



FFCANet: a frequency channel fusion coordinate attention mechanism network for lane detection

Shijie Li¹ · Shanhua Yao¹ · Zhonggen Wang¹ · Juan Wu¹

Accepted: 29 August 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Lane line detection becomes a challenging task in complex and dynamic driving scenarios. Addressing the limitations of existing lane line detection algorithms, which struggle to balance accuracy and efficiency in complex and changing traffic scenarios, a frequency channel fusion coordinate attention mechanism network (FFCANet) for lane detection is proposed. A residual neural network (ResNet) is used as a feature extraction backbone network. We propose a feature enhancement method with a frequency channel fusion coordinate attention mechanism (FFCA) that captures feature information from different spatial orientations and then uses multiple frequency components to extract detail and texture features of lane lines. A row-anchor-based prediction and classification method treats lane line detection as a problem of selecting lane marking anchors within row-oriented cells predefined by global features, which greatly improves the detection speed and can handle visionless driving scenarios. Additionally, an efficient channel attention (ECA) module is integrated into the auxiliary segmentation branch to capture dynamic dependencies between channels, further enhancing feature extraction capabilities. The performance of the model is evaluated on two publicly available datasets, TuSimple and CULane. Simulation results demonstrate that the average processing time per image frame is 5.0 ms, with an accuracy of 96.09% on the TuSimple dataset and an F1 score of 72.8% on the CULane dataset. The model exhibits excellent robustness in detecting complex scenes while effectively balancing detection accuracy and speed. The source code is available at <https://github.com/ljsj1012/FFCANet/tree/master>

Keywords Lane line detection · Frequency channel · Attention mechanism · Autonomous driving · Row anchor

1 Introduction

With increasing urbanization [1] and significant economic growth, the number of motor vehicles is rising rapidly. This growth has brought serious challenges to urban transportation and has made traffic safety issues more prominent. Consequently, autonomous vehicles (AVs) [2] and advanced

driver assistance systems (ADAS) [3] have emerged as critical technological solutions to alleviate traffic problems and enhance driving safety. AVs achieve the ability to drive autonomously without human intervention through a range of sensors, algorithms, and computing systems. ADAS significantly enhances the safety and convenience of conventional vehicles by providing features such as lane keeping, automatic emergency braking, and adaptive cruise control. Lane line detection technology, one of the core components of AVs and ADAS, plays a crucial role in vehicle positioning, navigation, and path planning. As technology continues to advance, AVs and ADAS will gradually become an integral part of future transportation systems. The optimization and improvement of lane line detection technology, as a core technology of AVs and ADAS, is of great significance for enhancing traffic safety, improving traffic flow, and promoting the development of intelligent transportation systems.

Although many lane line detection methods have been highly successful under certain conditions, vehicles face

✉ Shijie Li
li1696246496@163.com

Shanhua Yao
shhyao@aust.edu.cn

Zhonggen Wang
zgwang@ahu.edu.cn

Juan Wu
yj165@mail.ustc.edu.cn

¹ School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232001, China

complex scenarios such as heavy occlusion, inclement weather, and extreme lighting, as well as elongated features of lane lines that are inherently difficult to detect. This makes accurate and rapid detection of lane lines a challenging task. Currently, existing lane line detection methods are mainly classified into two categories: traditional methods [4, 5] and deep learning methods [6–10]. Traditional lane line detection methods primarily rely on image processing and feature extraction. These methods usually extract the color features [11–13] or edge features [14, 15] of the lane lines, use the inverse perspective transform to obtain a bird's eye view of the image [16], and finally use the Hough transform [17] or sliding window search [12, 13] to fit the lane lines. Traditional methods can achieve better results under simple conditions but often face greater limitations in complex and changing driving scenarios. They exhibit poor adaptability to environmental changes and complex road conditions, require frequent parameter adjustments, have high computational complexity, and offer poor real-time performance. In recent years, deep learning methods have received extensive attention and application in the field of lane line detection, gradually becoming the mainstream method. Deep learning-based lane line detection methods extract and understand features in images by learning autonomously from large amounts of data, allowing for more robust detection in different scenarios [18–21]. The mainstream deep learning methods fall into three categories: segmentation-based methods, anchor-based methods, and parameter-based methods. Segmentation-based methods usually formulate the lane line detection problem as a segmentation task to obtain a pixel-by-pixel predicted segmentation map. This method has higher detection accuracy but slower computation speed and poorer real-time performance, making it difficult to meet the real-time requirements of lane line detection in practice. Anchor-based methods generate candidate regions in an image using predefined anchor frames, then classify and regress these regions to detect and locate lane lines. This method is more efficient in detection but less effective in complex scenes. Parameter-based approaches achieve lane line detection by modeling the lane curve and regressing the model parameters. While parameter-based lane line detection is computationally efficient, it is less accurate in complex and non-regular road scenarios.

To effectively balance lane line detection accuracy and efficiency in complex environments, this paper proposes a frequency channel fusion coordinate attention mechanism network (FFCANet) for lane detection. FFCANet uses ResNet [22] as its backbone network. To enhance the model's ability to extract lane features in complex road scenes, we propose the FFCA module, which mitigates interference from similar external features and improves detection accuracy by capturing lane line details and texture information from multiple spatial directions. Additionally, to

achieve faster detection and address the issue of vision loss, a row-anchor-based prediction and classification method is employed to circumvent the high computational complexity of pixel-by-pixel segmentation. To further enhance the feature extraction capability and capture the dynamic dependencies between channels, the ECA module [23] is incorporated into the auxiliary segmentation branch. Finally, FFCANet was evaluated on the CULane [24] and Tusimple [25] datasets. Compared to other methods, FFCANet significantly enhances detection performance in complex scenes and achieves faster detection speeds while maintaining high accuracy.

The main contributions of the work in this paper are as follows:

- A feature enhancement method with a frequency channel fusion coordinate attention mechanism (FFCA) is proposed to increase feature diversity by capturing lane line details and texture information from multiple spatial directions.
- A row-anchor-based prediction and classification method is employed to improve detection efficiency and address the lack of visual cues.
- The ECA module is incorporated into the auxiliary segmentation branch to capture the dynamic dependencies between channels and further enhance feature extraction capability.
- This approach demonstrates effectiveness and applicability on publicly available benchmark datasets, achieving an effective balance between lane line detection accuracy and speed.

2 Related work

In recent years, lane line detection technology has attracted significant attention from both academia and industry. Existing lane line detection methods are mainly classified into traditional methods and deep learning-based methods.

2.1 Traditional methods

Traditional lane line detection methods primarily utilize algorithms such as edge detection and the Hough transform to identify and track lane lines through image processing and feature extraction techniques. Li et al. [26] employed a multi-channel threshold fusion method based on gradient and background differences combined with HSV color features of lane lines for lane line detection. However, the method exhibits limited adaptability to environmental changes and complex road conditions. Muthalagu et al. [27] utilized a combination of HLS and Lab for color threshold segmentation, applied the Sobel edge detection operator to extract

edge features, fused the color and edge features, and subsequently conducted lane line detection using a sliding window approach. However, this method necessitates frequent parameter adjustments and is slow in detection. Feng et al. [28] utilized enhanced gradient features to detect candidate edges of lane lines, obtained a bird's eye view through inverse perspective transformation, and finally employed an adaptive sliding window approach. However, the method suffers from high computational complexity and suboptimal real-time performance. While traditional lane line detection methods perform well under simple conditions, they exhibit significant limitations in complex and dynamic driving scenarios.

2.2 Deep learning methods

Deep learning-based lane line detection methods offer the advantages of high accuracy and robustness, enabling efficient lane line detection under various road conditions through feature learning from large-scale datasets. These methods can be categorized into three main types: segmentation-based methods, anchor-based methods, and parameter-based methods.

2.2.1 Segmentation-based methods

Segmentation-based methods typically frame the lane line detection problem as a segmentation task, where lane lines are identified through a segmentation map predicted pixel by pixel. Pan et al. [24] introduced a Spatial Convolutional Neural Network (SCNN), employing a layer-by-layer convolutional structure facilitating efficient information exchange between pixels across rows and columns of different layers, thereby enhancing feature capture in images. However, this method exhibits high computational complexity and slow detection speed. Hou et al. [29] proposed the Self Attention Distillation (SAD) module, enabling the network to self-learn and achieve substantial improvements without additional supervision or labeling. However, the method merely enhances the interlayer information flow within the lane area without offering additional monitoring signals for occlusion handling. Xu et al. [30] utilized Neural Architecture Search (NAS) to discover a more effective backbone network for capturing more accurate information to enhance curve lane detection. However, NAS entails extremely high computational expense [31], inefficiency, and time consumption. While segmentation-based methods can attain high-precision detection, they suffer from elevated computational complexity and reduced detection efficiency.

2.2.2 Anchor-based methods

Anchor-based methods utilize predefined anchor frames to generate candidate regions in an image, which are subsequently classified and regressed to detect and locate lane lines. Su et al. [32] introduced a novel vanishing point-guided anchor generator, leveraging multiple structural information related to lane lines to enhance performance. However, the performance of this method may be limited in complex environments. Qin et al. [33] proposed a lane line detection method based on row anchors, treating the lane line detection process as a row-based selection problem using global features, significantly reducing computational cost and addressing the visionless problem. The method struggles to accurately determine the shape of the lanes. Liu et al. [34] proposed a conditional lane line detection strategy employing conditional convolution and row anchors, where the conditional convolution module adaptively adjusts the convolution kernel size, locates the starting point of the lane line, and performs row anchor-based lane line detection. However, the method struggles to recognize the starting point in certain complex scenes, leading to diminished performance. Anchor-based methods offer real-time performance advantages; however, their performance may not be optimal in more complex scenarios.

2.2.3 Parameter-based methods

The parameter-based approach models lane curves with parameters and regresses these parameters to detect lane lines. PolyLaneNet [35] proposed predicting the polynomial coefficients of lane lines via deep polynomial regression, where the output represents the polynomial of each lane line in the image. Despite its computational efficiency, this method lacks accuracy. Liu et al. [36] introduced the application of Transformer to predict the parameters of the lane shape model for each lane line, taking into account road structure and camera pose. However, the method is highly sensitive to prediction parameters, and errors in higher-order coefficients may result in alterations to lane shape. Feng et al. [37] utilized parametric Bezier curves to represent lane lines as parametric curves through curve modeling and employed deep learning networks to directly predict these curve parameters for more efficient and accurate lane line detection. However, in extremely curved road scenarios, cubic curves may inadequately represent lane lines. Although increasing the curve's order partially addresses this issue, higher-order terms become challenging to predict, as small prediction errors can significantly alter lane line shapes. Consequently, the parameter-based method fails to surpass other lane detection methods in accuracy.

The aforementioned methods struggle to detect lane lines accurately and swiftly in complex environments, and

achieving both accuracy and speed simultaneously remains challenging. Consequently, we introduce a novel lane line detection network to address this issue. The proposed method is also anchor-based but successfully handles complex lanes and improves lane line detection accuracy by employing a frequency channel fusion coordinate attention mechanism.

3 Method

In this section, the lane line detection network proposed in this article is detailed. The network structure comprises a feature extraction backbone network (ResNet), a frequency channel fusion coordinate attention module (FFCA), an auxiliary segmentation branch, and a predictive classification branch.

3.1 FFCANet architecture

The overall architecture of FFCANet is shown in Fig. 1. Initially, image preprocessing techniques are employed to mitigate overfitting, and operations such as translation and rotation [38] are applied to the input lane image to obtain an RGB lane line image of size 288×800 . Subsequently, the ResNet backbone network extracts the preliminary features of the image. In this study, primarily ResNet-18 and ResNet-34 are utilized as the feature extraction networks, and the resulting preliminary feature map serves as the input for FFCA. The FFCA module processes the features independently in various spatial directions to capture lane line feature information diversely. It employs multiple frequency components to extract additional lane line information, enhancing feature diversity. This enables the model to more accurately detect and identify lane line features, thus capturing more lane line details and texture information and improving the model's feature extraction capability. The enriched feature map is then fed into the cell prediction classification branch. Here, lane marking anchors are selected in predefined row-oriented cells, and the probability of cells in different rows belonging to a lane line is predicted. With the potential existence of up to four lane lines, cells belonging to the same lane line are classified as one class. Three features are output through scales 2–4 of the ResNet network, serving as inputs for the auxiliary segmentation. The ECA module is integrated into the auxiliary segmentation to precisely highlight the more critical lane line features in the current task, further enhancing feature extraction capability. The auxiliary segmentation branch only functions during the training phase and is removed during the testing phase, thus not affecting runtime speed.

3.2 Frequency channel fusion coordinate attention mechanism

Due to the thin and sparse appearance of the lane lines, they lack distinctive features and are susceptible to interference from other objects with a similar localized appearance. To leverage the shape a priori information of lane lines fully and enhance the model's feature extraction capability, a frequency channel fusion coordinate attention mechanism (FFCA) approach is proposed. By utilizing multiple frequency components to extract more feature information, feature diversity is enhanced, enabling the model to more accurately capture lane line details and texture information. Incorporating both channel information and direction-related position information aids in better locating and identifying lane line features, thereby improving model detection.

3.2.1 Frequency channel attention

Frequency channel attention networks (FcaNet) is a channel attention mechanism based on the discrete cosine transform, which captures subtle texture variations and increases feature diversity by operating on features directly in the frequency domain [39]. Initially, FcaNet utilizes the discrete cosine transform (DCT) to convert the input feature maps into the frequency domain, allowing the network to learn and represent frequency-level information directly. The frequency channel attention module automatically learns and emphasizes the more crucial frequency components, thereby focusing more effectively on the frequency features pertinent to the classification task. Subsequently, the weighted frequency features are mapped back to the spatial domain using the inverse discrete cosine transform (IDCT), and the features are further refined by convolutional layers, ultimately producing a feature representation for classification.

The discrete cosine transform (DCT) is a commonly employed signal processing technique utilized to convert 1D or 2D signals from the spatial domain to the frequency domain. In image processing, DCT finds widespread application in tasks such as image compression and feature extraction. The basis function of the 2D DCT is:

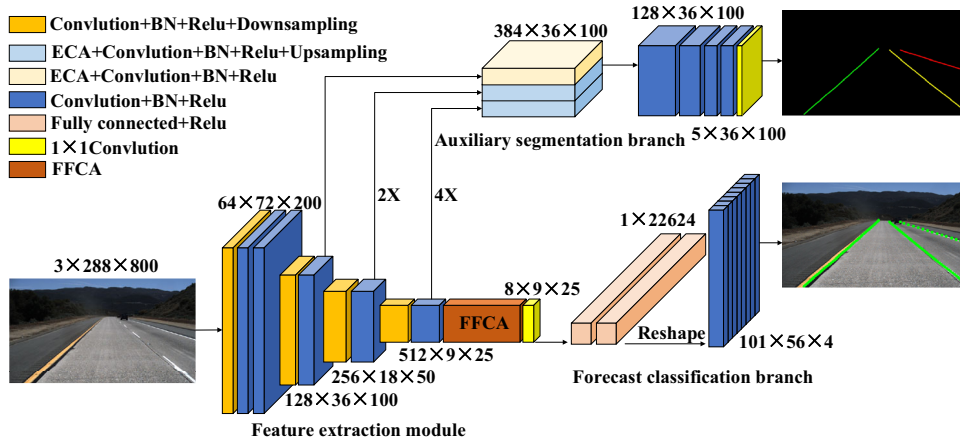
$$B_{h,w}^{i,j} = \cos\left(\frac{\pi h}{H}\left(i + \frac{1}{2}\right)\right) \cos\left(\frac{\pi w}{W}\left(j + \frac{1}{2}\right)\right) \quad (1)$$

The 2D DCT can be written as:

$$f_{h,w}^{2d} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} B_{h,w}^{i,j} \quad (2)$$

where $h, i \in \{0, 1, \dots, H-1\}$, $j, w \in \{0, 1, \dots, W-1\}$, H is the height of x^{2d} , W is the width of x^{2d} , $f^{2d} \in R^{H \times W}$ is the spectrum of the 2D DCT, $x^{2d} \in R^{H \times W}$ is the input. In

Fig. 1 FFCANet overall architecture



contrast, the 2D IDCT can be written as:

$$x_{h,j}^{2d} = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} f_{h,w}^{2d} B_{h,w}^{i,j} \quad (3)$$

Dividing the input feature X into n parts along the channel dimension, the split part is denoted as $[X^0, X^1, \dots, X^{n-1}]$, and $X^i \in R^{C' \times H \times W}, i \in \{0, 1, \dots, n-1\}, C' = C/n, C$ is divisible by n . For each section, assigning the corresponding 2D DCT frequency components, 2D DCT results as $Freq$ vectors for each part, the expression is:

$$Freq^i = 2DDCT^{u_i, v_i}(X^i) = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} X_{:,h,w}^i B_{h,w}^{u_i, v_i} \quad (4)$$

where $i \in \{0, 1, \dots, n-1\}, [u_i, v_i]$ is the 2D index corresponding to the frequency component of $X_i, Freq^i \in R^{C'}$ is the compressed C' -dimensional vector, the entire compression vector can be obtained by concatenating, denoted by:

$$Freq = \text{compress}(X) = \text{cat}([Freq^0, Freq^1, \dots, Freq^{(n-1)}]) \quad (5)$$

where $Freq \in R^C$ is a multispectral vector, the entire multispectral channel attention framework can be written as:

$$ms_att = \text{sigmoid}(fc(Freq)) \quad (6)$$

From Eqs. (5 and 6), it is evident that the frequency channel attention module extends the original global average pooling method to frameworks with multiple frequency components, effectively representing compressed channel information. The frequency channel attention is depicted in Fig. 2.

3.2.2 Coordinate attention

Since FcaNet lacks understanding and application of the spatial domain of the feature map, the coordinate attention (CA) mechanism is incorporated to enhance the model's comprehension of the spatial structure [40]. CA leverages the spatial structure of an image fully and processes features separately in various spatial directions to better capture information in different orientations. By decomposing the channel attention into two feature codes and aggregating the features along different spatial directions, the model can focus on the rows and columns of the image individually. Thus, it can efficiently capture long-range dependence of features in one direction while preserving precise positional information in the other direction. The coordinate attention mechanism is illustrated in Fig. 3.

Firstly, each channel of the input feature x is encoded along the x-axis and y-axis using two spatially scoped pooling kernels, $(H, 1)$ and $(1, W)$, respectively. The coding of the cth channel with height h and the coding of the cth channel with width w are shown in Eqs. (7) and (8), respectively.

$$z_c^h(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i) \quad (7)$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \leq j < H} x_c(j, w) \quad (8)$$

where W and H are the width and height of the input features, respectively; i, j are feature calculation coordinates, respectively; x_c is the feature of the input cth channel.

The two formulas above encode each channel along the x-axis coordinates and y-axis coordinates, respectively, resulting in a pair of orientation-aware feature maps that preserve the positional information of each channel of the feature map. Simultaneously capturing a long-range dependency along the coordinate direction for the attention module and maintaining precise positional information along the

Fig. 2 Frequency channel attention

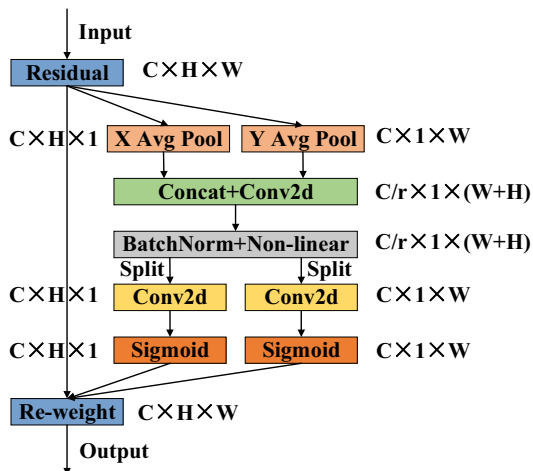
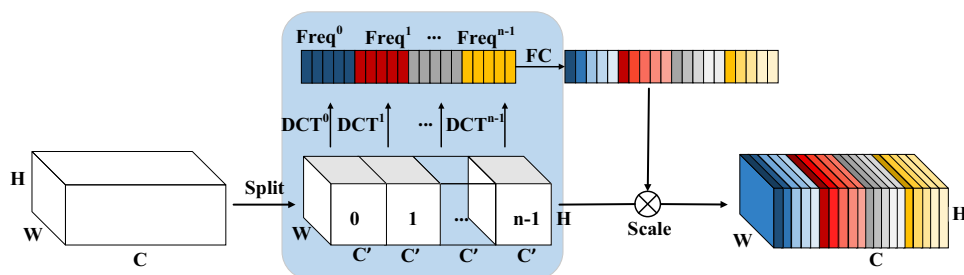


Fig. 3 Coordinate attention mechanism

other coordinate direction helps the network more accurately localize and identify important features.

The feature map is obtained from the embedding stage of coordinate information, which needs to be processed by splicing operation, followed by 1×1 convolution and non-linear activation, which is represented by the formula:

$$f = \delta(F_1([z^h, z^w])) \tag{9}$$

where δ is a nonlinear activation function; $f \in R^{C/r \times (H+W)}$; $[\cdot, \cdot]$ is the splicing operation along the spatial dimension; F_1 is a 1×1 convolution operation.

f is then split into two separate tensors, $f^h \in R^{C/r \times H}$ and $f^w \in R^{C/r \times W}$, along the spatial dimension. Two 1×1 convolutions F_h and F_w transform f^h and f^w to have the same number of channels as the input features x , respectively, and the formula is expressed as:

$$g^h = \sigma(F_h(f^h)) \tag{10}$$

$$g^w = \sigma(F_w(f^w)) \tag{11}$$

where σ is expressed as a sigmoid function. The output feature maps g^h and g^w are expanded and used as attentional weights for the x- and y-axis coordinates, respectively.

Finally, the process of re-weighting the original feature map x through the coordinate attention can be written as:

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{12}$$

3.3 Auxiliary segmentation branch

The auxiliary segmentation branch is a segmentation method that utilizes multi-scale features to emulate local features, primarily aimed at enhancing the network’s semantic analysis capability. This helps the model better comprehend the structural features of the lanes and enhances the feature extraction ability of the convolutional layer. Segmentation branches are utilized solely during the training phase of the model and are omitted during the testing phase, thereby ensuring that the speed of the run remains unaffected despite the addition of extra segmentation tasks. To effectively capture the dynamic dependencies between channels and accurately highlight the more significant lane line features in the current task, the efficient channel attention (ECA) module is introduced in the auxiliary segmentation branch to further enhance the performance of lane line segmentation networks.

3.3.1 ECA module

The fundamental mechanism of the ECA module is to adaptively capture dependencies between channels using simple and efficient 1D convolutions, eliminating the need for cumbersome downscaling and upscaling processes [23]. Compared to the traditional attention mechanism, it circumvents the complex multi-layer perceptual machine structure, thereby reducing model complexity and computational burden. By computing an adaptive convolution kernel size, the ECA module directly applies one-dimensional convolution on the channel features, enabling it to learn the importance of each channel relative to the others. The ECA module is depicted in Fig. 4.

The size k of a 1D convolutional kernel is proportional to the channel dimension C , and there is a mapping ϕ between k and C . The simplest mapping relationships are linear functions such as $\phi(k) = \gamma * k - b$. The channel dimension C is usually set to a power of 2. Using an exponential function to

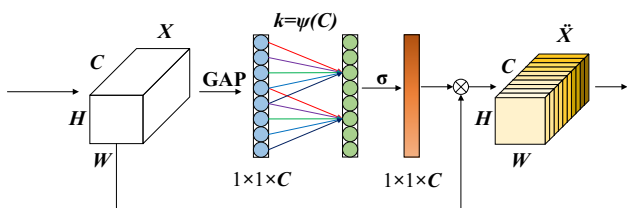


Fig. 4 ECA module

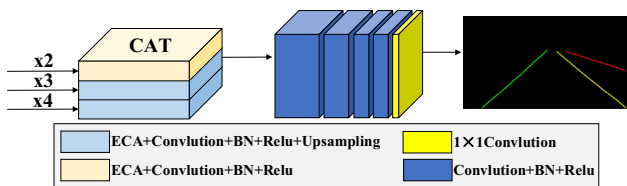


Fig. 5 Auxiliary segmentation network

approximate the mapping ϕ , this can be expressed as:

$$C = \phi(k) = 2^{(\gamma * k - b)} \tag{13}$$

Given the channel dimension C , the convolution kernel size k can be determined adaptively by the expression.

$$k = \psi(C) = \left\lfloor \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right\rfloor_{odd} \tag{14}$$

where $|t|_{odd}$ denotes the closest odd number to t . In the simulation experiment set $\gamma = 2, b = 1$ respectively. By mapping ϕ interactions, the higher dimensional channels have longer range interactions, while the lower dimensional channels have shorter range interactions by nonlinear mapping.

3.3.2 Auxiliary segmentation network

Through the ResNet backbone network layer2, layer3, layer4 layers will get three feature maps $\times 2, \times 3, \times 4$. These feature maps serve as the input for the segmentation network, enabling the segmentation method of local feature reconstruction using multi-scale features. The segmentation network is depicted in Fig. 5.

Initially, to capture the dependencies between feature channels, the $\times 2, \times 3$ and $\times 4$ feature maps undergo ECA operations individually. Following convolution, normalization, and activation operations, it is necessary to upsample $\times 3, \times 4$. Finally, the splicing operation is performed. The specific steps are illustrated in Fig. 6.

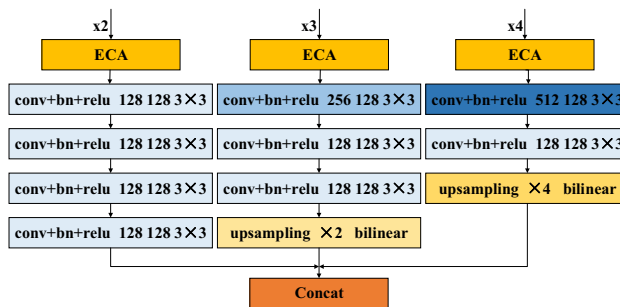


Fig. 6 Specific steps for segmented networks

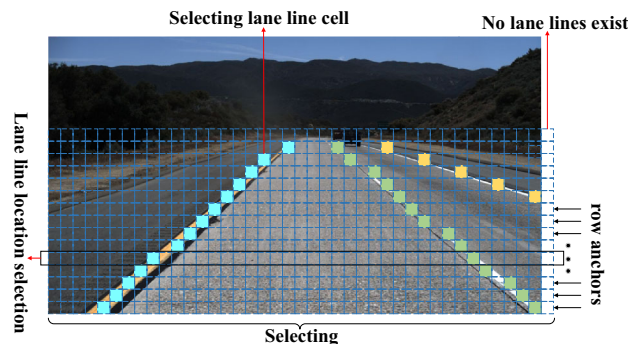


Fig. 7 Location selection and classification based on row anchors

3.4 Location selection and classification based on row anchors

Traditional semantic segmentation for lane line segmentation pixel by pixel point has high computational complexity and slow lane line detection. In order to enhance the efficiency of lane line detection and address issues such as lane occlusion, this paper employs a location selection and classification method based on row anchors [33].

Lane line detection has been transformed into the problem of selecting lane marking anchor points within predefined row-oriented cells using global features. Initially, the lane image is grid-divided into a specific number of rows, with each row further subdivided into a certain number of cells. Subsequently, for each row of cells, predictions are made regarding the cells containing lane lines. Finally, the cells identified to contain lane lines across all predefined rows are categorized, with those belonging to the same lane line grouped together. This approach to detecting lane line anchors circumvents the need to individually process each pixel point of the lane, thereby significantly enhancing detection efficiency. Moreover, the utilization of anchors on global image features results in a larger sensory field, which is more conducive to addressing challenging scenarios. The location selection and classification based on row anchors is depicted in Fig. 7.

Assuming the maximum number of lanes is C , h represents the number of divided rows (i.e., the number of row anchors), and w denotes the number of cells per row. If X represents a global image feature, then f^{ij} denotes the classifier used to select the lane position on the i th lane and j th row anchor. Then the lane is predicted to be:

$$P_{i,j,:} = f^{ij}(X) \quad (15)$$

where $i \in [1, C]$, $j \in [1, h]$. $P_{i,j,:}$ is a $(w+1)$ -dimensional vector, representing the probability of selecting $(w+1)$ cells for the i th lane and j th row anchor.

3.5 Loss function

During lane line detection, the background typically occupies most of the image, while the lane lines constitute only a small portion of the targets. To enhance the network's ability to learn and focus on critical and difficult-to-classify lane line samples, classification loss is introduced. The classification loss is defined as follows:

$$L_{cls} = \sum_{i=1}^C \sum_{j=1}^h L_{CE}(P_{i,j,:}, T_{i,j,:}) \quad (16)$$

where L_{CE} denotes the cross entropy loss. $T_{i,j,:}$ indicates that the marker is in the correct lane position.

Since the lane points in neighboring rows of anchors are close to each other, lane locations are represented by classification vectors. Thus, continuity is achieved by constraining the distribution of classification vectors across neighboring row anchors. The similarity loss function is defined as:

$$L_{sim} = \sum_{i=1}^C \sum_{j=1}^{h-1} \|P_{i,j,:} - P_{i,j+1,:}\|_1 \quad (17)$$

where $\|\cdot\|_1$ represents L_1 norm.

Another loss function emphasizes the shape of the lane. Given that most lanes are straight, second-order difference equations are employed to constrain their shape. First, the probabilities for different lane positions are computed using the softmax function, as expressed by:

$$Prob_{i,j,:} = \text{softmax}(P_{i,j,1:w}) \quad (18)$$

where $P_{i,j,1:w}$ denotes the w -dimensional vector. $Prob_{i,j,:}$ represents the probability of each lane position. Next, by using the expectation of the lane line prediction rather than an approximation, the expected lane position can be expressed

as:

$$Loc_{i,j} = \sum_{k=1}^w k \cdot Prob_{i,j,k} \quad (19)$$

where $Prob_{i,j,k}$ represents the probability of the i th lane, the j th row anchor, and the k th position.

According to Eq. (19), the second-order difference constraint can be formulated as:

$$L_{shp} = \sum_{i=1}^C \sum_{j=1}^{h-2} \|(Loc_{i,j} - Loc_{i,j+1}) - (Loc_{i,j+1} - Loc_{i,j+2})\|_1 \quad (20)$$

where $Loc_{i,j}$ denotes the position on the i th lane, j th row anchor. Finally, the structural loss can be expressed as:

$$L_{str} = L_{sim} + \lambda L_{shp} \quad (21)$$

where λ denotes the loss coefficient.

We employ cross entropy as an auxiliary segmentation loss, then the total loss function can be expressed as:

$$L_{total} = L_{cls} + \alpha L_{str} + \beta L_{seg} \quad (22)$$

where α and β denote the loss coefficients, and L_{seg} denotes the auxiliary segmentation loss.

4 Experiments and results

To validate the effectiveness and applicability of the FFCANet proposed in this paper, its performance is evaluated alongside other lane line detection methods on two public datasets, TuSimple and CULane, respectively. The following sections focus on experimental setup, ablation study, performance comparison and analysis, and visualization of results.

4.1 Experimental setting

4.1.1 Datasets

To assess the performance of the model proposed in this paper, it is trained and tested on two widely used benchmark datasets for lane line detection: TuSimple and CULane. The details of the two datasets are provided in Table 1.

TuSimple is one of the most widely used datasets for lane line inspection, collected under consistent motorway lighting conditions. It encompasses scenes captured in various weather conditions on motorways, with lane markings

Table 1 Details of the datasets

Dataset	#Frame	Train	Validation	Test	Resolution	#Lane	Environment
TuSimple	6408	3268	358	2782	1280 × 720	≤ 5	highway
CULane	133,235	88,880	9675	34,680	1640 × 590	≤ 4	urban and highway

annotated in each image. Lanes are annotated by the 2D coordinates of sampled points, with a uniform height interval of 10 pixels, and the dataset includes both straight and curved lanes. Created by the Chinese University of Hong Kong, the CULane dataset was primarily collected in cities and on motorways with nine complex driving scenarios including normal, crowd, night, no line, shadow, arrow, curve, dazzle light, cross.

4.1.2 Evaluation of indicators

The official evaluation metrics for both TuSimple and CULane are different. For the TuSimple dataset, the primary evaluation metrics are accuracy (Acc), false positive (FP), and false negative (FN). The expression for Acc is as follows:

$$Acc = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \quad (23)$$

where C_{clip} denotes the number of correctly predicted lane points, S_{clip} then denotes the total number of true and valid lane points in the image. A point prediction is considered correct if more than 85% of the predicted lane points are within the threshold of the ground truth. The FP and FN are calculated as:

$$FP = \frac{F_{pred}}{N_{pred}} \quad (24)$$

$$FN = \frac{M_{pred}}{N_{gt}} \quad (25)$$

where F_{pred} denotes the number of incorrectly predicted lanes and N_{pred} denotes the number of predicted lanes. M_{pred} denotes the number of missed lanes and N_{gt} denotes the number of ground truth lanes.

For the CULane dataset, each lane line is assumed to have a width of 30 pixels. The Intersection over Union (IOU) between the model predictions and the ground truth is computed, with predictions having an IOU > 0.5 considered as true positives. Finally, the F1 score is used as the evaluation metric for the CULane dataset. The expression is as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (26)$$

$$Precision = \frac{TP}{TP + FP} \quad (27)$$

Table 2 Parameter settings on different datasets

Dataset	TuSimple	CULane
Number of rows	56	18
Number of row anchors	100	200
Number of lane lines	4	4

$$Recall = \frac{TP}{TP + FN} \quad (28)$$

where TP , FP , and FN indicate true positives, false positives, and false negatives, respectively.

4.1.3 Implementation details

The models in this paper were trained and tested using NVIDIA GeForce RTX 4060 Laptop GPU 13th Gen Intel(R) Core(TM) i7-13650HX CPU, and the deep learning framework was PyTorch. We utilize a pre-trained ResNet as the backbone network, and the input images are resized to 288 × 800 pixels. To mitigate overfitting and enhance generalization, we apply a data augmentation strategy that includes scaling, panning, and flipping. The number of training epochs for the TuSimple dataset is set to 200, the batch size is set to 32, the AdamW optimiser [41] is utilized with an initial learning rate of 4e-4 and a weight decay rate of e-4, and the cosine annealing learning rate decay strategy [42] is employed during training. The number of training epochs for the CULane dataset is set to 50, the batch size is set to 32, and for training with a single GPU, the initial learning rate needs to be set to 0.025. For different datasets, the division of row anchor is also different, and the specific parameter settings are shown in Table 2.

4.2 Ablation study

To validate the effectiveness of the proposed FFCA module and the introduction of the ECA module in the auxiliary segmentation branch, ablation studies are conducted on the TuSimple dataset to show the performance of each module. ResNet-18 is selected as the baseline network, and FFCA module and ECA module are added to the baseline network in turn to assess the impact of each module on lane line detection. The models are trained and tested separately with the

Table 3 Results of ablation experiments on the TuSimple dataset

Baseline	FFCA	ECA	Acc	FP	FN
✓			95.82	0.1905	0.0392
✓	✓		95.98	0.1891	0.0372
✓		✓	95.93	0.1894	0.0379
✓	✓	✓	96.06	0.1883	0.0365

The optimal results for each indicator is indicated in bold

identical parameter settings, and the results of the ablation experiments are presented in Table 3.

From the data in the table above, it is evident that after the introduction of the proposed FFCA module to the original network, the accuracy is increased by 0.16% and the false positives and false negatives are reduced by 0.0014 and 0.0020 respectively. This demonstrates that the proposed FFCA module can make full use of the shape a priori information of lane lines to better locate and identify lane line features. With the introduction of the ECA module in the assisted segmentation branch, the accuracy improved by 0.11% and the false positives and false negatives were reduced by 0.0011 and 0.0013, respectively. It is shown that the ECA module is able to capture the dynamic dependencies between channels, more accurately highlighting the more important lane line features for the task at hand. The simultaneous introduction of the FFCA and ECA modules resulted in a 0.24% increase in accuracy and a 0.0022 and 0.0027 reduction in false positives and false negatives, respectively, affirming the efficacy of both modules. The visual comparison results of the ablation experiments are shown in Fig. 8. The figure indicates that lane line fitting is enhanced by the incorporation of the FFCA and ECA modules. This also demonstrates the validity and feasibility of incorporating these modules.

4.3 Performance comparison and analysis

The FFCANet of this paper is evaluated on two public datasets, TuSimple and CULane, using ResNet-18 and ResNet-34 as backbone networks, respectively, and compared with other lane line detection methods. The comparison methods were tested on the same hardware under the same conditions. For the TuSimple dataset, seven methods, SCNN [24], SAD [29], SGNNet [32], PolyLNet [35], UFLD [33], CondLNet [34], MAM [43], and BezierLNet [37] were selected for comparison. Acc, FP, FN, and Runtime (elapsed time per frame) are used as evaluation metrics and the results are shown in Table 4.

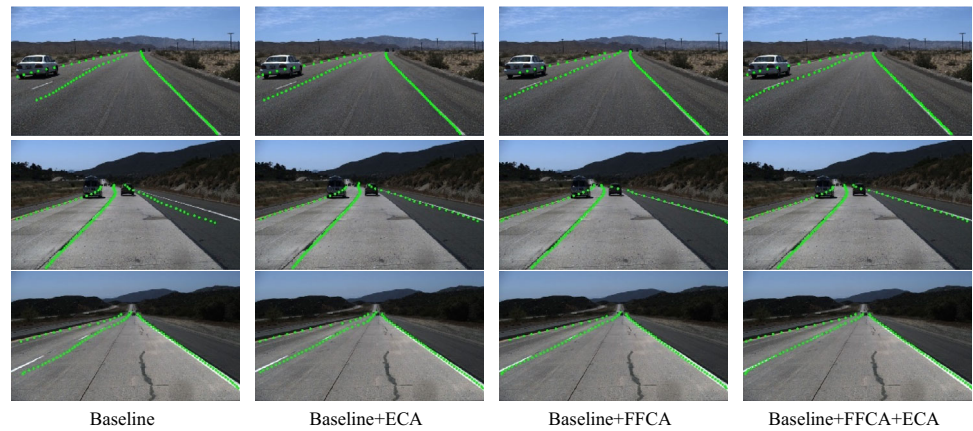
Since the TuSimple dataset only contains motorway scenes with adequate lighting conditions and favorable weather conditions, and the traffic scenarios are relatively uncomplicated, there is little difference in the accuracy of

each lane line detection method. When using the ResNet-18 backbone, the Acc of FFCANet is 96.06%, which is 0.19%, 2.7%, 0.24%, 0.58%, 0.23%, and 0.65% higher than SGNNet, PolyLNet, UFLD, CondLNet, MAM, and BezierLNet. Compared with UFLD, which is also based on row anchor detection method, FP and FN are reduced by 0.0022 and 0.0027, respectively. When using the ResNet-34 backbone, the Acc of FFCANet is 96.09%, which is 0.22%, 2.73%, 0.23%, 0.61%, 0.26%, and 0.44% higher than SGNNet, PolyLNet, UFLD, CondLNet, MAM, and BezierLNet. FP and FN were reduced by 0.0026 and 0.0032 over UFLD, respectively. Although the Acc of SCNN and SAD is inferior to that of FFCANet, the FFCANet proposed in this paper performs superior in terms of time consumed per image frame, which is 46.03 and 4.62 times faster than SCNN and SAD, respectively. In summary, FFCANet has a fast operating speed while maintaining a high Acc.

For the CULane dataset, seven methods, SCNN [24], SAD [29], Res34-VP [21], E2E [19], LSTR [36], DeepLab [20], GCSbn [44], STLNet [45], and UFLD [33] were selected for comparison. F1 values and FPS were used as assessment indicators and the results are shown in Table 5. From the table, it can be seen that FFCANet achieves superior results in terms of F1 value and speed, with the fastest speed of 345.8 FPS, with excellent real-time performance. When using the ResNet-34 backbone network, FFCANet has an F1 value of 72.8, outperforming all other comparison methods except STLNet and also has a faster speed advantage. Although STLNet has a slightly higher F1 score than our method, our approach is 3.17 times faster. The F1 value of FFCANet outperforms that of UFLD when using the same backbone network with little difference in speed, proving that the FFCA proposed in this paper as well as the introduction of ECA in the segmentation branch enhances the feature extraction capability. The above results demonstrate the excellent performance of FFCANet in challenging scenarios.

4.4 Visualization results

In order to visualize the detection effect of the method in this paper, the detection is performed on two public datasets, TuSimple and CULane. The visualization results of FFCANet on TuSimple dataset are depicted in Fig. 9. As

Fig. 8 Visual comparison results of ablation experiments**Table 4** Comparison with other methods on the TuSimple dataset

Method	Acc	FP	FN	Runtime	Params	GFlops
SCNN [24]	96.53	0.0617	0.0180	133.5	62.2	328.4
SAD [29]	96.64	0.0602	0.0205	13.4	11.8	43
SGNet [32]	95.87	–	–	10.9	–	–
PolyLNet [35]	93.36	0.0942	0.0933	8.7	22.8	25.1
UFLD(ResNet-18) [33]	95.82	0.1905	0.0392	2.6	27.2	9.2
UFLD(ResNet-34) [33]	95.86	0.1898	0.0386	4.8	54.6	16.9
CondLNet [34]	95.48	0.0218	0.0380	4.5	11.9	10.2
MAM [43]	95.83	0.1903	0.0389	3.6	32.5	11.6
BezierLNet(ResNet-18) [37]	95.41	0.0503	0.0385	4.0	4.1	13.4
BezierLNet(ResNet-34) [37]	95.65	0.0510	0.0390	5.7	9.5	24.6
FFCANet(ResNet-18) (Ours)	96.06	0.1883	0.0365	2.9	31.6	9.8
FFCANet(ResNet-34) (Ours)	96.09	0.1872	0.0354	5.0	56.3	17.5

The optimal results for each indicator is indicated in bold

Table 5 Comparison with other methods on the CULane dataset

Method	Normal	Crowded	Dazzle	Shadow	No line	Arrow	Curve	Night	Total	FPS
SCNN [24]	90.6	69.7	58.5	66.9	43.4	84.1	64.4	66.1	71.6	7.5
SAD [29]	90.1	68.8	60.2	65.9	41.6	84.0	65.7	66.0	70.8	74.6
Res34-VP [21]	90.4	69.2	61.4	62.5	43.1	83.5	64.7	63.8	70.9	38.5
E2E(ResNet-18) [19]	90.0	69.7	60.2	62.5	43.2	83.2	70.3	63.3	70.8	–
E2E(ResNet-34) [19]	90.4	69.9	61.5	68.1	45.0	83.7	69.8	63.2	71.5	–
LSTR [36]	86.8	67.3	56.6	59.8	40.1	78.7	56.6	59.9	68.7	–
DeepLab [20]	87.4	64.1	54.1	60.7	38.1	79.0	59.8	60.6	66.7	77.3
GCSbn [44]	89.4	67.7	59.8	62.5	40.9	83.1	60.6	63.4	69.5	189.6
STLNet [45]	91.8	70.2	65.9	69.3	48.8	85.3	67.5	68.2	73.6	62.8
UFLD(ResNet-18) [33]	87.7	66.0	58.4	62.8	40.2	81.0	57.9	62.1	68.4	384.2
UFLD(ResNet-34) [33]	90.7	70.2	59.5	69.3	44.4	85.7	69.5	66.7	72.3	209.7
FFCANet(ResNet-18) (Ours)	90.5	69.0	61.5	62.5	41.8	83.7	63.8	63.7	70.9	345.8
FFCANet(ResNet-34) (Ours)	91.1	70.4	61.7	69.6	42.9	84.8	69.4	67.1	72.8	198.2

The optimal results for each indicator is indicated in bold

Fig. 9 Visualization results on the TuSimple dataset



observed from the figure, the lane lines at the far end of the bend are accurately fitted, as shown by the yellow elliptical frames in the figure; and the issue of lane line occlusion is effectively addressed, as shown by the red elliptical frames in the figure. The network demonstrates excellent performance in detecting lane straightness, curves and vehicle occlusion.

The visualization results on the CULane dataset are illustrated in Fig. 10. A variety of lane line scenarios from the CULane dataset were selected. It is evident from the figure that FFCANet exhibits exceptional lane line detection performance in complex scenarios. Excellent detection of vehicle obstructions, glare interference, and curved lane lines is shown in the blue, red, and yellow ellipse frames in the figure respectively. Analysis of the visualization results demonstrates that FFCANet exhibits outstanding generalization ability and robustness in detecting lane lines in complex scenarios.

The visualisation of the CULane dataset presented in Fig. 11 shows that our method obtains smoother and more accurate lane lines in these challenging scenarios compared to other methods.

5 Conclusion

In this paper, we introduce FFCANet, a frequency channel fusion coordinate attention mechanism network for lane detection. FFCANet adopts ResNet as its backbone network. We introduce the FFCA module, which captures lane line details and texture information from different spatial orientations to enhance feature diversity. Additionally, in order to effectively address the challenges of detection efficiency and the absence of visual cues, we employ a row anchor-based prediction and classification method. This method avoids the high computational complexity associated with pixel-by-pixel segmentation by treating lane line detection as a problem of selecting lane marking anchors in row-oriented cells predefined by global features. To further enhance the feature extraction capability, an ECA module is introduced in the auxiliary segmentation branch to capture the dynamic dependencies between channels. We evaluated FFCANet on two publicly available benchmark datasets, Tusimple and CULane, and designed ablation experiments to validate the effectiveness of each module. Experimental results demonstrate that the proposed method achieves a balance between detection accuracy and efficiency in complex road scenarios. Furthermore, the lightweight ResNet-18 version achieves a processing speed of 345.8 frames per second.

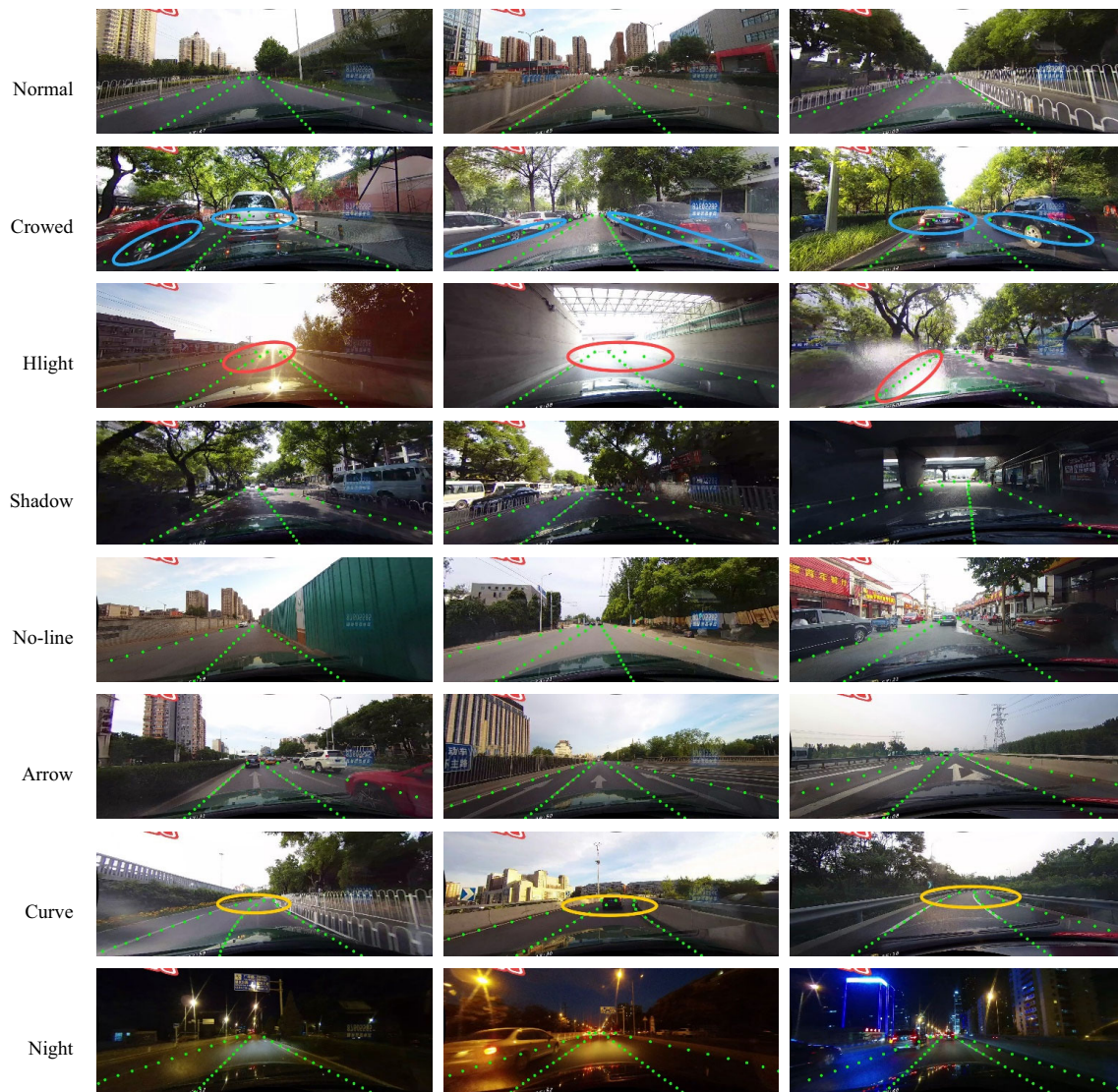


Fig. 10 Visualisation results on the CULane dataset

In future research, an integrated lane line detection scheme that combines vision algorithms with LiDAR technology could be explored. By integrating convolutional neural networks (CNNs) with LiDAR, 3D information of lane lines can be obtained to enhance detection accuracy, stability, and

real-time performance. This integration aims to enhance the perception and decision-making capabilities of autonomous vehicles in complex traffic environments, thereby providing crucial technical support for the advancement of self-driving cars.

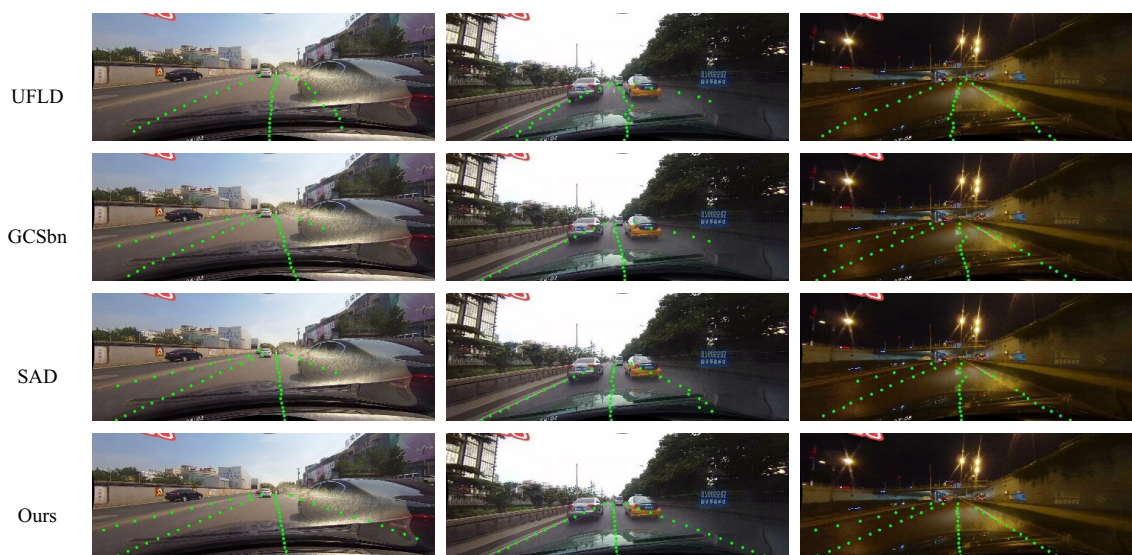


Fig. 11 Visual comparison results of UFLD, GCSbn, SAD and our method on the CULane dataset

Author contributions Shijie Li conceived and designed the study, participated in the survey, methodology, analysed the data, and wrote the original manuscript. Shanhua Yao and Zhonggen Wang provided expert guidance during the research process and revised the manuscript. Juan Wu is responsible for funding acquisition and review. All authors reviewed the manuscript.

Funding This work is supported by the National Natural Science Foundation of China under Grant 62105004.

Data availability Tusimple dataset is available at <https://www.kaggle.com/datasets/manideep1108/tusimple?resource=download>. Culane dataset is available at <https://xingangpan.github.io/projects/CULane.html>. Data is provided within the manuscript.

Declarations

Conflict of interest The authors declare no competing interests.

References

- Qin, Y., Zhao, N., Yang, J., Pan, S., Sheng, B., Lau, R.W.: Urbanevolver: function-aware urban layout regeneration. *Int. J. Comput. Vis.* **132**, 1–20 (2024)
- Wu, Z., Qiu, K., Yuan, T., et al.: A method to keep autonomous vehicles steadily drive based on lane detection. *Int. J. Adv. Robot. Syst.* **18**, 172988142110029 (2021)
- Gerstmaier, M., Gschwandtner, M., Findenig, R., et al.: Miniaturized advanced driver assistance systems: a low-cost educational platform for advanced driver assistance systems and autonomous driving. *IEEE Signal Process. Mag.* **38**(3), 105–114 (2021)
- Fan, G., Li, B., Han, Q., et al.: Robust lane detection and tracking based on machine vision. *ZTE Commun.* **18**(4), 69–77 (2020)
- Wu, H.Y., Zhao, X.M.: Multi-interference lane recognition based on IPM and edge image filtering. *Chin. J. Highw. Transp.* **33**(5), 153–164 (2020)
- Chen, Y., Xiang, Z., Du, W.: Improving lane detection with adaptive homography prediction. *Vis. Comput.* **39**, 581–595 (2023)
- Haris, M., Hou, J., Wang, X.: Lane line detection and departure estimation in a complex environment by using an asymmetric kernel convolution algorithm. *Vis. Comput.* **39**, 519–538 (2023)
- Wang, Q., Han, T., Qin, Z., et al.: Multitask attention network for lane detection and fitting. *IEEE Trans. Neural Netw. Learn. Syst.* **33**(3), 1066–1078 (2020)
- Zhang, J., Deng, T., Yan, F., et al.: Lane detection model based on spatio-temporal network with double convolutional gated recurrent units. *IEEE Trans. Intell. Transp. Syst.* **23**(7), 6666–6678 (2021)
- Wang, Y., Jing, Z., Ji, Z., et al.: Lane detection based on two-stage noise features filtering and clustering. *IEEE Sens. J.* **22**(15), 15526–15536 (2022)
- Lee, C., Moon, J.H.: Robust lane detection and tracking for real-time applications. *IEEE Trans. Intell. Transp. Syst.* **19**(12), 4043–4048 (2018)
- Yu, Z., Wu, X.B., Shen, L.: Illumination invariant lane detection algorithm based on dynamic region of interest. *Comput. Eng.* **43**(2), 43–56 (2017)
- Feng, Y., Li, J.: Robust lane detection and tracking for autonomous driving of rubber-tired gantry cranes in a container yard. In: 2022 IEEE 18th international conference on automation science and engineering (CASE), pp. 1729–1734 (2022)
- Yoo, H., Yang, U., Sohn, K.: Gradient-enhancing conversion for illumination-robust lane detection. *IEEE Trans. Intell. Transp. Syst.* **14**(3), 1083–1094 (2013)
- Li, Q., Zhou, J., Li, B., et al.: Robust lane-detection method for low-speed environments. *Sensors.* **18**(12), 4274 (2018)
- Borkar, A., Hayes, M., Smith, M.T.: A novel lane detection system with efficient ground truth generation. *IEEE Trans. Intell. Transp. Syst.* **13**(1), 365–374 (2011)
- Niu, J., Lu, J., Xu, M., et al.: Robust lane detection using two-stage feature extraction with curve fitting. *Pattern Recognit.* **59**, 225–233 (2016)
- Ran, H., Yin, Y., Huang, F., et al.: Flamnet: a flexible line anchor mechanism network for lane detection. *IEEE Trans. Intell. Transp. Syst.* **24**(11), 12767–12778 (2023)
- Yoo, S., Lee, H.S., Myeong, H., et al.: End-to-end lane marker detection via row-wise classification. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp. 1006–1007 (2020)

20. Chen, L.C., Papandreou, G., Kokkinos, I., et al.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(4), 834–848 (2017)
21. Liu, Y.B., Zeng, M., Meng, Q.H.: Heatmap-based vanishing point boosts lane detection. arXiv preprint [arXiv:2007.15602](https://arxiv.org/abs/2007.15602) (2020)
22. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778 (2016)
23. Wang, Q., Wu, B., Zhu, P., et al.: ECA-Net: Efficient channel attention for deep convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 11534–11542 (2020)
24. Pan, X., Shi, J., Luo, P., et al.: Spatial as deep: Spatial cnn for traffic scene understanding. In: 32nd AAAI Conference on Artificial Intelligence, pp 7276–7283 (2018)
25. TuSimple. <https://www.kaggle.com/datasets/manideep1108/tusimple?resource=download>. Accessed 28 March 2024
26. Li, J., Shi, X., Wang, J., et al.: Adaptive road detection method combining lane line and obstacle boundary. *IET Image Process.* **14**(10), 2216–2226 (2020)
27. Muthalagu, R., Bolimera, A., Kalaichelvi, V.: Lane detection technique based on perspective transformation and histogram analysis for self-driving cars. *Comput. Electr. Eng.* **85**, 106653 (2020)
28. Feng, Y., Li, J.: Robust accurate lane detection and tracking for automated rubber-tired gantries in a container terminal. *IEEE Trans. Intell. Transp. Syst.* **24**(10), 11254–11264 (2023)
29. Hou, Y., Ma, Z., Liu, C., et al.: Learning lightweight lane detection cnns by self attention distillation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp. 1013–1021 (2019)
30. Xu, H., Wang, S., Cai, X., et al.: Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In: European conference on computer vision (ECCV), Glasgow, UK, pp. 689–704 (2020)
31. Chen, Z., Qiu, G., Li, P., Zhu, L., Yang, X., Sheng, B.: Mngnas: distilling adaptive combination of multiple searched networks for one-shot neural architecture search. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(11), 13489–13508 (2023)
32. Su, J., Chen, C., Zhang, K., et al.: Structure guided lane detection. arXiv preprint [arXiv:2105.05403](https://arxiv.org/abs/2105.05403) (2021)
33. Qin, Z., Wang, H., Li, X.: Ultra fast structure-aware deep lane detection. In: Computer vision-ECCV 2020: 16th European conference, Glasgow, UK, pp. 276–291 (2020)
34. Liu, L., Chen, X., Zhu, S., et al.: Conclanenet: a top-to-down lane detection framework based on conditional convolution. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 3773–3782 (2021)
35. Tabelini, L., Berriel, R., Paixao, T.M., et al.: PolyLANet: Lane estimation via deep polynomial regression. In: 2020 25th international conference on pattern recognition (ICPR), pp 6150–6156 (2021)
36. Liu, R., Yuan, Z., Liu, T., et al.: End-to-end lane shape prediction with transformers. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 3694–3702 (2021)
37. Feng, Z., Guo, S., Tan, X., et al.: Rethinking efficient lane detection via curve modeling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 17062–17070 (2022)
38. Qu, Z., Jin, H., Zhou, Y., et al.: Focus on local: Detecting lane marker from bottom up via key point. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 14122–14130 (2021)
39. Qin, Z., Zhang, P., Wu, F., et al.: Fcanet: Frequency channel attention networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp 783–792 (2021)
40. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 13713–13722 (2021)
41. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)
42. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint [arXiv:1608.03983](https://arxiv.org/abs/1608.03983) (2016)
43. Song, Y., Huang, T., Fu, X., et al.: A novel lane line detection algorithm for driverless geographic information perception using mixed-attention mechanism ResNet and row anchor classification. *ISPRS Int. J. Geoinf.* **12**(3), 132 (2023)
44. Yang, X., Ji, W., Zhang, S., et al.: Lightweight real-time lane detection algorithm based on ghost convolution and self batch normalization. *J. Real-Time Image Process.* **20**(4), 69 (2023)
45. Du, Y., Zhang, R., Shi, P., et al.: ST-LaneNet: lane line detection method based on swin transformer and LaneNet. *Chin. J. Mech. Eng.* **37**(1), 14 (2024)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Shijie Li is currently pursuing his master's degree in the School of Electrical and Information Engineering at Anhui University of Science and Technology. His research interests include lane line detection and deep learning.



Shanhua Yao received the Ph.D. degree from the Anhui University, China, in 2010. Currently, he is a professor at the School of Electrical and Information Engineering, Anhui University of Science and Technology. His research interests include the study of coal mine communication theory and technology, and intelligent signal processing.



Zhonggen Wang received the Ph.D. degree from the Anhui University, China, in 2014. Currently, he is a professor at the School of Electrical and Information Engineering, Anhui University of Science and Technology. His research interests include computational electromagnetics, array antennas, and reflect arrays.



Juan Wu received the Ph.D. degree from the University of Science and Technology of China, in 2019. Currently, she teaches at the School of Electrical and Information Engineering, Anhui University of Science and Technology. Her research interests include quantum communications, signal and information processing, optical imaging.