

## Article

# Lane Line Detection and Object Scene Segmentation Using Otsu Thresholding and the Fast Hough Transform for Intelligent Vehicles in Complex Road Conditions

Muhammad Awais Javeed <sup>1,2,\*</sup>, Muhammad Arslan Ghaffar <sup>3</sup> , Muhammad Awais Ashraf <sup>1</sup>, Nimra Zubair <sup>1</sup>, Ahmed Sayed M. Metwally <sup>4</sup> , Elsayed M. Tag-Eldin <sup>5</sup> , Patrizia Bocchetta <sup>6</sup> , Muhammad Sufyan Javed <sup>7,\*</sup>  and Xingfang Jiang <sup>8,\*</sup>

- <sup>1</sup> School of Information Engineering, Chang'an University, Xi'an 710064, China  
<sup>2</sup> School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213164, China  
<sup>3</sup> School of Automobile, Chang'an University, Xi'an 710064, China  
<sup>4</sup> Department of Mathematics, College of Science, King Saud University, Riyadh 11451, Saudi Arabia  
<sup>5</sup> Faculty of Engineering and Technology, Future University in Egypt, New Cairo 11835, Egypt  
<sup>6</sup> Dipartimento di Ingegneria dell'Innovazione, Università del Salento, Via Monteroni, 73100 Lecce, Italy  
<sup>7</sup> School of Physical Science and Technology, Lanzhou University, Lanzhou 730000, China  
<sup>8</sup> School of Microelectronics and Control Engineering, Changzhou University, Changzhou 213164, China  
\* Correspondence: javedawais704@gmail.com (M.A.J.); safisabri@gmail.com (M.S.J.); xfjianj@cczu.edu.cn (J.X.)



**Citation:** Javeed, M.A.; Ghaffar, M.A.; Ashraf, M.A.; Zubair, N.; Metwally, A.S.M.; Tag-Eldin, E.M.; Bocchetta, P.; Javed, M.S.; Jiang, X. Lane Line Detection and Object Scene Segmentation Using Otsu Thresholding and the Fast Hough Transform for Intelligent Vehicles in Complex Road Conditions.

*Electronics* **2023**, *12*, 1079. <https://doi.org/10.3390/electronics12051079>

Academic Editors: David Valiente, Nuno Miguel Fonseca Ferreira and Olivier Sename

Received: 25 December 2022

Revised: 15 January 2023

Accepted: 17 February 2023

Published: 21 February 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Abstract:** An Otsu-threshold- and Canny-edge-detection-based fast Hough transform (FHT) approach to lane detection was proposed to improve the accuracy of lane detection for autonomous vehicle driving. During the last two decades, autonomous vehicles have become very popular, and it is constructive to avoid traffic accidents due to human mistakes. The new generation needs automatic vehicle intelligence. One of the essential functions of a cutting-edge automobile system is lane detection. This study recommended the idea of lane detection through improved (extended) Canny edge detection using a fast Hough transform. The Gaussian blur filter was used to smooth out the image and reduce noise, which could help to improve the edge detection accuracy. An edge detection operator known as the Sobel operator calculated the gradient of the image intensity to identify edges in an image using a convolutional kernel. These techniques were applied in the initial lane detection module to enhance the characteristics of the road lanes, making it easier to detect them in the image. The Hough transform was then used to identify the routes based on the mathematical relationship between the lanes and the vehicle. It did this by converting the image into a polar coordinate system and looking for lines within a specific range of contrasting points. This allowed the algorithm to distinguish between the lanes and other features in the image. After this, the Hough transform was used for lane detection, making it possible to distinguish between left and right lane marking detection extraction; the region of interest (ROI) must be extracted for traditional approaches to work effectively and easily. The proposed methodology was tested on several image sequences. The least-squares fitting in this region was then used to track the lane. The proposed system demonstrated high lane detection in experiments, demonstrating that the identification method performed well regarding reasoning speed and identification accuracy, which considered both accuracy and real-time processing and could satisfy the requirements of lane recognition for lightweight automatic driving systems.

**Keywords:** lane line detection; intelligent vehicles; Sobel operator; Gaussian blur filter; least-squares filter edge extraction; Canny algorithm; Otsu algorithm; fast Hough transform

## 1. Introduction

Previous decades saw a surge in interest in the study of safe and autonomous driving due to the prevalence of traffic accidents resulting from negligent or drowsy driving. Lane

detection is essential for accurate path planning for an autonomous vehicle. Recently, researchers have done much work on lane line detection. There are many methods for lane line detection, such as machine vision, convolutional neural networks (CNNs), and deep learning, as well as some conventional methods that use edge detection, the Hough transform (HT), Markov random fields (MRFs), the extended edge-linking algorithm (EELA), and the RANSAC algorithm to find the lane mark and lane line fitting. The convolutional techniques are a little slow and need a lot of knowledge to execute [1]. The Hough transform was created in the early 1960s; it gives fast and good results and solves many problems encountered while analyzing images, including correlation and a linear structuring feature. It provides quick and effective results and addresses numerous issues encountered during image processing.

Zhicheng Zhang worked on an HT lane recognition algorithm under difficult conditions [2]. The five parts of the calculation were a Gaussian blur, picture graying pre-processing, the DLD threshold (dark–light–dark threshold) technique, association filtering for edge separation, and the Hough transform. Dong Qiu and Meng Weng presented a path line detection technique that further refined the Hough change. Image complexity is decreased, and the processing time is shortened by using the wavelet's lifting procedure to retrieve the image's low-frequency wavelet coefficients. After this, the Canny operator is used to locate the image's border, and the edge data is used to select the threshold automatically [3]. In conclusion, the right lane lines are fitted using a linear regression method (LRM), and three restrictions are proposed from two angles and lane width perspectives using the HT.

Regarding recognizing lanes, edge detection techniques, such as the Sobel and Canny operators, are frequently employed. The Sobel operator was used to find the ROI in a greyscale image using the Otsu algorithm [4], and threshold segmentation and piecewise fitting were used to find the lane line. In the work of Li et al. [5], the Sobel operator and non-local maximum suppression (NLMS) were utilized to identify a significant number of potential lane lines in the ROI of an image. To finish the lane line extraction, it screened possible lines using a mix of line structure features and color features. Following lane line extraction through denoising and threshold segmentation, image ROI extraction was carried out using the Canny operator and the paired features of lane lines [6]. Canny operator edge detection was used for the initial processing [7]. Then, the Catmull-ROM spline control points were generated using the perspective principle, the Hough transform was merged with the maximum likelihood approach, and lane line extraction was completed. There is only one feature extraction pattern, yet all these methods share the characteristics of being fast, having a transparent mechanism, and being easily manipulated [8]. Despite some success with sound identification in specific situations, these systems are not robust and are easily thrown off by variations in the road environment. The requirements of lane line recognition are not met in real-world circumstances that are dynamic and complex.

Over the past few years, deep learning's theory and technology have flourished thanks to the proliferation of ever-faster computing power, leading to its increasing use in various contexts and, in particular, lane identification systems. For the foreseeable future, researchers will focus on deep learning as a development direction. The extended edge-linking algorithm (EELA) has directional edge gap closing, as suggested by Qing et al. [9]. The proposed method could be used to acquire a new edge. The Sobel edge operator presented by Mu and Ma [10] can be used in the adaptive ROI. However, even after edge identification, there are still some spurious edges. Any subsequent lane detection will be compromised due to these mistakes. Using an intelligent edge detection technique, Wang et al. proposed extracting features [11]. The program might adapt to challenging road conditions and achieve precise lane line fitting. Regarding roadside noise, Srivastava et al. (2014) suggested that more innovative edge detection might be helpful. The most popular and successful edge detection algorithms are variants of the Sobel and Canny edge operator (SCEO).

The HT detects the vertical straight lanes of interest. Images are pre-processed to pull out their edge features. Next, the image's focal area is isolated. The HT is applied to the region contour image when searching for a straight line in the image. A proposed partial Hough parameter space is used in the lane detection method, which was tested on various picture datasets. Methods for detecting lanes and edges are based on a partial HT proposed by P. Maya and Dr. C. Tharini [12] in which Canny edge detection and the Hough transform are used to fit the vanishing point. The ROI is divided by a vanishing point. The incline of the lane lines and the distance between the video's first and last frames are then considered.

The various techniques used in the literature to detect lane markings and generate lane departure warning signals are summarized in Table 1. The effectiveness of these approaches is evaluated in light of their performance under a variety of complex conditions, including the shadow effect, the occlusion of lane markings by other vehicles, arrows marked on the road surface, and various environmental factors (such as rain, fog, and sun glare).

**Table 1.** The various techniques used in the literature to detect lanes.

Authors	Methods	Advantages	Limitations
Xu and Li (2010) [13]	a: Straight-line model, median filtering, and histogram-based contrast enhancement. b: Sobel edge operator. c: Kalman filter and the Hough transform.	A failure judgment module is added to improve the algorithm's accuracy.	It is challenging to detect lanes in unstructured road conditions. In situations with shadows and water, lane detection fails.
Zhao et al. (2012) [14]	a: Model based on the spline. b: Kalman extension filter.	Spline model is straightforward, flexible, and robust. The first-time mix of spline-based path model and broadened Kalman filter for path-following.	Light impedance around evening time might degrade the presentation.
Obradovic et al. (2013) [15]	a: Set of fluffy places and smooth lines for street path detection. b: Modified edge ex-footing. c: Modified fluffy c-means grouping calculation (FLDetector).	Tried on PC-based stages and cell phones. Execution is autonomous of the accuracy of extraction of component focuses.	Execution time relies upon the places of fluffy focuses on a picture and the quantity of recognized smooth lines. Not appropriate for applications requiring countless element focuses, where Hough
Shang et al. (2013) [16]	a: Progression of 2D FIR filter + Gaussian noise decrease. b: Evaporating point-based steerable filter. c: Evaporating point-based Hough change. d: Kalman filters + takeoff cautioning unit executed on MicroBlaze Programming.	Fewer capacity prerequisites. Upgrade location ability.	In forcefully bent streets, variety in the place of the vanishing point seriously influences the path discovery results as follows: (1) Casting a ballot of pixels in lines does not collect at similar peaks; instead, it brings about a progression of discrete peaks. (2) The real vanishing point is not reasonable for the coordinate beginning.
Wu et al. (2014) [17]	a: Direct explanatory model. b: Canny edge locator + misleading edge end in light of slope direction and width of path mark + edge compensation. c: Local versatile thresholding + path marking verification given distance rules and following using a Kalman filter.	Distinguish path marks in complex situations such as shadow impact, night situations (streetlights, front light, and taillight of vehicle), and bent path markers.	Difficult to extricate path marks from the picture sequence that are genuinely debased by thick haze and the recorder runs at a shallow illumination level. Working continuously would be required for the proposed strategy regarding discovery rate and execution time.
Sohn et al. (2015) [18]	a: Vanishing point-based versatile ROI selection + invariance properties of path tone. b: Canny edge identification. c: Path bunching and lane fitting have given the most un-square strategy. d: Path offset and path mark heading point.	Handles various illumination conditions due to weather changes (rain, fog, etc.), streetlamps, and headlights of vehicles. Vanishing-point-based adaptive ROI selection reduces computational complexity.	Difficult-to-identify path checking in the accompanying areas of strength for circumstances reflection, path break, and obscure path marks.

Table 1. Cont.

Authors	Methods	Advantages	Limitations
Sohn et al. (2015) [18]	a: Vanishing point-based versatile ROI selection + invariance properties of path tone. b: Canny edge identification. c: Path bunching and lane fitting have given the most un-square strategy. d: Path offset and path mark heading point.	Handles various illumination conditions due to weather changes (rain, fog, etc.), streetlamps, and headlights of vehicles. Vanishing-point-based adaptive ROI selection reduces computational complexity.	Difficult-to-identify path checking in the accompanying areas of strength for circumstances reflection, path break, and obscure path marks.
Mammeri et al. (2016) [19]	a: MSER (maximally stable external region) masses utilized for return for ROI and feature extraction. b: Moderate probabilistic Hough transform. c: Kalman filter.	HSV variety space is utilized to recognize path tones (for example, white/yellow).	Traffic thickness and lighting frameworks on the street might affect path discovery execution around evening time.
Zhu et al. (2017) [20]	Principal advances in intelligent vehicles regarding path and street locations.	A straight line is an essential and robust model for short-range streets or highways. Splines are great models for bent roads.	To work on the presentation of the framework, combination-based strategies are required, which are complex.
Mohsen Ghafoorian et al. (2018) [21]	Division—CGAN organization. Classification—join earlier position and classification result to gauge path position.	The implanting misfortune in the discriminator can control the result limit nearer to the label.	An enormous number of parameters are required.
Gurghian et al. (2016) [22]	Classification—consolidate earlier position and classification result to gauge path position.	1: Quick location. 2: Basic organization structure.	1: Restricted application situations. 2: Fixed camera parameters.
Shriyash Chougule et al. (2018) [23]	Regression—a: Multi-branch. b: Coordinate relapse. c: Information augmentation.	1: This strategy does not need a clustering step. 2: Lightweight networks.	Fixed number of path lines can be distinguished.
Wang Ze et al. (2018) [24]	Regression—a: Edge proposition. b: Parameters relapse.	1: The LSTM fills in as a solution for an unsure number. 2: Need not bother with any post-handling.	The ordinate of the three focuses to be identified is predefined.
Van Gansbeke et al. (2019) [25]	Segmentation—a: Producing coordinate weight map. b: A differentiable least-squares fitting module.	This is a more general procedure with no predefined condition.	Fixed number of path lines can be identified. 2: While adding the number of weight maps, the exhibition will be debased.

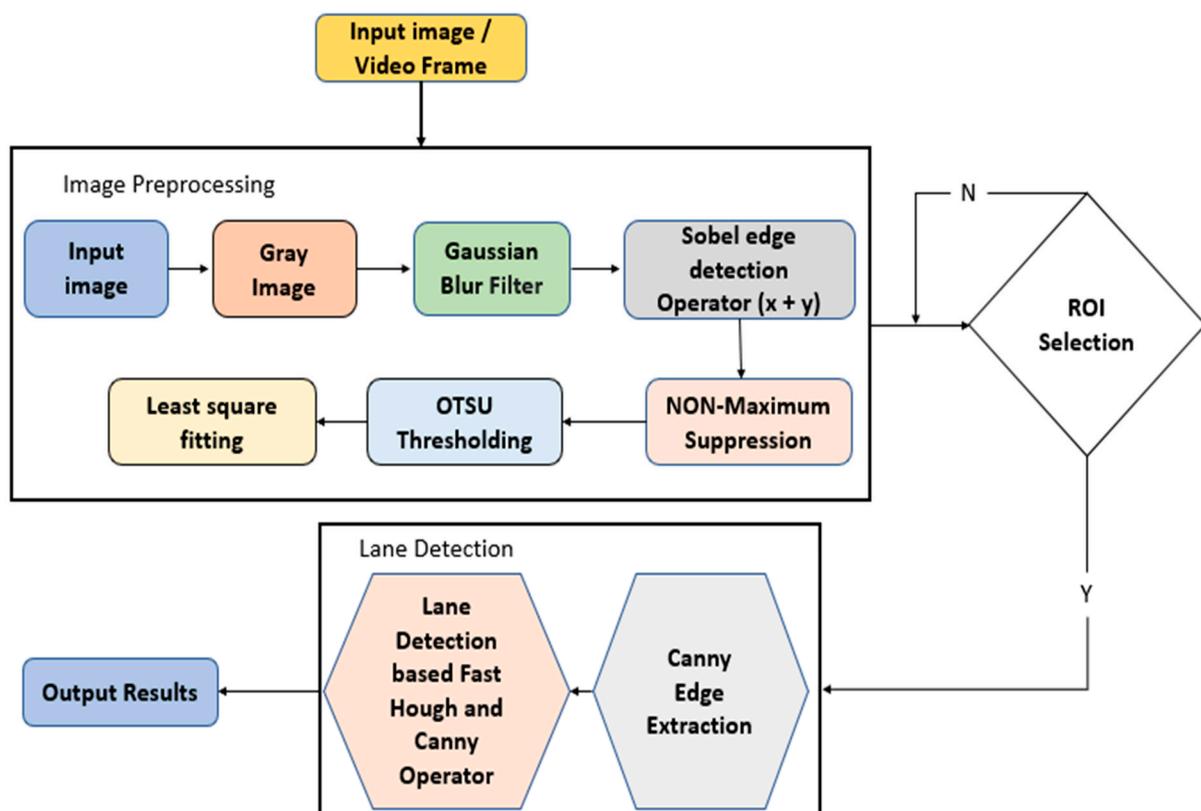
This task requires several steps, including noise reduction, a gradient calculator, and utilizing the HT to locate the ROI. The most significant applications of lane detection techniques are the detection of straight lines and curved lanes. Several researchers utilized the HT algorithm to attempt lane detection [26]. The HT generates additional data for analysis since it simultaneously examines all directions. In [27], the HT was used under several circumstances. We used the parallel HT, the improved HT, and the fast HT to achieve superior results. This study suggested processing time and data storage needs reductions by constraining the Hough space.

## 2. Methodology

This study used the traditional image processing method to detect the vertical lane line. The proposed methodology involves five stages, as shown in Figure 1: (a) image pre-processing, (b) ROI selection, (c) Canny edge extraction, (d) lane-detection-based fast Hough transform and Canny operator, and (e) results output.

We suggested a lane recognition technology that works in complex traffic conditions, which is especially important given how quickly people drive on roadways. First, we pre-processed each frame image and selected the processed images' region of interest (ROI). Finally, we only needed edge detection for vehicles and lines in the ROI area. We developed a new pre-processing method and ROI selection strategy for this study. First, in the pre-processing stage, we converted the RGB color to a grayscale image and then

used a Gaussian blue filter. At the same time, the preliminary edge feature detection was added in the preprocessing stage. Then, the part below the image was selected as the ROI area based on the proposed preprocessing. Only then is the ROI region of the image be Hough-transformed to identify all of the image's straight edges. There are, however, a lot of straight lines that are not continuous. Thus, many straight lines must be filtered and screened. The slope of the straight lines can determine the difference between the left and right lanes. This method is in contrast with the existing methods, whose preprocessing methods only perform graying, blurring, X-gradient, Y-gradient, global gradient, thresh, and morphological closure. Furthermore, the ways to select the ROI area are also very different. Some are based on the edge feature of the lane to select the ROI area, and some are based on the color feature of the lane to select the ROI area. These existing methods do not provide accurate and fast lane information, increasing lane detection difficulty. Experiments in this research demonstrated that the suggested lane identification approach was significantly superior to the current preprocessing and ROI selection methods.



**Figure 1.** The block diagram of the proposed methodology.

### 2.1. Image Editing before Uploading

Recognition of lanes is a crucial component of state-of-the-art driver assistance systems and a prerequisite for fully advanced driving assistance systems (ADASs). Modern cameras are sensitive to ambient light, temperature, and vibrations, all of which introduce noise into the image and make it difficult to obtain clear images. Therefore, algorithms for identifying lanes typically rely on the analysis of camera imagery. Therefore, image enhancement, also known as image pre-processing is required [28]. The current study suggested a methodology that uses a grayscale image with image leveling using a Gaussian blur filter, a Sobel edge detection operator, non-maximal suppression, Otsu thresholding, and a least-squares filter; furthermore, pre-processing images often involves a process of region extraction.

## 2.2. Grayscale Image

Our input image had three color channels (R, G, and B) and has some extra information. The color image covered ample space and was always affected via peripheral light. First, we converted the color digital image to grayscale so that it will have just two dimensions and be very easy to use [16]. The typical approach is the least complicated one. Taking the mean of three colors is all that is required. If the image is in RGB format, converting it to grayscale involves summing the red, green, and blue and dividing the result by 3. It often produces a digital color image by employing an RGB color model and then applying various weights to the initial three channels to produce the gray image depicted in Figure 2. Equation (1) illustrates the conversion formula.

$$\text{Grayscale} = \left( \frac{R + G + B}{3} \right) \quad (1)$$

$$\text{Grayscale} = R(x, y) * 0.299 + G(x, y) * 0.587 + B(x, y) * 0.114 \quad (2)$$



(a) Original road image



(b) Original road image



(a1) Grayscale image



(b1) Grayscale image

**Figure 2.** The original road input images (a,b) and the output grayscale images (a1,b1) are shown.

## 2.3. Noise Reduction

Since the image was converted to grayscale, it still had some noise and unwanted information. Therefore, the next step was removing the image's noise with the Gaussian filter. The Gaussian filter  $h(x, y)$  is typically used, and it can utilize a  $(3 \times 3)$  matrix.

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp \left[ -\frac{(x^2 + y^2)}{2\sigma^2} \right] \quad (3)$$

where  $\sigma$  is set to  $[\sigma_{min}]$  or  $\langle \sigma_{min} \rangle$ . In this instance, the image's noise level was assumed to be proportional to the scale parameter, which is also known as the standard deviation of the Gaussian filter. A large-scale (small-scale)  $\sigma$  is needed for large (small)  $\sigma_n$  cases [29]. Figure 3 depicts the road's final image after the Gaussian filter was applied.



(a) Gaussian filter

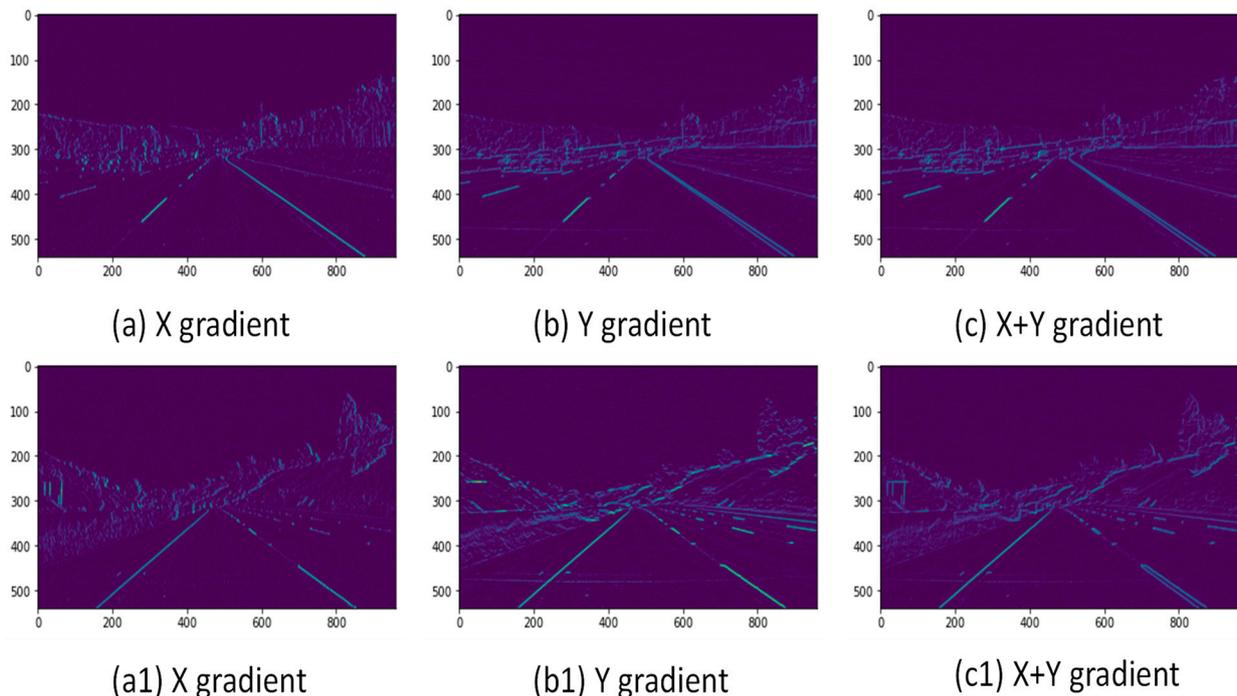
(b) Gaussian filter

**Figure 3.** Using Gaussian filter on (a) a1 and (b) b1 from Figure 2.

Although a significant filter variance is helpful for noise suppression when employing the Gaussian filter, it has the unintended side effect of distorting the parts of the image where there are sudden changes in pixel brightness. The Gaussian filter also creates edge position displacement, vanishing edges, and phantom edges.

2.4. Sobel Edge Detection Operator (SEDO)

The Sobel operator is a discrete differentiation operator. It estimates the image intensity function’s gradient, i.e., the derivatives  $G_X$  and  $G_Y$ . They are produced by applying a Sobel kernel filter to the smoothed image in both the X and Y directions. By combining these, we can determine the absolute magnitude and direction of the gradient at any given point. Figure 4 demonstrates the X, Y, and X + Y slopes.



**Figure 4.** Using an X gradient, Y gradient, and an X + Y gradient with the Sobel edge detection operator.

The Sobel operator has been widely used in image processing as a component of numerous popular edge detection algorithms. It approximates the gradient of the image intensity function and is an application of the discrete differentiation operator. At each pixel, the Sobel operator returns either the norm of the gradient vector or the gradient vector’s gradient. The Sobel operator, i.e., the x and y partial derivatives of the function

$f(x, y)$ , is essential for generating a three-dimensional (3D) map of the region surrounding an  $(x, y)$  point. The equations for its digital gradient approximation may look like the following when the central point emphasizes suppressing noise.

$$G_x = \{f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)\} - \{f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)\} \quad (4)$$

$$G_y = \{f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)\} - \{f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)\} \quad (5)$$

With the above results combined, they can determine the magnitude of the gradient at any given image location:

$$g(x, y) = \sqrt{G_x^2 + G_y^2} \quad (6)$$

Similarly, we can also adopt the following:

$$g(x, y) = |G_x| + |G_y| \quad (7)$$

Its convolution template operator is as follows:

$$T_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad T_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The Sobel operator is then used to identify the borders of image  $M$ . The linear template  $T_x$  and the vertical template  $T_y$  are then used to complete the image, possibly resulting in two gradient matrices of the same size as the original, namely,  $M_1$  and  $M_2$  [30]. The possible outcome is  $G$  when the two gradient matrices are added together.

### 2.5. Non-Local Maximum Suppression (NLMS)

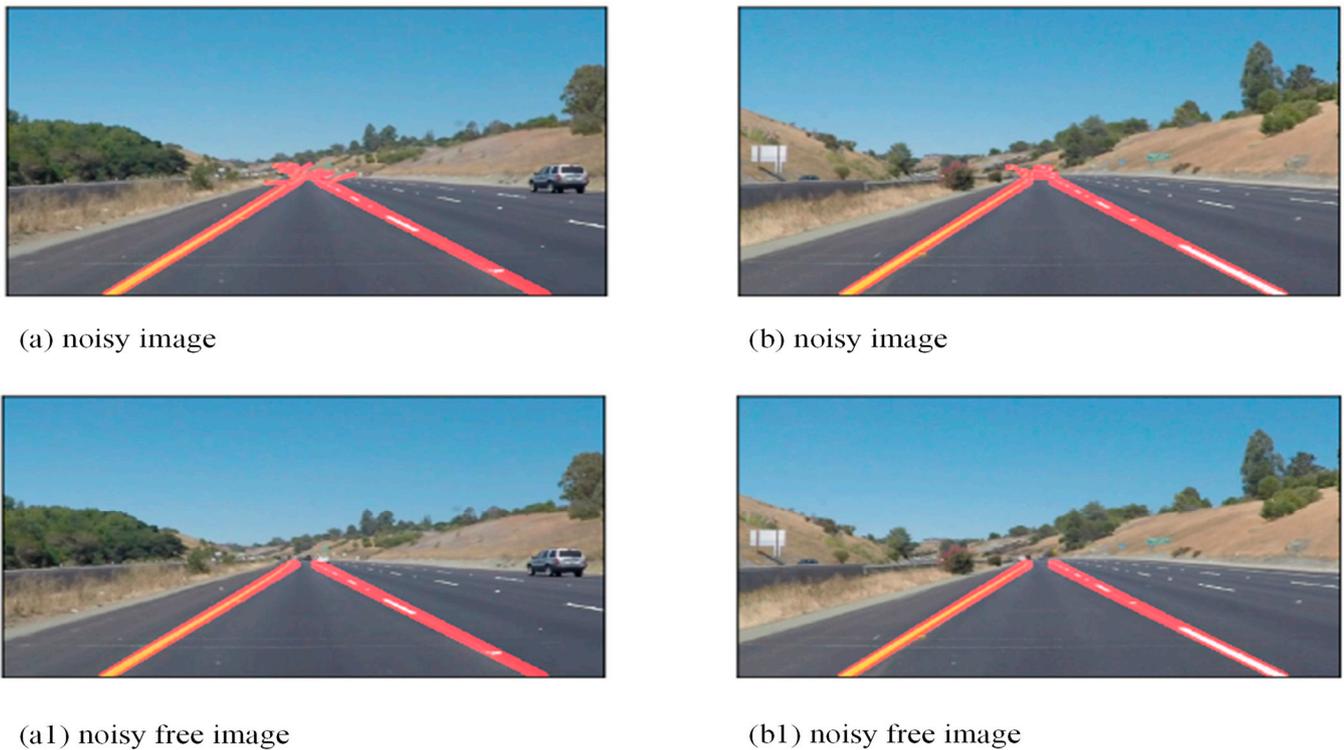
Many edge detection algorithms also use non-local maximum suppression. The pixels excluded from the nearby maxima are set to zero as the picture is examined along the picture slope directions. All image data outside of the local maxima is obliterated in this way. There should not be any thick borders in the final product. Therefore, non-maximal suppression is required for edge cleaning. The algorithm iteratively examines all the image's points to identify the pixel with the highest gradient value along the image's edges. Suppressing data at the non-maximum level removes points that are not located on critical borders. To observe its effects, in Figure 5a,b are noisy images, while the noise-free images (Figure 5a1,b1) were obtained after using the filter (NLMS).

### 2.6. Otsu Thresholding

Otsu thresholding is very helpful for edge detection because it changes the image to sharp black and white. Let us say the objective image they want to segment is  $f(x, y)$ ; its range in the grayscale is  $\{0, 1, \dots, L-1\}$ . The picture pixels will be separated into two classifications using the limit  $t$ :  $C_0 = \{0, 1, \dots, t\}$ ,  $\{t+1, t+2, \dots, L-1\}$ . The background and the target are represented by  $C_0$  and  $C_1$ . The following is the class-square error between  $C_0$  and  $C_1$ :

$$\sigma(t)^2 = \omega_0(t) * \bar{\omega}_1(t) * (\mu_0(t) - \mu_1(t))^2 \quad (8)$$

At this point, the threshold is  $t$ , and  $\omega_0(t)$  is the number of pixels in the image whose gray values are less than  $t$ .  $\bar{\omega}_1(t)$  is the quantity of pixels where the picture's dark worth is more critical than the limit esteem  $t$ .  $\mu_0(t)$  is the typical dark worth of the pixels where the picture's dim worth is not exactly the edge  $t$ . Furthermore,  $\mu_1(t)$  is the typical dim worth of the pixels where the picture dim qualities are more critical than the limit  $t$ . The best partition threshold is the  $t$  that makes  $\sigma(t)^2$  the greatest value.



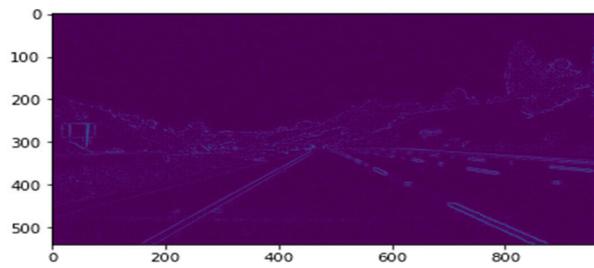
**Figure 5.** Multiple overlapping lines (a,b), and the results of the non-max suppression (a1,b1).

To find an appropriate threshold for each small block, we can change the color image to grayscale and use Otsu's technique. We do not compute the point of all the blocks because lane marks can only be found in regions with candidate lane mark edges. An edge orientation binarization block has four individual blocks. Only when one of those four blocks is true would we apply the thresholding to the entire region. Each pixel in the output image is assigned a color based on its intensity relative to a threshold; white (foreground) is used for pixels with powers higher than the threshold and black (background) is used for intensities lower than or equal to the point (background). The only parameter of the algorithm in an implementation is the stopping criterion at which the iterations are terminated. The algorithm improves on the standard Otsu's method for preserving weak objects by iteratively applying Otsu's process and gradually shrinking the to-be-determined (TBD) region for segmentation, as shown in Figure 6.

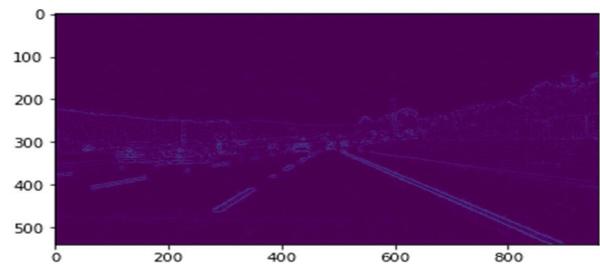
### 2.7. Canny Edge Detection

The Canny edge detector is an edge detection operator that can identify a wide range of image edges by employing a multi-stage algorithm. Using a method known as "Canny edge detection" on various visual objects, it is possible to significantly reduce the amount of data that must be processed. It has found many uses in different kinds of computer vision systems. In his research, Canny discovered that the requirements for implementing edge detection on various vision systems are generally consistent. Accordingly, a solution for edge detection that meets these needs can be applied practically anywhere. The Canny edge detection algorithm is one of the most precisely defined approaches that guarantee accurate detection. Since then, it has become a go-to edge detection algorithm [31]. Therefore, a situation can benefit from an edge detection solution that considers these needs.

Figure 7 depicts a Canny edge detector result obtained by applying this method to an original road image: two images were produced when the Canny edge detector was applied to a lane image, with one showing the edge pixels and the other showing the gradient orientations. Considering how well-defined and reliable the Canny edge detection algorithm is, it has quickly risen to the top of the most used edge detection algorithms.



(a) Least square filter



(b) Least square filter



(a1) Otsu thresholding



(b1) Otsu thresholding

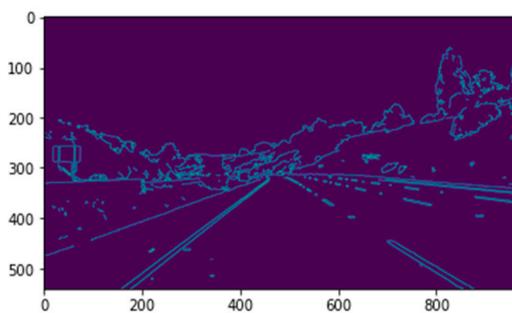
Figure 6. Initial entire pre-processing image least-squares filter (a,b) and the result after Otsu thresholding (a1,b1).



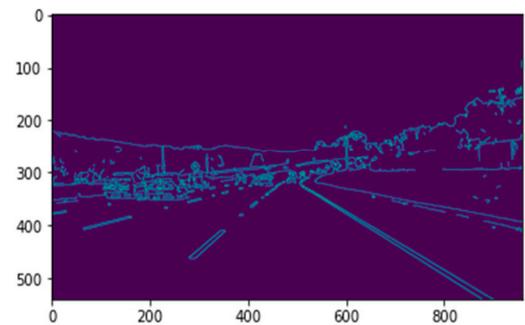
(a) Original road image



(b) Original road image



(a1) Canny edge detector



(b1) Canny edge detector

Figure 7. Canny edge feature extraction.

2.8. Region of Interest (ROI) Selection

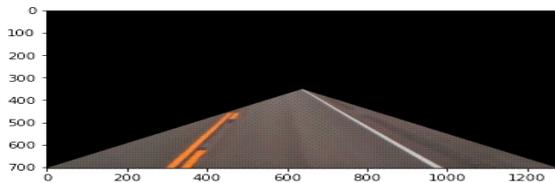
When a grayscale picture is obtained, the path detection algorithm is accelerated and made more accurate by establishing the grayscale view of the ROI investment [32]. An image’s relevant data can be extracted through the use of ROIs. When performing lane line identification, the lane line image in a specific area is sought after. Noise in the model can be reduced and the detection accuracy enhanced by masking other irrelevant information. As can be seen in Figure 8, the lane line is typically displayed in the following area of the camera. The focus of this study was on the lower-middle section of the image, which was comprised of a triangular field of view facing the camera. The coordinates for the triangle’s left vertex were (0, 700), its right vertex was at (1024, 700), and its center was at (560, 330) (for a 1024-by-720 image). The effectiveness of operations in this area could be significantly boosted by performing image pre-processing and a lane line search. After adjusting the contrast of the ROI, Otsu’s thresholding was used to convert the image using a binary conversion.



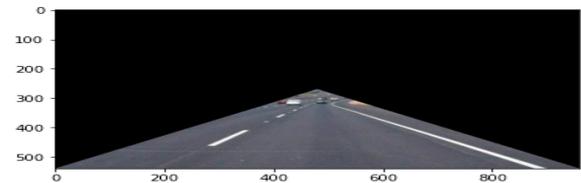
(a) Original road image.



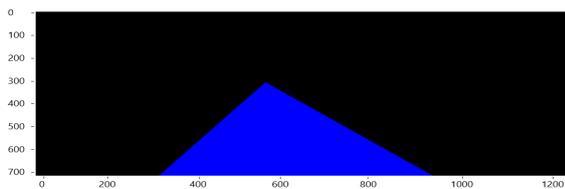
(b) Original road image.



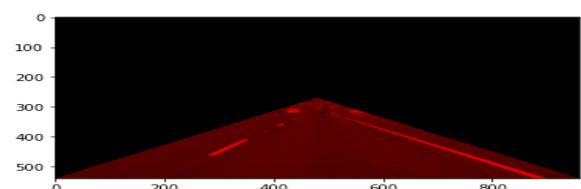
(c) Selection of the ROI.



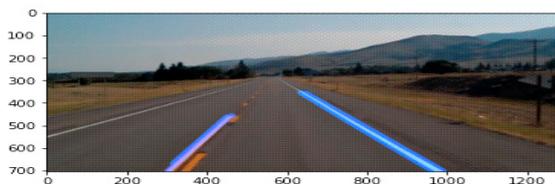
(d) Selection of the ROI.



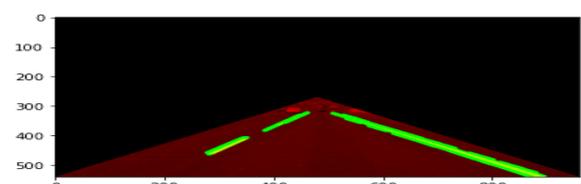
(e) Extraction of the ROI.



(f) Extraction of the ROI.



(g) Result.



(h) Result.

Figure 8. Edge detection and road region of interest extraction.

### 2.9. Fast Hough Transform (FHT)

The HT is a feature extraction technique for voting-based shape detection in images, including curves, circles, and straight lines. The HT is most widely used for detecting consecutive lines in a picture. The linear equation often has the following form when represented in the Cartesian coordinate system:

$$y = mx + b \quad (9)$$

where  $m$  is the slope and  $b$  is the intercept in this equation. Other parameters, denoted by  $\rho$  and  $\theta$ , which are polar coordinates, are proposed in light of this aspect:

$$\rho = x \cos\theta + y \sin\theta, \theta \in [-\pi, \pi) \quad (10)$$

As depicted in Figure 9, the  $\rho$  boundary addresses the distance between the line and the beginning. Additionally, the angle of the vector's closest point on the line is represented by the  $\theta$  parameter. This representation can map any line in the image space to a point  $(\rho, \theta)$  in the Hough space.

As displayed in Figure 10, straight lines that pass through the point  $(x_1, y_1)$  in the  $xy$  image space are mapped into the  $\rho\theta$  Hough parameter space to create a curve  $l_1$ , and straight lines that pass through the point  $(x_2, y_2)$  are mapped onto the  $\rho\theta$  Hough parameter space to create a curve  $l_2$ .

The straight line equation in the image space can be obtained from this, which is expressed by the following Equation (11). The intersection of the two curves is  $(\rho_0, \theta_0)$ , which corresponds to the spatial line L.

$$\rho_0 = x \cos\theta_0 + y \sin\theta_0 \quad (11)$$

### 2.10. The Algorithm

1. Decide on the range of  $\rho$  and  $\theta$ . Often, the range of  $\theta$  is  $[0, 180]$  degrees and the range of  $\rho$  is  $[-d, d]$ , where  $d$  is the length of the edge image's diagonal. It is important to quantize the range of  $\rho$  and  $\theta$  so that there should be a finite number of possible values.
2. Create a 2D array called the accumulator representing the Hough space with dimension  $(\text{num\_rhos}, \text{num\_thetas})$  and initialize all its values to zero.
3. Perform edge detection on the original image. This can be done with any edge detection algorithm of your choice.
4. For every pixel on the edge image, check whether the pixel is an edge pixel. If it is an edge pixel, loop through all possible values of  $\theta$ , calculate the corresponding  $\rho$ , find the  $\theta$  and  $\rho$  index in the accumulator, and increment the accumulator based on those index pairs.
5. Loop through all the values in the accumulator. If the value is larger than a certain threshold, obtain the  $\rho$  and  $\theta$  index, as well as the value of  $\rho$  and  $\theta$  from the index pair, which can then be converted back to the form of  $y = mx + b$ .

The same operation is performed for the ROI area's points of interest. If the curves representing those points in the coordinate system coincide, then it follows that there can be a straight line that connects all of the features. The main lane line depicted in the image can be obtained by utilizing the line in the Hough space bearing the intersection number [32]. In addition, there cannot be any fewer than one intersection and more than some fixed minimum number of intersections. After edges are found using Canny edge detection, the HT can easily mark the lane line by computing the edges in the image or source video.

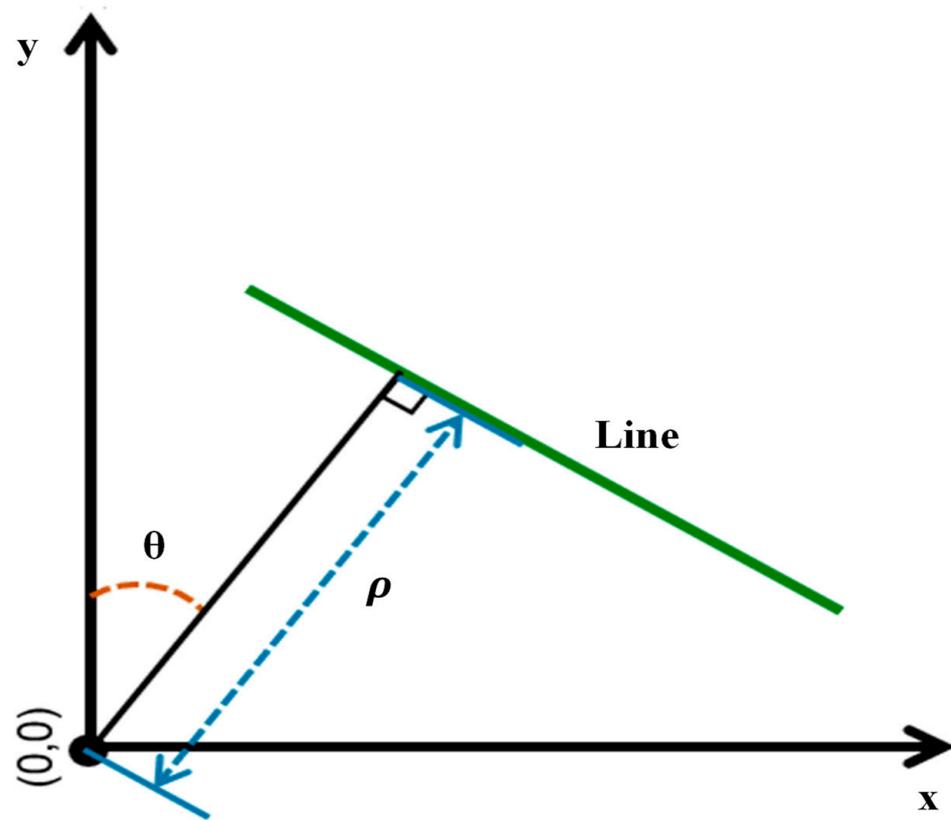


Figure 9. Schematic representation of  $\rho$  and  $\theta$ .

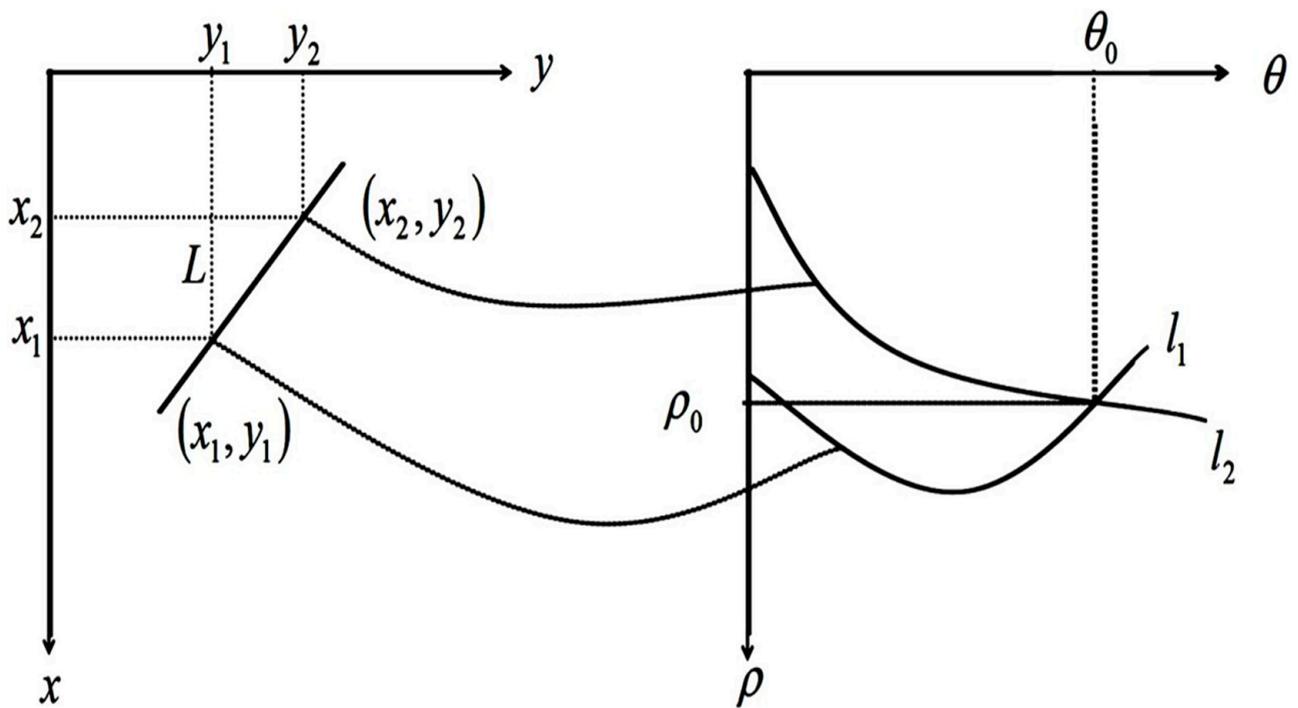


Figure 10. Geometric parameters and Hough space depicted schematically.

### 3. Experimental Results

Software settings: Python 3.7.0 64-bit, OpenCV 4.1.0, and FFMPEG 1.4 were used in this project. The software used in this study was Anaconda-Spyder IDE. The code is accessible [31] with the following hardware settings: 5 TB mechanical hard drive, 12 GB DDR4 RAM, and a 2.7 GHz Intel (R) Core™ i7-7500 processor. Furthermore, there was a possibility that we could come across unmarked, noise-interrupted frame images. Next-frame line data could be predicted using information from prior frames in the Hough space. It is possible to increase the precision of lane line detection in the next frame image without suffering from a large offset caused by the marked line detected in the previous image. Figure 11 shows the line cluster produced by joining the Hough transform's extracted points of interest.

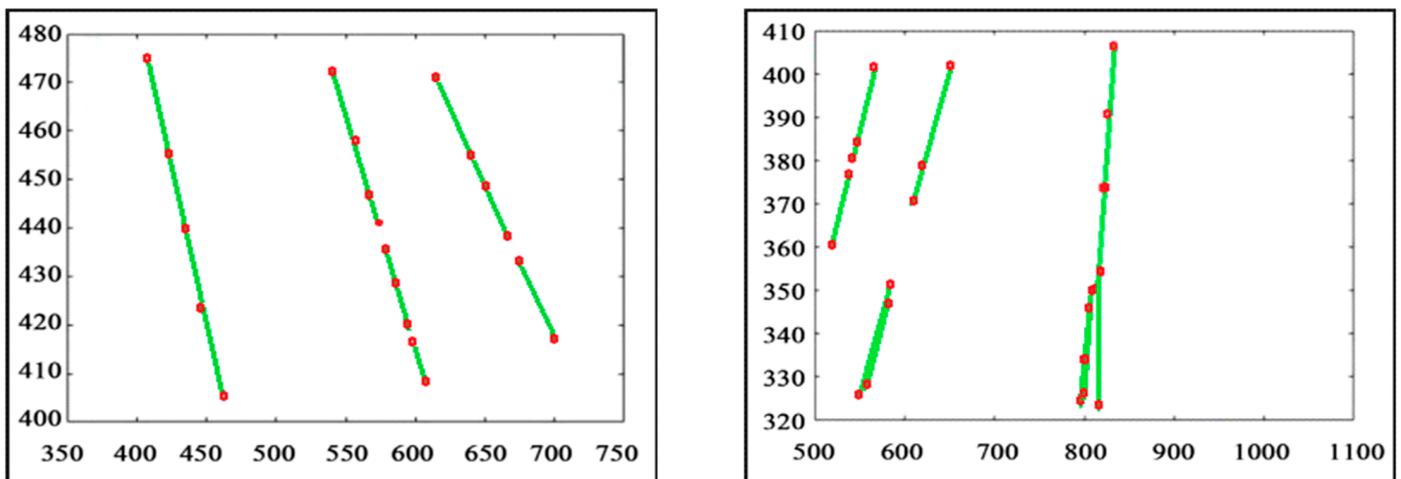
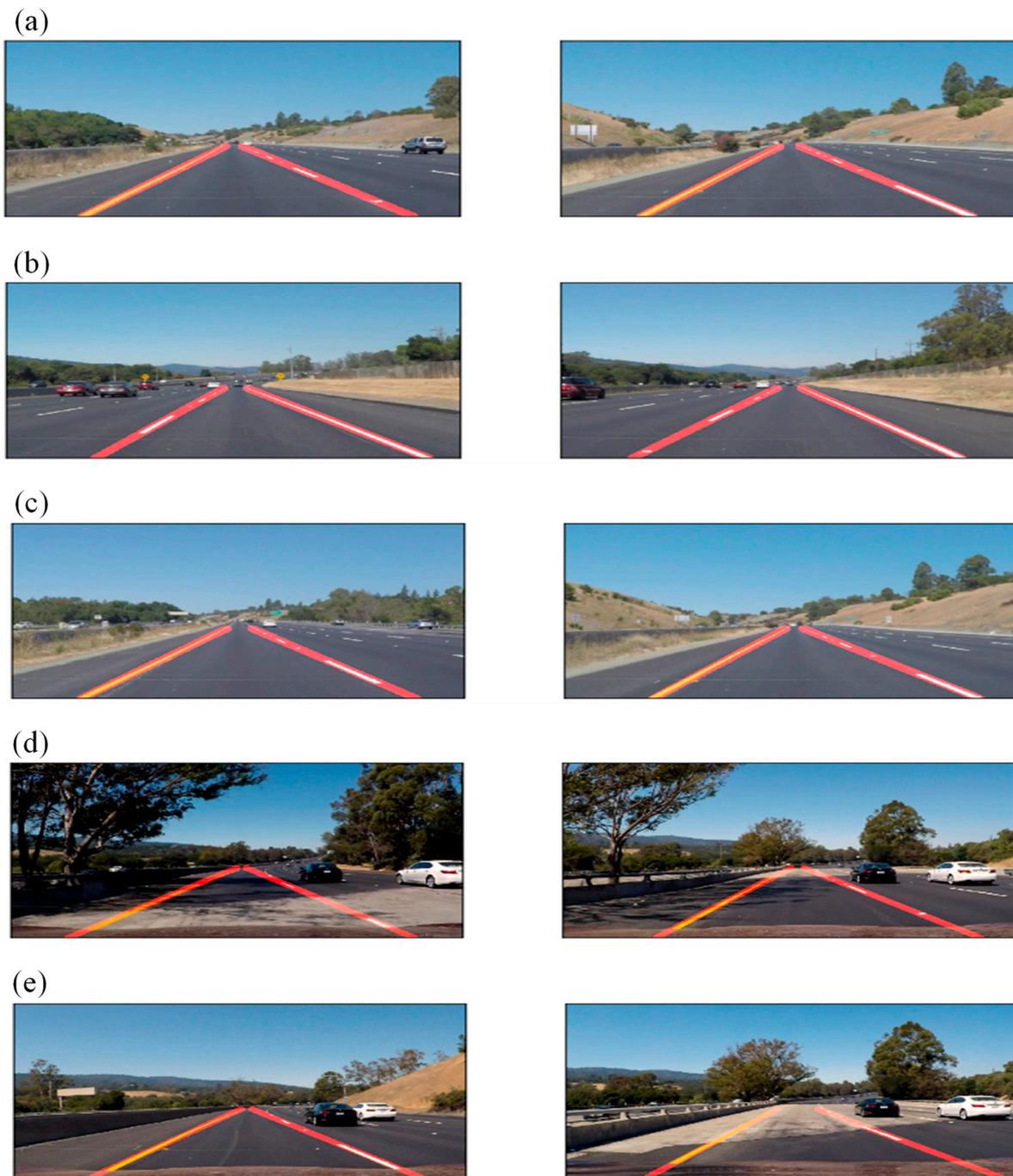


Figure 11. Road line clusters.

In addition to employing road-driving films for simulation tests, we also thoroughly tested the performance of the suggested method across a wide range of difficult road situations using the Tusimple dataset (a general purpose benchmark dataset) for lane detection; the Tusimple dataset, which consists of 3626 training images and 2782 testing images, is available in [33–35]. There are 20 frames made up of 1 s sequence images, where 19 are unmarked and 1 has a lane ground truth label. Based on the weather, the time of day, the number of lanes (two, three, or more), the time of night, and the traffic conditions, the images in the dataset can roughly be divided into five states [36]. Using 160 randomly selected images from the Tusimple dataset representing various conditions, we tested the proposed algorithm for lane detection in this work. Figure 12 depicts five experimental results obtained by placing complex signs in the center of the lane; it was evident that the lane line was detected alongside crosswalk lines, text reminder signs, and other marking lines. However, the lane line detection algorithm was more accurate. Standard filtering kept the values at  $9 \times 9$  and adjusted them like a mini length. Max length also changed  $\rho$ , where “The variable ( $\rho = rho$ ) is the perpendicular distance from the origin to the line” and obtained many different results. The max result was found using  $rho$  6, Gaussian blur  $9 \times 9$ , mini-length 20, max length 25, and the threshold at 160.



**Figure 12.** Fast Hough lane line detection results of the transformation on the dataset of typical video frames from Tusimple. Reddish lines indicate the ground truth lanes.

The maximum accuracy in our experiment in complex conditions was 96.7%. Moreover, the possible comparison processing time was 22.2 ms. Very high accuracy was achieved for normal road conditions. The results of lane detection in the road driving video were somewhat better than the average accurate classification performance. However, the data gathering time per frame was a little bit longer. Despite this, the proposed algorithm showed good precision and flexibility in recognizing the Tusimple dataset. The results can be seen in Table 2.

**Table 2.** Accuracy analysis of lane detection using the fast Hough transform.

Category	Algorithms	$\rho = rho$	Gaussian Blur	Mini Length	Max Length	Threshold	Accuracy Rate/%	Average Processing Time (ms)
No line	Fast Hough Transform	4	$5 \times 5$	20	15	160	89.5%	19.7
Crowded	Fast Hough Transform	5	$7 \times 7$	20	15	160	90.3%	19.9
Shadow	Fast Hough Transform	5	$9 \times 9$	20	15	140	92.4%	20.5
Curve	Fast Hough Transform	5	$9 \times 9$	20	15	160	93.5%	20.9
Crossroad	Fast Hough Transform	6	$9 \times 9$	20	15	160	94.2%	21.4
Normal	Fast Hough Transform	6	$9 \times 9$	20	25	160	96.7%	22.2

#### 4. Advantages, Limitations, and Real-Time Application Usage

Here are some potential advantages, limitations, and real-time application usage of the method for lane line detection presented in the paper.

##### 4.1. Advantages

1. Automatic threshold selection: The proposed method used Otsu's method for thresholding, which is an automated method for selecting a threshold value that separates the foreground from the background. This can be useful in cases where the lighting conditions are variable, allowing the algorithm to adapt to the current lighting conditions.
2. Robust line detection: The proposed method used the Hough transform, which is a powerful tool for detecting lines in images. This can allow the algorithm to accurately see lane lines, even in cases where the lines are somewhat curved or partially occluded.
3. Fast processing: the implementation of the Hough transform is rapid, which is essential for real-time lane detection in a vehicle.
4. Robust in complex road conditions: The method is powerful in challenging road conditions, such as when there are shadows or when the road is curved. This can allow the algorithm to perform well in various environments.

##### 4.2. Limitations

1. The proposed method may be sensitive to the parameters used for thresholding and line detection, which may need to be tuned for different conditions.
2. Further testing and evaluation are necessary to assess the performance and robustness of the proposed method thoroughly.
3. The proposed method may not perform well in lane lines that are faint or difficult to distinguish from the road surface.

##### 4.3. Real-Time Application Usage

1. Increased safety: lane line detection can help to keep a vehicle within its lane, improving safety in complex road conditions.
2. Enhanced driver assistance: by detecting lane lines, a vehicle can provide additional service to the driver, such as lane departure warnings or assisting with staying in lane.
3. Improved navigation: lane line detection can help a vehicle to accurately determine its position on the road, improving navigation accuracy.
4. Enhanced autonomous driving capabilities: lane line detection is an essential component of autonomous vehicle systems, as it allows the vehicle to identify its surroundings and make driving decisions.

5. Reduced driver fatigue: assisting the driver with staying in lane and lane line detection tasks can help to reduce driver fatigue.
6. Increased fuel efficiency: by helping the vehicle maintain a consistent speed and position within the lane, lane line detection can improve fuel efficiency.
7. Improved traffic flow: lane line detection can help vehicles to maintain a consistent speed and position within their lanes, improving traffic flow.
8. Enhanced intersection safety: lane line detection can help a vehicle to accurately determine its position at an intersection, improving safety.
9. Improved weather performance: lane lines may be harder to see in adverse weather conditions, but lane line detection can still help a vehicle stay within its lane.
10. Enhanced road maintenance: lane line detection can help a vehicle to accurately determine its position on the road, aiding in the care of roads by providing data on vehicle usage patterns.

## 5. Conclusions

The fast Hough transform and Otsu thresholding, which is an advanced canny edge detector (ACED), were used in this study to propose a method for a lane line identification FHT. It can only identify straight lines. Identifying curved road lanes (or the curvature of lanes) is a complex issue. Instead of strictly consecutive lines, we will employ perspective transformation and poly-fit lane lines. In most pictures, the lanes next to the vehicle are straight. Unless it is a very sharp curve, the bending only becomes apparent at a greater distance. Therefore, this elementary lane-finding strategy is beneficial. The raw camera data was converted to a grayscale image and image smoothing was achieved using a Gaussian blur filter, the Sobel edge detection operator, non-maximum suppression, Otsu thresholding, the minor square filter, and the extraction of the region of interest. We tested this algorithm with various inputs and conditions, and it consistently performed well. The algorithm made it possible to extract the lane line's pixels quickly and precisely, allowing for full semantic segmentation between the lane line and the road surface. This study ignored the effects of bright light and hard-to-read road signs on the driving environment. In the future, we can mix diverse conditions to improve lane recognition even further.

## 6. Future Scope

Lane line detection is a critical problem in intelligent vehicles and autonomous driving. There is a lot of potential for further research in this area, particularly in complex road conditions. Some areas that could be explored in more depth and receive further analysis are as follows:

- Improving the robustness and accuracy of lane line detection algorithms under various lighting and weather conditions, such as rain, snow, and fog.
- Developing methods for detecting and tracking lane lines in the presence of other visual distractions, such as pedestrians, vehicles, and road signs.
- Investigating machine learning techniques, such as deep learning, to improve the performance of lane line detection algorithms.
- Developing approaches for integrating lane line detection with other perception tasks, such as object detection and classification, to enable more comprehensive identification of the environment.
- Exploring the use of multiple sensors, such as cameras, lidar, and radar, to improve the accuracy and reliability of lane line detection in complex road conditions.

**Author Contributions:** Conceptualization, M.A.J., M.A.G., M.A.A., N.Z., A.S.M.M., E.M.T.-E., P.B., M.S.J. and X.J.; Methodology, M.A.J., M.A.G., M.A.A. and X.J.; Software, M.A.J., M.A.A., N.Z., E.M.T.-E., P.B. and M.S.J.; Validation, M.A.J., M.A.G., M.A.A., N.Z., A.S.M.M., P.B., M.S.J. and X.J.; Formal analysis, M.A.J., M.A.G., N.Z., A.S.M.M., E.M.T.-E., M.S.J. and X.J.; Investigation, P.B. and X.J.; Resources, A.S.M.M. and M.S.J.; Data curation, N.Z.; Writing—original draft, M.A.J., M.A.G., M.A.A., A.S.M.M. and E.M.T.-E.; Visualization, P.B.; Supervision, X.J.; Funding acquisition, E.M.T.-E. and X.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Researchers Supporting Project No. RSP-2023R363, King Saud University, Riyadh, Saudi Arabia.

**Data Availability Statement:** Not Applicable.

**Acknowledgments:** Authors are thankful for the Future university, King Saud University and Lanzhou University for Funding. This work was funded by the Researchers Supporting Project No. RSP-2023R363, King Saud University, Riyadh, Saudi Arabia.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Hou, H.; Guo, P.; Zheng, B.; Wang, J. An Effective Method for Lane Detection in Complex Situations. In Proceedings of the 2021 9th International Symposium on Next Generation Electronics (I.S.N.E.), Changsha, China, 9–11 July 2021; IEEE: Piscataway, NJ, USA, 2021.
- Zhang, Z.; Ma, X. Lane recognition algorithm using the hough transform based on complicated conditions. *J. Comput. Commun.* **2019**, *7*, 65. [[CrossRef](#)]
- Qiu, D.; Weng, M.; Yang, H.; Yu, W.; Liu, K. Research on Lane Line Detection Method Based on Improved Hough Transform. In Proceedings of the 2019 Chinese Control And Decision Conference (C.C.D.C.), Nanchang, China, 3–5 June 2019; IEEE: Piscataway, NJ, USA, 2019.
- Gu, K.X.; Li, Z.Q.; Wang, J.J.; Zhou, Y.; Zhang, H.; Zhao, B.; Ji, W. The Effect of Cryogenic Treatment on the Microstructure and Properties of Ti-6Al-4V Titanium Alloy. In *Materials Science Forum*; Trans Tech Publications Ltd.: Wollerau, Switzerland, 2013.
- Li, X.; Yin, P.; Zhi, Y.; Duan, C. Vertical Lane Line Detection Technology Based on Hough Transform. *IOP Conf. Ser. Earth Environ. Sci.* **2020**, *440*, 032126. [[CrossRef](#)]
- Wu, P.-C.; Chang, C.-Y.; Lin, C.H. Lane-mark extraction for automobiles under complex conditions. *Pattern Recognit.* **2014**, *47*, 2756–2767. [[CrossRef](#)]
- Wang, Y.; Shen, D.; Teoh, E.K. Lane detection using spline model. *Pattern Recognit. Lett.* **2000**, *21*, 677–689. [[CrossRef](#)]
- Savant, K.V.; Meghana, G.; Potnuru, G.; Bhavana, V. Lane Detection for Autonomous Cars Using Neural Networks. In *Machine Learning and Autonomous Systems*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 193–207.
- Borkar, A.; Hayes, M.; Smith, M.T. Robust Lane Detection and Tracking with Ransac and Kalman Filter. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (I.C.I.P.), Cairo, Egypt, 7–10 November 2009; IEEE: Piscataway, NJ, USA, 2009.
- Lee, J.W.; Yi, U.K. A lane-departure identification based on L.B.P.E., Hough transform, and linear regression. *Comput. Vis. Image Underst.* **2005**, *99*, 359–383. [[CrossRef](#)]
- Gabrielli, A.; Alfonsi, F.; del Corso, F.J.E. Simulated Hough Transform Model Optimized for Straight-Line Recognition Using Frontier FPGA Devices. *Electronics* **2022**, *11*, 517. [[CrossRef](#)]
- Zhao, Y.; Wen, C.; Xue, Z.; Gao, Y. 3D Room Layout Estimation from a Cubemap of Panorama Image via Deep Manhattan Hough Transform. In *European Conference on Computer Vision—ECCV 2022, Proceedings of the 17th European Conference, Tel Aviv, Israel, 23–27 October 2022*; Springer: Berlin/Heidelberg, Germany, 2022.
- Xu, H.; Li, H. Study on a Robust Approach of Lane Departure Warning Algorithm. In Proceedings of the IEEE International Conference on Signal Processing System (I.C.S.P.S.), Dalian, China, 5–7 July 2010; pp. 201–204.
- Zhao, K.; Meuter, M.; Nunn, C.; Muller, D.; Schneiders, S.M.; Pauli, J. A Novel Multi-Lane Detection and Tracking System. In Proceedings of the IEEE Intelligent Vehicles Symposium, Alcalá de Henares, Spain, 3–7 June 2012; pp. 1084–1089.
- Obradović, Đ.; Konjović, Z.; Pap, E.; Rudas, I.J. Linear fuzzy space based road lane model and detection. *Knowl.-Based Syst.* **2013**, *38*, 37–47. [[CrossRef](#)]
- An, X.; Shang, E.; Song, J.; Li, J.; He, H. Real-time lane departure warning system based on a single fpga. *EURASIP J. Image Video Process.* **2013**, *38*, 38. [[CrossRef](#)]
- Cheng, H.Y.; Jeng, B.S.; Tseng, P.T.; Fan, K.C. Lane detection with moving vehicles in the traffic scenes. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 571–582. [[CrossRef](#)]
- Son, J.; Yoo, H.; Kim, S.; Sohn, K. Realtime illumination invariant lane detection for lane departure warning system. *Expert Syst. Appl.* **2015**, *42*, 1816–1824. [[CrossRef](#)]
- Mammeri, A.; Boukerche, A.; Tang, Z. A real-time lane marking localization, tracking and communication system. *Comput. Commun.* **2016**, *73*, 132–143. [[CrossRef](#)]

20. Aziz, S.; Wang, H.; Liu, Y.; Peng, J.; Jiang, H. Variable universe fuzzy logic-based hybrid LFC control with real-time implementation. *IEEE Access* **2019**, *7*, 25535–25546. [[CrossRef](#)]
21. Ghafoorian, M.; Nugteren, C.; Baka, N.; Booi, O.; Hofmann, M. El-Gan: Embedding Loss Driven Generative Adversarial Networks for Lane Detection. In Proceedings of the European Conference on Computer Vision (E.C.C.V.), Munich, Germany, 8–14 September 2018.
22. Gurghian, A.; Koduri, T.; Bailur, S.V.; Carey, K.J.; Murali, V.N. Deeplanes: End-to-End Lane Position Estimation using Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 38–45.
23. Chougule, S.; Koznek, N.; Ismail, A.; Adam, G.; Narayan, V.; Schulze, M. Reliable Multilane Detection and Classification by Utilizing CNN as a Regression Network. In Proceedings of the European Conference on Computer Vision (E.C.C.V.), Munich, Germany, 8–14 September 2018.
24. Wang, Z.; Ren, W.; Qiu, Q. Lanenet: Real-time lane detection networks for autonomous driving. *arXiv* **2018**, arXiv:1807.01726.
25. Van Gansbeke, W.; De Brabandere, B.; Neven, D.; Proesmans, M.; Van Gool, L. End-to-End Lane Detection Through Differentiable Least-Squares fitting. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
26. Khan, M.A.M.; Haque, M.F.; Hasan, K.R.; Alajmani, S.H.; Baz, M.; Masud, M.; Nahid, A.A. LLDNet: A Lightweight Lane Detection Approach for Autonomous Cars Using Deep Learning. *Sensors* **2022**, *22*, 5595. [[CrossRef](#)] [[PubMed](#)]
27. Lin, Q.; Han, Y.; Hahn, H. Real-Time Lane Departure Detection Based on Extended Edge-Linking Algorithm. In Proceedings of the 2010 Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 7–10 May 2010; IEEE: Piscataway, NJ, USA, 2010.
28. Aziz, S.; Peng, J.; Wang, H.; Jiang, H. Admm-based distributed optimization of hybrid mt-dc-ac grid for determining smooth operation point. *IEEE Access* **2019**, *7*, 74238–74247. [[CrossRef](#)]
29. Ding, F.; Wang, A.; Zhang, Q. Lane Line Identification and Research Based on Markov Random Field. *World Electr. Veh. J.* **2022**, *13*, 106. [[CrossRef](#)]
30. Zhang, J.-Y.; Yan, C.; Huang, X.-X. Edge detection of images based on improved Sobel operator and genetic algorithms. In Proceedings of the 2009 International Conference on Image Analysis and Signal Processing, Linhai, China, 11–12 April 2009; pp. 31–35.
31. Available online: <https://github.com/SunnyKing342/Lane-line-detection-Hough-Transform.git> (accessed on 24 December 2022).
32. Aggarwal, N.; Karl, W.C. Line Detection in Images Through Regularized Hough Transform. *IEEE Trans. Image Process.* **2006**, *15*, 582–591. [[CrossRef](#)] [[PubMed](#)]
33. Marzougui, M.; Alasiry, A.; Kortli, Y.; Baili, J. A lane tracking method based on progressive probabilistic Hough transform. *IEEE Access* **2020**, *8*, 84893–84905. [[CrossRef](#)]
34. Ali, M.; Clausi, D. Using the Canny Edge Detector for Feature Extraction and Enhancement of Remote Sensing Images. In Proceedings of the I.G.A.R.S.S. 2001 Scanning the Present and Resolving the Future Proceedings IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No. 01CH37217), Sydney, NSW, Australia, 9–13 July 2001; IEEE: Piscataway, NJ, USA, 2001.
35. Traoré, C.A.D.G.; Séré, A. Straight-Line Detection with the Hough Transform Method Based on a Rectangular Grid. In Proceedings of the Information and Communication Technology for Competitive Strategies (I.C.T.C.S. 2020), Jaipur, India, 11–12 December 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 599–610.
36. Cao, J.; Song, C.; Xiao, F.; Peng, S. Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments. *Sensors* **2019**, *19*, 3166. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.