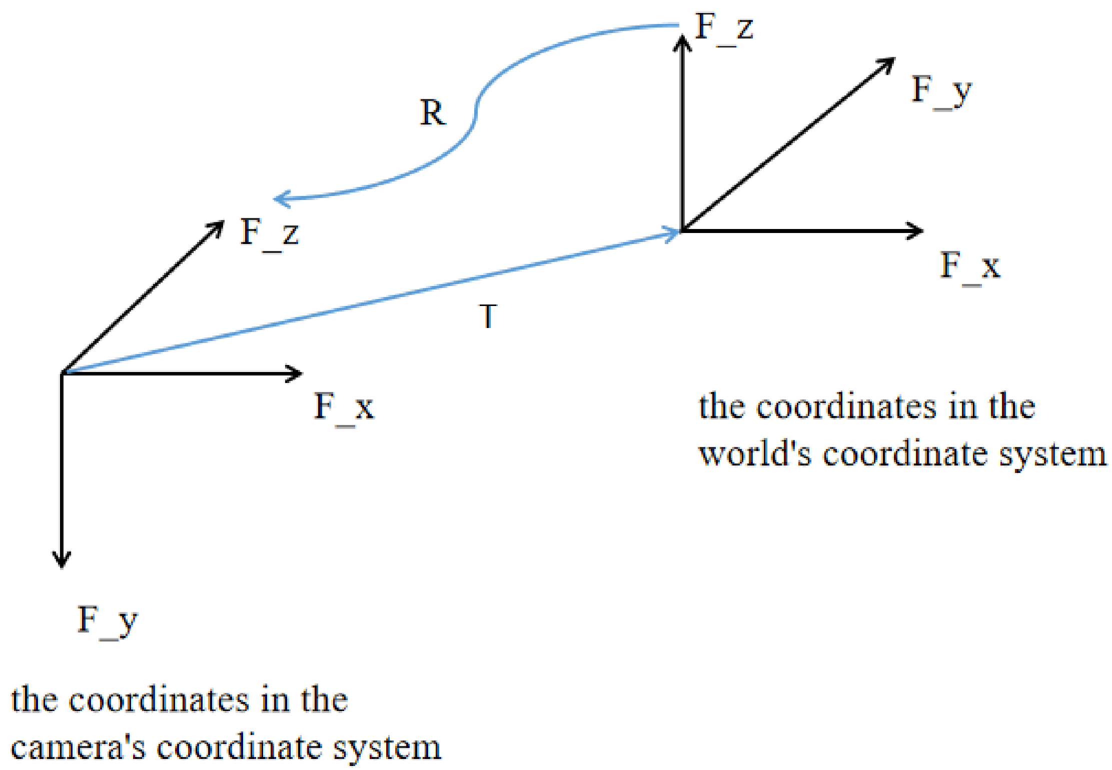

Figure 2.22: The relationship between the coordinates in the world's coordinate system and the coordinates in the camera's coordinate system

The point $F(F_x, F_y, F_z)$, which is in 3D space, can be represented in 2D image as f by the following equation:

$$f = C \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \\ 1 \end{bmatrix}, \tag{2.20}$$

where $C = \begin{bmatrix} 1/dx & 0 & x_0 \\ 0 & 1/dy & y_0 \\ 0 & 0 & 1 \end{bmatrix}$, dx and dy are the physical dimensions of each pixel in the X-axis and Y-axis, and (x_0, y_0) is the origin of a coordinate system in pixels.

2.3 Lane detection stage

Feature extraction is a significant part of the computer-vision-based lane detection technique [200, 201]. Lane detection is a process of detecting and extracting the lane lines in the ROI by using different feature extraction methods. Feature extraction is used to process the original image to extract the features in the given image which can improve the accuracy of the lane detection stage. As it focuses on rough handling of large amounts of the image which may contain a lot of jumbled information, the computational complexity is particularly high, and the computation time is long. There are three classic methods for extracting features in original images, i.e., color-based methods, texture-based methods, and edge-based methods.

2.3.1 Feature extraction based on colors

The feature of color is a global feature, which describes only the surface color properties of the objects in the image. As color-based feature extraction is easy to be affected by lighting and it is not sensitive to the size and the change of the objects in the image, this is a fatal weakness for lane detection [202, 203]. For instance, it may lose some important and necessary details in the night environment or heavy traffic when the road is full of vehicles. Based on this reason, the color-based feature extraction method is not widely used in the lane detection stage. Fig. 2.23 and 2.24 show the original image and the color histogram of it.

Figure 2.23: The original image with lane.

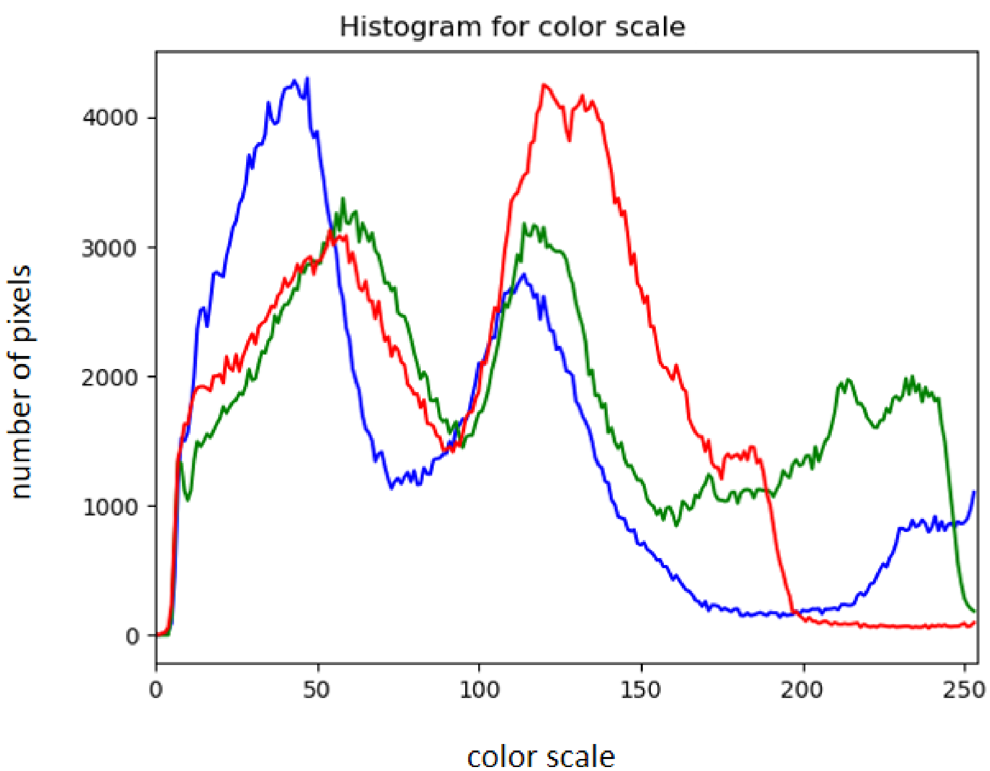

Figure 2.24: The result of global mapping method.

Shinde et al. [204] divided the entire image into 16 blocks with the same size of $w\dot{h}$. After that, they used the color-based feature extraction method by calculating the mean value of R, G, B respectively for each component in the block. At last, they calculate the average value of color components to extract the color features.

Khamisan et al. [205] used the product of masking image and the original thermal image to obtain the color information of the RBG space model which is implemented color-based feature extraction method. This multiplication will get the different intensity of color information.

2.3.2 Feature extraction based on textures

The feature of texture is also a kind of global feature, which also describes the surface properties of the objects in the image as color-based feature extraction [206, 207, 208, 209, 210]. However, different from feature extraction based on color, texture-based feature extraction does not focus on the pixel's property, it needs statistical calculations in areas that contain multiple pixels [211, 212, 213]. In the texture-based feature extraction method, the gray-level co-occurrence matrix approach (GLCM) is one of the most well-known approaches [214, 215]. GLCM is an approach that calculates the symbiosis matrix by calculating the gray image, and then the partial eigenvalues of the matrix are obtained by calculating the symbiosis matrix to represent the texture features of the image respectively [216, 217]. There are four steps in GLCM:

- Extract gray-image;
- Grayscale quantization;
- Calculate the parameter selection of the characteristic value;
- Calculate the texture feature value and generate texture feature image.

The advantage of it is that it can not be affected by the deviation of local details, and it can resist the noise in the image. Therefore, it is impossible to obtain high-level image content by using the texture-based feature extraction method. However, one of the disadvantages of texture-based feature extraction is that when the image is affected by the lighting or the watermarks on the road after rain in the lane detection stage, the result it reflects is different from the original image, which will reduce the accuracy in the field of detection [218, 219].

2.3.3 Feature extraction based on edges

Edge is a significant structural feature in the image, which often exists between the detection target and the background [220]. In an image, the luminance significant changes in background and foreground cause the edge of the image. Feature extraction based on edges aims to detect the pixel points in the image, and the value of the pixels around them changes sharply, which can greatly reduce the irrelevant information and maintain the structure of the image. We believe that different objects in the image lead to this change. Therefore, these pixel points can be taken as a set and used to mark the boundary of different objects in the image. In general, there are three steps in edge extraction:

1. Smoothing: Based on keeping the true edges, eliminate the noise as much as possible;
2. Enhancement: Make the boundary contour more obvious, which is easy to be detected;
3. Detection: Select the useful edge points for lane detection and discard the useless edge points which are considered as noise.

In this section, we will introduce three classic methods of feature extraction based on edges, i.e., Sobel edge detection, Canny edge detection, and Hough transform.

Sobel. In lane detection stage, Sobel operator uses two 3 x 3 matrices to calculate the approximate derivatives of two directions, i.e. vertical and horizontal, by convoluting the original image. We assume that the original image is IM, the approximate derivatives are I_x and I_y. The formula is as follows [221]:

$$I_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times IM, \tag{2.21}$$

$$I_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \times IM, \tag{2.22}$$

where $\times$ indicates the convolution operation.

For each point in the image, the result I and θ can be obtained by combining I_x and I_y by using the following formula:

$$I = \sqrt{I_x^2 + I_y^2}, \tag{2.23}$$

$$\theta = arctan(\frac{I_y}{I_x}). \tag{2.24}$$

In [222], the authors used the Sobel operator to remove the low-contrast part of the image, which improves the contrast of the detected object in ROI and the background. This operation makes the edge of the detected objects much clear, shortens the processing time of detecting the lane lines, and improves the accuracy of the lane detection stage.

In [223], Liu et al. applied different Sobel operator thresholds to the same IPM image and compare these images to find which threshold makes the lane lines in the detected image more clear and less noise. In different images, the suitable Sobel operator threshold is different. Even if the processing speed of Sobel edge detection is fast, the selection of threshold is limited, which may miss the detection in the image and cause the low-accuracy results [224]. Fig. 2.25, 2.26, 2.27 and 2.28 show the original image, the edge detection results in x and y direction, and the combined edge detection result. The advantages of Sobel edge detection are as follows:

- This method performs well in the image which has much noise;
- The processing speed is fast.

However, the accuracy of edge detection is non-ideal, because it uses only one threshold to determine the edge, which may misjudge the noise as the edge.

Figure 2.25: The original image with lane.

Canny. In general, we use the Canny edge detection technique to obtain the gradient of the image. In 1986, John F. Canny developed this method. There are four steps in Canny edge detection:

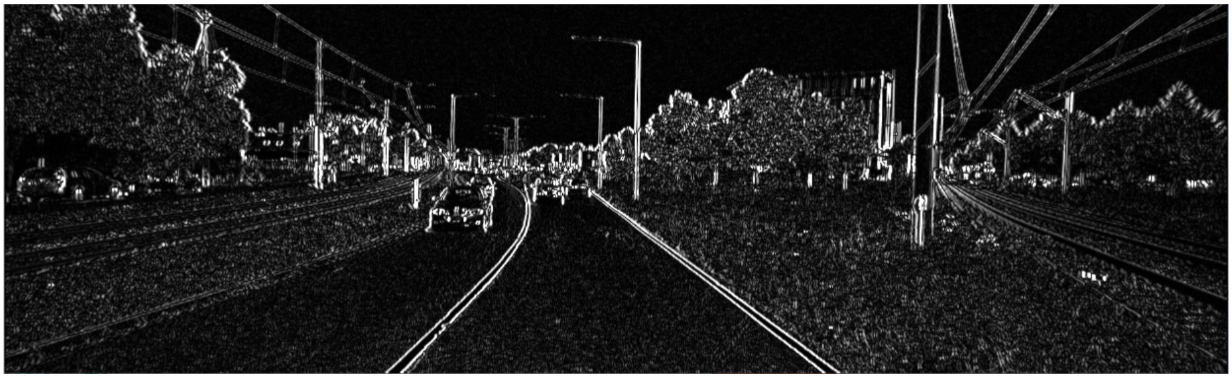

Figure 2.26: The result of Sobel edge detection in x direction.

Figure 2.27: The result of Sobel edge detection in y direction.

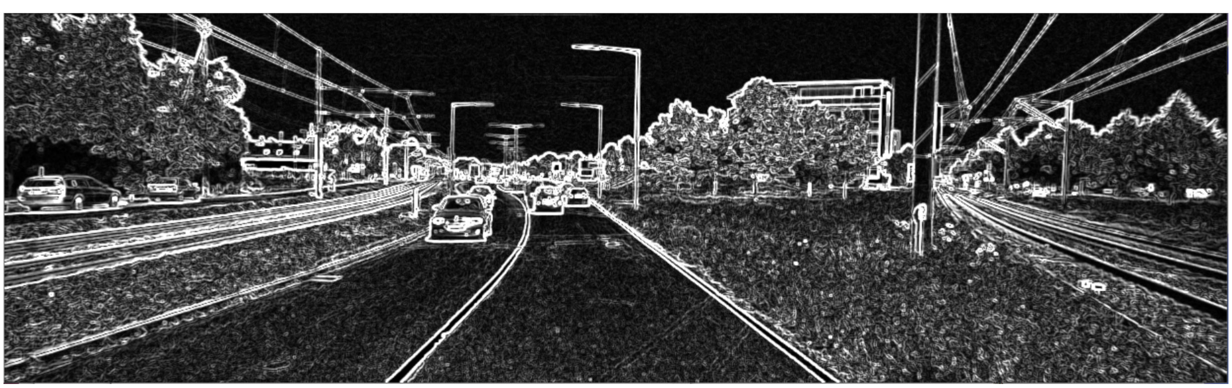

Figure 2.28: The result of Sobel edge detection.

1. Apply Gaussian filter for noise reduction;
2. Calculate the gradient value and gradient direction;
3. Non-maximum suppression;
4. Use high and low thresholds to detect edges.

In Canny edge detection, high and low thresholds need to combine to obtain the ideal edge extraction result. The high threshold is used to distinguish the object, whose edge will be extracted from the background. This operation decides the contrast between the object and the background. When the value of the high threshold is set too high, which causes the edge of the objects is not smooth enough, or discontinuous, a low threshold helps smooth the edge and cluster the discontinuous part.

In [5], the authors detected the lane lines by using Canny edge detection, in which the ratio of high thresholds and the low threshold is 1:2, which helps reduce the noise and improve the accuracy of the edge detection result. In [19], Bilal et al. simplified the Canny edge detection by using only one threshold [225]. This method extracts the edges from the grayscale image and uses the fixed threshold to remove the edges, which are shorter than the set threshold. After that, the simplified Canny edge detection method connects the disconnected segments into a complete lane line. This method is easy to realize, however, the threshold is difficult to be determined, which is easy to lose the image details.

The advantage of Canny edge detection is that this method combines high threshold and low threshold, which helps enhance the lane lines in the image. Besides that, this method also greatly reduces the probability of mistaking noise for edges, which improves the accuracy of lane detection. However, as Canny edge detection uses the fixed thresholds to extract the edge, this does not apply to all the images. The future work of it is to improve the method for setting the adaptive thresholds, which does not take too much processing time and maintain the same or even higher accuracy. Fig. 2.29 and 2.30 shows the result of Canny edge detection [19].

Hough transform

The aim of Hough transform [226, 5] is to detect the boundaries of the lane in the road image in LDWS, which includes straight lines and curves.

Figure 2.29: The original image with lane.

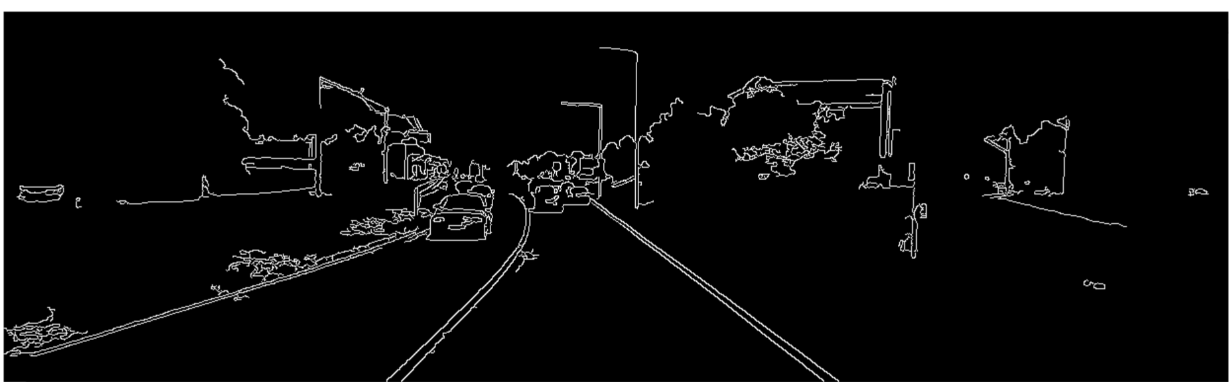

Figure 2.30: The result of Canny edge detection.

Generally, for each point in the coordinate system, it corresponds to a line in the polar coordinate system. When many points in the road image form one lane line, in the polar coordinate, the lines they correspond to certainly will intersect at one point, which indicates the detected lane line. Thus, the number of points of intersection equals the number of the lane lines in the image. Based on the transformation of coordinate, Hough transform is the problem of calculating the peak in the polar coordinate system instead of detecting the pixels of the original road image. Fig. 2.31 shows the process of the coordinate transform.

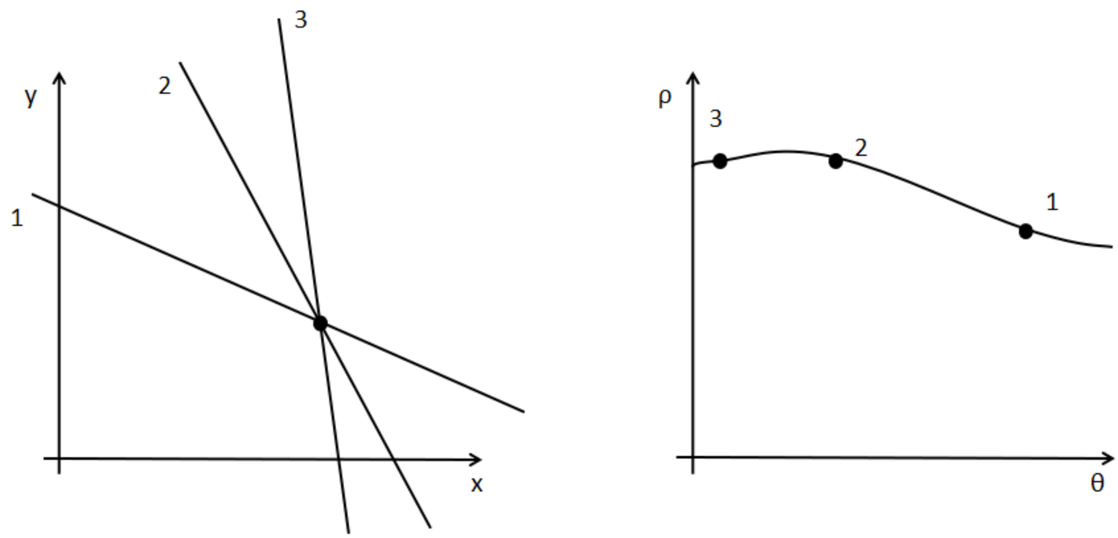

Figure 2.31: The process of the coordinate transformation.

In [227], the author proposed an improved Hough transform. In the classical Hough transform method, we determine the left and right lane line by the distribution of the lane line pixels. In this paper, the novel method limits the voting space for Hough transform, which improves the processing speed and accuracy of lane detection. In order to improve the performance of the Hough transform, several conditional constraints are applied to the Hough transform [228]. By setting the thresholds of left and right lanes and remove the lines adjacent to the lane lines by the distance and angle, the proposed Hough transform can detect the lane lines in real-time and improve the accuracy of detection.

As in most cases of the lane detection image, the lane lines are perpendicular to the bottom of the photo or have a small angle with the vertical line at the bottom of the photo. Based on this reason, in [8], Liu et al. constrained the angle θ in the polar coordinate as follows:

$$\theta \in [75°, 90°]. \tag{2.25}$$

After adding these constraint conditions, Hough transform only needs to detect the lane line in a small range, which shortens the processing time and removes the edges which are easily be identified as lane lines. Table 2.6 shows the comparison of feature extraction and Fig. 2.32 and 2.33 show the original image and the result of the Hough transform.

Figure 2.32: The original image with lane.

Figure 2.33: The result of Hough transform [8].

2.4 Lane departure recognition

When driving on the road, the vehicle keeps at a distance from both lane lines. Car accidents may occur when two vehicles driving in adjacent lanes approach the same lane line. In order to avoid this situation happening, LDWS has been developed in recent years. When the vehicle gets close to the boundary of the lane and the distance between

Table 2.6: Methods of feature extraction

Category	Pros/Cons	Methods	Accuracy(%)
Feature extraction based on colors	**Pros:** Color is the surface feature, which is easy to recognition. **Cons:** It is easy to be affected by lighting and the size of objects.	Li et al. [202] Shinde et al. [204] Khamisan et al. [205]	94.23 94.45 95.13
Feature extraction based on textures	**Pros:** This method can not be affected by the deviation of local details, and it can resist the noise in the image. **Cons:** The extraction results can be affected by lighting or other external factor.	Bagri et al. [206] Wei et al. [207] Vidyarthi et al. [208]	95.44 96.51 96.87
Feature extraction based on edges	**Pros:** Hough Transfer can be slightly affected by external factor, such as curve. **Cons:** The processing time is long because it needs to traverse all the pixels in the image.	Elsawy et al. [226] Deng et al. [5] Wei et al. [227]	97.46 98.14 98.23

the vehicle and the lane line is less than a set value, LDWS warns the driver back to the middle area of the lane. In general, LDWS can be categorized into two categories based on the position of the vanishing point and the origin point [229].

2.4.1 Land departure recognition based on vanishing point

In most of the situations in LDWS, the vanishing point is used to recognize lane departure. We use the vanishing point V_1 in the current image to compare the vanishing point V_2 in the previous frame, in order to obtain the horizontal displacement HD, which is used to determine that if this vehicle needs to be warned by LDWS. Fig. shows the basic method of LDWS based on a vanishing point. This method is flexible and high-accuracy as detecting the position of vanishing points in each frame. However, it also costs too much time for detecting, which is the drawback of this method. Therefore, how to figure out the positions of the vanishing points is the most important part of this method.

In order to improve the efficiency of LDWS based on the vanishing point, An et al. [230] recognized lane departure by using the improved vanishing point-based parallel Hough transform. In order to determine the positions of the vanishing points, the improved parallel Hough transform algorithm is applied. The proposed parallel Hough transform divides the angle θ into several pieces, each piece runs Houth transform in parallel. After getting the positions of the vanishing point, this paper calculates the horizontal displacement of each adjacent frame and determines if the warning is needed. This operator helps reduce the computation complexity of detection and only needs to detect the vanishing point once, which saves the detected time and improves the efficiency of lane departure recognition.

In [231], a robust method of estimating the position of the vanishing point is proposed. This method is summarized as follows:

- Extract the lines in the image from the background;
- Create the probabilistic voting function depends on the relevance of the extracted line segments;
- Obtain the position of the vanishing points from the function.

This method can be applied to images with a particularly noisy or complex background, which improves the efficiency and accuracy of detecting the position of the vanishing point.

In the adjacent frames, ensuring the vanishing point precisely helps improve the accuracy of the horizontal displacement, which improves the accuracy of LDWS. Fig. 2.34, 2.35, and 2.36 show the vanishing points in the adjacent frames and the horizontal displacement of these two vanishing points.

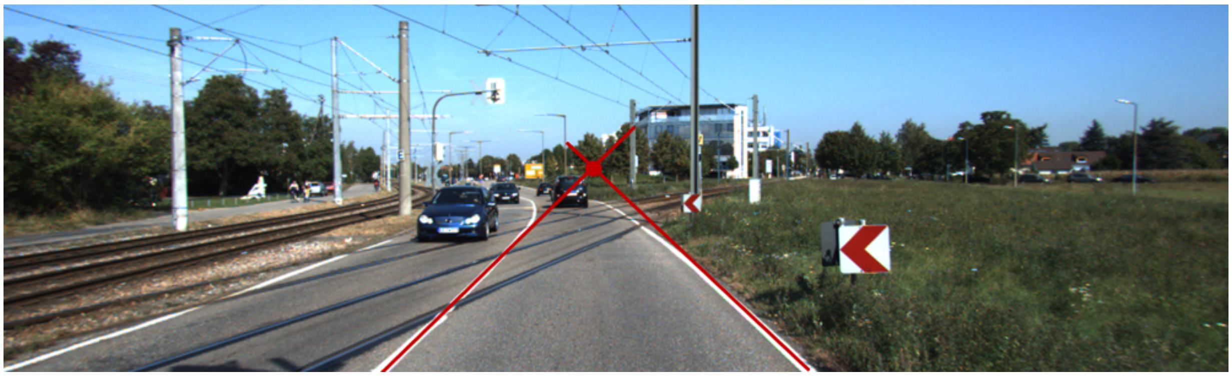

Figure 2.34: The vanishing point in the current frame.

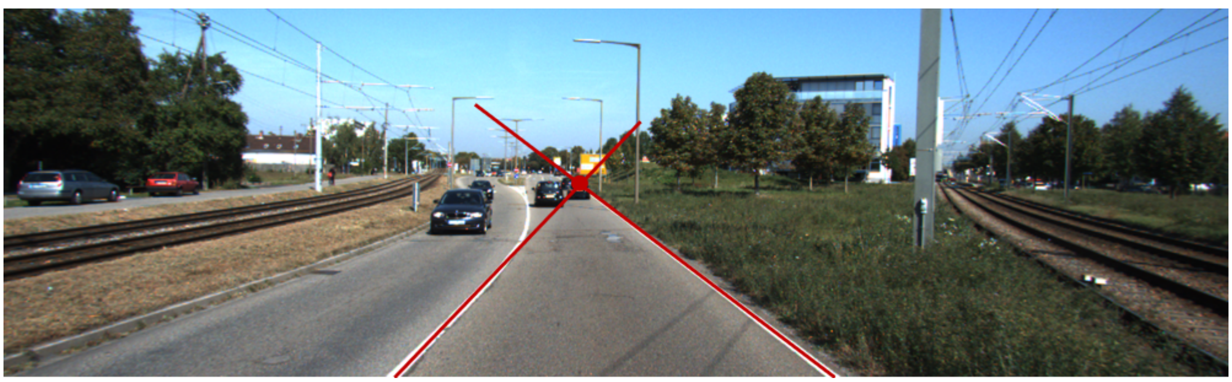

Figure 2.35: The vanishing point in the next frame.

2.4.2 Land departure recognition based on the position of origin

In general, the image is based on the driver's perspective, the position of the beginning of the lane lines is at the bottom of the image. This method sets three points in the image: left point, midpoint, and right point. The left point and right point indicate the two points where the horizontal straight line where the vehicle at, intersects the left and right lane lines, and the midpoint indicates the position of the vehicle, which is shown in Fig. 2.37. The equation of calculating offset is shown below:

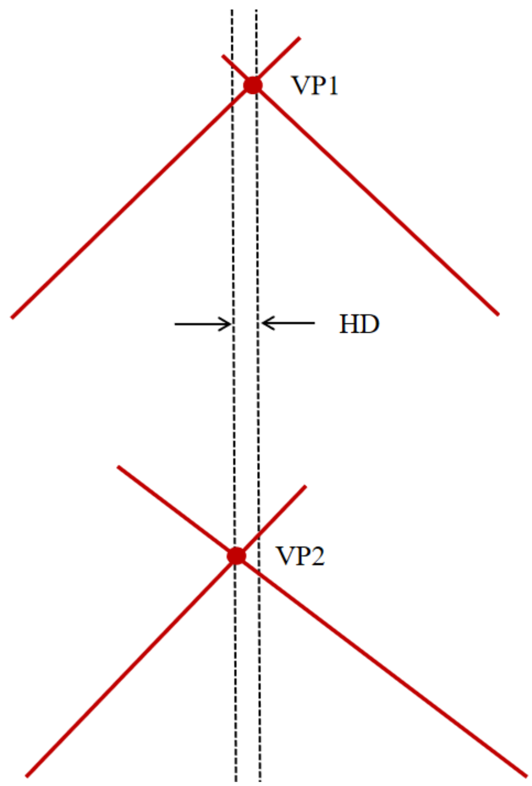

Figure 2.36: The horizontal displacement of the two vanishing points.

$$offset = \begin{cases} \frac{l}{d\times LW} - \frac{LW}{2}, & l \geq h, \\ \\ \frac{LW}{2} - \frac{h}{d\times LW}, & otherwise, \end{cases} \tag{2.26}$$

where LW is the width of the lane in the real world.

In [9], the author set the driver's perspective as the origin O, $MidL$ and $MidR$ are the midpoints on the left lane line and right lane line respectively, L' and R' indicate the distances between the O and $MidL$ and $MidR$, M' is the distance between $MidL$ and $MidR$, which is shown in Fig. 2.38. The value of L' and R' can be calculated as follows:

$$L' = \sqrt{MidL_x^2 + MidL_y^2}, \tag{2.27}$$

$$R' = \sqrt{MidR_x^2 + MidR_y^2}. \tag{2.28}$$

After obtaining the above two parameters, we can use them to ensure the difference D' determines by L' and R' with the following equations in order to determine that if the

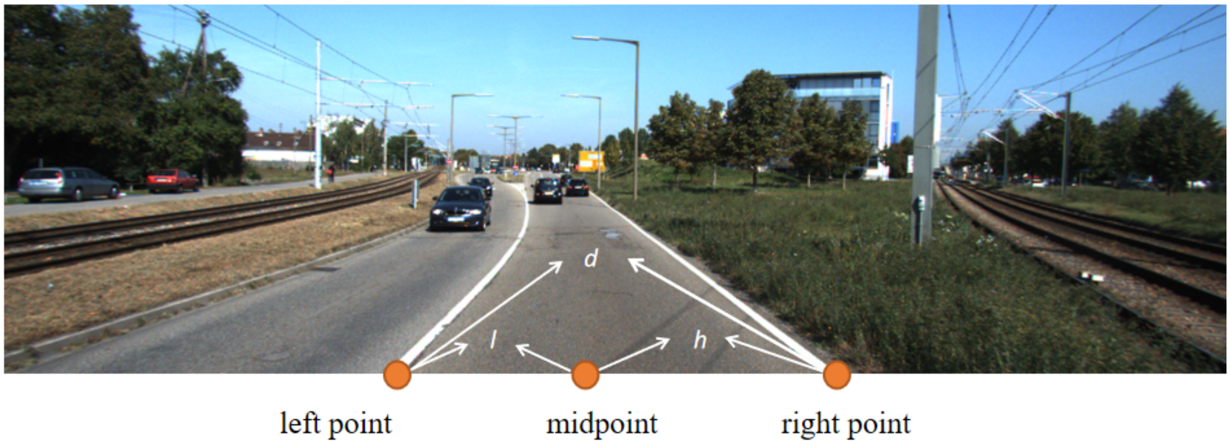

Figure 2.37: The left point, midpoint, right point and the distances between each of these points.

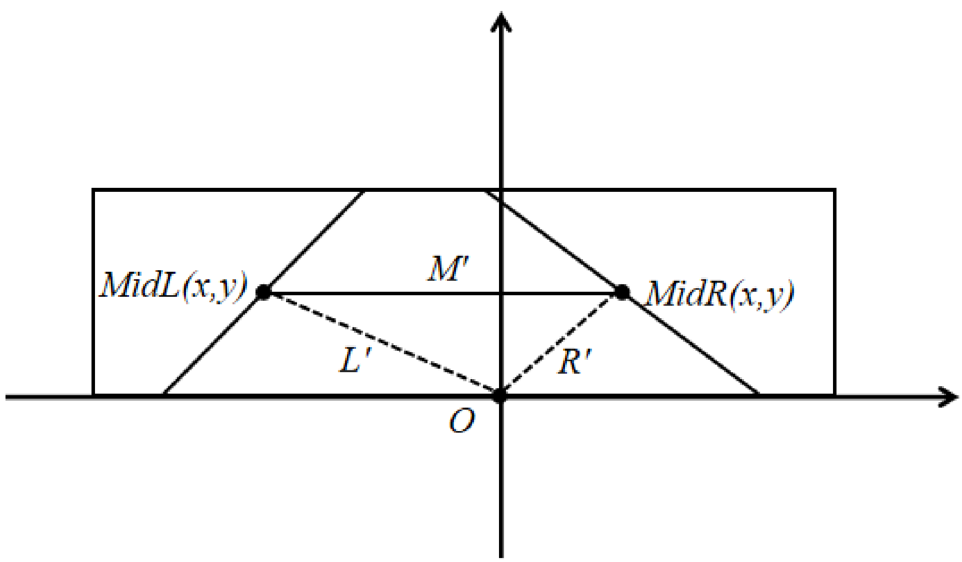

Figure 2.38: The coordinate in [9].

vehicle needs to be warned. Fig. 2.39 shows the identification rules, where threshold 1, threshold 2 are usually set as -20 and 20 respectively.

$$D' = \sqrt{||L'||^2 + ||R'||^2 - 2 \times ||L' \cdot R'||}. \tag{2.29}$$

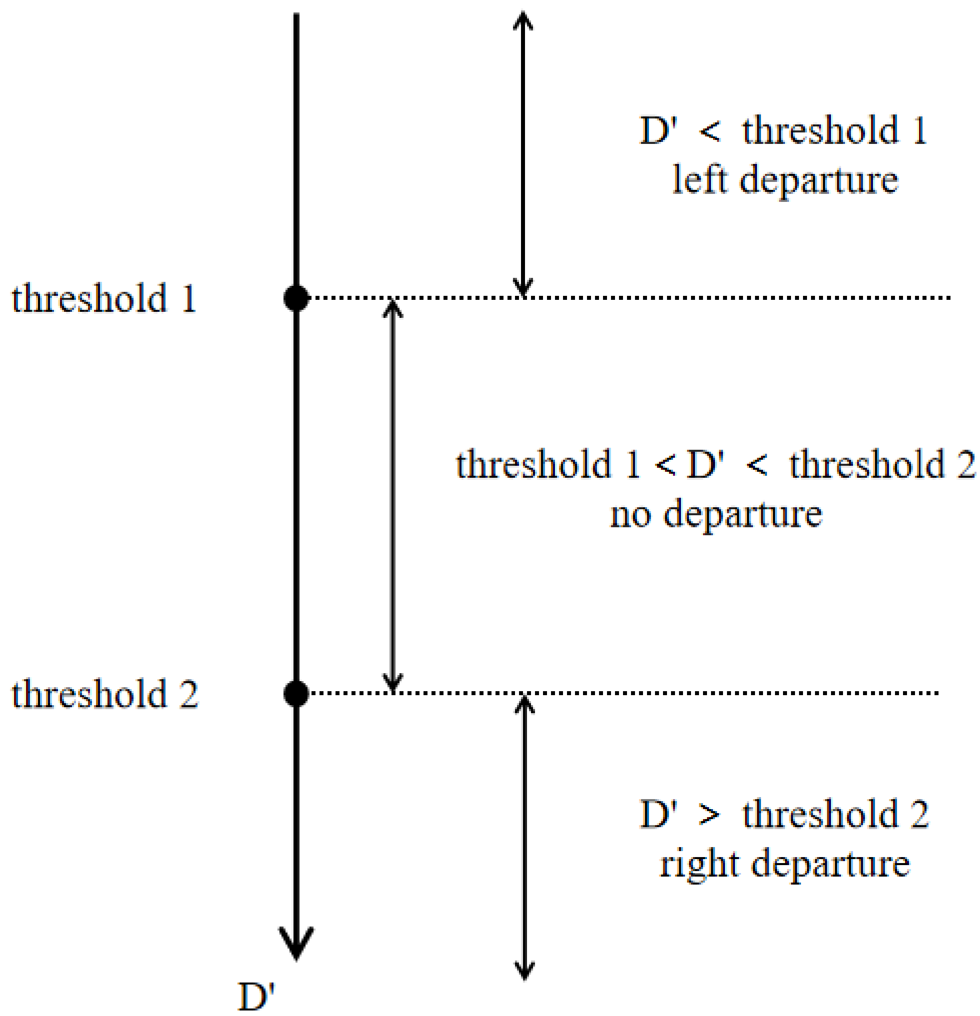

Figure 2.39: The method to detemine the departure direction in [9].

Chapter 3

The proposed work

In this section, we will introduce LEHA in detail. In our work, to improve the performance of LDWS, we propose a novel LDWS model LEHA which consisting of the image preprocessing module, lane detection module, and lane departure recognition module.

3.1 General procedure

The general procedure of our newly-designed LDWS model LEHA is shown in Fig. 3.1. At first, the original road image should be preprocessed by grayscale processing, binarization, and image smoothing sequentially. Among them, grayscale processing is the pre-step of binarization, and binarization aims to extract ROI from the background. Image smoothing focuses on removing the useless and irrelevant information of this image, such as noise. Then in the lane detection stage, the function of ROI is used to extract the part of the image which contains lane lines. After that, we apply edge detection to extract the edges of the lanes in ROI. The final step of lane detection is Hough transform, which aims to detect and mark the lane lines in the image. Finally, in lane departure recognition, we measure the distances between the vehicle and the lane lines on both sides of it and calculate the deviation distance between this vehicle and the centerline of the lane. Based on the result, LEHA determines if the warning is needed.

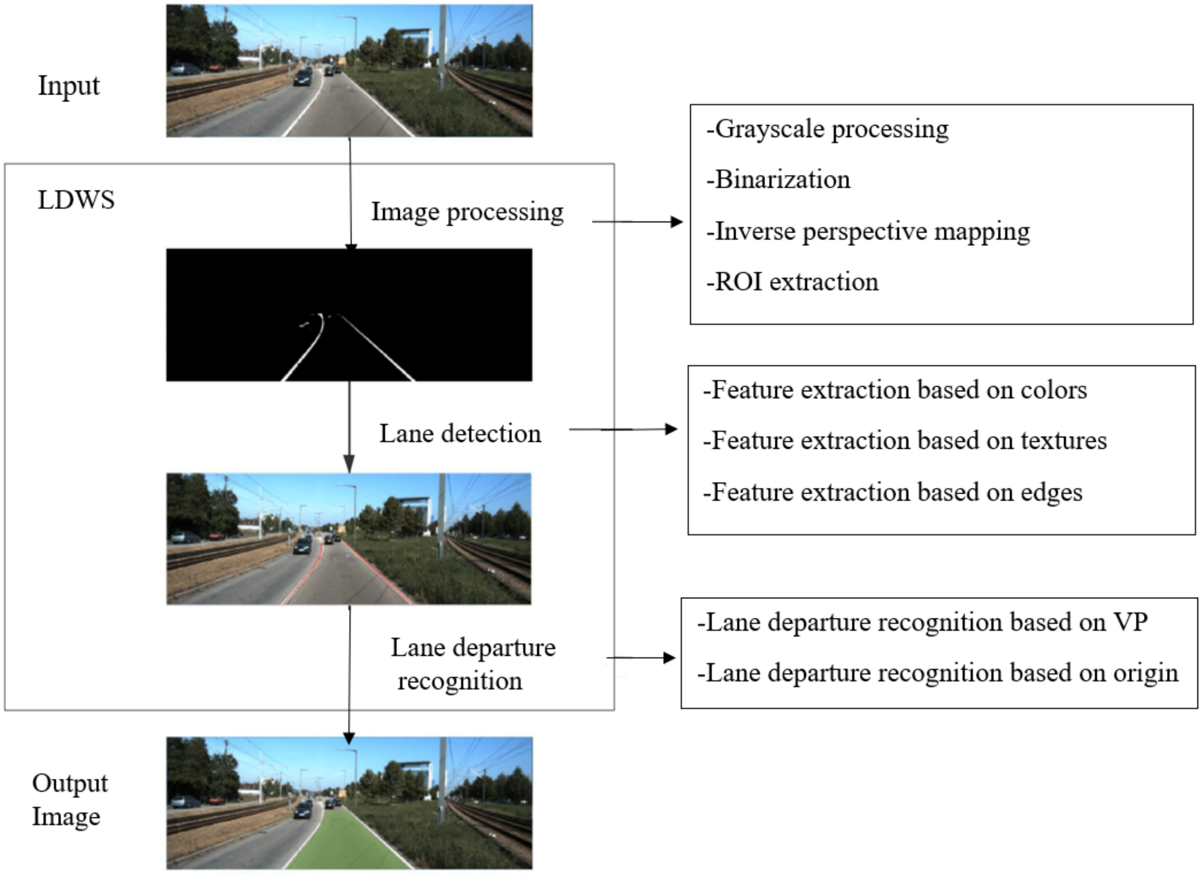

Figure 3.1: The proceduce of LEHA.

3.2 Image preprocessing

In our work, we focus on improving the speed of image preprocessing and eliminate the noise to the greatest extent. Therefore, we improve LEHA from the following aspects: grayscale processing, binarization, and image smoothing.

3.2.1 Grayscale processing

The objective of grayscale processing is to convert the original road image, which is in RGB color space, to a grayscale image. For each pixel a in RGB color space, its coordinate is R_a, G_a, and B_a, where R, G and B indicate red channel, green channel and blue channel respectively. The grayscale processing is the processing of making R_a, G_a, and B_a equal to a new-calculated GS_a, whose gray value is in the range of [0,255]. As the gray value GS_a increases, the color of this pixel tends to be white.

According to [222], the classical method of grayscale processing to derive the gray value GS_a is given by:

$$GS_a = r \cdot R_a + g \cdot G_a + b \cdot B_a, \tag{3.1}$$

where r, g, and b are assigned to 0.299, 0.587 and 0.114 and the sum of them is 1. The reason for the weight distribution is that the human eye has the highest sensitivity to green and the lowest sensitivity to blue. However, in practice, the road line tends to be marked by using yellow or white color. Accordingly, in the RGB model, the red channel and green channel provide much more information than the blue channel. Therefore, we change the assignment of these three channels, which assign a bigger value r to 0.36, g to 0.59, and less value b to 0.05. The proposed grayscale formula is shown as follows:

$$GS_a = 0.36 \cdot R_a + 0.59 \cdot G_a + 0.05 \cdot B_a. \tag{3.2}$$

Accordingly, relying on the newly assigned weights, we expect that the proposed grayscale processing method can enhance the contrast of the image, which will further provide convenience to the following binarization step and improves the accuracy of the image preprocessing stage. The original road image with straight lane and the result of our proposed grayscale processing is shown in Fig. 3.2 and 3.3, while the original road image with curve lane and the result of our proposed grayscale processing is shown in Fig. 3.6 and 3.7.

Figure 3.2: The original image with straight lane.

3.2.2 Image smoothing

Image smoothing aims to make the image more smooth by removing the noise from the image, by which the interference of detection caused by noise can be avoided, and the accuracy of the image preprocessing stage can be further improved. In general, image

Figure 3.3: The result of our proposed grayscale processing method.

Figure 3.4: The result of our proposed image smoothing method.

Figure 3.5: The result of our proposed binarization method.

smoothing uses the filter to remove the noise which is in high-frequency. One of the most wide-used filters in image smoothing is the median filter. The classic median filter uses the median, which is calculated by the current pixel and its neighborhood, to instead the current pixel value.

In this article, we design a novel median filter that aims to improve the efficiency and accuracy of image smoothing. The detailed procedure of the proposed image smoothing method is shown below:

1. We select a pixel from the left top corner of the image and set it as the center C.

2. Creating a square window W_{IS} around C with a width of $2w + 1$, which contains $(2w + 1)^2$ pixels. If the pixel is at the corner, we ignore the pixels outside the image. The pixels in the window are $(w + 1)^2$. If the pixel is at the boundary, the pixels in the window are $(2w + 1)(w + 1)$.

3. Since the total number of pixels in this window is odd, we can obtain the median M_{IS} of this window by implementing heap function[1] directly. In detail, we put all the pixels in an array and divide this array into two subsets H_1 and H_2 with a size difference of 1, which is shown as follows:

$$\begin{cases} H_1 = \frac{(2w+1)^2-1}{2}, \\ H_2 = \frac{(2w+1)^2+1}{2}. \end{cases} \tag{3.3}$$

For the pixel at the corner, the formula should be the following:

$$\begin{cases} H_1 = \frac{(w+1)^2-1}{2}, \\ & \text{if } w \text{ is even} \\ H_2 = \frac{(w+1)^2+1}{2}. \end{cases} \tag{3.4}$$

$$H_1 = H_2 = \frac{(w + 1)^2}{2}. \quad \text{if } w \text{ is odd} \tag{3.5}$$

For the pixel at the boundary of the image, the formula is given by:

$$\begin{cases} H_1 = \frac{(2w+1)(w+1)-1}{2}, \\ & \text{if } w \text{ is even} \\ H_2 = \frac{(2w+1)(w+1)+1}{2}. \end{cases} \tag{3.6}$$

$$H_1 = H_2 = \frac{(2w + 1)(w + 1)^2}{2}. \quad \text{if } w \text{ is odd} \tag{3.7}$$

[1]Here, the heap function is predefined in Python.

In our algorithm, H_1 is used to create a minimum heap. The factors in H_2 are used to replace the peak of this heap in sequence if this factor is greater than it. By comparing all the items in H_2, the peak of the heap is M_{IS} of this sliding window. The detailed process is shown in Algorithm 1.

Algorithm 1 The algorithm for searching the median of the window

```
set heapsize = (2w + 1)^2/2;
set arraylength = (2w + 1)^2;
for i = 0; i < heapsize; i++ do
    heap.add(array[i]);
end for
for i = heapsize; i < arraylength; i++ do
    if heap.peek() < array[i] then
        heap.poll();
        heap.add(array[i]);
    end if
end for
return heap.peek();
```

4. For each pixel $P_i \in W_{IS}$, we can obtain the mean value A_{IS} of this window by using the following formula:

$$A_{IS} = \frac{\sum_{i=1}^{(2w+1)^2} P_i}{(2w+1)^2}. \tag{3.8}$$

5. After getting the median and mean, we use their weighted sum MA_{IS} to replace the pixel value of C, which is given by:

$$MA_{IS} = \alpha \cdot M_{IS} + \beta \cdot A_{IS}, \tag{3.9}$$

where α and β are two user defined control parameters. Since the mean A_{IS} can be affected by the pixels that are too small or too large in the sliding window, which may deduce the accuracy of the processing result, we assign A_{IS} a smaller weight. In this article, we choose the value of α to 0.76 and β to 0.24 in trial-and-error manner based on the KITTI dataset.

Relying on the proposed smoothing method, the entire grayscale image is smoother. Therefore, in the following binarization step, the binarization result will contain less noise, which ensures the lane detection result accuracy is higher. Fig. 3.4 and Fig. 3.8 show the result of the road image with straight lane and curve lane respectively after image smoothing.

Figure 3.6: The original image with curve lane.

3.2.3 Binarization

As the last stage of image processing, binarization is used to convert the grayscale image generated by grayscale processing to an image consisted of two colors, i.e., white and black. The purpose of binarization is to separate the foreground from the road image. Typically, binarization is achieved by comparing the grayscale value of each pixel with the predefined threshold of T. According to the range of the grayscale value [0, 255], the range of the threshold T is also in this interval. The comparison process is shown below:

$$P(x_a, y_a) = \begin{cases} 255, & P(x_a, y_a) \geq T, \\ 0, & otherwise. \end{cases} \tag{3.10}$$

Figure 3.7: The result of our proposed grayscale processing method.

Figure 3.8: The result of our proposed image smoothing method.

Figure 3.9: The result of our proposed binarization method.

In the traditional global thresholding method [232, 233, 234], one threshold is applied to the entire image. While in the traditional local thresholding method, different sliding windows have different thresholds. In our work, we aim to obtain the threshold in the way of combining global and local thresholding methods. The proposed binarization combines the advantages of both of the two different methods and avoids their shortages, i.e., maintains the global structure and the local details of the entire image. The proposed method is shown below:

1. Similar to the proposed image smoothing method, to further analyze the information contained within the smoothed image, we adopt a sliding window W_B to traverse the entire image as well. The window size of this step is defined as $(2q+1)^2$. The total number of windows N_q can be indicated as follows:

$$N_q = \left\lceil \frac{N}{(2q+1)^2} \right\rceil, \tag{3.11}$$

   where N represents the number of pixels in the entire image.

2. Calculating the average value A_B of all the pixels $P_i \in W_B$ in each window in the image by using the following formula:

$$A_B = \frac{\sum_{i=1}^{(2q+1)^2} P_i}{(2q+1)^2}, \tag{3.12}$$

3. The average value A_B is considered to be the threshold. By comparing the threshold A_B to each pixel in this window, we use Eq.(3.10) to determine the grayscale value of this pixel.

4. Repeat all the above three steps N_q times to traverse the entire image by the sliding windows to obtain the proposed binarization result.

The results of our binarization method of the straight line and the curve lane are shown in Fig. 3.5 and Fig. 3.9 respectively.

3.3 Lane detection

In the lane detection stage, we design a novel ROI method to extract the part of the road image which contains the boundaries of the lane first. Then we use edge detection to identify all the edges in ROI. Finally, we apply a method for implementing Hough transform to detect and mark the lane lines, which is the most significant part of our lane detection module.

3.3.1 ROI extraction

ROI is the area that contains lane lines without the other irrelevant parts, such as the sky, buildings, and trees. In many of the existing approaches, the ROI extraction method can be categorized as static ROI and dynamic ROI. The static ROI extraction method [5] presupposes a geometric shape as ROI. Accordingly, the static ROI-based method has a fast process speed. However, since the geometric shape can not fit all the road images, the lane detection accuracy is low. Therefore, the application of the static ROI method is constrained. In the dynamic ROI extraction method [6], the ROI is determined based on the position of the boundaries of the lane. The most widely-used method is the vanishing point-based ROI method. Since the vanishing point is the point of intersection of the lane lines, the accuracy of the dynamic ROI extraction method is higher than the static ROI method. However, it takes much more time to find the vanishing point.

In our work, we design a novel method to ROI extraction method based on the coordinate. The detailed steps of our proposed method are summarized as follows and Fig. 3.10 and Fig. 3.11 show the procedure and the result of our proposed ROI extraction method respectively by using an example of the straight lane:

1. We take the point in the lower-left corner of the picture as the origin, extend to the right as the X-axis and extend up as the Y-axis.
2. Finding two endpoints $E_L(x_L, y_L)$ and $E_R(x_R, y_R)$ of the lane lines which are the closest to the vehicle and determine the coordinates of them.
3. Comparing the value of y_L and y_R and select the larger one. In this method, we assume this point as E_R.
4. We create a line L_1 that is parallel to the X-axis and through E_R.
5. We create a line L_2 that through both of the smaller endpoint $E_L(x_L, y_L)$ and the start point $S_P(x_P, y_P)$ of this lane line.
6. Obtaining the point of intersection P_S of L_1 and L_2.

7. We create a trapezoid whose lower base is the bottom of the road image, and the upper base is the segment between P_S and E_R. This trapezoid is the ROI in our work.

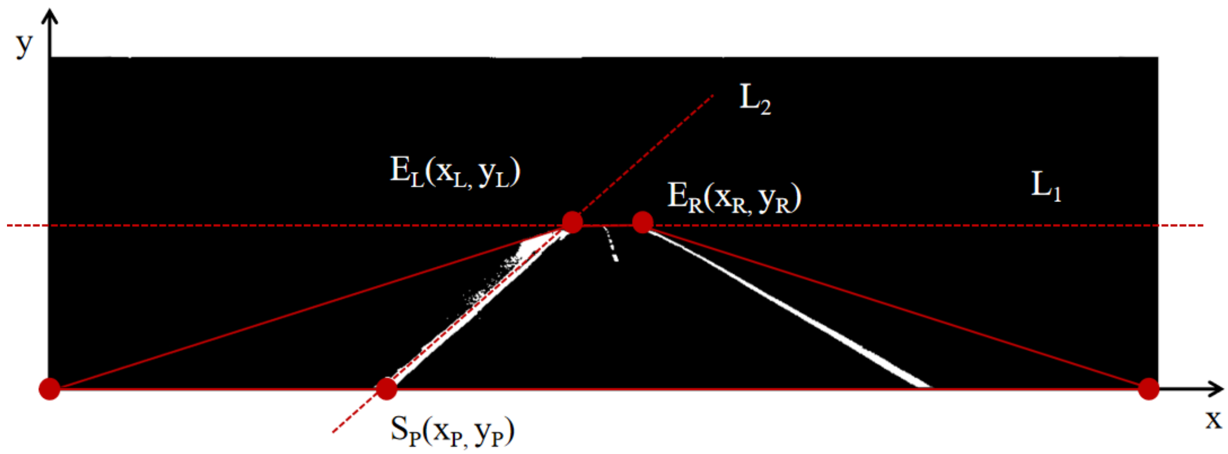

Figure 3.10: The detailed procedure of our proposed ROI extraction method.

Figure 3.11: The result of our proposed ROI extraction method.

The strength of this method is that this ROI contains the entire lane lines that need to be detected in the following steps. Meanwhile, this method makes the detection area smaller. Compared with the other dynamic ROI extraction methods such as the methods using vanishing point [6], our method is efficient as we do not need time to search the vanishing point. Compared with the static ROI extraction [5], the accuracy of our method is much higher than these kinds of methods because the shape of ROI changes based on the different road images. The results of both the straight lane and the curve lane are shown in Fig. 3.14 and Fig. 3.18.

3.3.2 Canny edge detection

Edge is a collection of the points on the road image where the grayvalue of the surrounding pixels changes rapidly [107, 235]. Since the color of the lane lines is very different from the road, we can identify the lane lines by using edge detection. Edge detection is a method that identifies the edges of lane lines and provides convenience to the following Hough transform to detect and mark the lane lines. In edge detection, the Canny operator is the most popular method. Thus, we adopt CED from [236] in our work, and the result of the straight lane and curve lane are shown in Fig. 3.15 and Fig. 3.19. The specific algorithm is shown below:

1. Using the Gaussian filter to remove the noise from the entire road image.
2. Applying the first-order differential partial derivative to calculate the gradient value and direction.
3. Suppressing the pixels that are not the maximum, which indicates that set these pixel values to 0. This step can greatly remove the weak edges and make the edges thinner.
4. Setting two thresholds T_1 and T_2, where T_1 indicates the lower threshold and T_2 indicates the higher threshold. If the gradient of the pixel is bigger than T_2, then it is the edge point. If the gradient of the pixel is smaller than T_1, then it is not the edge point. Otherwise, we use Breadth-First-Search (BFS) to search the edges based on the known edge points. If the point can be reachable by the known edge point, it can be assumed as the edge point. Otherwise, it is not.

3.3.3 Hough transform

Hough transform is a method for detecting and marking the lane lines based on the result of edge detection by using the coordinate transformation [237]. Coordinate transformation is a process of mapping a line, i.e., straight line or curve, from the Cartesian coordinate system to a point in polar coordinates in order to obtain a peak. This peak indicates the line in the Cartesian coordinate system [238]. Therefore, we transform the line-detection problem into the peak-searching problem.

In the classical Hough transform method, the common way is to transform every pixel between the two coordinates, which takes much time and reduces the efficiency of the entire

LDWS. In our work, we design a more efficient method to implement Hough transform which is shown as follows:

1. We divide the whole ROI horizontally into ten equal parts as shown in Fig. 3.12, each line has a point of intersection with the lane lines.
2. We use these points of intersection to transform from the Cartesian coordinate system to polar coordinates to obtain the peak. In general, as we usually detect two lane lines, there are two peaks in polar coordinates.
3. We map these two peaks to the first coordinate and obtain the detection result.

Figure 3.12: The method of our proposed Hough transform.

In our Hough transform method, we mainly focus on the non-ideal lane line condition. In the non-ideal lane line condition, we only need to transform twenty-two points, which saves much more time than the classical Hough transform method. As we divide the lane lines equally, this considers the distribution of different pixels on the lane lines, which ensures the accuracy of LDWS while improving the efficiency of detecting and marking the lane lines in this stage. Fig. 3.16 and Fig. 3.20 show the result of the Hough transform applied to the straight lane and the curve lane.

3.4 Lane departure recognition

Lane departure recognition is the final step of LDWS, which focuses on determining whether this vehicle has a risk of deviation from the safe driving area and whether the warning needs to turn on. In LDWS, the most common method for recognizing the departure of a vehicle

Figure 3.13: The original image with straight lane.

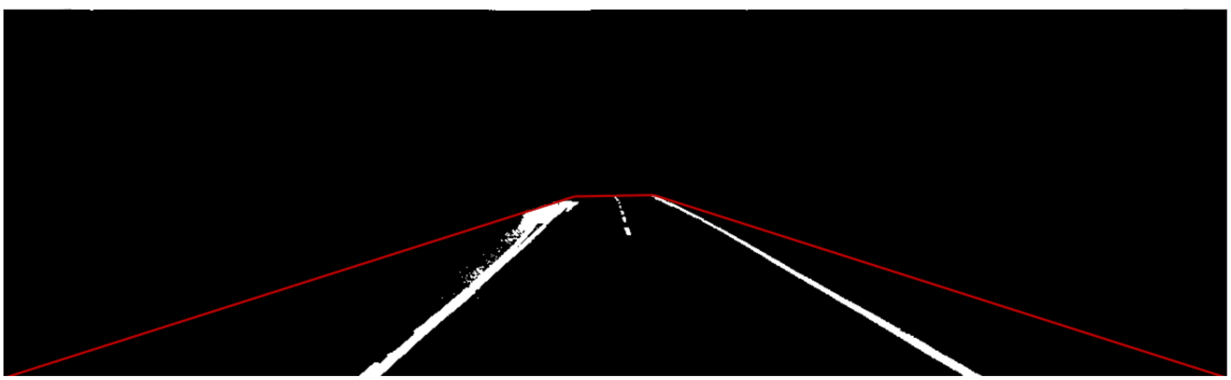

Figure 3.14: The result of our proposed ROI extraction method.

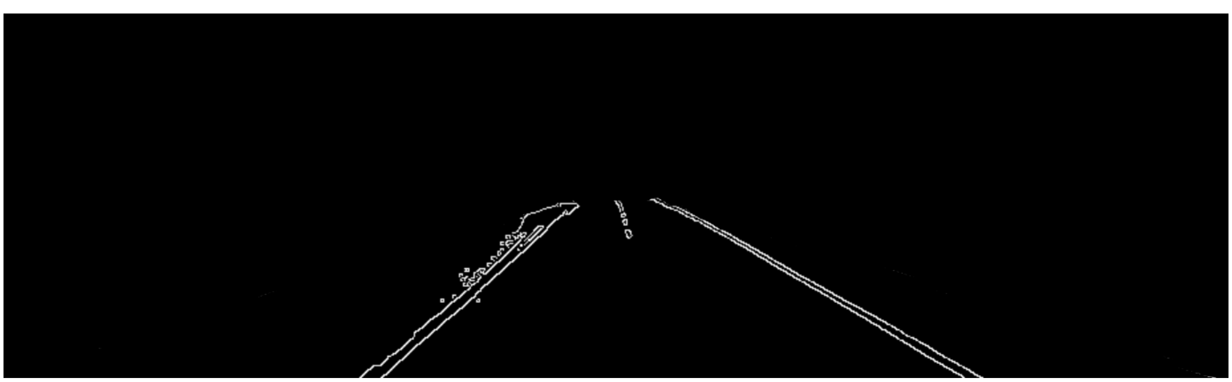

Figure 3.15: The result of Canny edge detection method in our work.

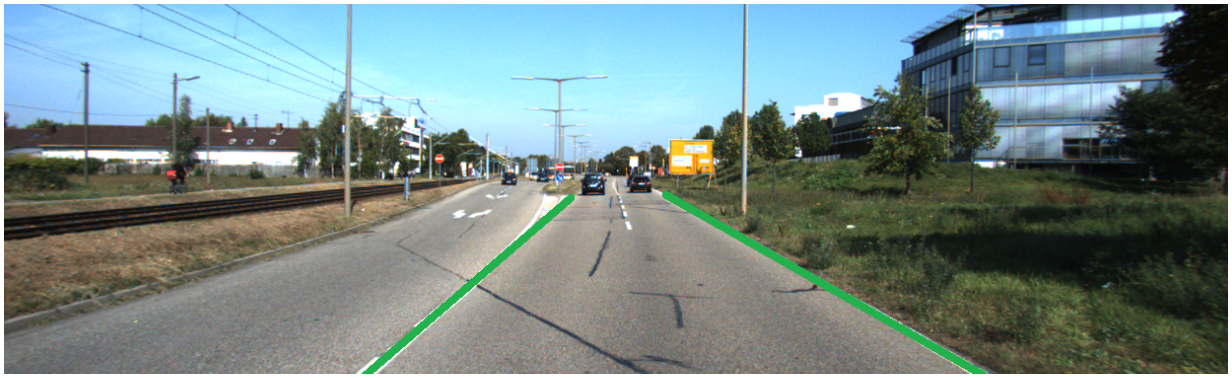

Figure 3.16: The result of our proposed Hough transform method.

Figure 3.17: The original image with curve lane.

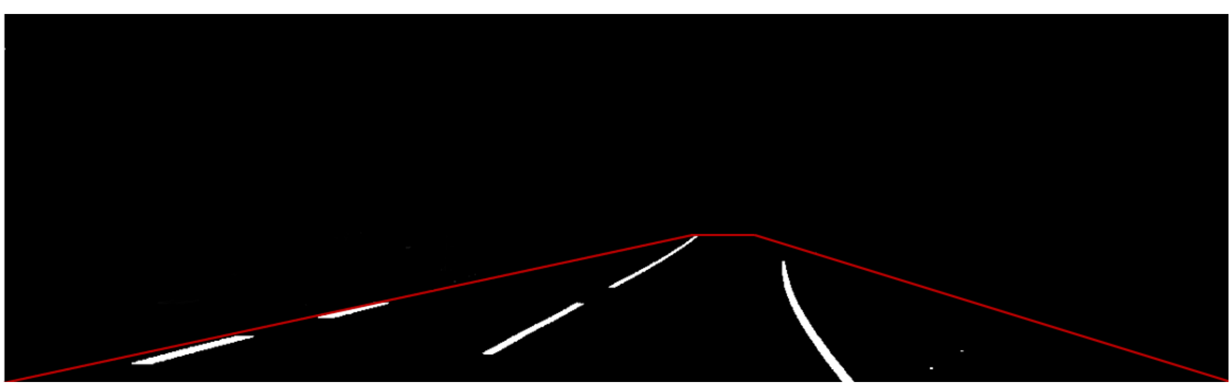

Figure 3.18: The result of our proposed ROI extraction method.

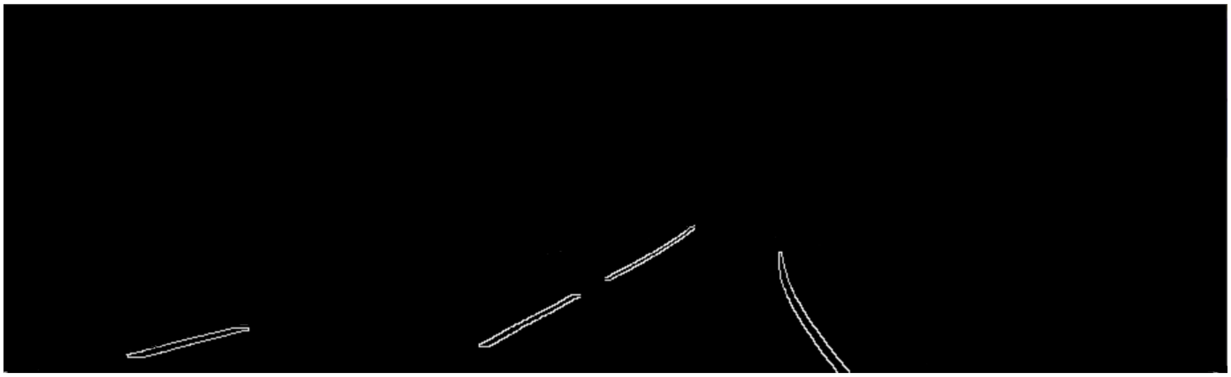

Figure 3.19: The result of Canny edge detection method in our work.

Figure 3.20: The result of our proposed Hough transform method.

is to measure the distances between the vehicle and both of the lane lines on either side of it. The advantage of our system is we calculate the deviation distance between the vehicle and the centerline of the lane, and present the deviation direction and the deviation distance to the driver directly. In our system, we assume the vehicle as a point in the image, which is the midpoint on the bottom boundary, and get the distance between the boundaries of the lane respectively. After obtaining the distances, we can calculate the actual deviation distance of the vehicle by using the ratio of the derivation distance and the width of the lane in the image, where deviation distance indicates the distance between the position of the vehicle and the centerline of the lane. The proposed lane departure recognition algorithm can be summarized as follows and Fig. 3.21 shows our lane departure recognition method:

1. We assume the left-bottom corner as the origin and create a coordinate of the image. The bottom boundary of the image is the X-axis, and the left boundary is the Y-axis.
2. We set the vehicle as the objective point OP and its horizontal line intersects the two-lane lines at left point OP_L and right point OP_R respectively.
3. The distance between OP and OP_L is m and the distance between OP and OP_R is n. The width of the lane is assigned to q. The actual width of the lane is WD.
4. The deviation distance D_d can be calculated by the following formula:

$$D_d = \begin{cases} \frac{m \times WD}{q} - \frac{WD}{2}, & m \geq n, \\ \\ \frac{WD}{2} - \frac{n \times WD}{d}, & otherwise. \end{cases} \tag{3.13}$$

Figure 3.21: The method of our proposed lane departure recognition.

The strength of our proposed lane departure recognition method is that it shows the actual deviation distance D_d clearly and directly, which alerts the driver to realize the degree of the risk of deviation from the safety driving area. Besides that, LEHA presents the deviation distance between the vehicle and the centerline of the lane. Our proposed lane departure recognition can greatly assist the driver back to the correct driving area in time by the warning. Fig. 3.22, Fig. 3.23, Fig. 3.24 and Fig. 3.25 show the result of our proposed LDWS in both the straight lane and the curve lane, which marks the boundaries of the lane and presents the derivation distance.

Figure 3.22: The original image with straight lane.

Figure 3.23: The result of LEHA in straight lane.

Figure 3.24: The original image with curve lane.

Figure 3.25: The result of LEHA in curve lane.

Chapter 4

Evaluation results

In this section, we present the comparison between LEHA and other four methods: [10], [11], [4] and [12]. These LDWS were proposed in recent years in terms of the final detection results under different lane conditions, the accuracy of lane detection rate and departure detection rate, and the processing time of these methods.

4.1 Testing environment

LEHA is based on Python 3.4 and tested on Lenovo V720-14 laptop, Intel Core i5-7200U 2.5GHz Dual-Core Processor, NVIDIA GeForce 940MX 2GB Dedicated GDDR5. The test road images are randomly selected from KITTI [239], which contains real road image data from different scenes, such as urban, rural, and highway.

Figure 4.1: Straight lane with ideal lane line condition in [10].

Figure 4.2: Straight lane with ideal lane line condition in [11].

Figure 4.3: Straight lane with ideal lane line condition in [4].

Figure 4.4: Straight lane with ideal lane line condition in [12].

Figure 4.5: Straight lane with ideal lane line condition in LEHA

Figure 4.6: Curve lane with ideal lane line condition in [10]

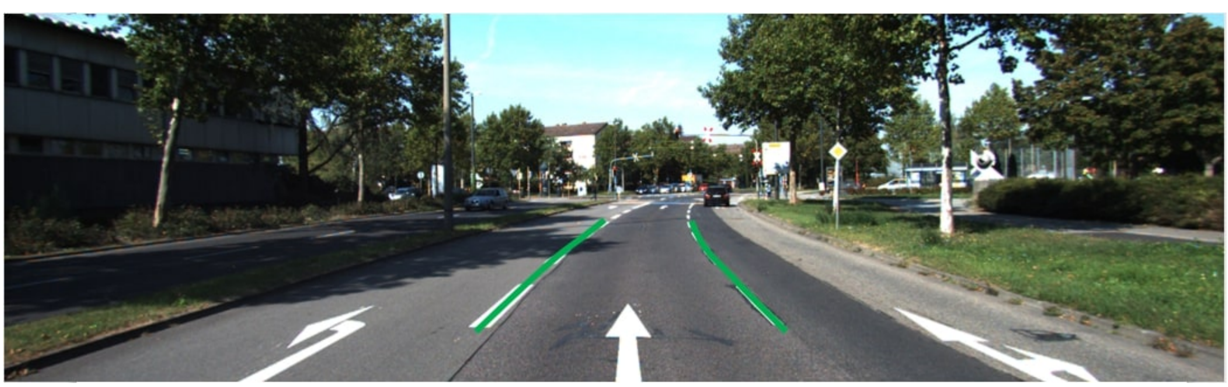

Figure 4.7: Curve lane with ideal lane line condition in [11]

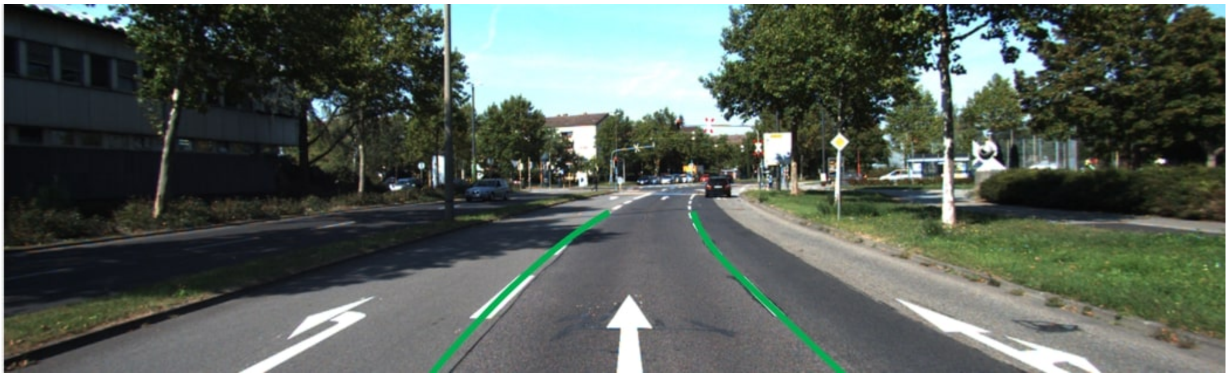

Figure 4.8: Curve lane with ideal lane line condition in [4]

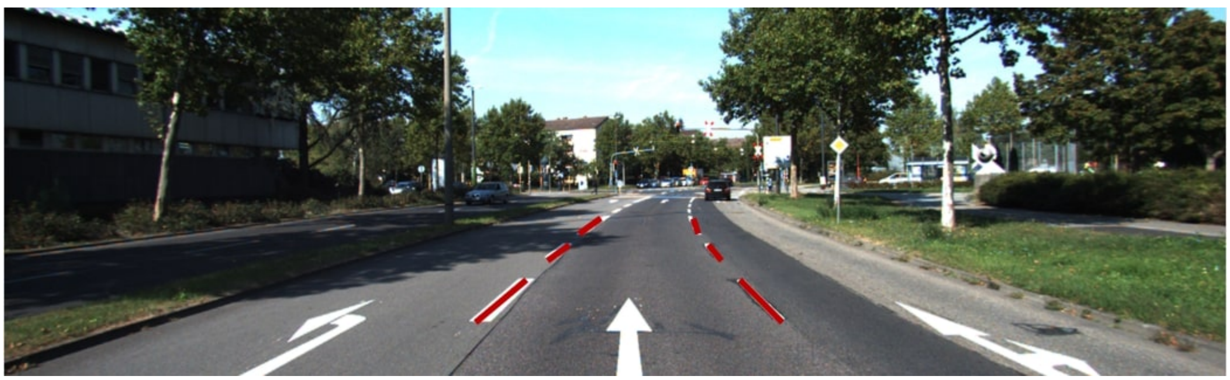

Figure 4.9: Curve lane with ideal lane line condition in [12]

Figure 4.10: Curve lane with ideal lane line condition in LEHA

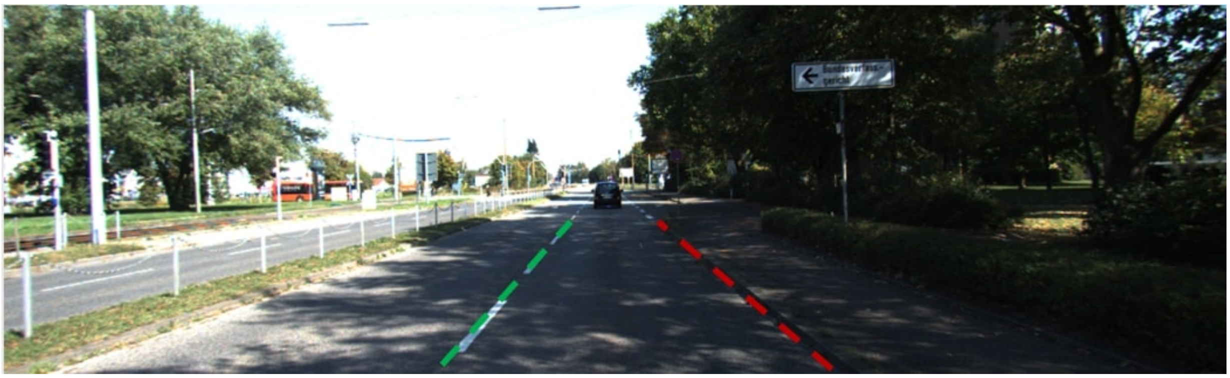

Figure 4.11: Straight lane with non-ideal lane line condition in [10]

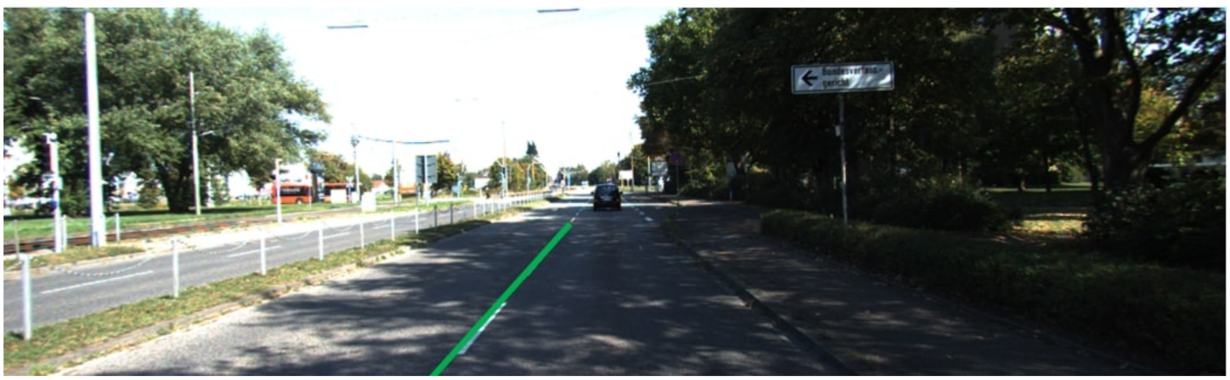

Figure 4.12: Straight lane with non-ideal lane line condition in [11]

Figure 4.13: Straight lane with non-ideal lane line condition in [4]

Figure 4.14: Straight lane with non-ideal lane line condition in [12]

Figure 4.15: Straight lane with non-ideal lane line condition in LEHA

Figure 4.16: Curve lane with ideal lane line condition in [10]

Figure 4.17: Curve lane with ideal lane line condition in [11]

Figure 4.18: Curve lane with ideal lane line condition in [4]

Figure 4.19: Curve lane with ideal lane line condition in [12]

Figure 4.20: Curve lane with ideal lane line condition in LEHA

4.2 Evaluation results

In practice, lane line conditions can not always be ideal, which may more or less be damaged or discolored to varying degrees. In LEHA, we focus on improving the accuracy of both the detection rate and the departure rate in non-ideal lane line conditions. Besides that, we are working to improve the processing speed.

In our paper, we compare our experimental result with the other four methods of LDWS in [10], [11], [4], and [12], which are represented as Gamal-LDWS, Chen-LDWS, Wu-LDWS, and Irshad-LDWS in the following diagrams. Figures above in Chapter 4.1 present the results of both the straight lane and the curve lane in the above methods in ideal lane line condition and non-ideal lane line condition respectively. From Fig. 4.1 to Fig. 4.10, we can clearly find that almost all the LDWS performs well in the ideal lane line condition as this condition can highly provide the lane lines information and details. Although the method in [12] can only mark the lane lines with white pixels, it marks the position of the lane lines correctly.

However, In the non-ideal conditions, the experiment results are really different. In lane detection, Gamal-LDWS [10], Wu-LDWS [4] and LEHA perform well in lane detection because all of the three methods can mark the lane lines completely. In Chen-LDWS [11] and Irshad-LDWS [12], the lane can be detected unless both of the left and right lane lines appear. The difference is that the method in Chen-LDWS [11] can connect the detected segments of lane lines and mark the entire lane lines completely, while the method in Irshad-LDWS [12] can only show the detected parts of the lane lines.

All the methods do not present the specific deviation distance and the direction between the vehicle and the centerline of the lane except LEHA, as we use the ratio of the deviation distance and the width of the road in the image to obtain the deviation distance in the real world in the lane departure recognition stage., which is one the advantages of us. Showing the deviation distance can help the driver recognize whether the deviation degree of the vehicle is serious and the deviation direction helps the driver adjust the driving direction, which is convenient for the vehicle to get back to the safe driving area in time.

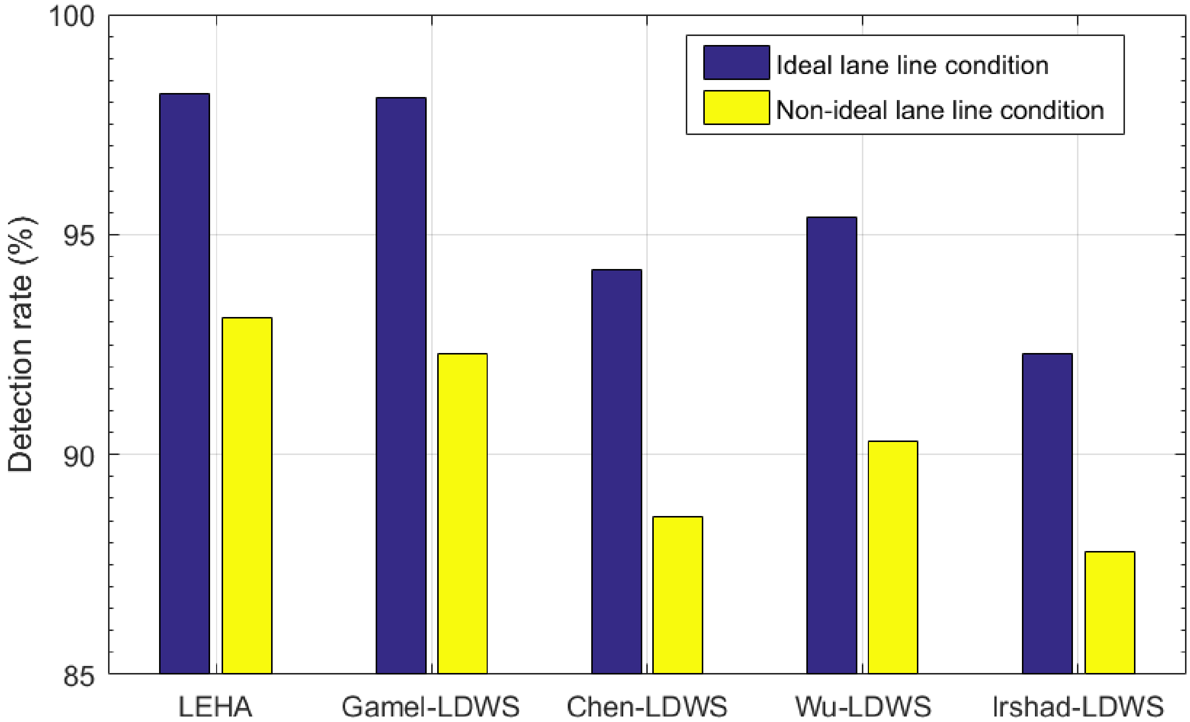

Figure 4.21: The comparison of detection rate.

Fig. 4.21 presents the detection rate of the five methods. As the information on lane lines does not fall completely into non-ideal conditions, all the detection rates are lower than the ideal one. Our proposed LDWS achieves the highest rate in both lane line conditions, which is 98.2% under ideal lane line conditions and 93.1% under non-ideal lane line conditions. The reason that LEHA can get high accuracy is that we focus on removing noise in the image preprocessing stage by combining two kinds of binarization methods, improving the grayscale processing formula, and adding image smoothing. Without the noise disturbing our experimental results, we can obtain a higher accuracy image. The second higher accuracy was obtained in Gamal-LDWS [10], at 98.1% and 92.3%, respectively; the experimental results under ideal lane line conditions are similar to ours. However, in the non-ideal conditions, their accuracy is 0.8% lower than ours because in their image preprocessing stage, they use only image smoothing, while we apply the newly-designed grayscale processing and binarization to improve the accuracy. As the methods in Chen-LDWS [11] and Irshad-LDWS [12] can only detect the lane lines with the marked parts, their detection rates are obviously lower than others in both of the conditions, whose accuracy is less than 95% under the ideal conditions and no more than 89% under non-ideal conditions.

The comparison of the departure rate of these methods is shown in Fig. 4.22. The departure rate of all the methods under the ideal lane line conditions is over 95%, which indicates that they perform well in detecting the vehicle deviation. Among them, LEHA obtains the highest departure rate at 99.1%, which means that LEHA can detect almost all the departure situations. The method in Gamal-LDWS [10] achieved second place, with a departure rate is 98.7% in the ideal conditions and 94.3% under non-ideal conditions, 0.4% and 0.9% lower than ours respectively. In the non-ideal lane line conditions, the reason that LEHA can get the first place at 95.2% is that we consider the vehicle as a point, and the deviation distance is the distance between the vehicle and the centerline of the lane, which requiring LEHA to be sensitive with the departure of the vehicle. The methods in Wu-LDWS [4] and Irshad-LDWS [12] obtained the third and fourth place in this comparison with the departure rate at 93.5% and 92.1%. There is not a great difference in accuracy between the four methods. However, in Fig. 4.22, we find that only the departure rate of Irshad-LDWS [12] is less than 90%, which is 89.6%. As the accuracy of the lane detection stage is low, this method cannot mark the correct position of the lane lines, especially under the non-ideal lane line conditions. The deviation distance cannot be measured precisely, which causes a low departure rate of this method.

Fig. 4.23 shows the comparison of the total processing time of each method in ideal and non-ideal conditions. In order to test the precise processing time, we traverse the KITTI dataset. From Fig. 4.23, we find that for all the methods, the minimum processing time for each image is 16.08 ms and the maximum processing time for each image is 25 ms. In LEHA, it takes 14.4 ms per image on average, and 23.4 ms per image on average. This is the optimal processing time all over these methods. As we design a newly-designed ROI extraction method, which removes the irrelevant part of the image to the greatest extent and proposed a novel Hough transform, which only needs to detect twenty-two lane line pixels in the image, we can see that both of these two parts help save much processing time. The method in Gamal-LDWS [10] takes 15.8 ms and 24.8 ms per road image under ideal and non-ideal images respectively. In Chen-LDWS [11], the average processing time under ideal conditions is 1.9 ms and 1.7 ms higher than ours. In Wu-LDWS [4], the average processing time in ideal conditions is 17.1 ms, which is the highest in comparison to the

Table 4.1: Comparison of recent deep learning LDWS methods

Methods	Dataset	Hardware	Accuracy (%)	Time (ms)
s-FNC-loc [240]	KITTI	Intel(R)CPU Xeon(R)E5-2697 v2 @ 2.70GHz 128GB RAM four NVIDIA Tesla K80 GPUs	98.1	400
DQLL [241]	NWPU	Intel Core i7-6800K @ 3.4 GHz CPU one NVIDIA GTX 1080Ti GPU	86.05	realtime
	TuSimple		93.36	realtime
CNN-LSTM [242]	/	NVIDIA Jetson Nano	94.4	89.3
Method in [243]	Road Vehicle Dataset (RVD)	NVIDIA TX2	97.067	realtime
	Caltech		98.067	realtime
	Tusimple		90	realtime

ideal conditions. Besides that, in Irshad-LDWS [12], the unideal processing time is 26.1 ms, which is the highest among all the methods.

Based on the above comparison of LEHA and other LDWS, we can conclude that LEHA performs better than them in terms of accuracy and efficiency. In addition, we present the deviation direction and the specific value of the deviation distance between the vehicle's position and the centerline of the lane, which can alert the driver back to the safe driving area more directly.

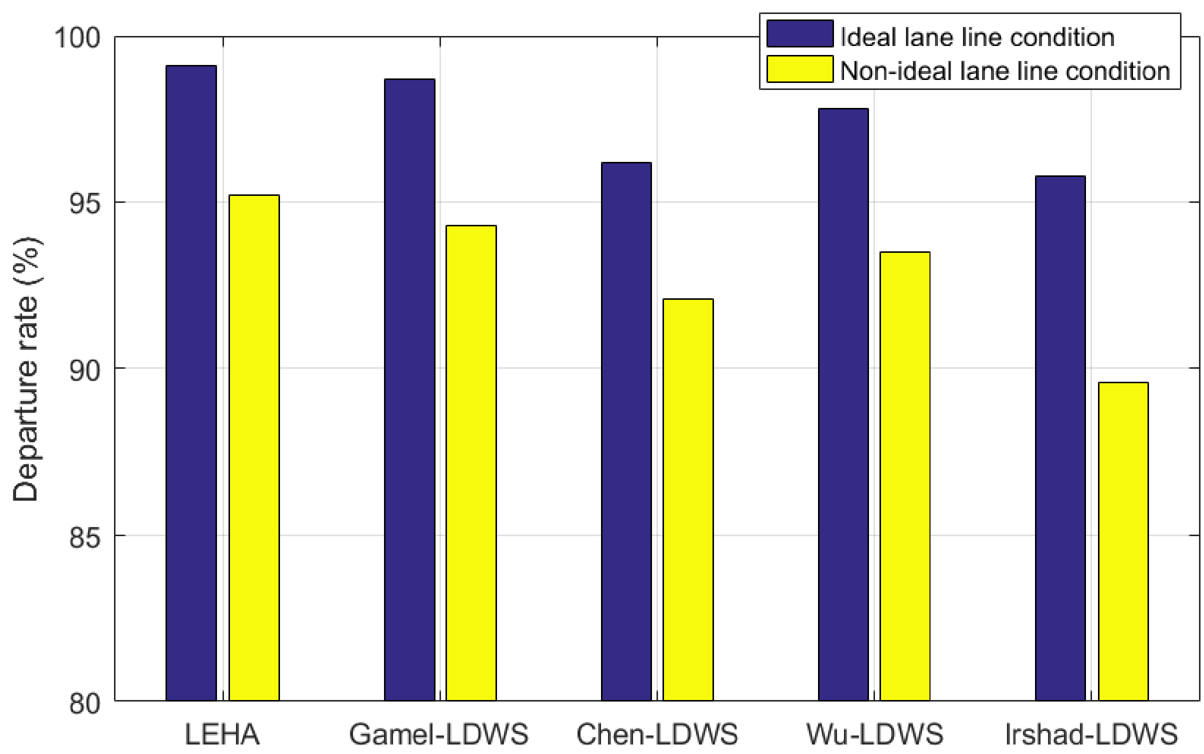

Figure 4.22: The comparison of departure rate.

Table 4.1 presents the deep learning methods which are proposed in recent years [240,

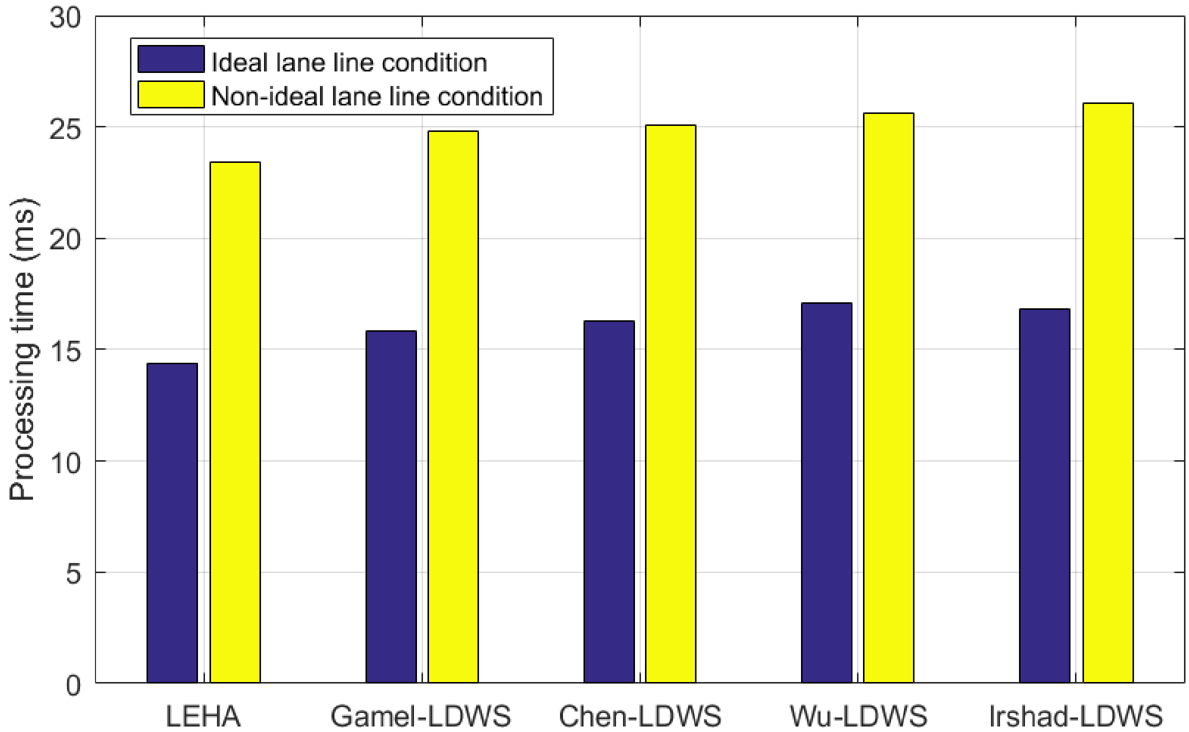

Figure 4.23: The comparison of processing time.

241, 242, 243]. From this table, we can find that all the deep learning methods use high-end hardware, such as NVIDIA GPU, which can not be widely used in vehicles. Besides that, based on the evaluation results above, we can achieve the same accuracy and efficiency as listed deep learning methods by using simple CPU hardware. Therefore, we can conclude that our LEHA is more suitable for practical use.

Chapter 5

Conclusion and future work

In this section, we will introduce some open challenges for improving the performance of LDWS, and then give a conclusion of the entire survey.

5.1 Emerging techniques

In this part, we introduce the following three open challenges in the above four stages for improving the efficiency and accuracy of LDWS.

5.1.1 Can LDWS be protected from noise to the greatest extent?

Image noise can be generated for various reasons, such as weather, camera, and the process of transfer. Image noise refers to the unnecessary and unimportant information details that the image may contain, which will affect the detected result to some extent. The existence of noise seriously affects the quality of the traffic image. Based on this reason, in order to get a clearer image and improve the efficiency and accuracy of the subsequent steps of lane detection, the original image needs to be denoised to make the details and edges in the image clearer, so as to facilitate improve the performance of LDWS.

5.1.2 Can LDWS achieve real-time performance?

Real-time is also one of the most important properties of LDWS. When the vehicle driving on the road, real-time ensures that the position of the vehicle is in the proper area of the lane. Besides that, when LDWS detects that the vehicle is too close to the lane line, it warns the driver to go back to the original driving trajectory in a timely manner. In [10, 244, 245], real-time is used to make the detection results more reliable and warn the potential danger, such as collision as fast as possible.

5.1.3 Are the calculated distance and direction messages presented to the driver easy to understand?

The messages shown on the system, which are presented to the driver need to be easy to understand. For instance, the driver needs to know clearly that if the system is engaged, if the distance between the position of the car and the lane line, the direction of the driving route, or if he is driving inside a proper area of the road lane. In [246, 247, 11, 248, 249], they used lane departure recognition to present the virtual messages in front of the driver clearly.

5.2 Conclusion

In this thesis, we introduce a novel proposed LDWS LEHA based on the improved methods of grayscale processing, binarization, image smoothing in the image preprocessing stage, the newly-designed ROI extraction, and Hough transform in the lane detection stage, and a precise calculation of the derivation distance and direction in the lane departure recognition stage, which focuses on improving accuracy and efficiency.

In the image stage, we remove most of the irrelevant noise from the image by using the above-proposed methods, which improve the accuracy of our LDWS. The novel ROI extraction and Hough transform are used to maintain the significant part of the image which contains the lane line information and reduces the detection time. Both of them work for reducing the processing time and improve efficiency. In the last part of LEHA, we present the deviation direction and distance between the vehicle and the centerline of the lane to alert the driver if he is out of the safe driving area and whether needs to drive approach the middle area of the lane.

The experimental results of LEHA are also presented in this paper. Under the ideal and non-ideal lane line conditions, the detection rates are 98.2% and 93.1%, and the departure rates are 99.1% and 95.2% respectively. For the processing time, in the ideal condition, the average processing time is 15.8 ms and in the non-ideal condition, the average processing time is 24.8 ms.

References

[1] Constantin Vertan, Corneliu Florea, Laura Florea, and Mihai-Sorin Badea. Reusing the otsu threshold beyond segmentation. In *International Symposium on Signals, Circuits and Systems*, pages 1–4, 2017.

[2] EYUPOGLU Can. Implementation of bernsen's locally adaptive binarization method for gray scale images. *The Online Journal of Science and Technology*, 7(2):68, 2017.

[3] Yang Lingxiao and Feng Qingxiu. The improvement of bernsen binarization algorithm for qr code image. In *IEEE International Conference on Cloud Computing and Intelligence Systems*, pages 931–934, 2018.

[4] Wu Chung-Bin, Wang Li-Hung, and Wang Kuan-Chieh. Ultra-low complexity block-based lane detection and departure warning system. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):582–593, 2018.

[5] Deng Ganlu and Wu Yefu. Double lane line edge detection method based on constraint conditions hough transform. In *International Symposium on Distributed Computing and Applications for Business Engineering and Science*, pages 107–110, 2018.

[6] Huo Chih-Li, Yu Yu-Hsaing, and Sun Tsung-Ying. Lane departure warning system based on dynamic vanishing point adjustment. In *IEEE Global Conference on Consumer Electronics*, pages 25–28. IEEE, 2012.

[7] Rathnayake Bandarage Shehani Sanketha and Ranathunga Lochandaka. Lane detection and prediction under hazy situations for autonomous vehicle navigation. In *International Conference on Advances in ICT for Emerging Regions*, pages 99–106, 2018.

[8] Jinyu Liu, Lu Lou, Darong Huang, Yu Zheng, and Wang Xia. Lane detection based on straight line model and k-means clustering. In *Data Driven Control and Learning Systems Conference*, pages 527–532. IEEE, 2018.

[9] Mandlik Pravin T and Deshmuk AB. Image processing based lane departure warning system using hough transform and euclidean distance. *International Journal of Research and Scientific Innovation*, III, 2016.

[10] Islam Gamal, Abdulrahman Badawy, Awab M.W. Al-Habal, Mohammed E.K. Adawy, Keroles K. Khalil, Magdy A. El-Moursy, and Ahmed Khattab. A robust, real-time and calibration-free lane departure warning system. In *IEEE International Symposium on Circuits and Systems*, pages 1–4, 2019.

[11] Peijiang Chen and Junhao Jiang. Algorithm design of lane departure warning system based on image processing. In *IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference*, pages 1–2501, 2018.

[12] Irshad Aman, Khan Anam Ahmad, Yunus Ibrar, and Shafait Faisal. Real-time lane departure warning system on a lower resource platform. In *International Conference on Digital Image Computing: Techniques and Applications*, pages 1–8, 2017.

[13] A Boukerche, H.A.B Oliveira, E.F Nakamura, and A.A.F Loureiro. Localization systems for wireless sensor networks. *IEEE Wireless Communications*, 14(6):6–12, 2007.

[14] Azzedine Boukerche. *Algorithms and protocols for wireless and mobile ad hoc networks*, volume 77. John Wiley & Sons, 2008.

[15] Jessica B Cicchino. Effects of lane departure warning on police-reported crash rates. *Journal of Safety Research*, 66:61–70, 2018.

[16] World Health Organization. *World report on ageing and health*. World Health Organization, 2015.

[17] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. Sdar: a secure distributed anonymous routing protocol for wireless and mobile ad hoc networks. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 618–624. IEEE, 2004.

[18] Assuncao Arthur N, de Paula Fabio O, and Oliveira Ricardo AR. Methodology to events identification in vehicles using statistical process control on steering wheel data. In *Proceedings of the 13th ACM International Symposium on Mobility Management and Wireless Access*, pages 1–4. ACM, 2015.

[19] Hazrat Bilal, Baoqun Yin, Jawad Khan, Luyang Wang, Jing Zhang, and Aakash Kuma. Real-time lane detection and tracking for advanced driver assistance systems. In *Chinese Control Conference*, pages 6772–6777, 2019.

[20] Azzedine Boukerche, Horacio A.B.F Oliveira, Eduardo F Nakamura, and Antonio A.F Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, 31(12):2838–2849, 2008.

[21] Chen Dong, Tian Zonghao, and Zhang Xiaolong. Lane detection algorithm based on inverse perspective mapping. In *International Conference on Man-Machine-Environment System Engineering*, pages 247–255, 2019.

[22] Jan Heller, Michal Havlena, and Tomas Pajdla. Globally optimal hand-eye calibration using branch-and-bound. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):1027–1033, 2016.

[23] *Handbook of algorithms for wireless networking and mobile computing*. Chapman & Hall/CRC computer and information science series. Chapman & Hall/CRC, Boca Raton, FL, 2006.

[24] A Boukerch, L Xu, and K EL-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer communications*, 30(11):2413–2427, 2007.

[25] Azzedine Boukerche, Xiuzhen Cheng, and Joseph Linus. Energy-aware data-centric routing in microsensor networks. In *Proc. ACM MSWiM workshop*, MSWIM '03, pages 42–49, 2003.

[26] Rodolfo Wanderson Lima Coutinho, Azzedine Boukerche, Luiz Filipe Menezes Vieira, and Antonio Alfredo Ferreira Loureiro. Geographic and opportunistic routing for underwater sensor networks. *IEEE transactions on computers*, 65(2):548–561, 2016.

[27] Yonglin Ren, Richard Werner, Nelem Pazzi, and Azzedine Boukerche. Monitoring patients via a secure and mobile healthcare system. *IEEE Wireless Communications*, 17(1):59–65, 2010.

[28] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Secure localization algorithms for wireless sensor networks. *IEEE Communications Magazine*, 46(4):96–101, 2008.

[29] Horacio Antonio Braga Fernandes De Oliveira, Azzedine Boukerche, Eduardo Freire Nakamura, and Antonio Alfredo Ferreira Loureiro. An efficient directed localization recursion protocol for wireless sensor networks. *IEEE Transactions on Computers*, 58(5):677–691, 2008.

[30] SAE On-Road Automated Vehicle Standards Committee et al. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J*, 3016:1–16, 2014.

[31] Osama Abumansoor and Azzedine Boukerche. A secure cooperative approach for nonline-of-sight location verification in vanet. *IEEE Transactions on Vehicular Technology*, 61(1):275–285, 2011.

[32] Azzedine Boukerche and Sajal K Das. Dynamic load balancing strategies for conservative parallel simulations. In *Proceedings 11th Workshop on Parallel and Distributed Simulation*, pages 20–28. IEEE, 1997.

[33] Azzedine Boukerche and Yonglin Ren. A trust-based security system for ubiquitous and pervasive computing environments. *Computer communications*, 31(18):4343–4351, 2008.

[34] Maram Bani Younes and Azzedine Boukerche. Intelligent traffic light controlling algorithms using vehicular networks. *IEEE transactions on vehicular technology*, 65(8):5887–5899, 2015.

[35] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina Borges Araujo. A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 157–164, 2004.

[36] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina Borges Araujo. Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments. *Journal of Parallel and Distributed Computing*, 66(4):586–599, 2006.

[37] Azzedine Boukerche and Amir Darehshoorzadeh. Opportunistic routing in wireless networks: Models, algorithms, and classifications. *ACM Computing Surveys (CSUR)*, 47(2):1–36, 2014.

[38] Azzedine Boukerche and Samer Samarah. A novel algorithm for mining association rules in wireless ad hoc sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(7):865–877, 2008.

[39] Rodolfo WL Coutinho, Azzedine Boukerche, Luiz FM Vieira, and Antonio AF Loureiro. Gedar: geographic and opportunistic routing protocol with depth adjustment for mobile underwater sensor networks. In *2014 IEEE International Conference on communications (ICC)*, pages 251–256. IEEE, 2014.

[40] Narote Sandipann P, Bhujbal Pradnya N, Narote Abbhilasha S, and Dhane, Dhiraj M. A review of recent advances in lane detection and departure warning system. *Pattern Recognition*, 73:216–234, 2018.

[41] Geiger Andreas, Lenz Philip, Stiller Christoph, and Urtasun Raquel. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[42] Xiangjing An, Mo Wu, and Hangen He. A novel approach to provide lane departure warning using only one forward-looking camera. In *International Symposium on Collaborative Technologies and Systems*, pages 356–362, 2006.

[43] Azzedine Boukerche and Yonglin Ren. A secure mobile healthcare system using trust-based multicast scheme. *IEEE Journal on Selected Areas in Communications*, 27(4):387–399, 2009.

[44] Azzedine Boukerche, Kathia Regina Lemos Jucá, Joao Bosco Sobral, and Mirela Sechi Moretti Annoni Notare. An artificial immune based intrusion detection model for computer and telecommunication systems. *Parallel Computing*, 30(5-6):629–646, 2004.

[45] Azzedine Boukerche and Mirela Sechi M Annoni Notare. Behavior-based intrusion detection in mobile phone systems. *Journal of Parallel and Distributed Computing*, 62(9):1476–1490, 2002.

[46] Cristiano Rezende, Azzedine Boukerche, Heitor S Ramos, and Antonio AF Loureiro. A reactive and scalable unicast solution for video streaming over vanets. *IEEE Transactions on Computers*, 64(3):614–626, 2014.

[47] Azzedine Boukerche, Xuzhen Cheng, and Joseph Linus. A performance evaluation of a novel energy-aware data-centric routing algorithm in wireless sensor networks. *Wireless Networks*, 11(5):619–635, 2005.

[48] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Analysis of a randomized congestion control scheme with dsdv routing in ad hoc wireless networks. *Journal of Parallel and Distributed Computing*, 61(7):967–995, 2001.

[49] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo Freire Nakamura, and Antonio AF Loureiro. Dv-loc: a scalable localization protocol using voronoi diagrams for wireless sensor networks. *IEEE Wireless Communications*, 16(2):50–55, 2009.

[50] Azzedine Boukerche, Xin Fei, and Regina B Araujo. An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange. *Computer Communications*, 30(14-15):2708–2720, 2007.

[51] Kuang Xiao, Wu Jiangtao, Chen Kaijuan, Zhao Zeang, Ding Zhen, Hu Fengjingyang, Fang Daining, and Qi H Jerry. Grayscale digital light processing 3d printing for highly functionally graded materials. *Science advances*, 5(5):eaav5790, 2019.

[52] Du Hao, He Shengfeng, Sheng Bin, Ma Lizhuang, and Lau Rynson WH. Saliency-guided color-to-gray conversion using region-based optimization. *IEEE Transactions on Image Processing*, 24(1):434–443, 2014.

[53] Azzedine Boukerche and Carl Tropper. A static partitioning and mapping algorithm for conservative parallel simulations. In *Proceedings of the eighth workshop on Parallel and distributed simulation*, pages 164–172, 1994.

[54] Azzedine Boukerche, Renato B Machado, Kathia RL Jucá, João Bosco M Sobral, and Mirela SMA Notare. An agent based and biological inspired real-time intrusion detection and security model for computer network operations. *Computer Communications*, 30(13):2649–2660, 2007.

[55] Azzedine Boukerche, Khalil El-Khatib, Li Xu, and Larry Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *computer communications*, 28(10):1193–1203, 2005.

[56] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. A distributed fault identification protocol for wireless and mobile ad hoc networks. *Journal of parallel and distributed computing*, 68(3):321–335, 2008.

[57] Azzedine Boukerche and E Robson. Vehicular cloud computing: Architectures, applications, and mobility. *Computer networks*, 135:171–189, 2018.

[58] Bian Wenjiao, Wakahara Toru, Wu Tao, Tang He, and Lin Jirui. Binarization of color character strings in scene images using deep neural network. In *Digital Image Computing: Techniques and Applications*, pages 1–6, 2018.

[59] Azzedine Boukerche and Xin Fei. A coverage-preserving scheme for wireless sensor network with irregular sensing range. *Ad hoc networks*, 5(8):1303–1316, 2007.

[60] Azzedine Boukerche and Xu Li. An agent-based trust and reputation management scheme for wireless sensor networks. In *GLOBECOM'05. IEEE Global Telecommunications Conference, 2005.*, volume 3, pages 5–pp. IEEE, 2005.

[61] Samer Samarah, Muhannad Al-Hajri, and Azzedine Boukerche. A predictive energy-efficient technique to support object-tracking sensor networks. *IEEE Transactions on Vehicular Technology*, 60(2):656–663, 2010.

[62] Rodolfo WL Coutinho, Azzedine Boukerche, Luiz FM Vieira, and Antonio AF Loureiro. Design guidelines for opportunistic routing in underwater networks. *IEEE Communications Magazine*, 54(2):40–48, 2016.

[63] Azzedine Boukerche, Sajal K Das, and Alessandro Fabbri. Swimnet: a scalable parallel simulation testbed for wireless and mobile networks. *Wireless Networks*, 7(5):467–486, 2001.

[64] Richard WN Pazzi and Azzedine Boukerche. Mobile data collector strategy for delay-sensitive applications over wireless sensor networks. *Computer Communications*, 31(5):1028–1039, 2008.

[65] Yonglin Ren and Azzedine Boukerche. Modeling and managing the trust for wireless and mobile ad hoc networks. In *2008 IEEE International Conference on Communications*, pages 2129–2133. IEEE, 2008.

[66] Azzedine Boukerche and Xin Fei. A voronoi approach for coverage protocols in wireless sensor networks. In *IEEE GLOBECOM 2007-IEEE Global Telecommunications Conference*, pages 5190–5194. IEEE, 2007.

[67] Rodolfo Bezerra Batista, Azzedine Boukerche, and Alba Cristina Magalhaes Alves de Melo. A parallel strategy for biological sequence alignment in restricted memory space. *Journal of Parallel and Distributed Computing*, 68(4):548–561, 2008.

[68] Azzedine Boukerche, Amber Roy, and Neville Thomas. Dynamic grid-based multicast group assignment in data distribution management. In *Proceedings Fourth IEEE International Workshop on Distributed Simulation and Real-Time Applications (DS-RT 2000)*, pages 47–54. IEEE, 2000.

[69] Azzedine Boukerche and Carl Tropper. A distributed graph algorithm for the detection of local cycles and knots. *IEEE Transactions on Parallel and Distributed Systems*, 9(8):748–757, 1998.

[70] Rodolfo WL Coutinho, Azzedine Boukerche, Luiz FM Vieira, and Antonio AF Loureiro. Underwater wireless sensor networks: A new challenge for topology control–based systems. *ACM Computing Surveys (CSUR)*, 51(1):1–36, 2018.

[71] Azzedine Boukerche, Anahit Martirosyan, and Richard Pazzi. An inter-cluster communication based energy aware and fault tolerant protocol for wireless sensor networks. *Mobile Networks and Applications*, 13(6):614–626, 2008.

[72] Azzedine Boukerche and Sotiris Nikoletseas. Protocols for data propagation in wireless sensor networks. In *Wireless communications systems and networks*, pages 23–51. Springer, 2004.

[73] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. An efficient synchronization scheme of multimedia streams in wireless and mobile systems. *IEEE transactions on Parallel and Distributed Systems*, 13(9):911–923, 2002.

[74] Azzedine Boukerche, Sajal K Das, Alessandro Fabbri, and Oktay Yildiz. Exploiting model independence for parallel pcs network simulation. In *Proceedings Thirteenth Workshop on Parallel and Distributed Simulation. PADS 99.(Cat. No. PR00155)*, pages 166–173. IEEE, 1999.

[75] Azzedine Boukerche and Steve Rogers. Performance of gzrp ad hoc routing protocol. *Journal of Interconnection Networks*, 2(01):31–48, 2001.

[76] Sumalee Agachai and Ho Hung Wai. Smarter and more connected: Future intelligent transportation system. *IATSS Research*, 42(2):67–71, 2018.

[77] Ahmad Abu Shanab and Taghi Khoshgoftaar. Filter-based subset selection for easy, moderate, and hard bioinformatics data. In *IEEE International Conference on Information Reuse and Integration*, pages 372–377, 2018.

[78] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Performance analysis of a distributed comparison-based self-diagnosis protocol for wireless ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems*, pages 165–172, 2006.

[79] Azzedine Boukerche, Noura Aljeri, Kaouther Abrougui, and Yan Wang. Towards a secure hybrid adaptive gateway discovery mechanism for intelligent transportation systems. *Security and Communication Networks*, 9(17):4027–4047, 2016.

[80] Y. Xu, X. Shan, B. Y. Chen, C. Chi, Z. F. Lu, and Y. Q. Wang. A lane detection method combined fuzzy control with ransac algorithm. In *International Conference on Power Electronics Systems and Applications - Smart Mobility, Power Transfer Security*, pages 1–6, 2017.

[81] Noura Aljeri and Azzedine Boukerche. A dynamic map discovery and selection scheme for predictive hierarchical mipv6 in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(1):793–806, 2019.

[82] Noura Aljeri and Azzedine Boukerche. A probabilistic neural network-based road side unit prediction scheme for autonomous driving. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[83] Yassine Akhiat, Mohamed Chahhou, and Ahmed Zinedine. Feature selection based on graph representation. In *IEEE International Congress on Information Science and Technology*, pages 232–237, 2018.

[84] Tat-Jun Chin and David Suter. Incremental kernel pca for efficient non-linear feature extraction. In *Proceedings of the British Machine Vision Conference*, volume 3, pages 939–948. Citeseer, 2006.

[85] Fang Leyuan, He Nanjun, Li Shutao, Plaza Antonio J, and Plaza Javier. A new spatial–spectral feature extraction method for hyperspectral images using local covariance matrix representation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3534–3546, 2018.

[86] Huifeng Wang, Yunfei Wang, Xiangmo Zhao, Guiping Wang, He Huang, and Jiajia Zhang. Lane detection of curving road for structural highway with straight-curve model on vision. *IEEE Transactions on Vehicular Technology*, 68(6):5321–5330, 2019.

[87] Qiao Tong, Yang Zhijing, Ren Jinchang, Yuen Peter, Zhao Huimin, Sun Genyun, Marshall Stephen, and Benediktsson Jon Atli. Joint bilateral filtering and spectral similarity-based sparse representation: a generic framework for effective feature extraction and data classification in hyperspectral imaging. *Pattern Recognition*, 77:316–328, 2018.

[88] Deepika Rani Bansaland Brij Kishore. Feature selection in support vector machines for outlier detection. In *International Conference on Electronics, Communication and Aerospace Technology*, pages 112–115, 2018.

[89] Dima El Zein and Ali Kalakech. Feature selection for android keystroke dynamics. In *International Arab Conference on Information Technology*, pages 1–6, 2018.

[90] N. Gopika and A. M. Kowshalaya M. E. . Correlation based feature selection algorithm for machine learning. In *International Conference on Communication and Electronics Systems*, pages 692–695, 2018.

[91] Sen Jia and Jiayue Zhuang. 3D-Gabor-based feature selection via enhanced fast density-peak-based clustering. In *International Workshop on Earth Observation and Remote Sensing Applications*, pages 1–5, 2018.

[92] N K Suchetha, Anupama Nikhil, and P Hrudya. Comparing the wrapper feature selection evaluators on twitter sentiment classification. In *International Conference on Computational Intelligence in Data Science*, pages 1–6, 2019.

[93] Atsushi Kawamura and Basabi Chakraborty. A new filter evaluation function for feature subset selection with evolutionary computation. In *International Conference on Awareness Science and Technology*, pages 101–105, 2018.

[94] Koda Satoru, Melgani Farid, and Nishii Ryuei. Unsupervised spectral-spatial feature extraction with generalized autoencoder for hyperspectral imagery. *Geoscience and Remote Sensing Letters*, 2019.

[95] Noura Aljeri and Azzedine Boukerche. An optimized link duration-based mobility management scheme for connected vehicular networks. In *Proceedings of the 16th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pages 7–14, 2019.

[96] Noura Aljeri and Azzedine Boukerche. A predictive collision detection protocol using vehicular networks. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–5. IEEE, 2017.

[97] Noura Aljeri and Azzedine Boukerche. Mobility management in 5g-enabled vehicular networks: Models, protocols, and classification. *ACM Computing Surveys (CSUR)*, 53(5):1–35, 2020.

[98] Noura Aljeri and Azzedine Boukerche. A novel online machine learning based rsu prediction scheme for intelligent vehicular networks. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE, 2019.

[99] Noura Aljeri and Azzedine Boukerche. An adaptive traffic-flow based controller deployment scheme for software-defined vehicular networks. In *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 191–198, 2020.

[100] Ali Moghimi, Ce Yang, and Peter M. Marchetto. Ensemble feature selection for plant phenotyping: A journey from hyperspectral to multispectral imaging. *IEEE Access*, 6:56870–56884, 2018.

[101] Xiangchenyang Su and Fang Liu. A survey for study of feature selection based on mutual information. In *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2018.

[102] M.S. Suresh Sumi and Athi Narayanan. Improving classification accuracy using combined filter+wrapper feature selection technique. In *IEEE International Conference on Electrical, Computer and Communication Technologies*, pages 1–6, 2019.

[103] Noura Aljeri and Azzedine Boukerche. A performance evaluation of time-series mobility prediction for connected vehicular networks. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 127–131, 2020.

[104] Noura Aljeri and Azzedine Boukerche. Advice-loc: An adaptive vehicle-centric location management scheme for intelligent connected cars. *Ad Hoc Networks*, 107:102223, 2020.

[105] Umar Ozgunalp. Combination of the symmetrical local threshold and the sobel edge detector for lane feature extraction. In *IEEE International Conference on Computational Intelligence and Communication Networks*, pages 24–28, 2017.

[106] Gonzalez Claudia I, Melin Patricia, Castro Juan R, Mendoza Olivia, and Castillo Oscar. An improved sobel edge detection method based on generalized type-2 fuzzy logic. *Soft Computing*, 20(2):773–784, 2016.

[107] Xuqin Yan and Yanqiang Li. A method of lane edge detection based on canny algorithm. In *Chinese Automation Congress*, pages 2120–2124, 2017.

[108] Yongfu Li and Zhanji Yang. Progressive probabilistic hough transform based nighttime lane line detection for micro-traffic road. In *Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems*, pages 1276–1281, 2018.

[109] Tianpeng Huang, Zhikai Wang, Xi Dai, Deqing Huang, and Hu Su. Unstructured lane identification based on hough transform and improved region growing. In *Chinese Control Conference*, pages 7612–7617, 2019.

[110] Houqiang Li, Zhaohui Zhang, Xiaoyan Zhao, and Mengzhong He. Two-stage hough transform algorithm for lane detection system based on tms320dm6437. In *IEEE International Conference on Imaging Systems and Techniques*, pages 1–5, 2017.

[111] Son Jongin, Yoo Hunjae, Kim Sanghoon, and Sohn Kwanghoon. Real-time illumination invariant lane detection for lane departure warning system. *Expert Systems with Applications*, 42(4):1816–1824, 2015.

[112] Peng Sun, Noura AlJeri, and Azzedine Boukerche. A novel passive road side unit detection scheme in vehicular networks. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–5. IEEE, 2017.

[113] Noura AlJeri and Azzedine Boukerche. An efficient movement-based handover prediction scheme for hierarchical mobile ipv6 in vanets. In *Proceedings of the 15th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, pages 47–54, 2018.

[114] Noura Aljeri, Kaouther Abrougui, Mohammed Almulla, and Azzedine Boukerche. A reliable quality of service aware fault tolerant gateway discovery protocol for vehicular networks. *Wireless Communications and Mobile Computing*, 15(10):1485–1495, 2015.

[115] Peng Sun, Noura AlJeri, and Azzedine Boukerche. Dacon: A novel traffic prediction and data-highway-assisted content delivery protocol for intelligent vehicular networks. *IEEE Transactions on Sustainable Computing*, 5(4):501–513, 2020.

[116] Noura Aljeri and Azzedine Boukerche. A two-tier machine learning-based handover management scheme for intelligent vehicular networks. *Ad Hoc Networks*, 94:101930, 2019.

[117] Peng Sun, Noura AlJeri, and Azzedine Boukerche. An energy-efficient proactive handover scheme for vehicular networks based on passive rsu detection. *IEEE Transactions on Sustainable Computing*, 5(1):37–47, 2018.

[118] Azzedine Boukerche, Alexander Magnano, and Noura Aljeri. Mobile ip handover for vehicular networks: Methods, models, and classifications. *ACM Computing Surveys (CSUR)*, 49(4):1–34, 2017.

[119] Ju Han Yoo, Seong-Whan Lee, Sung-Kee Park, and Dong Hwan Kim. A robust lane detection method based on vanishing point estimation using the relevance of line segments. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3254–3266, 2017.

[120] Noura Aljeri, Mohammed Almulla, and Azzedine Boukerche. An efficient fault detection and diagnosis protocolfor vehicular networks. In *Proceedings of the third ACM international symposium on Design and analysis of intelligent vehicular networks and applications*, pages 23–30, 2013.

[121] Noura Aljeri and Azzedine Boukerche. An efficient handover trigger scheme for vehicular networks using recurrent neural networks. In *Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, pages 85–91, 2019.

[122] Noura Aljeri and Azzedine Boukerche. Fog-enabled vehicular networks: A new challenge for mobility management. *Internet Technology Letters*, 3(6):e141, 2020.

[123] Azzedine Boukerche and Samer Samarah. An efficient data extraction mechanism for mining association rules from wireless sensor networks. In *2007 IEEE International Conference on Communications*, pages 3936–3941. IEEE, 2007.

[124] Maram Bani Younes and Azzedine Boukerche. An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wireless Networks*, 24(7):2451–2463, 2018.

[125] Azzedine Boukerche, Jan M Correa, Alba Cristina Magalhaes Melo, and Ricardo P Jacobi. A hardware accelerator for the fast retrieval of dialign biological sequence alignments in linear space. *IEEE Transactions on Computers*, 59(6):808–821, 2010.

[126] Ahmed Mostefaoui, Mahmoud Melkemi, and Azzedine Boukerche. Localized routing approach to bypass holes in wireless sensor networks. *IEEE transactions on computers*, 63(12):3053–3065, 2013.

[127] Jun-Juh Yan, Hang-Hong Kuo, Ying-Fan Lin, and Teh-Lu Liao. Real-time driver drowsiness detection system based on perclos and grayscale image processing. In *International Symposium on Computer, Consumer and Control*, pages 243–246, 2016.

[128] Peng Sun, Noura AlJeri, and Azzedine Boukerche. A fast vehicular traffic flow prediction scheme based on fourier and wavelet analysis. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[129] Peng Sun, Noura Aljeri, and Azzedine Boukerche. Machine learning-based models for real-time traffic flow prediction in vehicular networks. *IEEE Network*, 34(3):178–185, 2020.

[130] Noura Aljeri and Azzedine Boukerche. Movement prediction models for vehicular networks: an empirical analysis. *Wireless Networks*, 25(4):1505–1518, 2019.

[131] Noura Aljeri, Kaouther Abrougui, Mohammed Almulla, and Azzedine Boukerche. A performance evaluation of load balancing and qos-aware gateway discovery protocol for vanets. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, pages 90–94. IEEE, 2013.

[132] Noura Aljeri and Azzedine Boukerche. Performance evaluation of movement prediction techniques for vehicular networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2017.

[133] Noura Aljeri and Azzedine Boukerche. Mobility and handoff management in connected vehicular networks. In *Proceedings of the 16th ACM International Symposium on Mobility Management and Wireless Access*, pages 82–88, 2018.

[134] Azzedine Boukerche, Anis Zarrad, and Regina Araujo. A cross-layer approach-based gnutella for collaborative virtual environments over mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 21(7):911–924, 2009.

[135] Azzedine Boukerche, Nathan J McGraw, Caron Dzermajko, and Kaiyuan Lu. Grid-filtered region-based data distribution management in large-scale distributed simulation systems. In *38th Annual Simulation Symposium*, pages 259–266. IEEE, 2005.

[136] Azzedine Boukerche, Richard WN Pazzi, and Jing Feng. An end-to-end virtual environment streaming technique for thin mobile devices over heterogeneous networks. *Computer Communications*, 31(11):2716–2725, 2008.

[137] Abdelhamid Mammeri, Azzedine Boukerche, and Zongzhi Tang. A real-time lane marking localization, tracking and communication system. *Computer Communications*, 73:132–143, 2016.

[138] Azzedine Boukerche, Yan Du, Jing Feng, and Richard Pazzi. A reliable synchronous transport protocol for wireless image sensor networks. In *2008 IEEE Symposium on Computers and Communications*, pages 1083–1089. IEEE, 2008.

[139] Renfei Wang, Cristiano Rezende, Heitor S Ramos, Richard W Pazzi, Azzedine Boukerche, and Antonio AF Loureiro. Liaithon: A location-aware multipath video streaming scheme for urban vehicular networks. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 000436–000441. IEEE, 2012.

[140] Gooch Amy A, Olsen Sven C, Tumblin Jack, and Gooch Bruce. Color2gray: salience-preserving color removal. In *ACM Transactions on Graphics*, volume 24, pages 634–639. ACM, 2005.

[141] Grundland Mark and Dodgson Neil A. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recognition*, 40(11):2891–2896, 2007.

[142] Baraq Ghaleb, Ahmed Y Al-Dubai, Elias Ekonomou, Ayoub Alsarhan, Youssef Nasser, Lewis M Mackenzie, and Azzedine Boukerche. A survey of limitations and enhancements of the ipv6 routing protocol for low-power and lossy networks: A focus on core operations. *IEEE Communications Surveys & Tutorials*, 21(2):1607–1635, 2018.

[143] Hadi Habibzadeh, Tolga Soyata, Burak Kantarci, Azzedine Boukerche, and Cem Kaptan. Sensing, communication and security planes: A new challenge for a smart city system design. *Computer Networks*, 144:163–200, 2018.

[144] Zhou Mingqi, Sheng Bing, and Ma Lizhuang. Saliency preserving decolorization. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, 2014.

[145] Clayson Celes, Fabrício A Silva, Azzedine Boukerche, Rossana Maria de Castro Andrade, and Antonio AF Loureiro. Improving vanet simulation with calibrated vehicular mobility traces. *IEEE Transactions on Mobile Computing*, 16(12):3376–3389, 2017.

[146] Abdul Jabbar Siddiqui, Abdelhamid Mammeri, and Azzedine Boukerche. Real-time vehicle make and model recognition based on a bag of surf features. *IEEE Transactions on Intelligent Transportation Systems*, 17(11):3205–3219, 2016.

[147] Azzedine Boukerche and Caron Dzermajko. Performance evaluation of data distribution management strategies. *Concurrency and Computation: Practice and Experience*, 16(15):1545–1573, 2004.

[148] Nafchi Hossein Ziaei, Shahkolaei Atena, Hedjam Rachid, and Cheriet Mohamed. Corrc2g: Color to gray conversion by correlation. *IEEE Signal Processing Letters*, 24(11):1651–1655, 2017.

[149] Azzedine Boukerche and Yonglin Ren. A security management scheme using a novel computational reputation model for wireless and mobile ad hoc networks. In *Proceedings of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 88–95, 2008.

[150] Azzedine Boukerche, Sungbum Hong, and Tom Jacob. A distributed algorithm for dynamic channel allocation. *Mobile Networks and Applications*, 7(2):115–126, 2002.

[151] Bala Raja and Eschbach Reiner. Spatial color-to-grayscale transform preserving chrominance edge information. In *Color and Imaging Conference Final Program and Proceedings*, volume 2004, pages 82–86. Society for Imaging Science and Technology, 2004.

[152] Ancuti Codruta Orniana, Ancuti Cosmin, and Bekaert Philippe. Decolorizing images for robust matching. In *IEEE International Conference on Image Processing*, pages 149–152. IEEE, 2010.

[153] Lowe David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[154] Zhu Wei, Hu Ruizhen, and Liu Ligang. Grey conversion via perceived-contrast. *The Visual Computer*, 30(3):299–309, 2014.

[155] Rodolfo WL Coutinho, Azzedine Boukerche, Luiz FM Vieira, and Antonio AF Loureiro. A novel void node recovery paradigm for long-term underwater sensor networks. *Ad Hoc Networks*, 34:144–156, 2015.

[156] Elie El Ajaltouni, Azzedine Boukerche, and Ming Zhang. An efficient dynamic load balancing scheme for distributed simulations on a grid infrastructure. In *2008 12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, pages 61–68. IEEE, 2008.

[157] Vo Garret D and Park, Chiwoo. Robust regression for image binarization under heavy noise and nonuniform background. *Pattern Recognition*, 81:224–239, 2018.

[158] Hasan Mohammad Kamrul, Majumder Md Mujibur Rahman, Sarker Orvila, and Matin Abdul. Local contrast based thresholding for document binarization. In *International Conference on Electrical Engineering and Information Communication Technology*, pages 204–209, 2018.

[159] Horacio ABF Oliveira, Eduardo F Nakamura, Antonio AF Loureiro, and Azzedine Boukerche. Error analysis of localization systems for sensor networks. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 71–78, 2005.

[160] Azzedine Boukerche and Damla Turgut. Secure time synchronization protocols for wireless sensor networks. *IEEE Wireless Communications*, 14(5):64–69, 2007.

[161] Fabricio A Silva, Azzedine Boukerche, Thais RM Braga Silva, Linnyer B Ruiz, Eduardo Cerqueira, and Antonio AF Loureiro. Vehicular networks: A new challenge for content-delivery-based applications. *ACM Computing Surveys (CSUR)*, 49(1):1–29, 2016.

[162] Cristiano Rezende, Abdelhamid Mammeri, Azzedine Boukerche, and Antonio AF Loureiro. A receiver-based video dissemination solution for vehicular networks with content transmissions decoupled from relay node selection. *Ad Hoc Networks*, 17:1–17, 2014.

[163] Maram Bani Younes and Azzedine Boukerche. A performance evaluation of an efficient traffic congestion detection protocol (ecode) for intelligent transportation systems. *Ad Hoc Networks*, 24:317–336, 2015.

[164] Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range. *Computer communications*, 29(4):477–489, 2006.

[165] Anahit Martirosyan, Azzedine Boukerche, and Richard Pazzi. A taxonomy of cluster-based routing protocols for wireless sensor networks. In *2008 International Symposium on Parallel Architectures, Algorithms, and Networks (i-span 2008)*, pages 247–253. IEEE, 2008.

[166] Athanasios Bamis, Azzedine Boukerche, Ioannis Chatzigiannakis, and Sotiris Nikoletseas. A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks. *Computer Networks*, 52(1):130–154, 2008.

[167] Azriel Rosenfeld and Pilar De La Torre. Histogram concavity analysis as an aid in threshold selection. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(2):231–235, 1983.

[168] T. Pavlidis. Threshold selection using second derivatives of the gray scale image. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 274–277. IEEE, 1993.

[169] Garg Naresh. Binarization techniques used for grey scale images. *International Journal of Computer Applications*, 71(1), 2013.

[170] Azzedine Boukerche and Amber Roy. Dynamic grid-based approach to data distribution management. *Journal of Parallel and Distributed Computing*, 62(3):366–392, 2002.

[171] Richard W Pazzi and Azzedine Boukerche. Propane: A progressive panorama streaming protocol to support interactive 3d virtual environment exploration on graphics-constrained devices. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(1):1–22, 2014.

[172] Mourad Elhadef, Azzedine Boukerche, and Hisham Elkadiki. Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols. In *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, pages 18–27, 2006.

[173] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[174] Kim Hye Min, Lee Seung Hwan, Lee Chungkeun, Ha Jong-Won, and Yoon Young-Ro. Automatic lumen contour detection in intravascular oct images using otsu binarization and intensity curve. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 178–181. IEEE, 2014.

[175] Rafael Guillermo Gonzalez Acuña, Junli Tao, and Reinhard Klette. Generalization of otsu's binarization into recursive colour image segmentation. In *International Conference on Image and Vision Computing New Zealand, Auckland*, pages 1–6, 2015.

[176] Cunzhao Shi, Yanna Wang, Baihua Xiao, and Chunheng Wang. Otsu guided adaptive binarization of captcha image using gamma correction. In *International Conference on Pattern Recognition*, pages 3962–3967, 2016.

[177] Azzedine Boukerche, Cristiano Rezende, and Richard W Pazzi. Improving neighbor localization in vehicular ad hoc networks to avoid overhead from periodic messages. In *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pages 1–6. IEEE, 2009.

[178] Azzedine Boukerche, Richard Werner Nelem Pazzi, and Regina B Araujo. Hpeq a hierarchical periodic, event-driven and query-based wireless sensor network protocol.

In *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l*, pages 560–567. IEEE, 2005.

[179] Renê Oliveira, Carlos Montez, Azzedine Boukerche, and Michelle S Wangham. Reliable data dissemination protocol for vanet traffic safety applications. *Ad Hoc Networks*, 63:30–44, 2017.

[180] Horacio ABF Oliveira, Azzedine Boukerche, Eduardo F Nakamura, and Antonio AF Loureiro. Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm. *Performance Evaluation*, 66(3-5):209–222, 2009.

[181] Kaouther Abrougui, Azzedine Boukerche, and Richard Werner Nelem Pazzi. Design and evaluation of context-aware and location-based service discovery protocols for vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):717–735, 2011.

[182] Zhenxia Zhang, Richard W Pazzi, and Azzedine Boukerche. A mobility management scheme for wireless mesh networks based on a hybrid routing protocol. *Computer Networks*, 54(4):558–572, 2010.

[183] Robson E De Grande and Azzedine Boukerche. Dynamic balancing of communication and computation load for hla-based simulations on large-scale distributed systems. *Journal of Parallel and Distributed Computing*, 71(1):40–52, 2011.

[184] Khurshid Khurram, Siddiqi Imran, Faure Claudie, and Vincent Nicole. Comparison of niblack inspired binarization methods for ancient documents. In *Document Recognition and Retrieval XVI*, volume 7247, page 72470U. Society of Photo-Optical Instrumentation Engineers, 2009.

[185] Samorodova OA and Samorodov AV. Fast implementation of the niblack binarization algorithm for microscope image segmentation. *Pattern Recognition and Image Analysis volume*, 26(3):548–551, 2016.

[186] Michalak Hubert and Okarma Krzysztof. Improvement of image binarization methods using image preprocessing with local entropy filtering for alphanumerical character recognition purposes. *Entropy*, 21:562, 2019.

[187] Sauvola Jaakko and Pietikäinen Matti. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[188] Yu He and Yang Yang. An improved sauvola approach on QR code image binarization. In *IEEE International Conference on Advanced Infocomm Technology*, pages 6–10, 2019.

[189] Najafi MH and Salehi ME. A fast fault-tolerant architecture for sauvola local image thresholding algorithm using stochastic computing. *IEEE Transactions on Very Large Scale Integration Systems*, 2015.

[190] Azzedine Boukerche. A simulation based study of on-demand routing protocols for ad hoc wireless networks. In *Proceedings. 34th Annual Simulation Symposium*, pages 85–92. IEEE, 2001.

[191] Hadjadj Zineb and Meziane Abdelkrim. Binarization of document images with various object sizes. In *International Workshop on Arabic Script Analysis and Recognition*, pages 21–25. IEEE, 2017.

[192] Mustafa Wan Azani and Kader Mohamed Mydin M Abdul. Binarization of document image using optimum threshold modification. In *Journal of Physics: Conference Series*, volume 1019, page 012022, 2018.

[193] Nandy Mahua and Saha Satadal. An analytical study of different document image binarization methods. *IEEE National Conference on Computing and Communication Systems*, 2015.

[194] Talab Ahmed Mahgoub Ahmed and Junfei Wang and others. An enhanced bernsen algorithm approaches for vehicle logo detection. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 7(4):203–210, 2014.

[195] Sergio Correia, Azzedine Boukerche, and Rodolfo I Meneguette. An architecture for hierarchical software-defined vehicular networks. *IEEE Communications Magazine*, 55(7):80–86, 2017.

[196] Amir Darehshoorzadeh and Azzedine Boukerche. Underwater sensor networks: A new challenge for opportunistic routing protocols. *IEEE Communications Magazine*, 53(11):98–107, 2015.

[197] Yang Weibin, Fang Bin, and Tang Yuan Yan. Fast and accurate vanishing point detection and its application in inverse perspective mapping of structured road. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(5):755–766, 2016.

[198] Adamshuk Rodrigo, Carvalho David, Neme João HZ, Margraf Erick, Okida Sergio, Tusset Angelo, Santos Max M, Amaral Rodrigo, Ventura Artur, and Carvalho, Saulo. On the applicability of inverse perspective mapping for the forward distance estimation based on the hsv colormap. In *IEEE International Conference on Industrial Technology*, pages 1036–1041, 2017.

[199] Kim Youngseok and Kum Dongsuk. Deep learning based vehicle position and orientation estimation via inverse perspective mapping image. In *Intelligent Vehicles Symposium*, pages 317–323. IEEE, 2019.

[200] Yiming Tian, Xitai Wang, Peng Yang, Jie Wang, and Jie Zhang. A single accelerometer-based robust human activity recognition via wavelet features and ensemble feature selection. In *International Conference on Automation and Computing*, pages 1–6, 2018.

[201] G. S. Thejas, Sajal Raj Joshi, S. S. Iyengar, N. R. Sunitha, and Prajwal Badrinath. Mini-batch normalized mutual information: A hybrid feature selection method. *IEEE Access*, 7:116875–116885, 2019.

[202] Li Zhixi, He Yifan, Keel Stuart, Meng Wei, Chang Robert T, and He Mingguang. Efficacy of a deep learning system for detecting glaucomatous optic neuropathy based on color fundus photographs. *Ophthalmology*, 125(8):1199–1206, 2018.

[203] Getaneh Gezahegne Tiruneh and Aminah Robinson Fayek. Feature selection for construction organizational competencies impacting performance. In *IEEE International Conference on Fuzzy Systems*, pages 1–5, 2019.

[204] Shinde Sandhya R, Sabale Sonali, Kulkarni Siddhant, and Bhatia Deepti. Experiments on content based image classification using color feature extraction. In *International Conference on Communication, Information Computing Technology*, pages 1–6, 2015.

[205] Khamisan Norliana, Ghazali Kamarul Hawari, and Zin Aufa Huda Muhammad. A thermograph image extraction based on color features for induction motor bearing fault diagnosis monitoring. *ARPN Journal of Engineering and Applied Sciences*, 10(22):17095–17101, 2015.

[206] Bagri Neelima and Johari Punit Kumar. A comparative study on feature extraction using texture and shape for content based image retrieval. *International Journal of Advanced Science and Technology*, 80(4):41–52, 2015.

[207] Wei Li and Hong-ying Dai. Real-time road congestion detection based on image texture analysis. *Procedia Engineering*, 137:196–201, 2016.

[208] Vidyarthi Ankit and Mittal Namita. Texture based feature extraction method for classification of brain tumor mri. *Journal of Intelligent & Fuzzy Systems*, 32(4):2807–2818, 2017.

[209] Humeau-Heurtier Anne. Texture feature extraction methods: A survey. *IEEE Access*, 7:8975–9000, 2019.

[210] Aytuğ Onan. Ensemble learning based feature selection with an application to text classification. In *IEEE Signal Processing and Communications Applications Conference*, pages 1–4, 2018.

[211] Janvier Omar Sinayobye, Kyanda Swaib Kaawaase, Fred N. Kiwanuka, and Richard Musabe. Hybrid model of correlation based filter feature selection and machine learning classifiers applied on smart meter data set. In *IEEE/ACM Symposium on Software Engineering in Africa*, pages 1–10, 2019.

[212] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. *IEEE Transactions on Image Processing*, 28(11):5596–5609, 2019.

[213] Alper Kursat Uysal. On two-stage feature selection methods for text classification. *IEEE Access*, 6:43233–43251, 2018.

[214] Afsaneh Mahanipour, Hossein Nezamabadi-pour, and Bahareh Nikpour. Using fuzzy-rough set feature selection for feature construction based on genetic programming. In *Conference on Swarm Intelligence and Evolutionary Computation*, pages 1–6. IEEE, 2018.

[215] Andrzej SeulEmail and authorKrzysztof Okarma. Classification of textures for autonomous cleaning robots based on the glcm and statistical local texture features. In *Computer Science On-line Conference*, pages 405–414, 2018.

[216] Lan Zeying and Liu Yang. Study on multi-scale window determination for glcm texture description in high-resolution remote sensing image geo-analysis supported by gis and domain knowledge. *ISPRS International Journal of Geo-Information*, 7(5):175, 2018.

[217] Mohanaiah P, Sathyanarayana P, and GuruKumar L. Image texture feature extraction using glcm approach. *International Journal of Scientific and Research Publications*, 3(5):1, 2013.

[218] Yujia Zhai, Wei Song, Xianjun Liu, Lizhen Liu, and Xinlei Zhao. A chi-square statistics based feature selection method in text classification. In *IEEE International Conference on Software Engineering and Service Science*, pages 160–163, 2018.

[219] Zhao Guodong and Zhou Zhiyong. Efficient linear feature extraction based on large margin nearest neighbor. *IEEE Access*, 7:78616–78624, 2019.

[220] Huilin Zheng, Hyun Woo Park, Dingkun Li, Kwang Ho Park, and Keun Ho Ryu. A hybrid feature selection approach for applying to patients with diabetes mellitus: Knhanes 2013-2015. In *NAFOSTED Conference on Information and Computer Science*, pages 110–113, 2018.

[221] Yu Yang and Jo Kang-Hyun. Lane detection based on color probability model and fuzzy clustering. In *International Conference on Graphic and Image Processing*, volume 10615, page 1061508, 2018.

[222] Yenİaydin Yasin and Schmidt Klaus Werner. A lane detection algorithm based on reliable lane markings. In *Signal Processing and Communications Applications Conference*, pages 1–4, 2018.

[223] Liu Wanjia, Gong Jun, and Ma Botan. Research on lane detection method with shadow interference. In *Chinese Control Conference*, pages 7704–7709. IEEE, 2019.

[224] Anwar Ul Haq, Defu Zhang, He Peng, and Sami Ur Rahman. Combining multiple feature-ranking techniques and clustering of variables for feature selection. *IEEE Access*, 7:151482–151492, 2019.

[225] Liying Yuan and Xue Xu. Adaptive image edge detection algorithm based on canny operator. In *International Conference on Advanced Information Technology and Sensor Application*, pages 28–31, 2015.

[226] Elsawy Amr, Abdel-Mottaleb Mohamed, and Shousha Mohamed Abou. Segmentation of corneal optical coherence tomography images using randomized hough transform. In *Society of Photo-Optical Instrumentation Engineers*, volume 10949, page 109490U, 2019.

[227] Wei Xianwen, Zhang Zhaojin, Chai Zongjun, and Feng Wei. Research on lane detection and tracking algorithm based on improved hough transform. In *IEEE International Conference of Intelligent Robotic and Control Engineering*, pages 275–279, 2018.

[228] Dong Qiu, Meng Weng, Hongtao Yang, Weibo Yu, and Keping Liu. Research on lane line detection method based on improved hough transform. In *IEEE Chinese Control And Decision Conference*, pages 5686–5690, 2019.

[229] Liu X T, Zou Y, and Guo, HW. An improved vision-based lane departure warning system under high speed driving condition. In *Journal of Physics: Conference Series*, volume 1267, page 012053, 2019.

[230] An Xiangjing, Shang Erke, Song Jinze, Li Jian, and He Hangen. Real-time lane departure warning system based on a single fpga. *EURASIP Journal on Image and Video Processing*, 2013(1):38, 2013.

[231] Yoo Ju Han, Lee, Seong-Whan, Park Sung-Kee, and Kim Dong Hwan. A robust lane detection method based on vanishing point estimation using the relevance of line segments. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3254–3266, 2017.

[232] Nasiri Sanaz, Amirfattahi Rassoul, Sadeghi Mohammad Taghi, and Mortaheb Sepehr. A new binarization method for high accuracy handwritten digit recognition of slabs in steel companies. In *Iranian Conference on Machine Vision and Image Processing*, pages 26–30, 2017.

[233] Chen Mingming, Tang Chen, Xu Min, and Lei Zhenkun. Binarization of optical fringe patterns with intensity inhomogeneities based on modified fcm algorithm. *Optics and Lasers in Engineering*, 123:14–19, 2019.

[234] Mehta Nihaal, Liu Keke, Alibhai A Yasin, Gendelman Isaac, Braun Phillip X, Ishibazawa Akihiro, Sorour Osama, Duker Jay S, and Waheed Nadia K. Impact of

binarization thresholding and brightness/contrast adjustment methodology on optical coherence tomography angiography image quantification. *American Journal of Ophthalmology*, 205:54–65, 2019.

[235] Ma Li-Yong, Hua Chun-Sheng, He Yu-Qing, Liu Yun-Jing, and Yan Pei-Lun. A lane detection technique based on adaptive threshold segmentation of lane gradient image. In *Annual International Conference on Network and Information Systems for Computers*, pages 182–186, 2018.

[236] Mochamad Vicky Ghani Aziz, Ary Setijadi Prihatmanto, and Hilwadi Hindersah. Implementation of lane detection algorithm for self-driving car on toll road cipularang using python language. In *International Conference on Electric Vehicular Technology*, pages 144–148, 2017.

[237] Soonhong Jung, Junsic Youn, and Sanghoon Sull. Efficient lane detection based on spatiotemporal images. *IEEE Transactions on Intelligent Transportation Systems*, 17(1):289–295, 2016.

[238] Xiaolong Liu, Zhidong Deng, Hongchao Lu, and Lele Cao. Benchmark for road marking detection: Dataset specification and performance baseline. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1–6, 2017.

[239] Jianhao Jiao, Rui Fan, Han Ma, and Ming Liu. Using dp towards a shortest path problem-related application. In *2019 International Conference on Robotics and Automation*, pages 8669–8675, 2019.

[240] J. Gao, Q. Wang, and Y. Yuan. Embedding structured contour and location prior in siamesed fully convolutional networks for road detection. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 219–224, 2017.

[241] Zhiyuan Zhao, Qi Wang, and Xuelong Li. Deep reinforcement learning based lane detection and localization. *Neurocomputing*, 413:328 – 338, 2020.

[242] Wenwei Wang, Zhipeng Zhang, Yue Gao, and Yiding Li. Lane detection using cnn-lstm with curve fitting for autonomous driving. *DEStech Transactions on Environment, Energy and Earth Sciences, ICEEE*, 10 2019.

[243] Yuhao Huang, Shitao Chen, Yu Chen, Zhiqiang Jian, and Nanning Zheng. Spatial-temproal based lane detection using deep learning. volume 519 of *IFIP Advances in*

Information and Communication Technology, pages 143–154. Springer International Publishing, 2018.

[244] Aviral Petwal and Malaya Kumar Hota. Computer vision based real time lane departure warning system. In *IEEE International Conference on Communication and Signal Processing*, pages 0580–0584, 2018.

[245] Ayhan Küçükmanisa, Orhan Akbulut, and Oğuzhan Urhan. Robust and real-time lane detection filter based on adaptive neuro-fuzzy inference system. *IET Image Processing*, 13(7):1181–1190, 2019.

[246] Pratik Bhope and Pinakini Samant. Use of image processing in lane departure warning system. In *International Conference for Convergence in Technology*, pages 1–4, 2018.

[247] Xing Ma, Chunyang Mu, Xiaolong Wang, and Jianyu Chen. Projective geometry model for lane departure warning system in webots. In *International Conference on Control, Automation and Robotics*, pages 689–695. IEEE, 2019.

[248] Ankita Kamble and Sandhya Potadar. Lane departure warning system for advanced drivers assistance. In *International Conference on Intelligent Computing and Control Systems*, pages 1775–1778, 2018.

[249] Wenshuo Wang, Ding Zhao, Wei Han, and Junqiang Xi. A learning-based approach for lane departure warning systems with a personalized driver model. *IEEE Transactions on Vehicular Technology*, 67(10):9145–9157, 2018.