

## Research Article

# Vision-Based Lane Departure Detection Using a Stacked Sparse Autoencoder

Zengcai Wang <sup>1,2</sup>, Xiaojin Wang,<sup>1,2</sup> Lei Zhao <sup>1</sup> and Guoxin Zhang <sup>1</sup>

<sup>1</sup>School of Mechanical Engineering, Shandong University, Jinan, 250061, China

<sup>2</sup>Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education, School of Mechanical Engineering, Shandong University, Jinan, 250061, China

Correspondence should be addressed to Zengcai Wang; wangzc@sdu.edu.cn

Received 2 October 2017; Revised 8 August 2018; Accepted 27 August 2018; Published 16 September 2018

Academic Editor: Konstantinos Karamanos

Copyright © 2018 Zengcai Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a lane departure detection approach that utilizes a stacked sparse autoencoder (SSAE) for vehicles driving on motorways or similar roads. Image preprocessing techniques are successfully executed in the initialization procedure to obtain robust region-of-interest extraction parts. Lane detection operations based on Hough transform with a polar angle constraint and a matching algorithm are then implemented for two-lane boundary extraction. The slopes and intercepts of lines are obtained by converting the two lanes from polar to Cartesian space. Lateral offsets are also computed as an important step of feature extraction in the image pixel coordinate without any intrinsic or extrinsic camera parameter. Subsequently, a softmax classifier is designed with the proposed SSAE. The slopes and intercepts of lines and lateral offsets are the feature inputs. A greedy, layer-wise method is employed based on the inputs to pretrain the weights of the entire deep network. Fine-tuning is conducted to determine the global optimal parameters by simultaneously altering all layer parameters. The outputs are three detection labels. Experimental results indicate that the proposed approach can detect lane departure robustly with a high detection rate. The efficiency of the proposed method is demonstrated on several real images.

## 1. Introduction

Road safety is a major social issue. The 2015 Global Status Report on Road Safety provided by the World Health Organization shows that the total number of road traffic deaths worldwide has plateaued at 1.25 million per year, with over 3,400 people dying on roads all over the world every day [1]. A considerable fraction of these accidents is due to the unintentional deviation of vehicles from their traveling lane. Unexpected lane departure usually occurs because of the temporary and involuntary fading of a driver's vision caused by fatigue, use of a mobile phone, operation of devices on the instrument panels of vehicles, or chatting. Lane departure is also the secondary cause of road traffic accidents [2]. Therefore, identifying lane departure occurrence is important.

Vision-based systems, which involve installing video cameras (and/or other sensors) in the interior of vehicles to sense the environment and provide useful information for the driver, can be used to improve road safety. Equipping

drivers with an effective vision-based lane departure warning system (LDWS) can rapidly and effectively prevent or reduce lane departure accidents. Lane departure detection has elicited much attention, and several vision-based LDWSs have been successfully developed in the past 20 years [3–18]. These vision-based LDWSs rely on different computer vision techniques. With preparatory work that includes image preprocessing, lane detection, and matching, lane departure detection was conducted in this study by using stacked sparse autoencoders (SSAEs) to create a softmax classifier in gray-level images obtained by a camera with a charge coupled device (CCD); the camera was mounted on a test vehicle. The input features of the developed SSAE system were six parameters (i.e., left lane slope  $k_l$ , right lane slope  $k_r$ , left lane intercept  $b_l$ , right lane intercept  $b_r$ , left lateral offset  $D_l$ , and right lateral offset  $D_r$ ) extracted in the previous processing. The output results were three labels (i.e., normal driving, left lane departure, and right lane departure).  $k_l$ ,  $k_r$ ,  $b_l$ , and  $b_r$  were obtained by using our improved Hough transform with

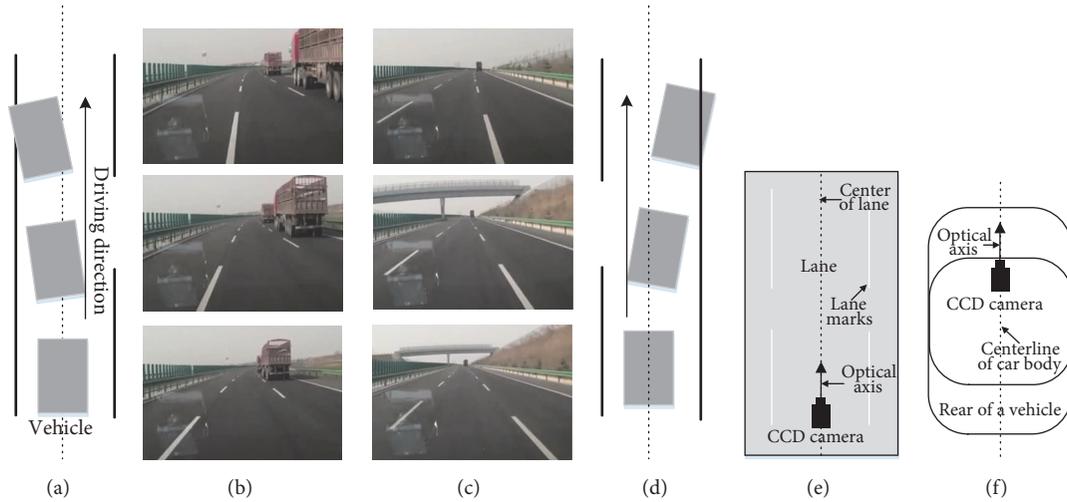


FIGURE 1: Lane departure description: (a) description of a vehicle moving to the left side, (b) change in the driving direction of lane marks when a vehicle approaches the left direction, (c) change in the driving direction of lane marks when a vehicle approaches the right direction, (d) description of a vehicle moving to the right side, (e) description of the relationship between the CCD camera and the lane, and (f) CCD camera installation on a vehicle.

a polar angle constraint (hereinafter referred to as PACTH).  $D_l$  and  $D_r$  are the distances of the vehicle to the left and right lanes; they were computed without internal and external camera parameters [19]. The SSAE neural network is a highly effective classification method and has been widely employed for classification and pattern recognition problems [20] since its proposal. An SSAE consists of multiple layers of SAEs, in which the output of the previous layer is the input of the next layer. Greedy layer-wise SSAEs can pretrain the weights of the entire deep network (DN) by training each layer. Numerous optimization algorithms have been proposed to optimize the parameters of neural networks. The steepest descent algorithm [21] was selected in the current study because of its practicality. A pictorial description of the lane departure of a vehicle is shown in Figures 1(a) and 1(d). The lane boundary description changes when the driving direction of the vehicle deviates from the center of its moving lane in the left or right direction, as shown in Figures 1(b) and 1(c).

To implement the proposed method, we assumed the following: (1) the optical axis of the CCD camera, the lane center, and the centerline of the car body nearly coincide, as shown in Figures 1(e) and 1(f); (2) the processed video sequences are collected on highways and similar roads where the lane curvatures are small; (3) the lane marks are denoted with a color that is brighter than that of the other parts; and (4) the left and right lane marks are parallel to the lane center. With these assumptions, we obtained the following advantages. First, the proposed approach reduces the noise-related effects of a dynamic road scene. Second, the approach allows for lane departure detection without camera-related parameters (i.e., camera calibration is unnecessary). Third, the six input features of our SSAE system can be obtained in the image pixel coordinate without coordinate transformation between world coordinates.

The proposed algorithm involves four steps. Figure 2 illustrates the basic procedure. The first step is image preprocessing, which includes graying, filtering, binarization, and extraction of the region of interest (ROI). These processes are presented in Section 3. The second step is lane detection and matching (presented in Section 4), in which  $k_l$ ,  $k_r$ ,  $b_l$ , and  $b_r$  are obtained. The third step is the calculation of lateral offset (LO)  $D_l$  and  $D_r$  and the design of a softmax classifier according to SSAE. The last step is lane departure detection with three labels using SSAE. The third and last steps are presented in Section 5. The experimental results and comparison with other methods are provided in Section 6. The conclusions are provided in Section 7.

## 2. Related Work

An important problem with captured images is how to effectively obtain robust ROI extraction parts while using image processing techniques to eliminate noise factors. Various feature extraction techniques have been utilized in literature. These techniques include filtering and denoising methods (i.e., mean, median, adaptive [22], and finite impulse response (FIR) filters [23]), gradient operators (Sobel [24], Canny [25], Roberts [26], and Prewitt [27]), binarization methods [28], and vanishing point detection methods [29]. The high-performance approach proposed in the current study was verified through an experimental comparison.

The next step is a robust type of detection and tracking of lane marks. Ruyi et al. [30, 31] used inverse perspective mappings to obtain top-view images for lane boundary detection. Various shape models, which include piecewise linear segments [32], parabolas [19], hyperbolas [33], splines [34, 35], snakes [36], clothoids [37], or their combinations [38], have been applied to determine the mathematical parameters

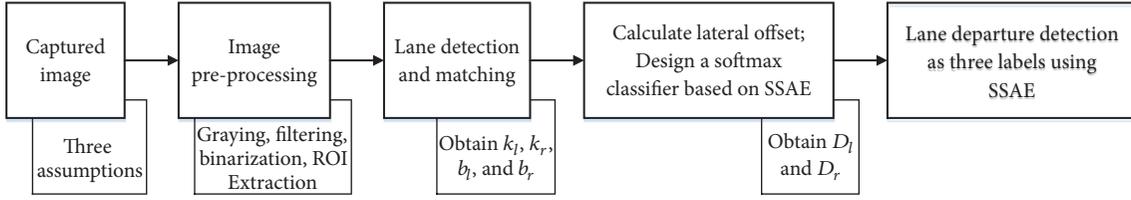


FIGURE 2: Basic procedure of lane departure detection.

that fit lane borders. These models focus on obtaining such parameters but are complex and time consuming. Sehestedt et al. [39] proposed a lane tracking system based on a weak model and particle filtering. Borkar A. et al. [40] applied the Kalman filter to track lane marks. The group intelligent algorithm was also utilized in lane boundary tracking by Cheng [41]. Several lane detectors and trackers, such as GOLD [3], RALPH [4], SCARF [5], MANIAC [6], and LANA [7], have been implemented and cited in literature.

Other researchers worked on LDWS by using several types of lane boundary estimation and lane tracking techniques. LeBlanc et al. [8] proposed an LDWS that predicts a vehicle's path and compared this path with the sensed road geometry to estimate the time-to-lane-crossing (TLC). Kwon et al. [9] developed a vision-based LDWS that considers two warning criteria: LO and TLC. Lee [10] proposed an LDWS that estimates lane orientation through an edge distribution function and identified changes in the driving direction of a vehicle; a modification of this technique [11] includes a boundary pixel extractor to improve robustness. Jung and Kelber [12] also provided an LDWS using LO with an uncalibrated camera, and the change rate was considered. Hsu [13] applied radial basis probability networks as a pattern recognition mechanism that measures and records a vehicle's lateral displacement and its change rate. Then, the trajectory is compared with the training patterns to determine the classification that fits most and to check if the vehicle is about to perform lane departure. Fardi et al. [14] proposed an LDWS based on the ratio of the lane angles and distances of two-lane boundaries. Hsu et al. [15] proposed an LDWS based on angle variation. Kibbel et al. [16] also used lateral position and lateral velocity based on road marking detection for lane departure detection. Kim and Oh [17] proposed an LDWS based on fuzzy techniques by combining LO information with TLC. Wang et al. [18] also applied the fuzzy method to vision-based lane detection and LDWS by using the angle relations of the boundaries.

In sum, the approaches for lane departure detection can generally be classified into two main classes according to whether camera calibration is required or not. The two classifications are (1) lane departure detection with necessary camera calibration to obtain the internal and external parameters of camera and link the world coordinates of 3D real-world and image pixel coordinates via coordinate transformation [8, 9, 13–17] and (2) lane departure detection that relies solely on image pixel coordinates and does not provide accurate estimates of the vehicle's position in the world coordinates [10–12]. Lane departure detection methods

may also be classified into four main classes according to the lane departure discriminant parameters. The four classes are (1) TLC method [8], (2) LO method [12, 13, 16], (3) lane angle variation method [14, 15, 18], and (4) the combination of these three methods [9–11, 17]. With the four assumptions established in this study, we explored the combination of LO, lane slope, and intercept solely in the image pixel coordinates as the feature inputs of our softmax classifier. This scenario was considered because of the important observation that the left and right lane slopes and the intercept change in the image pixel coordinates when lane departures to the left or right side occur. PACTH was utilized to estimate the lane slope and intercept. The proposed approach is unaffected by the lens optics parameters related to the camera, vehicle-related data (e.g., vehicle width and weight), and width of the driving lane. Furthermore, coordinate transformation, a complex road model, curvature, and TLC are unnecessary in the proposed approach.

### 3. Image Preprocessing

The input vision sequences include not only lane information but also nonlane information, such as road obstructions and the sky, which affect lane detection. Therefore, roadway image preprocessing is necessary to highlight the lane lines and detect lanes in real-time with accurate and low-error rates.

**3.1. Graying.** The collected sequences are RGB images. All RGB images were converted into gray images  $I_G$  according to [42]

$$I_G = 0.2989R + 0.5870G + 0.1140B \quad (1)$$

This conversion reduces the computational burden by three times and contributes to real-time image processing.

**3.2. Filtering.** The gray input images  $I_G$  contained a large amount of noise, so noise removal was necessary. A 2D FIR filter was used in our test [23]. This 2D filter's time consumption is much lower than that of other filters, and its filtering result is much better than that of other filters. The FIR filtering result is shown in Figure 3.

**3.3. Binarization.** Binarization is required to highlight the lane of the filtering images. The core problem of binarization is how to determine the optimal threshold; if the threshold is excessively large, then a lane edge point will be missed or some redundant information will be detected. This study

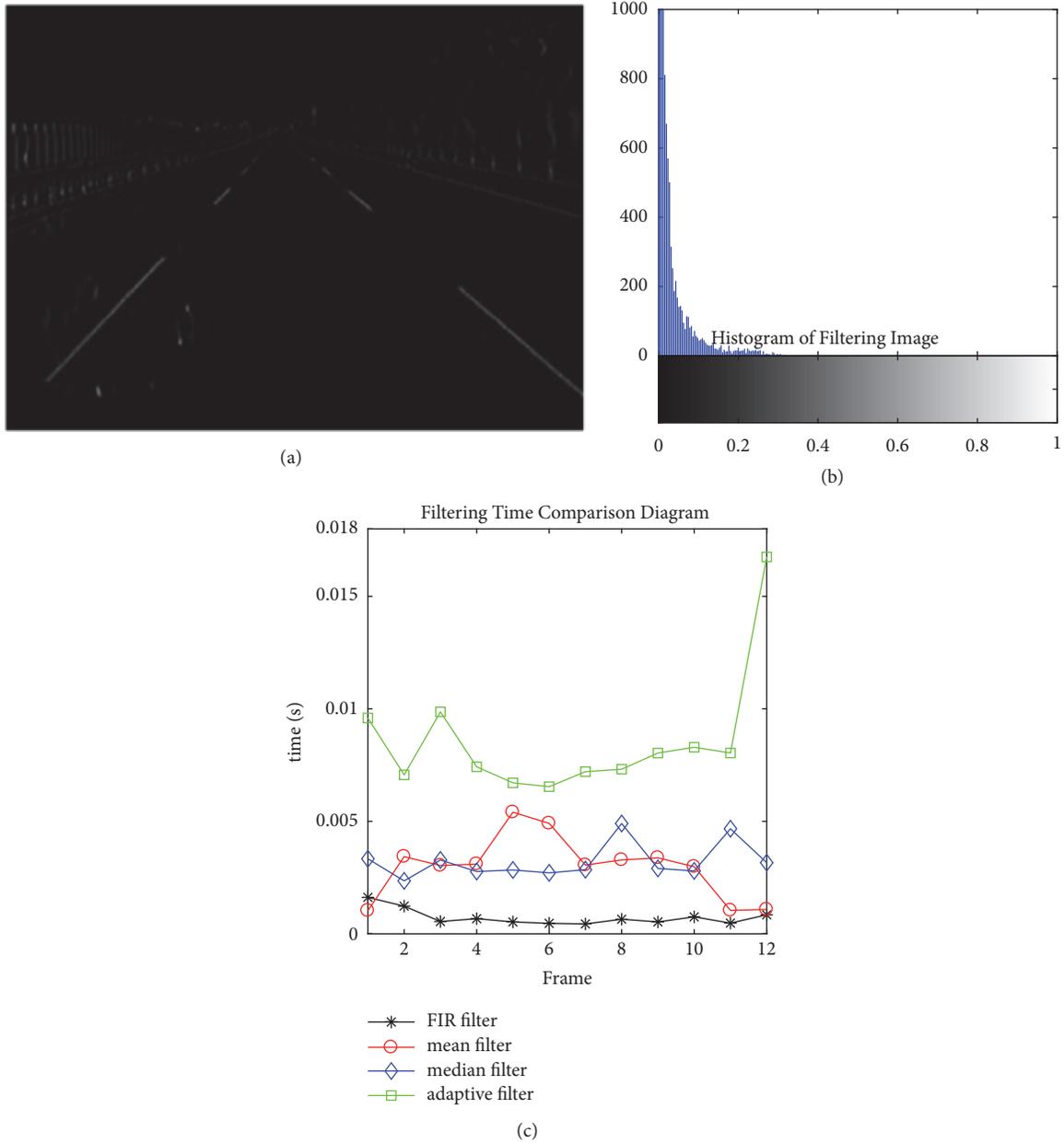


FIGURE 3: FIR filtering result: (a) filtering result image, (b) filtering image histogram, and (c) comparison of filtering time with that of three other filters.

employed adaptive Otsu's method to perform binarization [28]. The method assumes the presence of two pixel distributions (one for the lane and another for the background) and calculates a threshold value  $t$  to minimize the variance between the two pixel distributions according to

$$t = \max [\omega_1(t)(u_1(t) - u)^2 + \omega_2(t)(u_2(t) - u)^2] \quad (2)$$

where  $t$  is the threshold value of the two pixel distributions,  $\omega_1$  denotes the probability of the background pixels in the total image,  $\omega_2$  denotes the probability of the lane pixels in the entire image,  $u_1$  is the average grayscale of  $\omega_1$ ,  $u_2$  is the average grayscale of  $\omega_2$ , and  $u$  is the average grayscale of the entire image. Adaptive Otsu's method performs better

than the other compared thresholding methods, as shown in Figure 4.

**3.4. Vanishing Point Detection to Set Dynamic ROI.** The vanishing point detection method was used to segment the binarization images, set the ROI that contains lane markings, and cut off several parts with useless information, including the sky in the upper half of the image. We only provided the ROI with improved real-time performance and robustness in the follow-up processing.

The lower half of a binarized image was processed through Hough transform, and the left and right lane parameters (slope and intercept) in the image pixel coordinates were



FIGURE 4: Binarization result.

obtained as  $(k_l, b_l)$  and  $(k_r, b_r)$ .  $k_l$ ,  $k_r$ ,  $b_l$ , and  $b_r$  were introduced to (3) to calculate the intersection vertical coordinates  $y$  of two straight lines (i.e., the vanishing point ordinate in the current frame).

$$y = \frac{k_l \cdot b_r - k_r \cdot b_l}{k_l - k_r} \quad (3)$$

where  $k_l$  is the slope of the left lane,  $k_r$  is the slope of the right lane,  $b_l$  is the intercept of the left lane, and  $b_r$  is the intercept of the right lane.

The part where the ordinate  $y$  is maintained is the dynamic ROI. The schematic in Figure 5(a) shows that ROI-II is the dynamic ROI, whereas ROI-I is the truncated part. Figure 5(b) shows the experiment result.

## 4. Lane Detection and Matching

**4.1. Linear Model.** Selecting a good model is necessary in the lane recognition process, which begins with the hypothesis of a road model. A linear model is better than parabolic, hyperbola-pair, and spline models in terms of algorithm simplicity and computational burden. A linear model was selected in this study and adopted for the subsequent processing. Below, we verified the selected model meets the accuracy requirements.

The highway engineering technical standard states that the minimum turning radius on a freeway is 650 m, as used in the method in [43], and the assigned lane curvature radius  $R$  is 650 m. The lane line is replaced with a straight line in the region of length ( $H = 5\text{m}$ ). The resulting error  $\varepsilon$  can be calculated as follows:

$$\varepsilon = R - \sqrt{R^2 - \left(\frac{H}{2}\right)^2} \quad (4)$$

The result is about 4.8 mm less than 5 mm, which is less than one-thousandth of the intercept length. Therefore, a small stadia highway curve line can be approximated as a straight line. The linear model can thus meet lane detection accuracy requirements for highways.

**4.2. PACHT.** The classic Hough transform utilizes point-line duality to convert the straight line detection problem in the

image space to a cumulative peak problem of the point in the Hough domain by using

$$\rho = x \cos \theta + y \sin \theta \quad (5)$$

where  $\rho$  is the polar radius denoting the normal distance of the line from the image space origin,  $\theta$  is the polar angle that denotes the angle between the  $x$ -axis and the normal line,  $(x, y)$  is the coordinate in the image space,  $\rho$ - $\theta$  denotes the Hough domain, and  $x$ - $y$  denotes the image space.

Many nontarget lane edge points (e.g., trees, road signs, and other interference points) are commonly noted after binarization. This study improved the traditional Hough transform by restraining  $\rho$  and  $\theta$  values to limit the scope of the voting space and minimize the interference.

We assumed that  $\rho$  and  $\theta$  of the left lane are  $\rho_l$  and  $\theta_l$  and those of the right lane are  $\rho_r$  and  $\theta_r$ . Through an analysis of many image samples, we defined the constrained region of target points as  $\theta_{l1} \leq \theta_l \leq \theta_{l2}$ ,  $\theta_{r1} \leq \theta_r \leq \theta_{r2}$ ,  $\rho_{l1} \leq \rho_l \leq \rho_{l2}$ , and  $\rho_{r1} \leq \rho_r \leq \rho_{r2}$ , where  $\theta_{l1}$  and  $\theta_{l2}$  are the upper and lower limits of the polar angle  $\theta_l$  of the left lane,  $\theta_{r1}$  and  $\theta_{r2}$  are the upper and lower limits of the polar angle  $\theta_r$  of the right lane,  $\rho_{l1}$  and  $\rho_{l2}$  are the upper and lower limits of the polar radius  $\rho_l$  of the left lane, and  $\rho_{r1}$  and  $\rho_{r2}$  are the upper and lower limits of the polar radius  $\rho_r$  of the right lane. The PAC area diagram is shown in Figure 6(a). Figure 6(b) shows that many interference points in the areas involved in the follow-up processing were effectively removed.

**4.3. Lane Detection and Matching.** Use PACHT to detect lanes. Although the aforementioned processing technique has set the region, a frame contains at least two-lane lines; it can even contain four, six, or more lanes on a high-grade road in real lane detection. We assumed that a maximum of  $n$  lines are detected per frame, and all lines constitute a straight line repository  $\{(\rho_{ik}, \theta_{ik})\}$ ,  $i = 1, 2, \dots, n$  to perform the following matching operation.

The left and right lane information detected in one frame is  $[(\rho_1, \theta_1), (\rho_2, \theta_2)]$ . Assuming that the maximum number of lanes is  $n$ , a count value is set for each lane and all count values are represented by a counter vector of  $1 \times n$ . The lane lines identified in the current frame are then matched with those in the repository. If a line in the repository matches an input line, then it is replaced with the input line and the count number increases by one. Otherwise, the count number decreases by one. The value of the counter that corresponds to each line is then checked if the count values are saturated. Given that the lines selected in the repository are matched at least  $N$  times, the line matching process is also the lane tracking process.

The lane detection and matching process is presented as follows:

(i) The repository  $\{(\rho_{ik}, \theta_{ik})\}$  and match counter  $N_i$  are initialized.  $\{(\rho_{ik}, \theta_{ik})\}$  is assigned to an all-zero vector:  $N_i = 0$ ,  $k = 0$ ,  $i = 1, 2, \dots, n$ .

(ii) The input lines  $\{(\rho_{ik}, \theta_{ik})\}$  are obtained by PACHT at time  $k$ ,  $i = 1, 2, \dots, n$ .

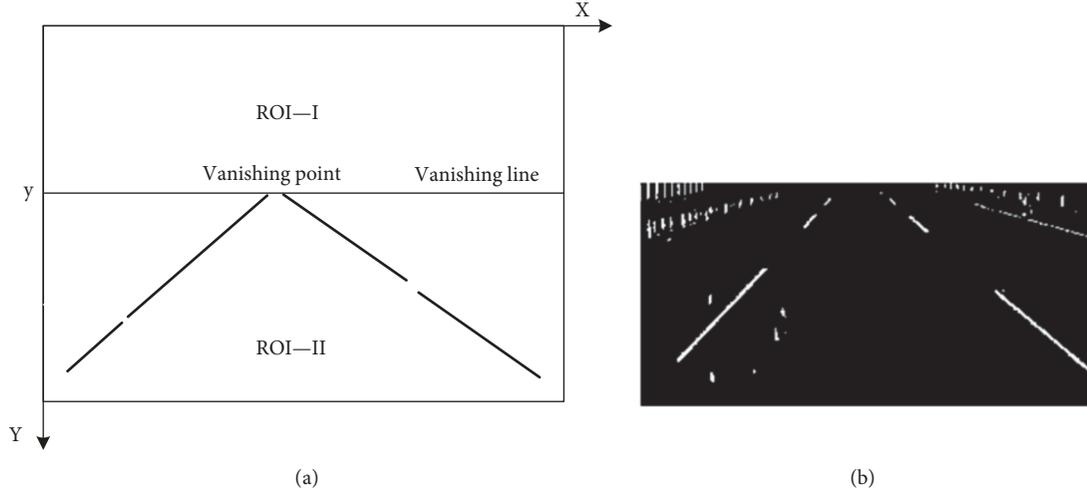


FIGURE 5: Vanishing point detection to set the dynamic ROI: (a) vanishing point detection schematic and (b) ROI of the experiment.

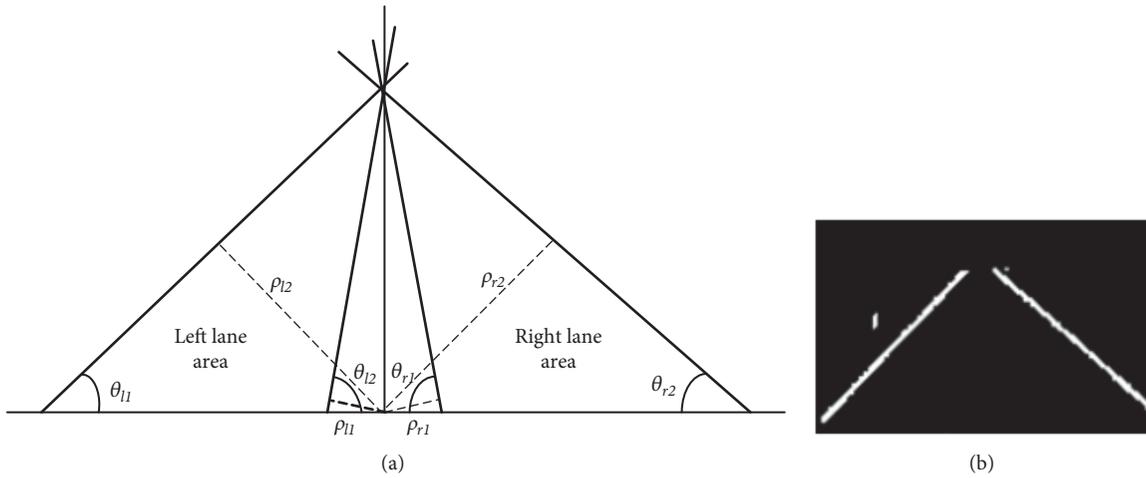


FIGURE 6: PAC area: (a) PAC area diagram and (b) PACTH experimental result.

(iii) The distances  $D_i$  between the lines identified in the current frame and those in the repository at time  $k - 1$  are calculated one by one according to the following formula:

$$D_i = \left| \rho_{ik} - \rho_{j(k-1)} \right| + \left| \theta_{ik} - \theta_{j(k-1)} \right| \cdot \text{Size}_{image} \quad (6)$$

where  $\rho_{ik}$  is the polar radius of lines detected in the current frame,  $\theta_{ik}$  is the polar angle of lines detected in the current frame,  $\rho_{j(k-1)}$  is the polar radius of lines in the repository at time  $k - 1$ ,  $\theta_{j(k-1)}$  is the polar angle of lines in the repository at time  $k - 1$ , and  $\text{Size}_{image}$  is the width of the road image;  $i, j = 1, 2, \dots, n$ .

(iv) The best matches between the current lines and those in the repository are then identified. If  $D_i < D_{\min}$ , then the matching is successful, the input lines replace the lines that correspond to them, and the count value is increased by one. Otherwise, the count number decreases by one until it saturates.  $D_{\min}$  is a predetermined matching distance threshold.

(v) The count number of each line is then checked. If  $N_i < 0$ , let  $N_i = 0$ ; if  $N_i > N$ , let  $N_i = N$ . This condition continues to the next cycle until the image sequences end.

The counting process of the counter is shown in Figure 7, which shows that two active lanes are detected in the system. We set  $N$  to 25 in this process. Two horizontal lines were observed at a value of 25, and the two lines indicate two detected lanes. The vehicle changed its driving lanes at frames 250 and 750, where the curve cross phenomenon occurred.

Figure 8 shows the lane detection and matching flow chart.

Convert the two lanes from polar to Cartesian space. Left and right lane slopes  $k_l$  and  $k_r$ , and lane intercepts  $b_l$  and  $b_r$  are obtained.

## 5. Lane Departure Detection Based on SSAE

5.1. LO Computation. We computed LO according to the two lanes' detection and matching result. We regarded the vehicle as a rectangular rigid object in our model, so LO

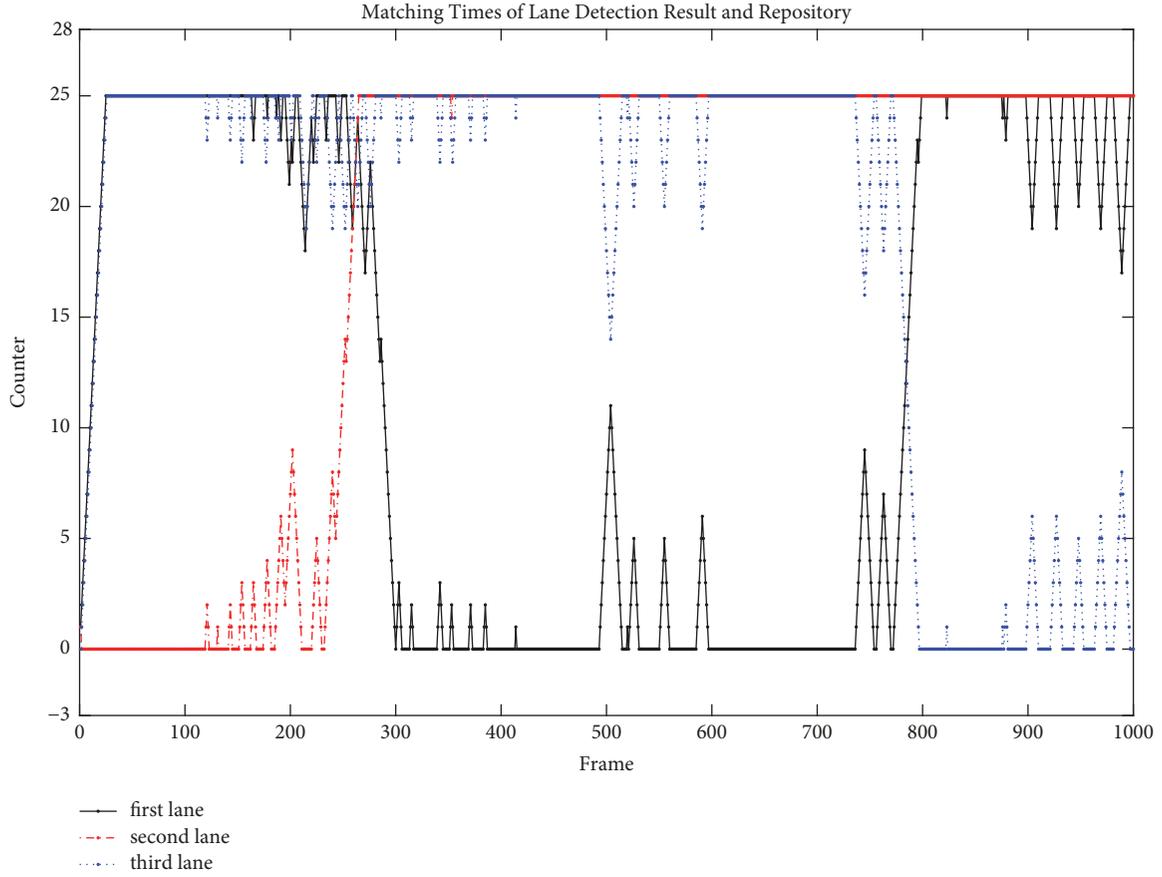


FIGURE 7: Lane detection and matching result.

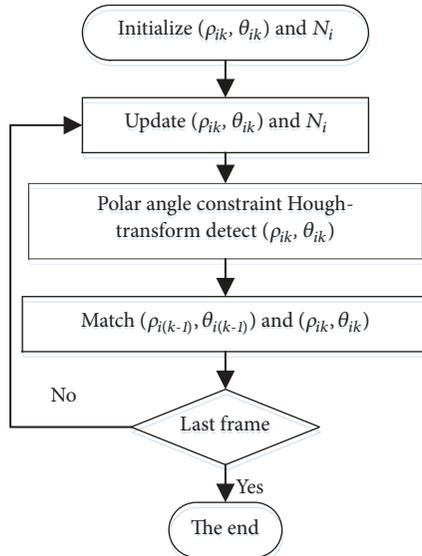


FIGURE 8: Lane detection and matching flow chart.

is the distance between the corner of the rigid body and two lanes. Strict calculation of this distance has a hysteresis effect when the onset time, which is the driver's initial action time, is considered. Therefore, we presumed a virtual lane

that is narrower than the actual lane. Our LO is the distance between the center and virtual lane. Figure 9(a) presents the description of our LO in the world coordinates, where  $D_l$  is the left LO,  $D_r$  is the right LO, and  $R$  is the width of the reserved area.

Given that the images captured by the CCD camera are projected through the perspective,  $D_l$  and  $D_r$  can be directly calculated in the image pixel coordinates without any intrinsic or extrinsic camera parameters. Figure 9(c) shows the perspective projection image of Figure 9(a). The upper left corner is set as the coordinate origin. Figures 9(b), 9(c), and 9(d) show the intersections  $(u_i, v_i)$ ,  $i = l, r$ , where the left (right) lane intersects with the image bottom boundary.  $D_l$  and  $D_r$  can be computed with

$$D_l = \frac{W}{2} - \left(R + \frac{V}{2}\right) - u_l \tag{7}$$

$$D_r = u_r - \left(R + \frac{V}{2}\right) - \frac{W}{2}$$

where  $D_l$  is the left LO,  $D_r$  is the right LO,  $W$  is the width of the image,  $R$  is the width of the reserved area,  $V$  is the width of the vehicle,  $u_l$  is the abscissa of the intersection of the left lane and image bottom boundary, and  $u_r$  is the abscissa of the intersection of the right lane and image bottom boundary.

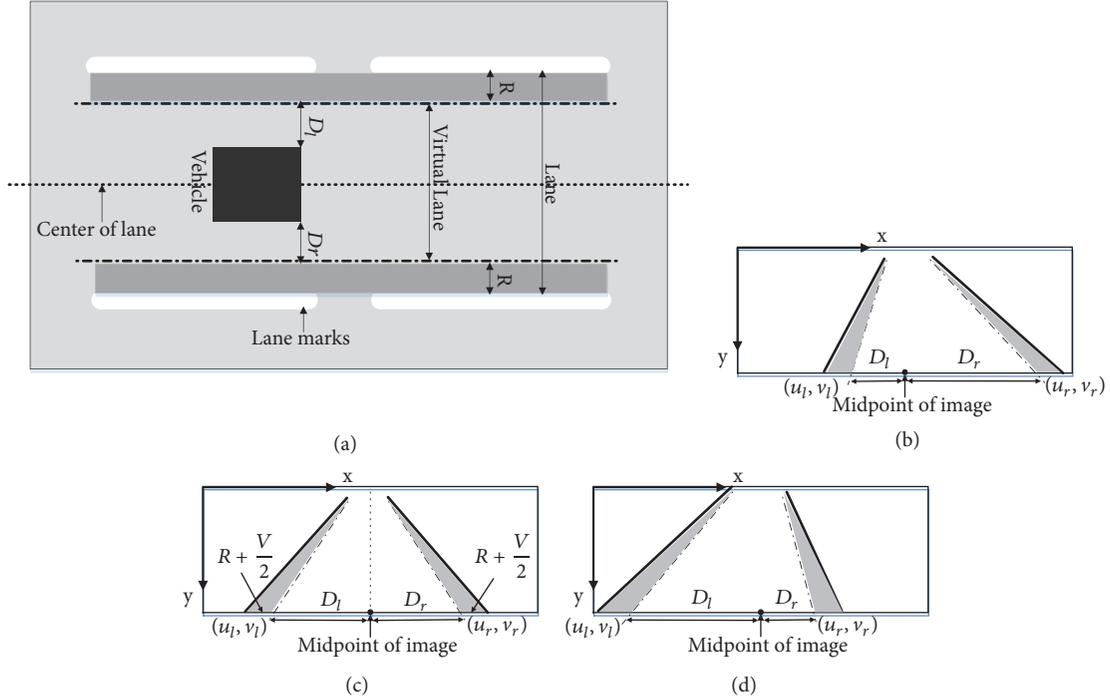


FIGURE 9: LO computation: (a) description of our LO in the world coordinates, (b) description of our LO in the image pixel coordinates to the left side, (c) description of our LO in the image pixel coordinates of normal driving, and (d) description of our LO in the image pixel coordinates to the right side.

Figure 9(a) shows that the width of vehicle  $V$  is approximately considered. Figures 9(b), 9(c), and 9(d) show that  $R+V/2$ , which is a constant, can be regarded as a whole due to mathematical transformation. LO is not influenced by the parameters of lens optics, vehicle type, width of the traveling lane, and localization of lane marks throughout its computation process.

**5.2. SSAE.** Neural networks have been applied in many classification problems and have obtained favorable results. This study proposed a novel neural network model called SSAE to classify lane departure. SSAE initializes the parameters and then utilizes feed-forward and back-propagation by a batch gradient descent algorithm to identify the minimum cost function for obtaining the global optimum parameters. The progression, which involves unsupervised learning, is called pretraining. Afterward, fine-tuning is employed to obtain improved results. SSAE possesses a powerful expression and enjoys all the benefits of DNs. Furthermore, the model can accomplish hierarchical grouping or part-whole decomposition of the input.

Unlike other neural networks, an SAE neural network is an unsupervised learning algorithm that does not require labeled training examples. Applying back-propagation makes the target values equal to the inputs. A schematic of an SAE with  $m$  inputs and  $n$  units in the hidden layer is shown in Figure 10.

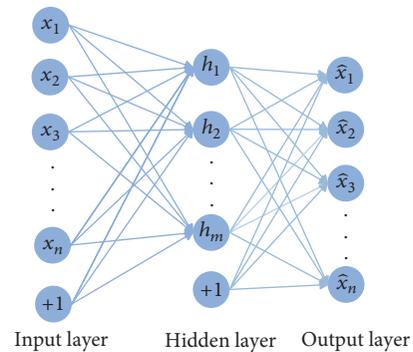


FIGURE 10: Schematic of an SAE.

The overall cost function in an AE is as follows:

$$J(W, b) = \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{2} \|h_{W,b}(x^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \quad (8)$$

where  $m$  is the number of inputs,  $h_{W,b}(x^{(i)})$  is the output of the activation function when the raw input is  $x^{(i)}$ ,  $y^{(i)}$  is the raw output,  $\lambda$  is the relative importance of the second term, and  $W_{ji}^{(l)}$  is the weight associated with the connection between unit  $j$  in layer  $l$  and unit  $i$  in layer  $l+1$ .

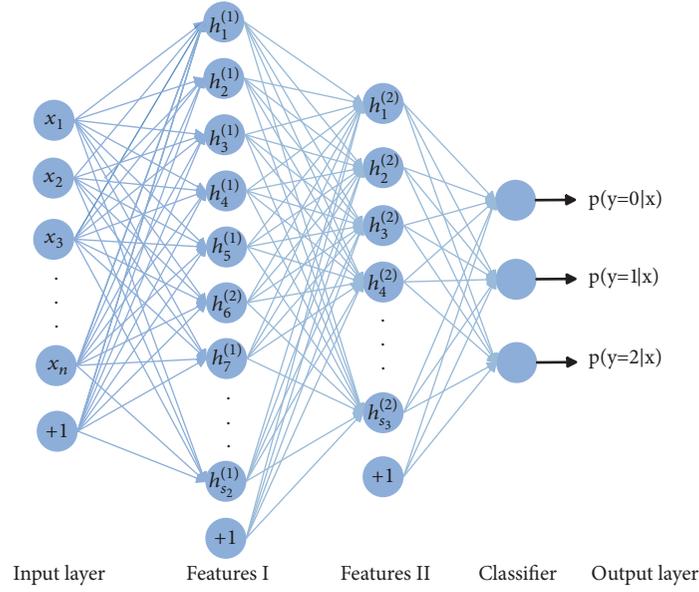


FIGURE 11: Establishment of an SSAE composed of two SAEs.

Several correlations of the input features in an SAE can be determined by imposing constraints on the network. Sparsity  $\rho$  is imposed to constrain the hidden units as follows:

$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m [\alpha_j^2(x^{(i)})] \quad (9)$$

where  $\alpha_j^2$  denotes the activation of hidden unit  $j$  in the SAE when the network is given a specific  $x^{(i)}$ . The parameter is the average activation of the hidden unit  $j$ . Moreover, we set the two parameters equal as follows:

$$\hat{\rho}_j = \rho \quad (10)$$

This parameter commonly has a small value (e.g., 0.05). Thus, the activation of the hidden unit must be close to 0. A penalty term is added to penalize the situation of  $\hat{\rho}_j$  deviating significantly from  $\rho$  to optimize the objective. The penalty term is as follows:

$$\sum_{j=1}^{s_2} KL(\rho \parallel \hat{\rho}_j) \quad (11)$$

The overall cost function is

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{s_2} KL(\rho \parallel \hat{\rho}_j) \quad (12)$$

The SSAE consists of multiple SAE layers. The outputs of each layer are wired to the inputs of the succeeding layers. The detailed establishment of SSAE, pretraining, and fine-tuning were conducted in the following steps.

*Step 1.* An SAE was trained on raw input  $x^{(k)}$  to learn primary features  $h^{(1)(k)}$ . The structure of the first SAE is  $[n, s_2, n]$ ,

which corresponds to  $n$  inputs,  $s_2$  units in hidden layers, and  $n$  outputs, as shown in Figure 11.

*Step 2.* The trained SAE was adopted to obtain primary feature activations  $h^{(1)(k)}$  for each input  $x^{(k)}$ .

*Step 3.*  $h^{(1)(k)}$  was utilized as the “raw input” to the second SAE to learn the secondary features  $h^{(2)(k)}$ . The second SAE structure is described by  $[s_2, s_3, s_2]$ , which corresponds to  $s_2$  inputs,  $s_3$  units in hidden layers, and  $s_2$  outputs.

*Step 4.*  $h^{(1)(k)}$  was fed to the second SAE to obtain the secondary feature activations  $h^{(2)(k)}$  for each  $h^{(1)(k)}$ .

*Step 5.* The secondary features were regarded as “raw inputs” to a sigmoid classifier and trained to map the secondary features to digit labels.

*Step 6.* All three layers were combined to form an SSAE with two hidden layers and a classifier layer.

*Step 7.* Back-propagation was conducted to improve the results by adjusting the parameters of all layers simultaneously through a process called fine-tuning.

*Step 8.* Step 7 was performed repeatedly until the set training times were attained.

### 5.3. Lane Departure Detection Process Description

*5.3.1. Entire Process Description.* Vision-based lane departure detection based on SSAE is an essential pattern recognition system (Figure 2). This system consists of data collection and

TABLE 1: Six parameter changes from frames 40 to 44.

Frame No.	$D_l$	$D_r$	$k_l$	$k_r$	$b_l$	$b_r$
40	159	160	1	-1	-81	229
41	159	160	1	-1	-81	229
42	159	160	1	-1	-81	229
43	159	160	1	-1	-79	229
44	159	160	1	-1	-82	227

acquisition, feature extraction, and lane departure identification and prediction.

**5.3.2. Basic Parameters.** One of the most important parameters during the reduction of the cost function by batch gradient descent is learning rate  $\alpha$ . If this rate is excessively large, it would result in an excessively large step size, and the gradient descent can overshoot the minimum and deviate increasingly farther from the minimum. If this rate is exceedingly small, it would slow down the computing speed required to reduce the cost function. Another drawback is the potential trapping in the local optima and potential resultant inability to reach the global optimal solution. We set  $\alpha$  to  $5 \times 10^{-6}$  in the proposed model.

The other parameter values, such as the sample number in the batch training  $MaxIter$ , sparsity parameter  $\rho$ , and sparse penalty factor weight  $\beta$ , can be obtained by changing only one parameter and fixing the others. We set  $MaxIter = 300$ ,  $\rho = 0.3$ ,  $\beta = 3$ , and the number of SAEs to 2.

**5.3.3. Our Feature Input.** The description and feature extraction in the previous chapters indicated that the feature inputs are left lane slope  $k_l$ , right lane slope  $k_r$ , left lane intercept  $b_l$ , right lane intercept  $b_r$ , left LO  $D_l$ , and right LO  $D_r$ , all of which change when the vehicle deviates from the lane. When the vehicle approaches the left boundary,  $k_l$ ,  $b_l$ , and  $D_l$  decrease simultaneously, whereas  $k_r$ ,  $b_r$ , and  $D_r$  increase. Conversely, when the vehicle approaches the right boundary,  $k_r$ ,  $b_r$ , and  $D_r$  decrease simultaneously, whereas  $k_l$ ,  $b_l$ , and  $D_l$  increase. Table 1 shows the six parameter changes in several frames.

## 6. Experiment and Result

The proposed system was evaluated with images captured by a CCD camera mounted on a vehicle. While driving the vehicle, road tests were conducted on structured roads paved with asphalt or cement and with lane marks. The number of image sequences in the tests was 5309 frames, and the image size was  $320 \times 240$  pixels.

**6.1. Lane Detection and Matching Experiment.** The experimental lane detection and matching results of the proposed system are presented in Figure 12. Figure 12(a) presents the lane detection and matching result of a sequence from frames 1259 to 1278. Figure 12(b) shows the values of left lane slope  $k_l$  and right lane slope  $k_r$  for the 20 frames;  $k_l$  is approximately 1, whereas  $k_r$  is approximately -1 in the case of normal driving.

Figure 12(c) shows the values of left lane intercept  $b_l$  and right lane intercept  $b_r$  for the 20 frames;  $b_l$  is approximately 80 pixels, whereas  $b_r$  is approximately 220 pixels during normal driving. When lane departure occurs, the values of  $k_l$ ,  $k_r$ ,  $b_l$ , and  $b_r$  change.

Lane detection and matching in our experiment failed in 176 frames because of a large amount of white noise on the road, so the recognition rate of the proposed method is 96.69%. The real-time performance is approximately 15 ms in a computer with 3.88 GB of RAM.

**6.2. Comparison with Different Hidden Layer Structures.** We determined the number of first and second AE hidden units in this experimental procedure. We set the number of the first AE hidden units to vary from 5 to 1,000 at intervals of 25. The number of the second AE hidden units varied from 5 to 200 at intervals of 5. A total of 1,600 different structures were assessed to determine the most suitable structure for the proposed model. The recognition accuracy results are shown in Figure 13.

Figure 13 shows the accuracy of a certain combination in which the number of the first SAE hidden units is 205 and the number of the second SAE hidden units is 160. The maximum value of 90.74% is reached. Hence, we finalized our lane departure detection model as [6, 205, 160, 3], followed by the structure of [6, 155, 150, 3] (90.67%) and the structure of [6, 180, 195, 3] (90.61%).

**6.3. Comparison of with and without Pretraining.** Pretraining plays an important role in the recognition of neural networks. The experimental results showed that the accuracy rates with pretraining are 90.74%, and the accuracy rates without pretraining are 88.69%. Approximately 2.05% was improved by pretraining. This result is relatively easy to understand. The network can only fine-tune the parameters without pretraining, which generally comprise several random initial points. This notion explains why the network may be confined to the local optima. The network can be fine-tuned on the pretrained parameters to obtain the global optima. Thus, pretraining the SSAE is necessary.

**6.4. Comparison with Different Classifiers.** We utilized the best model [6, 205, 160, 3] to detect lane departure given the basic parameters, as shown in Figure 14. Figure 14(a) shows the value change of right LO  $D_r$  from frames 187 to 260 where right departure occurs. Figure 14(c) shows the right departure detection result.  $D_r$  decreases gradually before frame 251 and

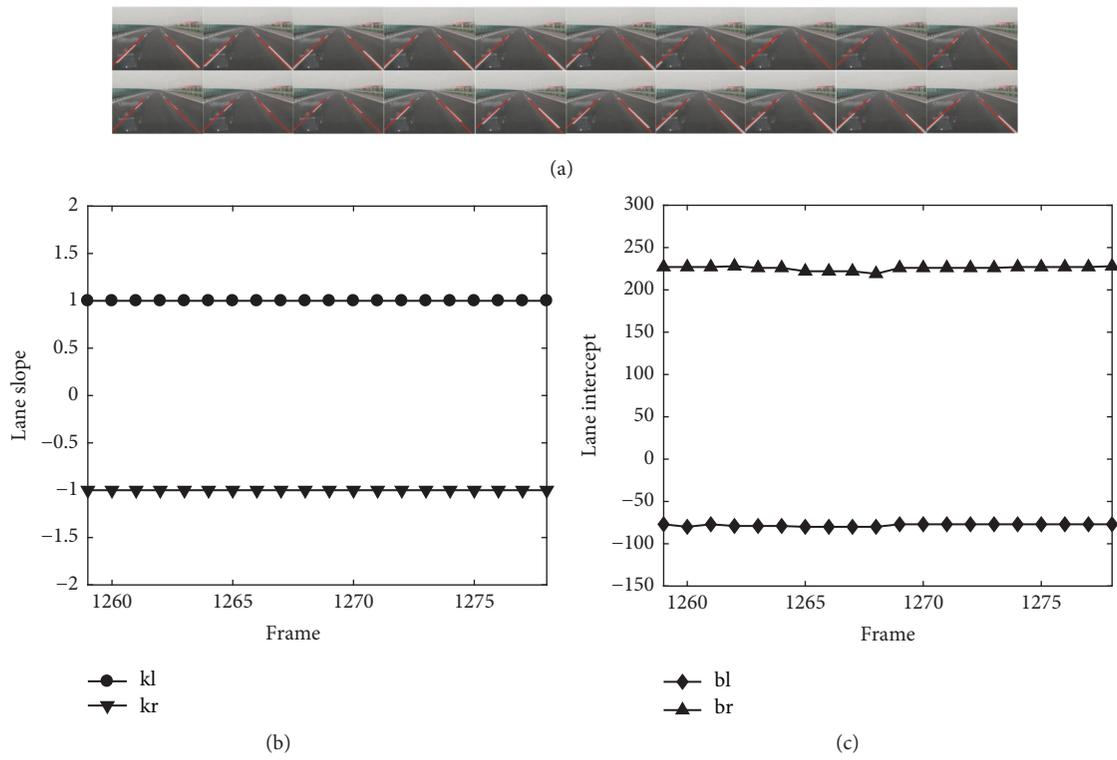


FIGURE 12: Lane detection and matching results: (a) recognition result mark in the original images, (b) values of the left and right lane slopes, and (c) values of the left and right lane intercepts.

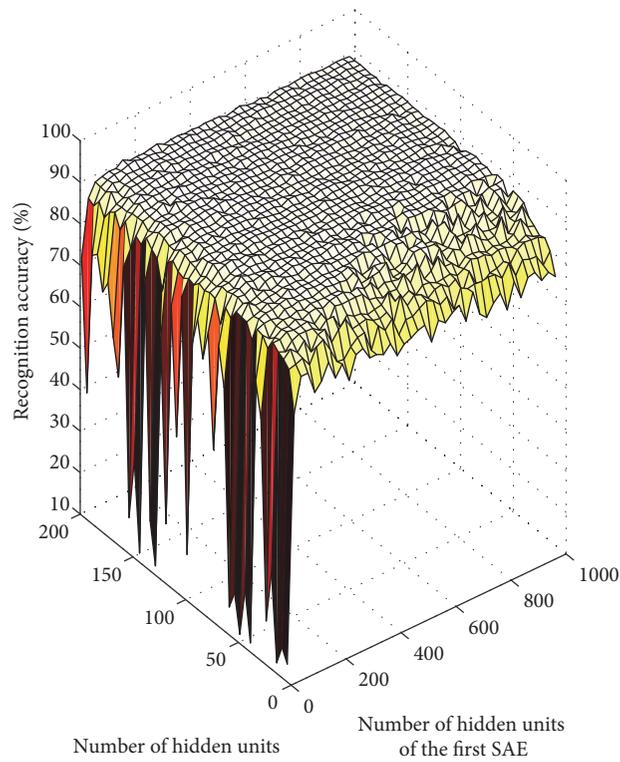


FIGURE 13: Recognition accuracy rates for different hidden layer structures.

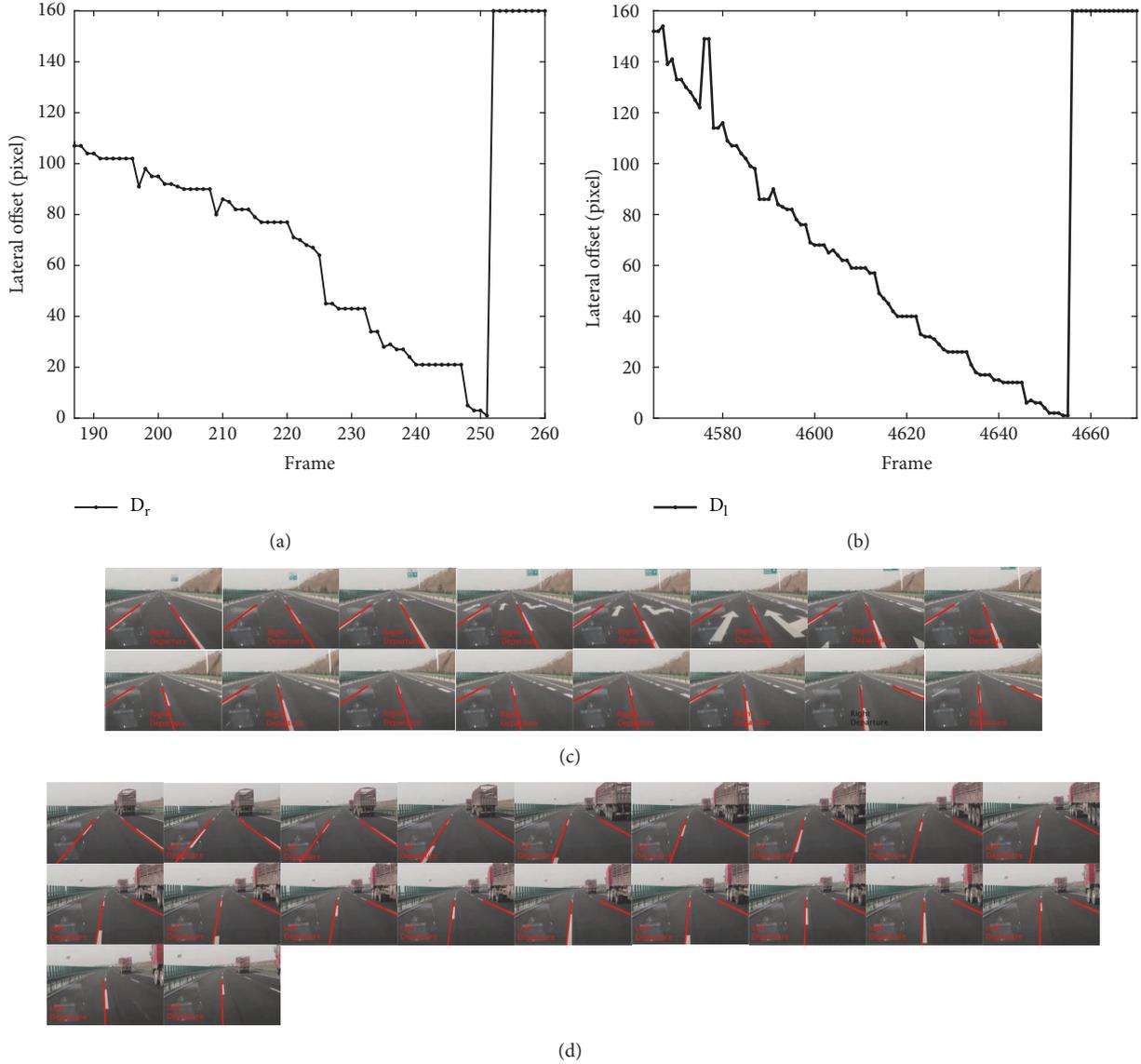


FIGURE 14: Results of right and left lane departure detection. (a) Value change of right LO  $D_r$  from frames 187 to 260 where right departure occurs. (b) Value change of left LO  $D_l$  from frames 4565 to 4670 where left departure occurs. (c) Right departure detection result. (d) Images of left departure detection.

then a step change occurs because the vehicle offsets to a certain extent and changes lanes, as shown in the last two images of Figure 14(c). Figure 14(b) shows the value change of left LO  $D_l$  from frames 4565 to 4670 where left departure occurs. Figure 14(d) shows the images of the left departure detection.  $D_l$  also decreases gradually before frame 4655. A step change also occurs because the vehicle offsets to a certain extent and changes lanes, as shown in the last two images of Figure 14(d).

Then, we compared our method with other classifications. The first comparison experiment only covered LO with no other parameters (i.e.,  $k_l$ ,  $k_r$ ,  $b_l$ , and  $b_r$ ). This method failed to detect 926 frames, and the detection accuracy rate is 81.59%, as shown in Table 2. The SSAE is better than LO by 9.15% because regular changes in  $k_l$ ,  $k_r$ ,  $b_l$ , and  $b_r$  emerge

when lane departures occur. Therefore,  $k_l$ ,  $k_r$ ,  $b_l$ , and  $b_r$ , which are SSAE inputs, are required. The second comparison experiment covered the classifier function changes from softmax to sigmoid in SSAE. The sigmoid performed worse than the softmax and only attained 71.63% accuracy, as shown in Table 2. The third experiment compared our algorithm with SAE-DN, whose weights were pretrained only by one SAE. SAE-DN attained 86.26% accuracy, which is 4.48% lower than that of SSAE, as shown in Table 2. The fourth experiment compared SSAE with a linear separable support vector machine (SVM-LS). The accuracy rate of SVM-LS is 81.78%, which means that it failed to detect 968 frames, as shown in Table 2. The last experiment compared SSAE with a nonlinear support vector machine (SVM-NL). It failed to detect 559 frames and attained 89.47% accuracy, which is

TABLE 2: Lane departure detection accuracy of six algorithms.

Algorithm	SSAE no pre-training	LO	Sigmoid	SAE-DN	SVM-LS	SVM-NL	SSAE
Total Number	5309 frames						
Failed	600	977	1506	729	968	559	491
Detection	frames	frames	frames	frames	frames	frames	frames
Accuracy	88.69%	81.59%	71.63%	86.26%	81.78%	89.47%	90.74%

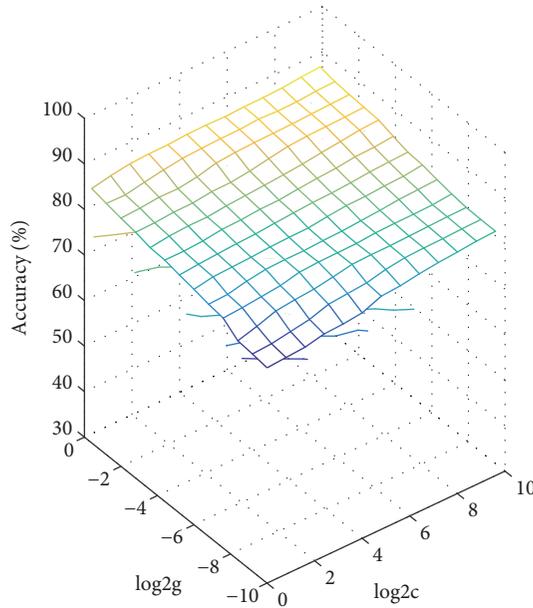


FIGURE 15: Parameter selection result of SVM-NL.

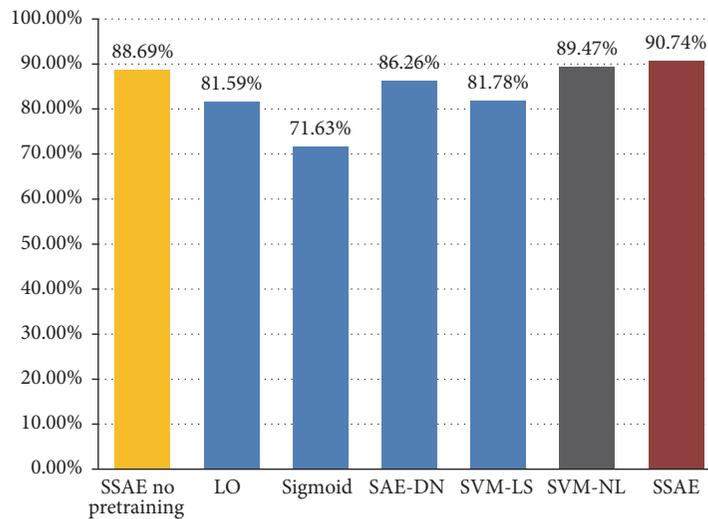


FIGURE 16: Accuracy rate of seven different classifiers.

1.27% lower than that of SSAE, as shown in Table 2. The parameter selection result of SVM-NL is shown in Figure 15.

To visually present that SSAE is superior to other classifiers, a bar chart is drawn in Figure 16. The SSAE method, represented by the red column, obtained the highest recognition rate of 90.74%. It is followed by SVM-NL (89.47%), represented by the black column, and the SSAE with no

pretraining (88.69%), represented by the yellow column. The other methods have a lower accuracy rate.

### 7. Conclusion

The fundamental issue in a lane departure detection system is to robustly identify lane boundaries and design a robust

lane departure detection algorithm. The proposed method of vision-based lane departure detection based on SSAE is relatively successful from these perspectives. Our lane detection and matching method achieved a high recognition rate of 96.69%, and its real-time performance was good. The parameter setting and performance of the SSAE algorithm were proven by the experimental results under different conditions. The proposed method obtained a high accuracy rate of 90.74%. Furthermore, we compared the performance of the proposed approach with that of five other algorithms and proved the superiority of the SSAE model over the competing models.

This paper presents SSAE DNs for the lane departure detection problem. We selected right and left LO, lane slope, and lane intercept as the feature inputs of SSAE based on our observations and experiment. As a result, a satisfactory experimental result was obtained. More importantly, this study determined approaches to obtain feature inputs and perfect solutions. In the future, we can obtain more feature inputs to improve pattern recognition accuracy. Collecting more video sequences is also necessary to achieve convincing experimental results.

The proposed algorithm was processed at a speed of 61 frames per second on a Pentium PC (3.00 GHz, 3.88 GB of RAM).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Open Foundation of State Key Laboratory of Automotive Simulation and Control (China, Grant no. 20161105).

## References

- [1] World Health Organization. "Global status report on road safety 2015". (2015), [http://www.who.int/violence\\_injury\\_prevention/road\\_safety\\_status/2015/en/](http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/).
- [2] Fatality Analysis Reporting System (FARS). <https://www.nhtsa.gov/research-data/fatality-analysis-reporting-system-fars>.
- [3] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.
- [4] D. Pomerleau, "RALPH: Rapidly Adapting Lateral Position Handler," in *Proceedings of the 1995 Intelligent Vehicles Symposium*, pp. 506–511, September 1995.
- [5] J. D. Crisman and C. E. Thorpe, "SCARF: A Color Vision System that Tracks Roads and Intersections," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 49–58, 1993.
- [6] M. Jochem Todd, D. A. Pomerleau, and C. E. Thorpe, "MANIAC: A Next Generation Neurally Based Autonomous Road Follower," in *Proceedings of the International Conference on Intelligent Autonomous Systems*, 1993.
- [7] C. Kreucher and S. Lakshmanan, "LANA: A lane extraction algorithm that uses frequency domain features," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 343–350, 1999.
- [8] D. J. Leblanc, "CAPC: A Road-Departure m," *Control Systems IEEE*, vol. 6, pp. 61–71, 1996.
- [9] W. Kwon, J.-W. Lee, and D. Shin, "Experiments on decision making strategies for a lane departure warning system," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation, ICRA99*, pp. 2596–2601, May 1999.
- [10] J. W. Lee, "A machine vision system for lane-departure detection," *Computer Vision and Image Understanding*, vol. 86, no. 1, pp. 52–78, 2002.
- [11] J. W. Lee and U. K. Yi, "A lane-departure identification based on LBPE, Hough transform, and linear regression," *Computer Vision and Image Understanding*, vol. 99, no. 3, pp. 359–383, 2005.
- [12] C. R. Jung and C. R. Kelber, "A lane departure warning system using lateral offset with uncalibrated camera," in *Proceedings of the Intelligent Transportation Systems, 2005. Proceedings IEEE*, pp. 102–107, 2005.
- [13] C. S. Hsu, *The decision strategies of irregular vehicle behavior warning system [MS thesis]*, Graduate Institute of Civil Engineering, National Taiwan University, Taipei, Taiwan, 2003.
- [14] B. Fardi, U. Scheunert, H. Cramer, and G. Wanielik, "A new approach for lane departure identification," in *Proceedings of the IEEE IV2003 Intelligent Vehicles Symposium. Proceedings*, pp. 100–105, Columbus, OH, USA.
- [15] H. Pau-Lo et al., "The adaptive lane-departure warning system," in *Proceedings of the SICE 2002. 41st SICE Annual Conference*, pp. 2867–2872, Osaka, Japan.
- [16] J. Kibbel, W. Justus, and K. Fürstenberg, "Lane estimation and departure warning using multilayer laserscanner," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 777–781, Austria, September 2005.
- [17] S.-Y. Kim and S.-Y. Oh, "A driver adaptive lane departure warning system based on image processing and a fuzzy evolutionary technique," in *Proceedings of the 2003 IEEE Intelligent Vehicles Symposium, IV 2003*, pp. 361–365, USA, June 2003.
- [18] J.-G. Wang, C.-J. Lin, and S.-M. Chen, "Applying fuzzy method to vision-based lane detection and departure warning system," *Expert Systems with Applications*, vol. 37, no. 1, pp. 113–126, 2010.
- [19] C. R. Jung and C. R. Kelber, "A lane departure warning system based on a linear-parabolic lane model," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 891–895, University of Parma, Parma, Italy, June 2004.
- [20] F. Gu, F. Flórez-Revuelta, D. Monekoso, and P. Remagnino, "Marginalised stacked denoising autoencoders for robust representation of real-time multi-view action recognition," *Sensors*, vol. 15, no. 7, pp. 17209–17231, 2015.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS '12)*, pp. 1097–1105, Lake Tahoe, Nev, USA, December 2012.
- [22] G. Gupta, "Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter," *International Journal of Soft Computing & Engineering*, vol. 1, no. 5, 2011.
- [23] P. Chandrasekar and J.-Y. Ryu, "Design of programmable digital FIR/IIR filter with excellent noise cancellation," *International Journal of Applied Engineering Research*, vol. 11, no. 15, pp. 8467–8470, 2016.
- [24] S. Gupta and S. Mazumdar, "Sobel edge detection algorithm," *International journal of computer science and management Research*, vol. 2, no. 2, pp. 1578–1583, 2013.

- [25] J. A. Canny, "Computational Approach to Edge Detection," *IEEE Computer Society*, 1986.
- [26] Rashmi, M. Kumar, and R. Saxena, "Algorithm and Technique on Various Edge Detection : A Survey," *Signal Image Processing*, vol. 03, pp. 65–75, 2013.
- [27] I. Corke Peter, "Machine Vision," *Moldes*, vol. 3, pp. 0219–6131, 2000.
- [28] R. Farrahi Moghaddam and M. Cheriet, "AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization," *Pattern Recognition*, vol. 45, no. 6, pp. 2419–2431, 2012.
- [29] W. Ding and Y. Li, "Efficient vanishing point detection method in complex urban road environments," *IET Computer Vision*, vol. 9, no. 4, pp. 549–558, 2015.
- [30] J. Ruyi, K. Reinhard, V. Tobi, and W. Shigang, "Lane detection and tracking using a new lane model and distance transform," *Machine Vision and Applications*, vol. 22, no. 4, pp. 721–737, 2011.
- [31] Zheng R. Y., J. Yuan Z., and H. Z. Liu, "An Algorithm of Lane Detection Based on IPM-DVS," *Journal of Beijing Union University (Natural Science Edition)*, vol. 29, no. 2, pp. 41–46, 2015.
- [32] M. A. Nasirudin and R. A. Mohd, "International Symposium on Intelligent Transport System," in *Proceedings of the International Symposium on Intelligent Transport System*, 2007.
- [33] Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 4, pp. 570–579, 2008.
- [34] K. Zhao, M. Meuter, C. Nunn, D. Müller, S. Müller-Schneiders, and J. Pauli, "A novel multi-lane detection and tracking system," in *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, IV 2012*, pp. 1084–1089, Spain, June 2012.
- [35] Y. Wang, S. Dinggang, and K. T. Eam, "Lane detection using catmull-rom spline," in *Proceedings of the IEEE Intelligent Vehicle Symposium*, pp. 51–57, 1998.
- [36] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-Snake," *Image and Vision Computing*, vol. 22, no. 4, pp. 269–280, 2004.
- [37] D. Khosla, "Accurate estimation of forward path geometry using two-clothoid road model," in *Proceedings of the 2002 IEEE Intelligent Vehicle Symposium, IV 2002*, pp. 154–159, June 2002.
- [38] C. R. Jung and C. R. Kelber, "Lane following and lane departure using a linear-parabolic model," *Image and Vision Computing*, vol. 23, no. 13, pp. 1192–1202, 2005.
- [39] S. Sehestedt, *Efficient Lane Detection and Tracking in Urban Environments*, EMCR, 2007.
- [40] A. Borkar, M. Hayes, and M. T. Smith, "Robust lane detection and tracking with ransac and Kalman filter," in *Proceedings of the 2009 16th IEEE International Conference on Image Processing ICIP 2009*, pp. 3261–3264, Cairo, Egypt, November 2009.
- [41] W.-C. Cheng, "PSO algorithm particle filters for improving the performance of lane detection and tracking systems in difficult roads," *Sensors*, vol. 12, no. 12, pp. 17168–17185, 2012.
- [42] S. Mallik, R. Mallik, N. Roy, and A. Chatterjee, "Face localization by closed loop discriminator estimation and improved detection using contemporary feature extraction techniques," in *Proceedings of the IEEE International Conference on Computer Graphics, Vision and Information Security, CGVIS 2015*, pp. 33–38, India, November 2015.
- [43] H.-Y. Yu, W.-G. Zhang, and Z.-P. Zhao, "Lane departure detection based on the slope of lane lines," *Opto-Electronic Engineering*, vol. 39, no. 7, pp. 43–48, 2012.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

