Chinese Journal of Mechanical Engineering

ORIGINAL ARTICLE

Open Access

ST-LaneNet: Lane Line Detection Method Based on Swin Transformer and LaneNet

Yufeng Du[1], Rongyun Zhang[1*], Peicheng Shi[2], Linfeng Zhao[3], Bin Zhang[1] and Yaming Liu[1]

Abstract

The advancement of autonomous driving heavily relies on the ability to accurate lane lines detection. As deep learning and computer vision technologies evolve, a variety of deep learning-based methods for lane line detection have been proposed by researchers in the field. However, owing to the simple appearance of lane lines and the lack of distinctive features, it is easy for other objects with similar local appearances to interfere with the process of detecting lane lines. The precision of lane line detection is limited by the unpredictable quantity and diversity of lane lines. To address the aforementioned challenges, we propose a novel deep learning approach for lane line detection. This method leverages the Swin Transformer in conjunction with LaneNet (called ST-LaneNet). The experience results showed that the true positive detection rate can reach 97.53% for easy lanes and 96.83% for difficult lanes (such as scenes with severe occlusion and extreme lighting conditions), which can better accomplish the objective of detecting lane lines. In 1000 detection samples, the average detection accuracy can reach 97.83%, the average inference time per image can reach 17.8 ms, and the average number of frames per second can reach 64.8 Hz. The programming scripts and associated models for this project can be accessed openly at the following GitHub repository: https://github.com/Duane711/Lane-line-detection-ST-LaneNet.

Keywords Autonomous driving, Lane line detection, Deep learning, Swin transformer

1 Introduction

Future research should also focus on traffic safety. The National Bureau of Statistics of China reported that in 2019, there were 247646 traffic incidents, causing 62763 fatalities and 256101 injuries, along with a direct economic impact amounting to 1.346 billion yuan [1]. Recently, the fatalities resulting from vehicular accidents in China has exceeded the total number of deaths from various production safety accidents. Road traffic accidents are the largest non-etiogenic and unintended cause of death in China. Several academic and industrial research teams have devoted significant efforts and resources to create sophisticated algorithms aimed at enhancing the understanding of driving environments. This initiative is part of a broader goal to achieve the development of self-driving vehicles and improve advanced driver assistance systems (ADAS). Autonomous driving technology is important for improving vehicle safety. In the past few years, with the advancement of deep learning architectures, the artificial intelligence industry has risen, and autonomous driving technology based on artificial intelligence has developed vigorously. Lane line detection is fundamental to autonomous driving systems. Through lane line detection, an automatic driving system can achieve an active safety control function for the lateral movement of a car. An algorithm for detecting lane lines forms a fundamental element in the architecture of autonomous driving systems. Its information collection channels are equipped with multiple

*Correspondence:
Rongyun Zhang
hanfengzhiwei@163.com
[1] School of Mechanical Engineering, Anhui Polytechnic University, Wuhu 241000, China
[2] Automotive New Technology Anhui Engineering and Technology Research Center, Anhui Polytechnic University, Wuhu 241000, China
[3] School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei 230009, China

Springer Open

© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

cameras in the automatic driving system, which collect and process more information. Therefore, for a lane line detection algorithm to be more widely used, it should be able to effectively reduce information processing costs. In addition, when a vehicle enters a scene with severe occlusions and extreme lighting conditions, insufficient information can lead to poor detection results. This problem, called no visual cue, is another problem encountered in lane line detection. In this scenario, there is an urgent need for a higher-level semantic segmentation of lanes, which should have stronger representation capabilities than traditional image processing methods. Although a spatial convolutional neural network (SCNN) [2] significantly improves the performance of the depth segmentation method by transferring information among adjacent pixels, it incurs higher computational costs owing to the dense pixel-level information transfer. Although deep segmentation methods dominate the field of lane line detection, this type of algorithm has difficulty effectively extracting lane line information in various scenarios. To address the problems of current lane line localization, including low edge segmentation accuracy, high computational costs, poor detection effects in scenes without visual cues, and difficulty in extracting lane line detection features, this study proposes a novel approach has been developed for detecting lane lines, utilizing the capabilities of the Swin Transformer in conjunction with LaneNet.

The main contributions of this method are: (1) The multitask detection network has high accuracy and detection efficiency; (2) the lane line detection network is lightweight and the calculation costs are reduced; and (3) ST-LaneNet solves the problems of special scenarios (such as scenes with no visual cues), poor detection effects, and difficult feature extraction of lane lines. The subsequent sections of this article are organized in the following manner: Section 2 presents a concise summary and review of various other methods used for detecting lane lines; Section 3 outlines the architecture of the model; Section 4 offers a comprehensive overview of the training dataset and the methodologies employed during the training phase. Section 5 delineates the conducted experiments and discusses their outcomes. Lastly, Section 6 encapsulates the essence of the entire document.

2 Related Work

In recent years, the field of lane line detection has garnered considerable attention from the academic community, resulting in noteworthy advancements in research. These methodologies, primarily centered around feature extraction, can be broadly classified into two distinct categories. The first encompasses traditional methods of lane line feature extraction, which rely on established algorithms and techniques. The second category, on the other hand, represents a more contemporary approach, utilizing neural network learning for the detection of lane lines. This bifurcation signifies the evolving nature of research in this area, highlighting a shift from conventional methods to more advanced, AI-driven techniques [3–6].

2.1 Traditional Methods

The conventional methodologies for detecting lane lines primarily rely on sensors mounted on vehicles. These systems are designed to identify and extract the geometric characteristics of objects within captured images.

Chen et al. [7] introduced an innovative approach for lane line detection, leveraging a Kalman filter to dynamically identify the region of interest (ROI), which aligns parallel to the lane line's surrounding area. This methodology specifically targets the ROI inclusive of the lane line based on its positional data, consequently augmenting the detection robustness. Furthermore, this technique necessitates minimal data processing, which facilitates the real-time execution of the algorithm. Nonetheless, it is important to note that variations in lighting conditions can impact the image quality captured by the camera, posing a challenge to this method. The scope of the application of a single image process and boundary extraction algorithm is limited, which affects the accuracy and robustness of lane line detection.

In their study, Wang et al. [8] introduced a novel approach for detecting lane lines in a distributed manner, leveraging the distinct curvature characteristics of lane lines in proximal and distal fields of view. This method obtains the vanishing line of the lane plane in an image by camera calibration, considers 2/3 of the area below the vanishing line of the lane plane as ROI I, and employs an edge distribution function to accurately determine the slope of the lane line's linear model within a specified region. The approach subsequently employs the directional Haar feature to extract edge feature points. These points are then utilized to align with the linear model representing the lane lines, ensuring a precise fit. Finally, in the remaining 1/3 of the area below the vanishing line of the lane plane, the parameters were used to further determine ROI II. The edge feature points were obtained by the unidirectional search algorithm, and hyperbolic fitting was performed to obtain the complete lane line model. Owing to reliance on the selection of feature points, the effectiveness of lane line detection diminishes notably in scenarios where the lane markings are incomplete and the illumination of the scene is suboptimal.

Wu et al. [9] introduced an innovative lane line detection algorithm, characterized by its utilization of dynamic ROIs and independence from varying illumination conditions. Based on the dynamic ROI, yellow and white lane line regions were found in the YCbCr color space, and the lane lines were accurately extracted by combining the Hough transform and the polar angle constraint algorithm. However, the algorithm momentarily fails in the presence of strong reflections resulting from water on the road surface or direct sunlight. To conclude, while conventional methods for lane line detection have somewhat enhanced the precision of lane line identification on structured roadways, the accuracy in localizing lane lines and segmenting their edges is still inadequate. Moreover, there is a significant need for further enhancement in the robustness of these methods.

2.2 Deep Learning Methods

In the contemporary era, marked by the ascendancy of deep learning, neural networks have gained extensive application within computer vision. The innate capability of deep convolutional neural networks (CNNs) for autonomous feature extraction has led to the scholarly proposition of various deep learning-based methodologies for lane line detection.

Neven et al. [10] presented LaneNet, a novel framework that transforms the process of lane line detection into an instance segmentation task.

He et al. [11] developed a novel lane detection methodology utilizing a dual-view convolutional neural network framework. This method is particularly effective in mitigating the impact of gradient textures, thereby significantly decreasing the likelihood of false detections.

Zhang et al. [12] developed a sophisticated spatiotemporal network, incorporating dual convolutional gated recurrent units (ConvGRUs), aimed at enhancing lane detection in complex driving scenarios. The network integrates two structurally identical ConvGRUs, each assigned distinct roles and positions within the framework. The first ConvGRU is strategically employed for the extraction of probable low-level features pertinent to lane markings. Subsequently, these extracted features are amalgamated with outputs from specified blocks, then fed into the succeeding layer of the comprehensive end-to-end network. Conversely, the second ConvGRU is designed to process spatiotemporal information, handling continuous frame inputs to effectively interpret driving dynamics.

Lee et al. [13] introduced an innovative approach through the development of a vanishing point-guided network (VPGNet). This network uniquely integrates vanishing point data into its training methodology, resulting in enhanced detection capabilities in challenging conditions such as rainy and low-light environments. This integration has been instrumental in achieving notable detection performance under these specific scenarios. Andrade et al. [14] presented a comprehensive framework categorizing digital image processing into three distinct hierarchical levels. In the initial, low-level stage, the process involves reducing the dimensionality of the input image from three to a single layer, enhancing image sharpness, and defining the region of interest (ROI) contingent on the minimum safe distance from the preceding vehicle. The development of a feature extractor, specifically designed for lane-edge detection, is a crucial component of the mid-level processing phase. At the advanced, high-level stage, the focus shifts to the development of a lane-tracking strategy. This stage incorporates the utilization of the Hough Transform and a shape-preserving spline interpolation method, aimed at facilitating a seamless and accurate lane fitting.

In their research, Zou et al. [15] conducted an in-depth exploration into lane detection, focusing on the utilization of multiple frames from a continuous driving scenario. They proposed a novel hybrid deep learning architecture that synergistically combines the convolutional neural network (CNN) with the recurrent neural network (RNN). This approach involves a detailed abstraction of information from each frame using a CNN block. Following this, the CNN-derived features from several consecutive frames, inherently possessing time-series characteristics, are integrated and processed through an RNN block. This methodology facilitates advanced feature learning and enables accurate lane prediction, marking a significant advancement in the field.

In their research, Zhang et al. [16] introduced an innovative network known as the ripple lane line detection network (RiLLD-Net), which leverages rapid connections and gradient mapping for the efficient acquisition of lane line characteristics. The RiLLD-Net is adept at managing typical scenarios encountered in lane line detection. To tackle more intricate situations, such as obscured or complex lane markings, a more robust framework named Ripple-GAN has been proposed. This advanced network amalgamates the functionalities of RiLLD-Net with the confrontational training methodology inherent in Wasserstein generative adversarial networks, complemented by multitarget semantic segmentation. This integration facilitates enhanced performance in challenging lane detection contexts.

Chen et al. [17] introduced a sophisticated algorithm designed for the detection and tracking of multiple lanes, irrespective of their shapes. This algorithm represents an advancement over traditional methods by transitioning from the spatial domain to a more comprehensive temporal-spatial domain. This transition leverages a robust, generalized model for multiple-lane analysis. Distancing from conventional image preprocessing techniques, the approach involves the use of slice images derived from the decomposition and reconstruction of image sequences, which encapsulate vital structured historical data. Within this domain, the authors devised specific assumptions to streamline the algorithm's complexity and computational demands. Foremost among these is the adoption of a lane-marker detector with minimal constraints, as opposed to standard binarization methods, to yield a more refined binary image. These lane-marker candidates are then utilized to initiate a particle filtering tracking process. Furthermore, Chen et al. [17] proposed the generation of a confidence map. This map is computed from binary slice images, employing an enhanced distance transform for more accurate particle sampling and weight calculations. Significantly, the proposed multilane model allows for the deployment of multiple independent particle filters, each dedicated to tracking individual lane boundaries.

3 The Proposed Method

In this section, we introduce the conceptualization and implementation of an innovative hybrid neural network.

3.1 System Overview

This network is the result of an integrative fusion between the Swin Transformer and an enhanced version of LaneNet, meticulously engineered to effectively execute lane detection tasks. Figure 1 illustrates the integration of lane line edge proposal and localization networks within lane line detection frameworks. A key component of this system is the Cyclic Shift Window, which represents an efficient method for batch computation of self-attention in shifted-window partitioning schemes. This process involves dividing the window into parts and implementing a cyclic shift, effectively repositioning the original Windows 1 and 3 to the locations depicted in Figure 1. Subsequent to this cyclic displacement, the original window undergoes a 180° counterclockwise rotation. Notably, the total number of windows is maintained, and the images across the four windows are evenly distributed.

The computational dynamics are particularly noteworthy. In Window 0, the patch computes self-attention by interacting with another. However, the patches in Windows 1, 2, and 3, stemming from disparate regions, preclude the feasibility of direct self-attention computation between them. To address this, patches from different regions are overlaid, enabling the computation across varied areas. The culmination of this process involves the application of a softmax function, as delineated in Ref. [18], to refine the output. The front view of the vehicle was obtained through lane line edge proposal and lane localization networks. The specific steps of the lane line edge proposal network are as follows:

(1) Perform inverse perspective mapping on the input front view of the vehicle and perform binarization processing to obtain a feature map.

(2) Apply a depth-separable convolution to the feature map obtained in step (1) to achieve the effects of progressive feature extraction channels and information aggregation.

(3) The content-aware reassembly of features restores feature resolution and generates a pixel-level lane line edge map.

The specific steps of the lane line localization network are as follows:

(1) Input the front view of the vehicle into the lane line localization network and implement the down-sampling of the front view of the vehicle through patch partition and linear embedding.

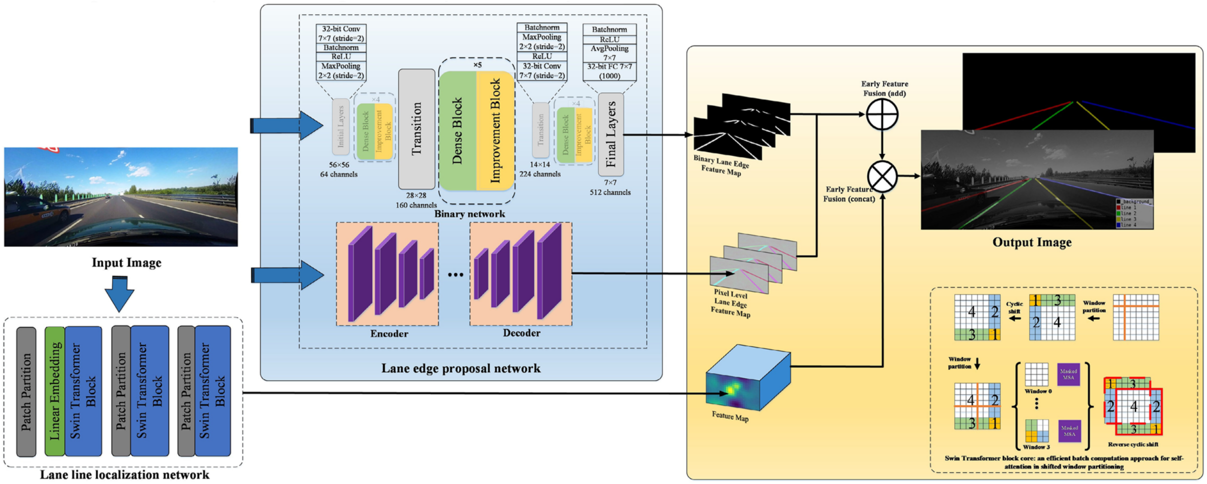


Figure 1 ST-LaneNet: the Lane line detection network structure based on the Swin Transformer and LaneNet

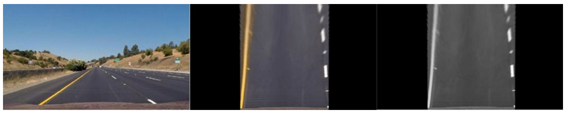


Figure 2 Effect diagram of the front view of the vehicle after passing through the IPM

Figure 3 Binary lane line edge feature map

(2) After the three Swin transformer block processing steps in the lane localization network, the lane line localization task was completed through the Swin Transformer network according to the steps. Finally, the lane line edge feature map obtained by the proposal network and the lane line location feature map were feature cascaded to complete the detection task.

3.2 Network Design

3.2.1 Lane Line Edge Proposal Network

The proposed lane line edge detection framework encompasses a multifaceted structure comprising a coupling network, a binary classification network, and an efficient lightweight encoder-decoder architecture. This system is designed to process the frontal view imagery captured by the vehicle's camera, which is concurrently fed into the integrated backbone network of the model.

(1) Binary Network

The frontal perspective of the vehicle was integrated as an input into the network dedicated to proposing lane line edges and inverse perspective mapping (IPM) [19] is performed to obtain the top view of the vehicle. Binarization is then performed on the results to obtain the corresponding feature map. A top view of the vehicle obtained using the IPM is shown in Figure 2.

As shown in Figure 3, the pixels of the lane line edge in the top view of the vehicle obtained by the IPM were binarized. A series of irrelevant information from the image was filtered, indicating the pixels, which could theoretically be the edge of the lane line. Thereafter, a binary lane line edge feature map was generated. The foundational element of the proposed architecture comprises a dense block and an improvement block, as illustrated in Figure 1. This configuration is a binary variation of the DenseNet framework, with the dense block designed to augment the feature capacity, and the improvement block focused on refining the quality of these features.

Within the dense block, a binary convolution process is employed to derive 64 new feature channels from an input feature map, such as one consisting of 256 channels. These newly derived features are then concatenated with the original feature map, resulting in a composite feature map encompassing 320 channels. This process significantly enhances the feature capacity of the network.

Conversely, the improvement block is tasked with enhancing the quality of the concatenated feature channels. It utilizes binary convolution to process the 320-channel input feature map, generating 64 output channels. These output channels are subsequently integrated with the earlier computed 64 channels via a residual connection. This integration occurs without altering the initial 256 channels of the feature map, as depicted in Figure 1, thus ensuring the preservation of the original feature information while simultaneously improving its overall quality. Using this approach, a feature map was precisely generated.

Pixel binarization segmentation of the edges of lane lines was also used to generate a binarized lane line edge feature map, and a series of irrelevant information in the images was filtered. Pixels that may theoretically be the edges of the lane segmentation and the resulting feature map are indicated. This research employs advanced deep neural network architectures for the extraction of features from image data given the information and powerful feature extraction capabilities of neural networks. In other words, the feature encoder-decoder using convolutional and deconvolutional layers (transposed convolutional layers) is extensively employed in a variety of predictive tasks, including but not limited to semantic segmentation [20–22]. To improve the prediction efficiency, the lane line edge network proposed in this study adopts a lightweight encoder-decoder architecture [23], inputs the inverse perspective mapping (IPM) image, capturing the vehicle's frontal view, into the decoder. This process involves a meticulous layer-by-layer extraction of features. Subsequently, the feature map's resolution is progressively reconstructed utilizing the decoder, and a pixel-level lane line edge feature map was generated. The encoder-decoder architecture is employed within a lane detection model, conceptualized as a task of semantic segmentation. In scenarios where the encoder-decoder network is architected to accommodate identical input and output dimensions, it facilitates the possibility of training the entire network through an end-to-end approach.

(2) Encoder

To reduce the computational costs, a depthwise separable convolution was employed as a substitution for the conventional convolutional approach [24], as shown in Figure 3. Depthwise separable convolution includes two parts: deepwise and pointwise convolution. The specific operations are as follows: As shown in Figure 4(a), deep convolutional layers, each equipped with a kernel size of 3, were methodically stacked to facilitate a progressive and nuanced extraction of features. As shown in Figure 4(b), each layer of depthwise convolution is succeeded by a 1 × 1 pointwise convolutional layer, a strategic design choice aimed at facilitating the aggregation of channel information.

In the encoding stage, features are extracted, and context information is preserved, which can effectively reduce the probability that objects whose local appearance is similar to a lane line are predicted as lane lines. The receiving field of the encoder was expanded while controlling the calculation costs. The encoder uses an expanded convolutional kernel, and the convolutional module contains three deepwise convolutional layers and a 1 × 1 pointwise convolutional layer.

The initial depthwise convolutional layer in the architecture was characterized by an expansion rate set at a value of 1, which corresponded to the standard separated convolutional layer. The remaining two deepwise convolutional layers introduce a dilated convolutional kernel [25] without adding additional parameters or computational costs and increase the effective receptive field, thus properly balancing the efficiency and effectiveness of feature extraction.

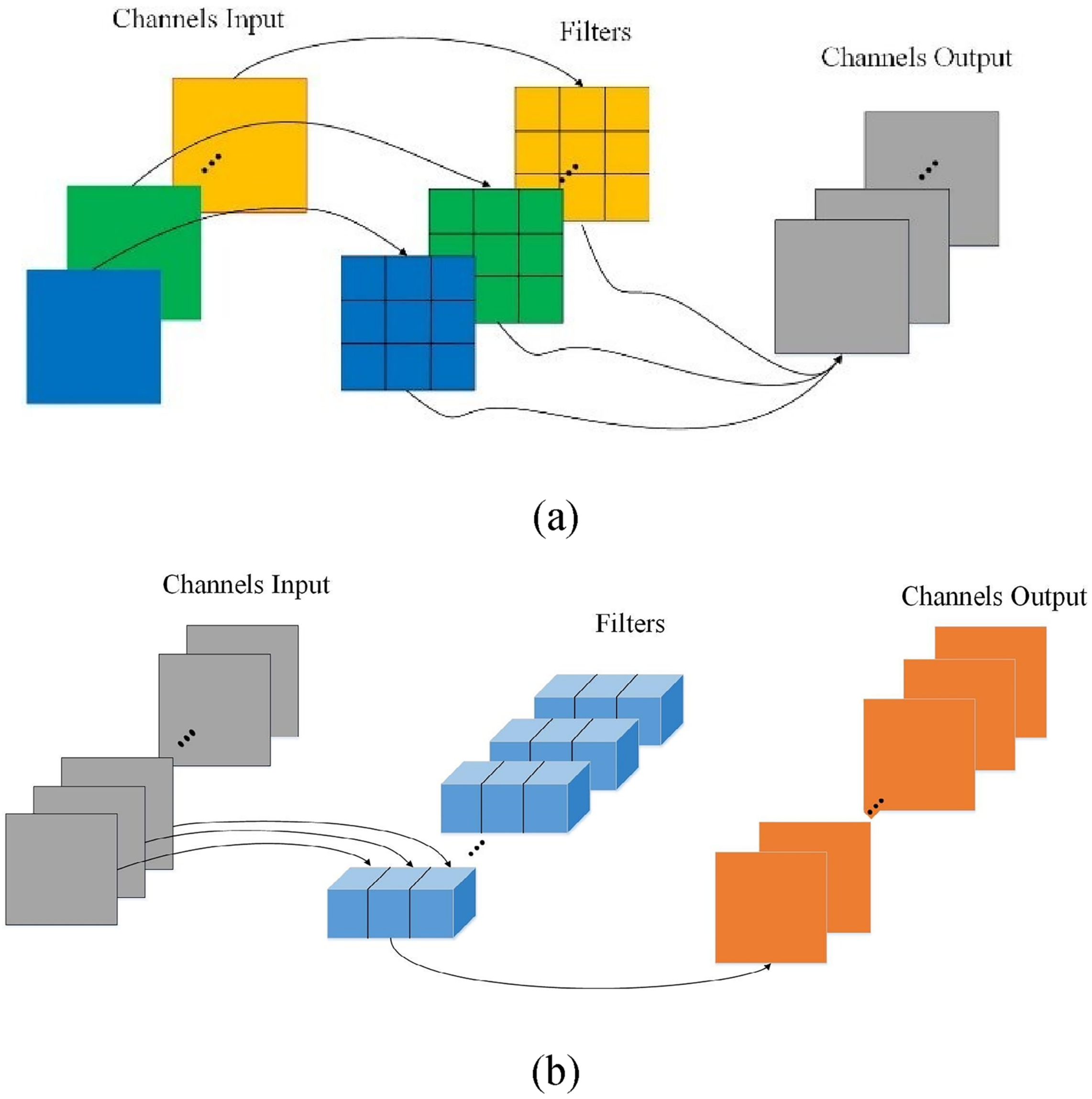


Figure 4 Depthwise separable convolution consists of (**a**) deepwise convolution and (**b**) pointwise convolution

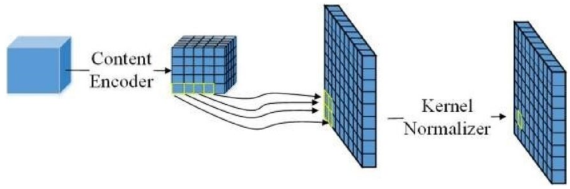


Figure 5 Kernel prediction module

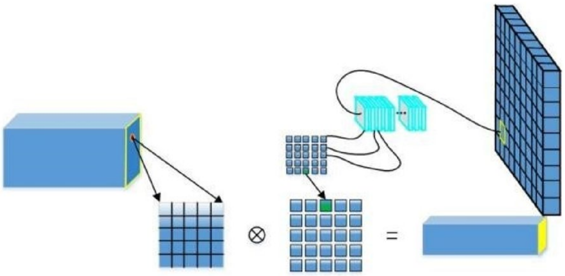


Figure 6 Content-aware reassembly module

(3) Decoder

To reconstruct the resolution of the features and produce a detailed, pixel-level map that highlights the edge features of lane lines, we designed an encoder-decoder structure. Although the deconvolutional layer (transposed convolution layer) is widely used to amplify intermediate features, this approach is characterized by its significant computational demand and the complexity inherent in its training process. Therefore, this article adopts a content-aware up-sampling method through the reorganization of features: Content-aware reassembly of features (CARAFE) [26], a novel structure comprising two integral components: the kernel prediction module and the reassembly module. CARAFE operates by initially utilizing the kernel prediction module to forecast the up-sampling kernel. Subsequently, the reassembly process is executed through the content-aware assembly module, completing the up-sampling procedure. The kernel prediction is divided into three steps, as shown in Figure 5.

First, the channel of the feature-map was compressed to diminish the computational requirements. Second, the convolutional layer was used to predict the squeezed input feature map, and an up-sampling kernel was applied. The content-aware reassembly is divided into two steps, as illustrated in Figure 6.

Initially, the process involves a backward mapping of each spatial location on the output feature map to its corresponding position on the input feature map, followed by the excision of the area centered around this mapped location. Subsequently, the algorithm employs a dot product calculation between the up-sampling kernel and this specific point to predict and yield an output value. It is noteworthy that, at identical spatial locations, different channels utilize a shared up-sampling kernel. Furthermore, these channels exclusively incorporate the parameters of the convolutional kernel present in the content encoder, emphasizing a streamlined and focused approach in feature processing. Therefore, the CARAFE has fewer network parameters, making its network structure lightweight.

Finally, a residual connection [27] is introduced to solve the problems of gradient disappearance and weight matrix degradation to provide high-resolution features and obtain a more accurate lane line edge feature map.

3.2.2 Lane Line Localization Network

The vehicle's frontal perspective is fed into the lane line localization network. Given the intricate nature of lane lines, discerning them from feature maps presents a significant challenge. Consequently, the deployment of a convolutional neural network (CNN) for detection incurs considerable computational expense. Therefore, a Swin Transformer [28] was introduced to replace the CNN to achieve lane line detection, as shown in Figure 7.

The architectural design of the network adopts a hierarchical methodology, encompassing four distinct stages. In each stage, there is a systematic reduction in the resolution of the input feature map, concurrently expanding the receptive field on a layer-by-layer basis. Stage 1: First, patch partition was used to cut the front view of the input vehicle. Second, linear embedding, which achieves down-sampling

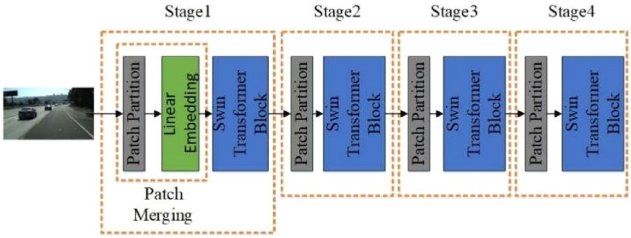


Figure 7 Swin Transformer structure

of the front view of the vehicle, reduces the resolution, modifies the channel count and constructs a hierarchical structure, thereby diminishing the computational requirements, was used to reduce its dimensionality. Finally, the outcome of the initial phase is subsequently channeled through a Swin Transformer block. This process constitutes the first stage. In stages 2, 3, and 4, each stage uniformly comprises a sequence of patch merging followed by the application of a Swin Transformer block.

As shown in Figure 8, the foundational architecture of the Swin Transformer block is centered around two pivotal components: the window multihead self-attention (W-MSA) and the shifted-window multihead self-attention (SW-MSA).

The architecture of the Swin Transformer block is composed of dual multi-layer perceptrons (MLPs). Prior to each multihead self-attention (MSA) module and MLP, layer normalization (LN) is systematically implemented, enhancing the stability and performance of the network. Post each module, a residual connection strategy is employed, crucial for mitigating error propagation. The integration of window-based multihead self-attention (W-MSA) within this framework addresses scalability concerns, while the shifted window multihead self-attention (SW-MSA) is ingeniously designed to reduce computational complexity, thereby optimizing the efficiency of the system.

4 Training Strategy

When we build an end-to-end trainable neural network, we can train it in a fully supervised manner and perform end-to-end optimization through stochastic gradient descent [29].

4.1 Lane Edge Proposal Network

In their methodology, the researchers utilized the vehicle's front view as the input, generating a lane edge probability map corresponding in size to the input image as the output. This process involved supervised training of parameters. For each training image, corresponding annotated images were provided, with "1" denoting pixels located at the lane's edge and "0" representing other areas. The lane-edge proposal network underwent optimization through an end-to-end process using stochastic gradient descent.

A notable aspect in the annotation process was the disproportionate number of positive (lane-edge) to negative (non-lane-edge) points. This imbalance necessitated a careful calibration of weights for both positive and negative samples during training. Furthermore, the model required a distinct approach for managing the weights of samples that were either challenging or straightforward to classify.

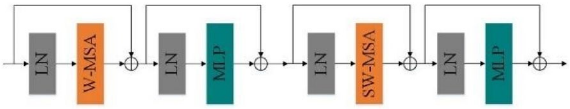


Figure 8 Swin transformer block structure

To address these complexities, the researchers introduced a focal loss function [30] as a replacement for the traditional cross-entropy loss function. This focal loss function strategically reduces the weight assigned to samples that are easy to classify, thereby directing the model's training focus towards samples that are more challenging to classify. The focal loss function, simplifying the cross-entropy loss using p_t, which is expressed in Eq. (1):

$$\mathrm{CE}(p,y)=\mathrm{CE}(p_t)=-\log(p_t). \tag{1}$$

To reduce the influence of negative samples, increasing coefficient α_t before the conventional loss function is similar to p_t. When label = 1, $\alpha_t=\alpha$; and when label = others, $\alpha_t=1-\alpha$. By controlling the value of α, the influence of the positive and negative samples on the loss was controlled, as shown in Eqs. (2), (3):

$$\mathrm{CE}(p_t)=-\alpha_t\log(p_t), \tag{2}$$

$$\alpha_t=\begin{cases}\alpha, & \text{if } y=1,\\ 1-\alpha, & \text{otherwise.}\end{cases} \tag{3}$$

Then, Eqs. (2) and (3) are combined to obtain Eq. (4):

$$\mathrm{CE}(p,y,\alpha)=\begin{cases}-\log(p)\times\alpha, & \text{if } y=1,\\ -\log(1-p)\times(1-\alpha), & \text{if } y=0.\end{cases} \tag{4}$$

The modulating factor $(1-p_t)^\gamma$ is implemented to modulate the weights assigned to samples that present significant challenges in terms of complexity or variability, and Eq. (5) is obtained:

$$\mathrm{FL}(p_t)=-(1-p_t)^\gamma\log(p_t). \tag{5}$$

When γ tends to 0, $(1-p_t)^\gamma$ tends to 1, which makes a greater contribution to $\mathrm{FL}(p_t)$. When p_t tends to 1, $(1-p_t)^\gamma$ tends to 0, which makes a smaller contribution to $\mathrm{FL}(p_t)$. By adjusting the value of γ to control the modulating factor, the formulation of Eq. (6) was achieved by an integrated approach, wherein the weights assigned to both positive and negative samples were amalgamated with those designated for samples varying in classification difficulty, encompassing both challenging and readily classifiable instances.

$$\mathrm{FL}(p_t)=-\alpha_t(1-p_t)^\gamma\log(p_t). \tag{6}$$

The objective extends beyond merely recalibrating the weights of the positive and negative samples; it encompasses the nuanced regulation of weights assigned to samples that are challenging to classify as well as those that are readily classifiable. In the course of training, the lane-edge proposal network autonomously produces a lane-edge proposal map, directly corresponding to the input image. Within this map, the value assigned to each pixel reflects the degree of confidence regarding whether that pixel is situated at the boundary of a lane segment.

4.2 Lane Line Localization Network

In the presented methodology, the initial step involves processing the vehicle's frontal view. This image is methodically segmented into discrete, non-overlapping sections, referred to as 'tokens', each measuring 4×4 pixels. Subsequently, these tokens are analyzed in terms of their feature dimensions, which, considering the RGB color space, culminates in a feature vector with a dimensionality of $4\times4\times3$, equating to 48 distinct elements; the number of patches was $H/4\times W/4$ (W: image width, H: image height). The pixel-resolution image is converted to the same resolution as that in the patch-by-patch partition. At this point, the original feature values are fed into the linear embedding and projected onto any dimension, which is denoted as C. In the proposed architecture, the token is enhanced with a Swin Transformer block, which has undergone modifications in its self-attentive computation mechanism. The first stage of this process encompasses three key components: patch merging, the integration of the modified Swin Transformer block, and the implementation of linear embedding.

The architecture incorporates a patch-merging layer, which significantly diminishes the quantity of tokens, thereby facilitating a hierarchical representation of data. This reduction is methodically achieved by concatenating adjacent patch features into groups, followed by the application of a linear layer onto a feature cascade with a dimensionality of $4C$. Consequently, the token count is effectively reduced by a factor of four, resulting in a more compact representation with dimensions $H/8\times W/8$. The output feature dimension was compressed from $4C$ to $2C$ by linear embedding and input to the Swin transformer blocks. The resolution was kept constant during the feature transformation process by the Swin transformer blocks. Stage 2 is formed by merging a patch with the first Swin transformer block through patch merging, and Stages 3 and 4 repeat the process of Stage 2. These results formed a layered representation, and the feature map's resolution, post-processing through the Swin transformer, was found to be analogous to that achieved via a conventional convolutional network.

Taking into account the computational intricacies, it is noteworthy that the complexity inherent in the global MSA computation exhibits a quadratic relationship with the token count. Consequently, to facilitate dense predictions or to accurately represent images of high resolution, a substantial aggregation of tokens was employed. The employment of global computation in this context results in a quadratic increase in complexity, directly proportional to the quantity of tokens involved. This escalation in computational demand potentially limits the network's applicability when dealing with a substantial number of tokens, resulting in dense predictions and high-resolution images. To facilitate efficient modeling, the study employed a non-overlapping technique for the uniform segmentation of an image (W-MSA) during the computation of self-attention within localized windows. The computational complexities of both the MSA and W-MSA modules were quantitatively determined using Eqs. (7) and (8).

$$\Omega(\text{MSA}) = 4HWC^2 + 2(HW)^2C, \tag{7}$$

$$\Omega(\text{W-MSA}) = 4HWC^2 + 2M^2HWC. \tag{8}$$

As shown in Eqs. (7) and (8), calculations using the global MSA module are not applicable when the values of hw are large. To address the limitations posed by the absence of inter-window connections in the W-MSA (window-based multi-head self-attention) module mechanism, which restricts the model's potential, a novel approach is introduced involving the integration of cross-window connections. This method alternates between two distinct partitioned configurations within consecutive Swin Transformer blocks, thereby enhancing the model's capability by facilitating broader interaction across different window segments.

Furthermore, to preserve the computational efficiency associated with non-overlapping windows, the technique of SW-MSA partitioning was employed. The sequential processing of Swin Transformer blocks was executed in accordance with the formulations presented in Eqs. (9)–(12):

$$\hat{z}^l = \text{W-MSA}\left(\text{LN}\left(z^{l-1}\right)\right) + z^{l-1}, \tag{9}$$

$$z^l = \text{MLP}\left(\text{LN}\left(\hat{z}^l\right)\right) + \hat{z}^l, \tag{10}$$

$$\hat{z}^{l+1} = \text{SW - MSA}\left(\text{LN}\left(z^l\right)\right) + z^l, \tag{11}$$

$$z^{l+1} = \text{MLP}\left(\text{LN}\left(\hat{z}^{l+1}\right)\right) + \hat{z}^{l+1}. \tag{12}$$

As shown in Eqs. (9)–(12), $\hat{z}^l$ represents the output characteristics of the (S) W-MSA module and z^l

represents the output characteristics of the MLP module of block l. The employment of a shielding mechanism is pivotal in confining the scope of self-attention computations to individual sub-windows. Additionally, this mechanism encompasses the implementation of a cyclic shift strategy, which is used to ensure that the batch processing window has the same number of partitions as the regular window and a lower delay, which results in better real-time performance. To enhance the model's characterization capabilities without the need for modifying training hyperparameters, and to maintain detection accuracy, a strategy of incorporating relative position coding is introduced, as delineated in Eq. (13):

$$\mathrm{Attention}(Q, K, V) = \mathrm{SoftMax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d}} + B\right)V, \quad (13)$$

where $Q, K, V \in R^{M^2 \times d}$ are the matrices of the query, key, and value, respectively; d is the dimension of the query and key; M^2 is the number of patches in the window; and the relative position bias is $B \in R^{M^2 \times M^2}$. As the relative positions along each axis are in the range $[-M+1, M-1]$, a smaller size bias matrix, with values in B taken from $\hat{B}$, is parameterized and fine-tuned by three double interpolations using different window sizes.

5 Experiments and Results

In this section, a series of experiments are presented to demonstrate the efficacy of the lane line detection method, which is grounded in the integration of the Swin Transformer and an enhanced version of the LaneNet network. The following sections mainly focus on four aspects: (1) the dataset, (2) hyperparameter settings and hardware environment, (3) performance evaluation, and (4) visualization.

5.1 Dataset

To evaluate this detection method, experiments were conducted on a widely used benchmark dataset (the TuSimple dataset). We utilized the TuSimple dataset, which comprises 72000 annotated front-view images, all gathered under consistent lighting conditions on highways. This dataset is detailed in Table 1. To further assess the network's detection robustness, the authors curated an additional dataset derived from CULane. This dataset presents a more formidable challenge than the TuSimple dataset, encompassing not only normal scenes but also eight complex scenarios, including crowded situations, night-time conditions, and environments with dazzling lights. For the purposes of testing and training, new sets were delineated distinctly. The dataset was partitioned such that 80% constituted the training set and the remaining 20% formed the test set. Due to the unavailability of real labels for the test segments to the public, the training set was maintained in its original form. However, for testing purposes, the pre-existing validation set from the original dataset was employed.

The testing set was divided into two subsets, namely simple and difficult samples, as shown in Table 1. The row anchor points defined by the dataset were used for each dataset. Specifically, the row anchor points ranged from 160 to 710, with a step length of 10. The number of grid cells was set to 100.

5.2 Hyperparameter Settings and Hardware Environment

Drawing from the methodology outlined in Ref. [31], the image resolution for the TuSimple dataset was modified to 368×640 pixels, a measure taken to optimize memory usage. In our experimental setup, the network training was facilitated using stochastic gradient descent (SGD) [32], with a predefined set of hyperparameters: the deployment of 2 GPUs, a training duration encompassing 80000 epochs, and a batch size of 32. We employed a polynomial decay strategy for learning-rate scheduling. The initial learning rate was set at 0.1, with 1000 warm-up steps, while the momentum and weight decay parameters were fixed at 0.9 and 0.005, respectively. The activation function utilized was LReLU, and TensorFlow 1.6 served as the development framework. Performance evaluation involved recording the running time on an RTX3060 GPU, with the final time derived from the average of 1000 sample runs. All experimental procedures were conducted on a training platform equipped with an NVIDIA RTX3060 GPU and an 11th Generation Intel Core i7-11700K CPU.

5.3 Performance Evaluation

The accuracy was used as the official evaluation standard. In addition, false positive (FP), false negative (FN), true positive rate (TPR), and false positive rate (FPR) were also reported. The TPR calculation method is given in Eq. (14), FPR calculation method is expressed by Eq. (15),

Table 1 Dataset information description of TuSimple and CULane

Dataset	#Frame	Train	Test	Resolution	#Lane	Time	Environment
TuSimple	72000	57600	14400	1280 × 720	< 4	Daytime	Highway
CULane	133000	106400	26600	1640 × 90	≤ 4	Daytime and night	City, rural, highway

Precision is given by Eq. (16), Recall is given by Eq. (17), the F1-Measure is shown by Eq. (18), and the accuracy calculation method [29] is expressed by Eq. (19). The calculation formulas are as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{14}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{15}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{16}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{17}$$

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \tag{18}$$

$$\mathrm{Accuracy} = \frac{N_{pred}}{N_{gt}}. \tag{19}$$

The results are shown in Tables 2 and 3.

In this study, a TP denotes an instance accurately classified and predicted as belonging to the positive class. Conversely, a FP represents an instance originating from the negative class but erroneously predicted as positive. A FN is an instance of the positive class incorrectly forecasted as negative. Lastly, a true negative (TN) refers to an instance correctly identified and predicted within the negative class.

The TPR, defined as the ratio of the number of detected lanes to the number of target lanes, quantifies the fraction of actual positive instances correctly identified by the network among all positive instances. Meanwhile, the FPR, calculated as the ratio of the number of false alarms to the number of target lanes, measures the proportion of actual negative instances misclassified as positive by the network in relation to all negative instances.

Each lane is detected only once, and overestimation or underestimation of the total number of lanes is not advisable. In this research, we critically examined a phenomenon where repeated detection of a dotted lane markedly increases the FPR, while the identification of segmented lanes within a single lane notably diminishes the TPR. To provide a comprehensive analysis, the LaneNet lane line detection methodology was employed as a comparative benchmark to evaluate its detection efficacy against the detection approach proposed in this study.

The data in Table 4 show that the TPR of ST-LaneNet is higher and the FPR is lower. Tables 5 and 6 show that the performance of ST-LaneNet improved compared with the others. The F1 values for the different scenario categories are listed in Table 5. The FP values are presented in

Table 2 Comparison of the TPR and FPR based on the TuSimple dataset

Difficulty	Easy lanes (1190 lanes)			Hard lanes (1050 lanes)		
	Detected	TPR (%)	FPR (%)	Detected	TPR (%)	FPR (%)
ResNet-18 [27]	1096	92.12	8.21	965	91.91	9.52
ResNet-34 [27]	1112	93.43	6.83	975	92.83	8.31
ENet [33]	1135	95.42	5.11	997	94.92	7.14
SCNN [2]	1140	95.83	4.62	1002	95.41	7.42
LaneNet [10]	1138	95.64	4.51	998	95.03	7.33
ST-LaneNet	1160	97.53	3.82	1032	96.83	6.82

Table 3 Comparison of the TPR and FPR based on the CULane dataset

Difficulty	Easy lanes (1190 lanes)			Hard lanes (1050 lanes)		
	Detected	TPR (%)	FPR (%)	Detected	TPR (%)	FPR (%)
ResNet-18 [27]	1096	90.08	9.86	965	88.94	11.05
ResNet-34 [27]	1112	91.24	8.45	975	89.56	10.07
ENet [33]	1135	92.89	6.95	997	91.06	8.51
SCNN [2]	1140	93.15	5.67	1002	91.42	7.03
LaneNet [10]	1138	93.03	5.42	998	91.23	7.22
ST-LaneNet	1160	94.87	4.15	1032	93.17	5.97

Table 4 Performance of different algorithms on the TuSimple test set

Algorithm	Val_Accuracy (%)	Test_ Accuracy (%)	FP	FN
ResNet-18	93.12	94.38	0.0935	0.0813
ResNet-34	93.43	95.11	0.0906	0.0785
ENet	94.62	95.98	0.0875	0.0716
LaneNet	97.12	98.08	0.0706	0.0212
SCNN	97.22	98.12	0.0598	0.0173
ST-LaneNet	97.81	98.85	0.0565	0.0165

Table 5 Performance of different algorithms on the CULane test set

Category	ResNet-18	ResNet-34	ENet	LaneNet	SCNN	ST-LaneNet
Normal	90.4	90.4	91.3	91.8	90.6	93.1
Crowded	65.4	67.8	68.3	70.5	69.7	74.1
Dazzle light	87.4	60.3	61.1	60.2	58.5	70.6
Shadow	65.6	68.1	67.2	67.4	66.9	78.3
No line	40.8	42.7	43.5	44.7	43.4	50.3
Arrow	83.7	84.2	85.2	85.3	84.1	84.9
Curve	61.4	62.3	63.9	65.2	64.4	65.3
Crossroad	1845	1985	2045	2006	1990	1887
Night	64.7	65.9	67.2	67.5	66.1	69.2
Total	70.5	71.4	72.1	72.6	71.3	74.8
Framesper second (Hz)	25.9	37.5	50.3	48.9	8	63.9

Table 6 Comparison of the inference time, frames per second and model size

Network	Inference time (ms)	Frames per second (Hz)
ResNet-18	27.2	30.9
ResNet-34	53.2	40.3
ENet	18.8	61.5
SCNN	135.8	29.6
LaneNet	19.5	63.6
ST-LaneNet	17.8	64.8

the context of the crossroads scenario. Given the absence of straight lines within this scenario, it is noted that any predicted point is classified as a FP. Although the model capacities of ResNet-18 and ResNet-34 are larger, their performance is slightly inferior to that of ST-LaneNet because ResNet-18 and ResNet-34 use only spatial upsampling as the decoder. Tables 5 and 6 show that the ST-LaneNet proposed in this study has a faster inference time, less frames per second, and smaller model size, which means that ST-LaneNet has faster image transmission and road lane detection speed. The losses of the ENet, SCNN, LaneNet, and ST-LaneNet training in this study are shown in Figure 9. In summary, ST-LaneNet has the advantage of greater precision. Compared with the SCNN, the architectural design of ST-LaneNet has been optimized, resulting in a substantial reduction in the quantity of its parameters. This refinement has concurrently led to a marked enhancement in its operational speed. Therefore, the effectiveness of ST-LaneNet was demonstrated.

5.4 Visualization

The proposed lane line edge detection framework, rooted in deep neural network technology, exhibits a

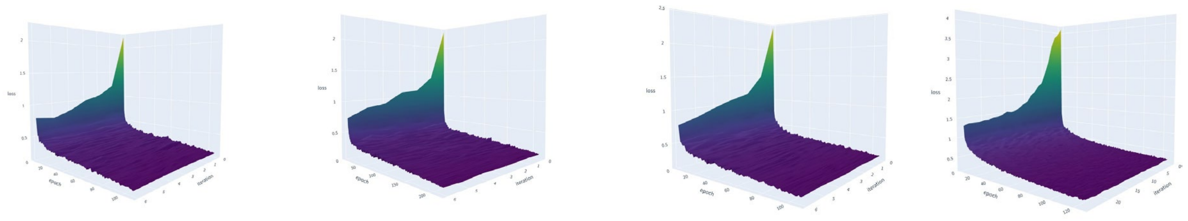

(a) ENet (b) SCNN (c) LaneNet (d) ST-LaneNet

Figure 9 Loss function of training

robust capability for extracting contextual information. This methodology, leveraging the depth of neural networks, facilitates decision-making over an expanded contextual spectrum, thereby significantly reducing the likelihood of erroneous traffic sign detections on roadways. Therefore, lane line detection is robust in various scenarios. Without relying on the shape and number of lanes, the accuracy of the lane line localization network detection was improved, proving the effectiveness of the lane line localization network in handling various road scenes.

The integrated network for lane line detection is composed of two primary components: the lane line edge proposal network and the lane line localization network. This synergistic architecture is specifically designed to efficiently identify lane lines. To evaluate the efficacy of various algorithms within this framework, their performances have been methodically analyzed and are presented in Figure 10, utilizing the TuSimple test set as the benchmark. Top row: Lane line prediction overlapped with the original image. Middle row: White-lane line forecast. Bottom row: Segmented image of a lane line instance.

In the SCNN (from left to right, first column) and LaneNet (from left to right, second column) detection scenes, the lanes were straight, the lighting conditions were good, and the lane lines were marked intact. In the ST-LaneNet (from left to right, third and fourth columns) detection scene, the lane is a curve, the light conditions are normal, and the lane markings are incomplete.

Additionally, in an effort to preserve the structural integrity of lane lines, the described technique involves either expanding or cropping the lane line to align with the image's boundary, as illustrated in Figure 11. In the bottom row of this figure, expansion of the lane line is executed to uphold the lane's structure, complemented by the application of an image enhancement method. The impact of this enhancement is highlighted by a red ellipse. The efficacy of ST-LaneNet on the CULane test set is further demonstrated in Figure 12, showcasing its performance.

Due to the absence of delineations at the intersection, the detection outcome is not as depicted in Figure 12(g). Within each figure, the initial row represents the input data, while the subsequent row illustrates the instance segmentations that have been subjected

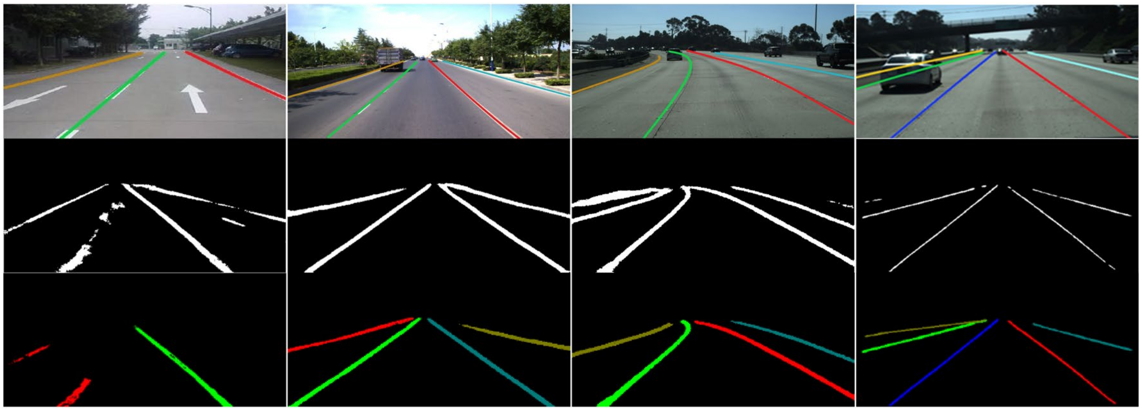


Figure 10 Performance of different algorithms on the TuSimple test set

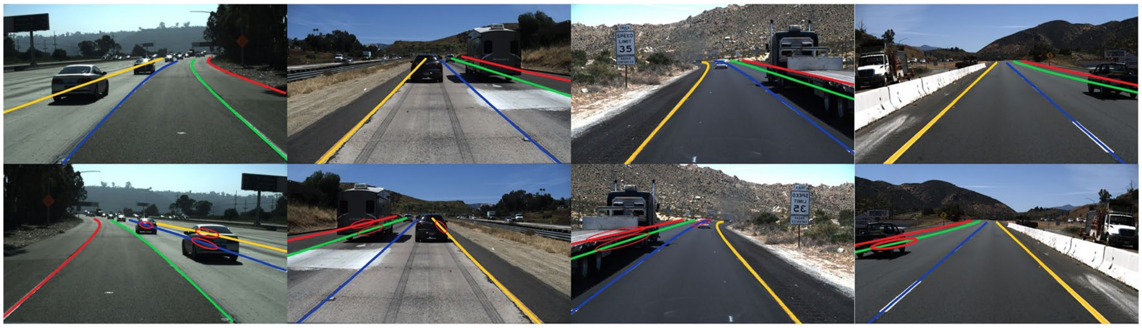
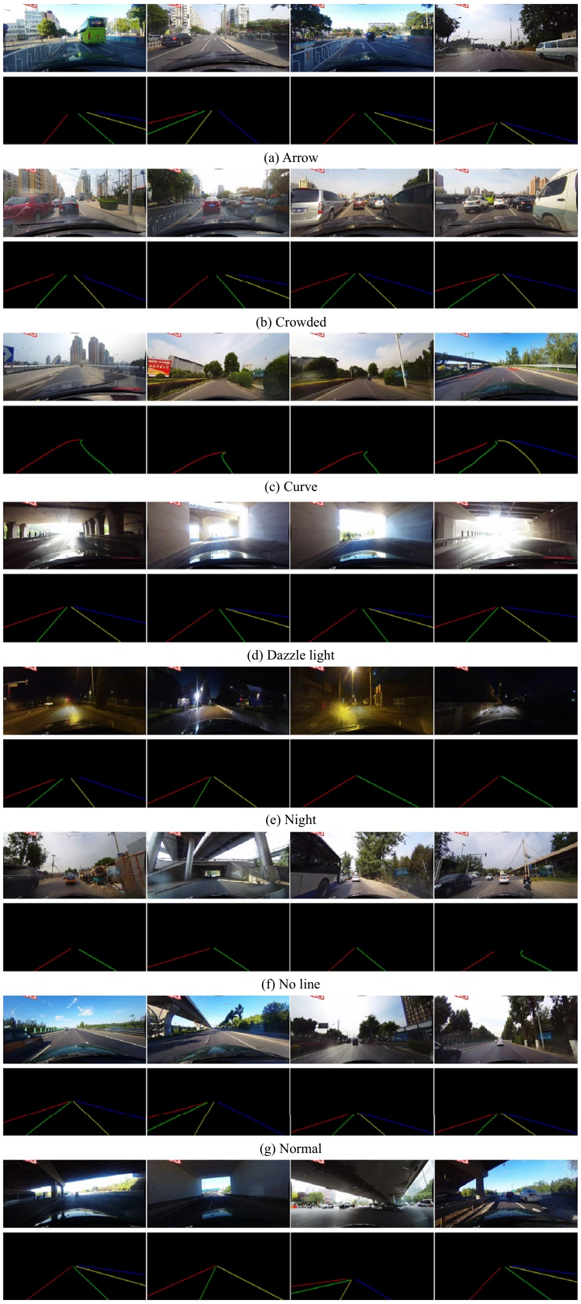


Figure 11 Top row: The original annotation

to clustering. The LaneNet detection effect was better than the SCNN detection effect. Among the three lane line detection methods, ST-LaneNet had the best detection effect. The results demonstrate that the ST-LaneNet lane line detection network can realize lane line detection in a variety of scenarios. The ground truth lanes are drawn on the input image.


(a) Arrow

(b) Crowded

(c) Curve

(d) Dazzle light

(e) Night

(f) No line

(g) Normal

(h) Shadow

Figure 12 Performance of different algorithms on the CULane test set: (**a**) Arrow, (**b**) Crowded, (**c**) Curve, (**d**) Dazzle light, (**e**) Night, (**f**) No line, (**g**) Normal, (**h**) Shadow

Because of the inherent structure of lane lines, classification-based networks can easily overfit the training set and perform poorly on the test set. Furthermore, in image scenes where driving vehicles or other objects obstruct lane lines, the detection of lane lines is reduced. To prevent this phenomenon and improve the generalization ability of the network, we adopted an image enhancement method composed of rotation, and vertical and horizontal shifts.

6 Conclusions

(1) This paper proposes a lane line detection method based on the Swin Transformer and LaneNet network, which achieves significant speed, efficiency, robustness, and accuracy improvements, and solves the problem of detection in scenes without visual cues. It was also trained using the popular TuSimple dataset.

(2) The method consists of two steps. One is the feature extraction of lane line edges based on the lane line edge proposal network. In this part, the standard convolution is replaced by the depthwise separable convolution to reduce the computational cost; and CARAFE is adopted to make the network lightweight, restore the feature resolution, and finally generate the pixel-level lane line edge feature map.

(3) In the other step, the lane line localization network based on traditional convolution is replaced by the lane line localization network based on the Swin Transformer to achieve accurate localization of lane lines and obtain the feature map of lane line localization. Finally, our designed lane line detection network was experimentally tested.

(4) Through benchmark tests, it was verified that the laneline detection method can ensure robustness when completing detection tasks in various scenarios. However, the current lane line detection method does not consider detection tasks in nighttime scenes or extreme weather scenes. The next work, based on the Swin Transformer's powerful performance in various visual problems, gradually considers using it to completely replace the deep neural network to achieve a more accurate, efficient, and widely used lane line detection method.

Acknowledgements

Not applicable.

Authors' Contributions

YD was responsible for literature retrieval, research design, data collection and analysis, charts production and manuscript writing; RZ was responsible for innovation point proposal and refinement, technical support, financial support, and manuscript proofreading; PS and LZ were responsible for comparative experimental design; BZ and YL were responsible for literature retrieval and language polish. All authors read and approved the final manuscript.

Funding
Supported by National Natural Science Foundation of China (Grant Nos. 51605003, 51575001), Natural Science Foundation of Anhui Higher Education Institutions of China (Grant No. KJ2020A0358), and Young and Middle-Aged Top Talents Training Program of Anhui Polytechnic University of China.

Availability of Data and Materials
The datasets supporting the conclusions of this article are included within the article.

Declarations

Competing Interests
The authors declare no competing financial interests.

Received: 1 December 2021 Revised: 28 December 2023 Accepted: 4 January 2024
Published online: 26 February 2024

References

[1] Y Wang, J Hu, F Wang, et al. Tire road friction coefficient estimation: Review and research perspectives. *Chinese Journal of Mechanical Engineering*, 2022, 35: 6. https://doi.org/10.1186/s10033-021-00675-z.
[2] X Pan, J Shi, P Luo, et al. Spatial as deep: Spatial CNN for traffic scene understanding. *32nd AAAI Conference on Artificial Intelligence,* 2018: 7276-7283.
[3] D Yang, W Bao, K Zheng. Lane detection of smart car based on deep learning. *Journal of Physics: Conference Series*, 2021, 1873 (1): 012068. https://doi.org/10.1088/1742-6596/1873/1/012068.
[4] H Li, X Li. Flexible lane detection using CNNs. *Proceedings-2021 International Conference on Computer Technology and Media Convergence Design,* 2021: 235–238. https://doi.org/10.1109/CTMCD53128.2021.00057.
[5] Z Zhang. Z-Net: A novel way of lane detection. *Journal of Physics: Conference Series*, 2020, 1682 (1): 012013. https://doi.org/10.1088/1742-6596/1682/1/012013.
[6] J Hur, S N Kang, S W Seo. Multi-lane detection in urban driving environments using conditional random fields. *IEEE Intelligent Vehicles Symposium,* 2013: 1297–1302. https://doi.org/10.1109/IVS.2013.6629645.
[7] T Chen, H Zhang, D Chen, et al. Lane detection based on high priority pixels and tracking by kalman filter. *Qiche Gongcheng/Automotive Engineering*, 2016, 38 (2): 200-205.
[8] H Wang, Y F Cai, G Y Lin, et al. Lane-line detection method based on orientation variance haar feature and hyperbolic model. *Journal Traffic Transportation Engineering*, 2014, 14 (5): 119–126.
[9] H Y Wu, X M Zhao. Multi-interference lane recognition based on IPM and edge image filtering. *China Journal Highway Transportation*, 2020, 33 (5): 153–164. https://doi.org/10.19721/j.cnki.1001-7372.2020.05.014.
[10] D Neven, B De Brabandere, S Georgoulis, et al. Towards end-to-end lane detection: An instance segmentation approach. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2018: 286–291. https://doi.org/10.1109/IVS.2018.8500547.
[11] B He, R Ai, Y Yan, et al. Accurate and robust lane detection based on dual-view convolutional neutral network. *IEEE Intelligent Vehicles Symposium, Proceedings*, 2016: 1041–1046. https://doi.org/10.1109/IVS.2016.7535517.
[12] J Zhang, T Deng, F Yan, et al. Lane detection model based on spatio-temporal network with double convolutional gated recurrent units. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23 (7): 6666–6678. https://doi.org/10.1109/TITS.2021.3060258.
[13] S Lee, J Kim, J Yoon, et al. VPGNet: Vanishing point guided network for lane and road marking detection and recognition. *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, October 29, 2017: 1965–1973. https://doi.org/10.1109/ICCV.2017.215.
[14] DC Andrade, F Bueno, F Franco, et al. A novel strategy for road lane detection and tracking based on a vehicle's forward monocular camera. *IEEE Transactions on Intelligent Transportation Systems*, 2019, 20 (4): 1497–1507. https://doi.org/10.1109/TITS.2018.2856361.
[15] Q Zou, H Jiang, Q Dai, et al. Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Transactions on Vehicular Technology*, 2020, 69 (1): 41–54. https://doi.org/10.1109/TVT.2019.2949603.
[16] Y Zhang, Z Lu, D Ma, et al. Ripple-GAN: Lane line detection with ripple lane line detection network and Wasserstein GAN. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 22 (3): 1532-1542. https://doi.org/10.1109/TITS.2020.2971728.
[17] S Chen, L Huang, H Chen, et al. Multi-lane detection and tracking using temporal-spatial model and particle filtering. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23 (3): 2227–2245. https://doi.org/10.1109/TITS.2020.3035614.
[18] E Grave, A Joulin, M Cissé, et al. Efficient softmax approximation for GPUs. *The 34th International Conference on Machine Learning,* 2017, 3: 2111–2119.
[19] M Bertozzi, A Broggi, A Fascioli. Stereo inverse perspective mapping: theory and applications. *Image and Vision Computing*, 1998, 16 (8): 585–590. https://doi.org/10.1016/s0262-8856(97)00093-0.
[20] J Long, E Shelhamer, T Darrell. Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA July 12, 2015: 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.
[21] V Badrinarayanan, A Kendall, R Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39 (12): 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615.
[22] Z Wang, W Ren, Q Qiu. LaneNet: Real-time lane detection networks for autonomous driving. *IEEE Conference on Computer Vision and Pattern Recognition,* Salt Lake City, Utah, USA, June 22, 2018: 1–9. Available: http://arxiv.org/abs/1807.01726.
[23] K Cho, B Merrienboer, C Gulcehre, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014: 1724–1734. https://doi.org/10.3115/v1/D14-1179.
[24] F Chollet. Deep learning with depthwise separable convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, USA, January 25, 2017: 1800–1807. https://doi.org/10.1109/CVPR.2017.195.
[25] F Yu, V Koltun. Multi-scale context aggregation by dilated convolutions. *4th International Conference on Learning Representations*, San Juan, Puerto Rico, May 2-4, 2016: 1–13. arXiv:1511.07122, 2015.
[26] J Wang, K Chen, R Xu, et al. CARAFE: Content-aware reassembly of features. *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, Korea, November 3, 2019: 3007–3016. https://doi.org/10.1109/ICCV.2019.00310.
[27] K He, X Zhang, S Ren, et al. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, June 30, 2016: 770–778. https://doi.org/10.1109/CVPR.2016.90.
[28] Z Liu, Y Lin, Y Cao, et al. Swin Transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE International Conference on Computer Vision*, 2021: 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986.
[29] P Netrapalli. Stochastic gradient descent and its variants in machine learning. *Journal of the Indian Institute of Science*, 99 (2): 201–213. https://doi.org/10.1007/s41745-019-0098-4.
[30] T Y Lin, P Goyal, R Girshick, et al. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 42 (2): 318–327. https://doi.org/10.1109/TPAMI.2018.2858826.
[31] M Ghafoorian, C Nugteren, N Baka, et al. EL-GAN: embedding loss driven generative adversarial networks for lane detection. *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, California, 2019: 256–272.
[32] L Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings of COMPSTAT 2010*, 2010: 177–186. https://doi.org/10.1007/978-3-7908-2604-3_16.

[33] A Paszke, A Chaurasia, S Kim, et al. ENet: A deep neural network architecture for real-time semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition,* Las Vegas, USA, June 30, 2016: 1–10

Yufeng Du received his B.E. degree in material forming and control engineering from *College of Mechanical and Electrical Engineering, Anhui Polytechnic University, China,* in 2019. Now he is pursuing his mater degree in automotive engineering at *Anhui Polytechnic University, China*. His research interests are intelligent networked vehicle, machine learning and computer vision.

Rongyun Zhang is an associate professor at *School of Machinery Engineering, Anhui Polytechnic University, China*. He received his B.S. and Ph.D. degree in vehicle engineering from *Hefei University of Technology, China,* in June 2009 and June 2015, respectively. His research interests focus on vehicle safety control, vehicle intelligent control, pmsm control and electric vehicle technology, etc.

Peicheng Shi is a professor at *School of Machinery and Automobile Engineering, Anhui Polytechnic University, China*. He received his Ph.D. degree in vehicle engineering from *HeFei University of Technology, China,* in 2010. Now he is working at *School of Mechanical and Automotive Engineering Anhui Polytechnic University, China,* his research interests are vibration and control of machinery and automobile vibration.

Linfeng Zhao received the Ph.D. degree from *Hefei University of Technology, Hefei, China,* in 2010. He is currently an associate professor at *School of Automotive and Transportation Engineering, Hefei University of Technology, China*. His current research interests include vehicle dynamics and control, electric power steering system, and intelligent vehicles.

Bin Zhang received his B.E. degree in vehicle engineering from *Anhui Polytechnic University, China,* in 2020. Now he is pursuing a master degree in automotive engineering at *Anhui Polytechnic University, China*. His research interests are vehicle dynamics control, path planning and path tracking.

Yaming Liu received his B.E. degree in vehicle engineering from *Anhui Polytechnic University, China,* in 2020. Now he is pursuing a master degree in automotive engineering at *Anhui Polytechnic University, China*. His research interest is electric vehicle state parameter estimation.