High-order deep infomax-guided deformable transformer network for efficient lane detection

High-Order Deep Infomax Guided Deformable Transformer Network for Efficient Lane Detection

Rong Gao[1,2], Siqi Hu[1], Lingyu Yan[1], Li Zhang[3*], Hang Ruan[4], Yonghong Yu[5] and Zhiwei Ye[1]

[1]School of Computer Science, Hubei University of Technology, Wuhan, 430068, China.
[2]State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing University, Nanjing, 210023, China.
[3]Department of Computer Science, Royal Holloway, University of London, Surrey, TW20 0EX, UK.
[4]School of Mathematics, University of Edinburgh, Edinburgh, EH9 3FD, UK.
[5]College of Tongda, Nanjing University of Posts and Telecommunications, Nanjing, China.

*Corresponding author(s). E-mail(s): li.zhang@rhul.ac.uk;
Contributing authors: gaorong@hbut.edu.cn; husiqi3112@163.com; yanlingyu@hbut.edu.cn; hruan@ed.ac.uk; yuyh@njupt.edu.cn; hgcsyzw@mail.hbut.edu.cn;

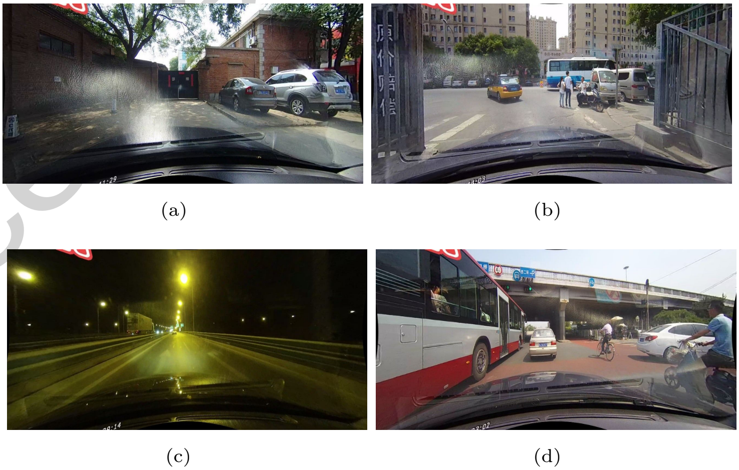
Abstract

With the development of deep learning, lane detection models based on deep convolutional neural networks have been widely used in autonomous driving systems and advanced driver assistance systems. However, in case of harsh and complex environment, the performances of detection models degrade greatly due to the difficulty of merging long-range lane points with global context and exclusion of important higher-order information. To address these issues, we propose a new learning model to better capture lane features, called Deformable Transformer with High-Order Deep Infomax model (DTHDI). Specifically, we propose a deformable transformer neural network model based on segmentation techniques for high accuracy detection, in which local and global contextual information is seamlessly fused and more information about the diversity of lane line shape features is retained, resulting in extraction of rich lane features. Meanwhile, we introduce a mutual information maximization approach for mining higher-order correlations among global shape, local shape, and lane position of lane lines to learn more discriminative representations of lane lines. In addition, we employ a row classification approach to further reduce the computational complexity for robust lane line detection. Our model is evaluated on two popular lane detection datasets. The empirical results show that the proposed DTHDI model outperforms the state-of-the-art methods.

Keywords: lane detection; semantic segmentation; multi-scale; spatial pyramid; Mutual Information

1 Introduction

In the field of computer vision, a lane detection task is a designation that the location of the lane lines presents in an image. For autonomous driving systems and advanced assisted driving systems, the algorithms of lane detection are often used to locate lane lines and ensure stable vehicle movement within the driving area. Therefore, the lane detection task is important for autonomous driving and advanced assisted driving systems. In addition, to avoid various emergencies on the driving road, autonomous driving systems and advanced assisted driving systems require high demands on the real-time and accuracy of lane line detection algorithms.


(a)

(b)

(c)

(d)

Fig. 1 Extreme circumstances

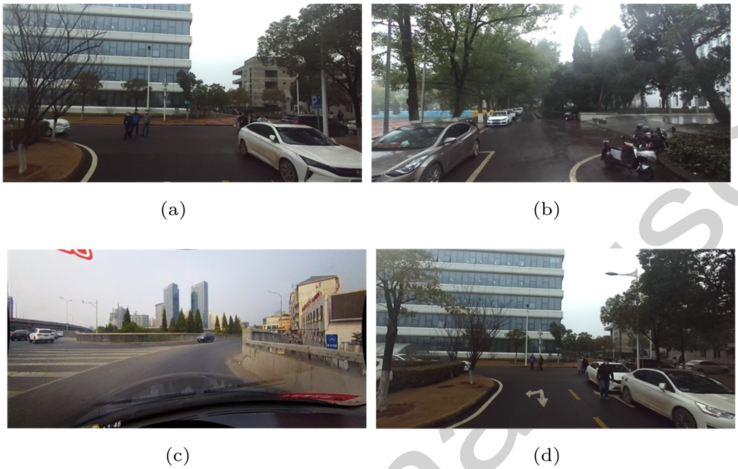
Traditional lane detection methods [1–3] rely on a combination of highly specialized feature selection and heuristics. When the road scene changes, such methods are limited by simple and low-level features and mostly suffer from poor robustness. Moreover, when encountering light changes, external occlusion, or broken lane lines, the problem of false recognition or failure in recognition easily occurs. As shown in Figure 1, large obstacles, extreme lighting conditions, aging roads, and other extreme environments, the lane lines are blurring even non-existent, which limits the performance of lane detection. With the widespread of artificial intelligence [4, 5], many pioneers have started to explore deep learning techniques for lane detection. Many lane detection-methods[6–8] utilized deep learning techniques and thus go beyond traditional methods [9, 10]. These methods treat lane detection as a semantic segmentation problem where the lane line is detected by pixel decoding based on CNN neural networks. The above segmentation-based methods provide a feasible solution to the lane line detection problem. They have been proven to have higher detection accuracy and stronger anti-interference capability than conventional methods.

The above research works have made some progress. Nevertheless, it still has several issues as follows.

1. Bad and complex road scenes and other conditions, different lighting, weather conditions, and occlusion phenomena caused by objects around the road (e.g., pedestrians, vehicles) further require detection models with stronger global context-awareness to improve detection accuracy. While the lane line shape is long and thin, the CNN-based lane line detection model [11–13] is limited by the fixed perceptual field in its network, which cannot combine the local information (e.g., long-range points, etc.) in lane line detection with the global context information caused by surrounding objects (e.g., pedestrians, vehicles, etc.). On the other hand, in recent years, several researchers have introduced attention mechanisms in lane detection models [11, 14, 15] to capture relevant remote information and contextual information. However, the use of a fixed attention model in these models cannot adapt to the complex shape features of lane lines, which leads to poor detection performance.

2. As shown in Figure 1, objects on the road and lane lines have different semantics. In lane line detection, it is difficult to distinguish them in the absence of high-level semantics and description of global contextual information. Meanwhile, local information is important for lane detection due to the thin and long shape of lane lines. Some research works model local geometric features of lanes and integrate them into global results [16], while others favor the construction of a fully connected layer with global features to predict lanes [7]. These works are demonstrating the importance of local or global information for lane detection. However, few works are exploiting both features. On the other hand, since key submodules of autonomous driving systems (e.g., human-vehicle detection) usually coexist with lane detection, which indicates some implicit association among objects on the road (e.g., pedestrians,

vehicles), surrounding buildings, and in the lane, i.e., there is some interrelationship between local and global information [17, 18]. Therefore, it is essential for in-depth research on how to model higher-order correlations between local and global contextual information with joint use of local and global contextual information for more powerful learning.


(a)

(b)

(c)

(d)

Fig. 2 Harsh and complex lane line scenarios

Inspired by recent advances in transformer encoder-decoder architectures for various vision tasks[11, 15, 19], we propose a novel end-to-end lane detection model called Deformable Transformer with High-Order Deep Infomax model DTHDI to tackle the above problems.

Specifically, First of all, we obtain an accurate lane line representation by designing a pyramid transformer model with multi-stage extraction of rich features of lane lines and outputting a multi-scale feature map. Then, we propose a row-wise deformable transformer module based on a two-branch strategy to extract lane line features. In the segmentation branch, we propose a deformable transformer encoder-decoder structure. The global information of the multi-scale features is encoded in the proposed encoder-decoder structure, and the adaptive fusion of the multi-scale features with the global information is achieved based on the deformable transformer. Subsequently, these output features are up-sampled layer by layer to recover to the original image size thereby achieving segmentation. Moreover, in the classification branch, the accuracy of the detection model is further improved by introducing an enhanced row classification technique without affecting the real-time efficiency. In addition, inspired by the idea of self-supervised learning, we design a lane feature enhancement aggregator. This aggregator achieves multi-granularity high-order mutual information maximization of global shape, local shape, and lane position based on the mutual information maximization theory, which mines the correlation between local and global information and improves the detection model's ability to capture features. With the proposed DTHDI, our model outperforms the state-of-the-art methods on benchmark datasets. With the proposed DTHDI, our algorithm outperforms the state-of-the-art methods on the benchmark dataset.

The contributions of this work are summarized as follows:

1) We propose a fast and accurate transformer-based lane detection model to achieve high accuracy in lane detection while ensuring runtime efficiency. It captures the shape features of lanes well and effectively incorporates more global contextual information. High real-time efficiency is maintained using a line-by-line classification technique proposed in this paper with guaranteed high accuracy.

2) We develop an efficient lane feature enhancement aggregator, which extracts as many local lane features as possible with strong discriminative power by maximizing the higher-order mutual information among the global shape, local shape, and lane line position of the lane lines.

3) The DTHDI model achieves state-of-the-art performance for the Tusimple and CULane datasets and outstanding performance gains under blurred scenes.

The rest of the paper is organized as follows. In Section 2, an overview of existing studies on lane detection is discussed. In Section 3, the proposed DTHDI model will be presented in detail. Section 4 presents, the experimental studies, and result analysis to indicate the efficiency of the proposed model. Finally, we summarize the finding of this research and identify future work in Section 5.

2 Related Work

In this section, we briefly review three lines of research that related to our work: Lane Detection, Transformer, and Mutual Information.

2.1 Lane Detection

Early lane detection algorithms used traditional methods [20–22], including key steps such as image preprocessing and feature extraction. Such algorithms require manual adjustment of the operator and its related parameters according to the characteristics of different scenes, with a high workload and poor robustness. Currently, deep learning-based methods have become the current mainstream due to their remarkable performance, but there are still many challenges.

Earlier deep learning-based methods detected lane lines mainly by segmentation [6, 23]. In the segmentation-based methods[24–27] lane lines were obtained by classifying each pixel as a lane or background and then fitting them pixel by pixel. Since the segmentation-based approach performs well in terms of detection accuracy, it has been adopted by many existing studies. A special slice-by-slice convolution approach was recommended in SCNN[6],which generalized the traditional depth-by-layer convolution to slice-by-slice convolution in the feature map. CurveLane-NAS[28],on the other hand, aimed to capture lane-sensitive structures for both long- and close-range curve information by a search framework. EDA-FSS[25]consumed feature size selection to extract detailed features. Meanwhile, a series of dilation convolutions with decreasing dilation rates were used to obtain fine-grained spatial information for multi-lane segmentation. ESA[29] aimed to distill important global spatial information by predicting the occlusion position in an image.

However, these segmentation-based methods are a bottleneck in terms of processing time. In order to further improve the recognition for semantic segmentation of images as well as the computational speed. Several studies [7, 30, 31] converted lane detection into a row classification problem by dividing the image into a defined number of rows and cells per row and predicting which cell contains the lane with the highest probability. E2E-LMD [31] converted the output of a segmented backbone network into a row-by-row representation, thus achieving a reduced amount of computer. UFast [7] used a row-by-row classification-based network under global and structural information to solve the problem of no-lane lines in lane detection. These methods are efficient and fast, but lose accuracy to some extent. The method proposed in this paper strikes a balance between efficiency and accuracy and ensures a high detection accuracy with a small computational effort.

2.2 Transformer

The Transformer structure was first designed by Vaswani et al. [32] in 2017 and has subsequently been widely used in different fields. The transformer shows amazing potential for processing vision tasks that require capturing global relationships. (e.g., image classification [33]). In contrast to the simple application of the Transformer in image classification tasks, Wang et al.[34] introduced the Pyramid Vision Transformer structure, which aims to train densely distributed regions of image features for outputting high-resolution features. LSTR [35] applies the Transformer to the lane detection task, where it expands the extracted features into a one-dimensional vector that is subsequently used as input to the Transformer Encoder. In[35], an end-to-end approach using the Transformer allows learning direct parameters to describe the shape model. Lee et al. [11] improve the final lane detection performance by enhancing the attention to partial lane lines with a self-attentive module. Liu et al. [15], on the other hand, propose a variant of the transformer model that can use a self-attention mechanism to capture the slender structure and the global context. All the above transformer-based approaches have achieved good performance in the lane line. However, the commonly adopted fixed attention routines models are unable to adaptively fit complex lane line features, especially when the lanes change.

2.3 Mutual Information in Supervised Learning

Self-supervised learning is one of the key directions of recent research in the CV field. As one of the theoretical foundations for self-supervised learning, the Informax principle proposes that the Informax principle proposes that maximizing the mutual information of input and output can be used to learn better generative models. Hjelm et al. [36] demonstrated that integrating knowledge about the input location into the target can greatly affect the applicability of the representation to downstream tasks, thus proposing the Deep Infomax model. Chen et al. [37] propose a new sampling algorithm based on the Deep Infomax estimation and maximization algorithm in the field of person re-localization research. Ji et al. [36] argue that unsupervised clustering and segmentation are done by maximizing the transformation or spatial proximity of image associations between mutual information. Mukherjee et al. [38] designed a conditional mutual information neural estimator for classifiers. Bachman et al. [39] developed a self-supervised representation learning method based on the maximization idea of mutual information between features extracted from multiple views of a shared context.

3 Method

In this section, the structure of the proposed model is described in detail.

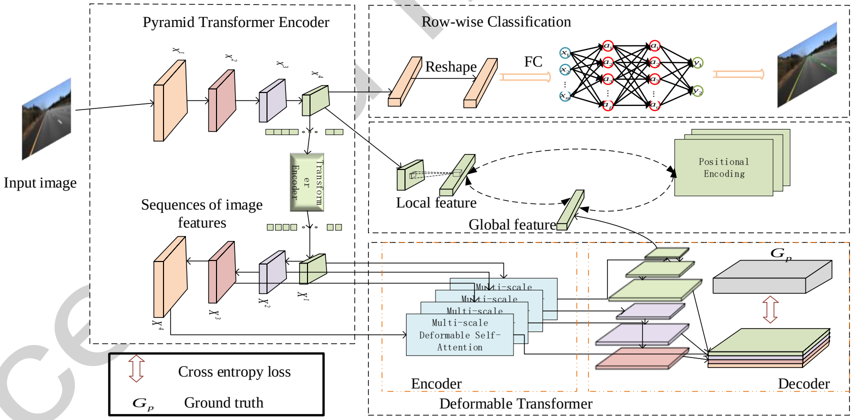


Fig. 3 The overall framework of the model.

DTHDI is a single-stage anchor-based model for lane detection. An overview of our method is shown in Fig.3. First, to obtain a robust feature representation, we use the pyramid transformer encoder to extract features from the input image, which outputs a multi-scale feature map based on a feature pyramid to obtain a richer representation. Then, the multi-scale feature map is fed into two separate branches, the segmentation branch, and the classification branch. In the segmentation branch, we segment the input image using the end-to-end structure of the Deformable Transformer, which merges the multi-scale rich global contextual features and outputs the predicted lane masks. In the classification branch, the obtained multiscale features are fully connected for classification based on an improved line-by-line classification scheme to obtain the probability distribution of line anchors, which satisfies the lane detection real-time requirement. Subsequently, we perform multi-granularity high-order mutual information optimization on the features of the two branches to maximize the global, local, and lane location mutual information, thus characterizing the lanes more accurately. Finally, the lane line detection results are obtained by training the fusion loss of the two branches as well as the higher-order information maximization loss.

3.1 Pyramid Transformer Encoder

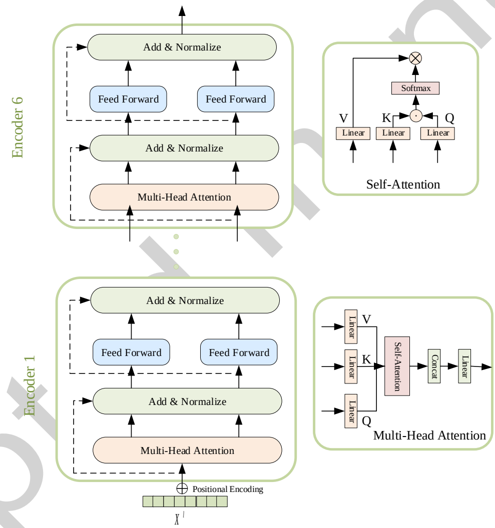


Fig. 4 The center suppressed cropping Module

In the module, we design a pyramid transformer encoder model to advance accurate features. First of all, we adopt Resnet as the basis for feature extraction based on the Encoder structure, then hybridize the FPN to acquire multi-scale features. Transformer Encoder is added to the last layer of Resnet to extract richer multi-scale features from complex scene images. The structure of the pyramid transformer encoder is shown in Fig.4 below.

The image features are obtained by applying the multi-layer convolution of the Resnet for the input image size $I \in R^{C\times H\times W}$. The feature map x^l of the final layer is subjected to several up-sampling procedures to yield $\left\{X^l\right\}_{i-1}^{L}$ for the different dimensional features $\left\{x^{l}\right\}_{i-1}^{L}$ acquired by convolution. Finally, it is linked one by one to the feature maps created using Resnet convolution. Meanwhile, the feature matrix of the last layer of Resnet is expanded into a one-dimensional feature vector, and then the position encoding E_q is added as the Transformer Encoder's input X^l:

$$X^{i}=\left\{x^{l}\right\}+E_{q} \tag{1}$$

X^l is entered into Encoder:

$$\operatorname{Attention}(Q, K, V)=\operatorname{softmax}\left(\frac{Q K^{T}}{\sqrt{d_{k}}}\right) V \tag{2}$$

where $Q=\operatorname{Linear}\left(X^{l}\right)=X^{l} W_{Q}, K=\operatorname{Linear}\left(X^{l}\right)=X^{l} W_{k}, V=\operatorname{Linear}\left(X^{l}\right)=X^{l} W_{V}$. The final output is a multi-scale feature map $\left\{X^{l}\right\}_{i-1}^{L}$, and $X^{l} \in R^{C \times H \times W}$.

3.2 Deformable Transformer

This section inputs the multi-scale feature from Section 3.1 into the segmentation branch, uses the Deformable Transformer to extract the contextual information, and produces the lane mask. Deformable Transformer converts the original input structure in Transformer into the input for multi-scale features to preserve the original information of multi-scale features. Deformable Transformer, on the other hand, only concentrates on a limited number of key sample points surrounding the current query, and by assigning a small number of fixed keys to each query, it avoids the long training time problem that exists in classical Transformer.

Specifically, a Deformable Transformer is made up of two components: Deformable Transformer Encoder and Decoder.

Deformable Transformer Encoder. Both the input and output of the encoder are multi-scale feature maps with the same resolution. The classical Transformer Encoder structure is shown in Fig.4. To join the location encoding as Encoder input, multi-scale features *th* are first joined by upsampling fusion and then expanded into feature vectors. The multi-head attention module adaptively acquires the key content based on modifying the key-query pair weights are given a set of key elements and a query element, where both key elements and query elements are pixels in the multi-scale feature map. It has the following formula:

$$\text{MultiHeadAttn }\left(T_{q}, X\right)=\sum_{m=1}^{M} W_{m}\left[\sum_{k \in \Omega_{k}} A_{m q k} \cdot W_{m}^{\prime} X_{k}\right] \tag{3}$$

where T_q is the target feature of the query; X is the input feature vector; m and M are the attention head index and the total number, respectively; Ω_k

refers to the Key element; and W_m and W'_m are the learnable weights; and q and k are the indexes of the Query element and the Key element, respectively. The attention weights $A_{mgk} \propto \exp\left\{\frac{T_q^T U_m^T V_m X_k}{\sqrt{C/M}}\right\}$ are normalized to $\sum_{k\in\Omega_k} A_{mqk} = 1$, where C is the feature dimension and, U_m and V_m are the learnable weights. The Deformable Transformer Encoder eliminates the fusion coupling procedure. It transforms the previously described multi-head attention into multi-scale deformable attention that adjusts to multi-scale feature input, improving the original information reading of various scale maps. At the same time, it differs from the multi-head attention in Equation (3). Each pixel point is treated as a target feature, which is compared to other pixel points (sample points) in the image. Only a tiny percentage of critical characteristics (reference points) around the target features are focused by multi-scale deformable attention. The following is the updated multi-scale deformable attention formula.

$$\text{MSDeformAttn}\left(T_q, \hat{p}_q, \left\{X^l\right\}_{l-1}^{L}\right) = \sum_{m=1}^{M} W_m \left[\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk} \cdot W'_m x^l\left(\phi_l\left(\hat{p}_q\right) + \Delta p_{mlqk}\right)\right] \tag{4}$$

where $\left\{X^l\right\}_{l=1}^{L}$ denotes the multi-scale feature map of the input and $X^l \in R^{C\times H\times W}$. $\hat{p}_q$ are the normalized coordinates of the reference points of each query element, where $\hat{p}_q \in [0,1]^2$, m denotes the number of attention heads, l denotes the input feature layer, and k denotes the sample points. And A_{nigh} denote the sample offset and attention weight for the $k\ th$ sample point in the $l\ th$ feature layer and the $m\ th$ attention head, respectively.

For each query pixel, the reference point is itself. A scale-level embedding, indicated as e_l , is added to the deformable attention in addition to the location embedding. Unlike the location embedding, which is placed according to the different scale sizes of the feature maps, the numerous scale-level embeddings $\{e_l\}_{l=1}^{L}$ are randomly initialized and participate in network training as parameters to achieve optimality in the network training.

Decoder. By combining the multi-scale feature maps extracted based on progressive upsampling with the encoder component, we obtain the final feature representation and output the lane line example maps.

We investigate the lane mask output by the decoder to be close to the ground-truth masks of binary lane mask, assuming that the lane segmentation branch is denoted as $P_p = \Phi p(\pi_p)$, with π_p as the parameter. Then its loss function is:

$$L_{seg} = CE(P_p, G_p) \tag{5}$$

where $P_p = \Phi p(\pi_p)$ is the output of the split branch and G_p is the ground-truth masks of binary lane mask.

3.3 Row-wise Classification

In this section, we propose a line-by-line classification technique to further reduce the computational effort of our method while detecting lane lines. As shown in Fig.5. the area of the image containing the lanes is divided into a predetermined number of row anchors (h) . Each row anchor is divided into a predefined number of grid cells (w). The number of lanes C is also predetermined. The grid of $h * w$ is used to denote the lane location for each lane. Therefore, we classify each divided grid for the input image attributes and then output the probability distribution of the presence of lanes in the grid. It is worth noting that after each row anchor in the row-by-row classification method proposed in this paper, a grid is added to display the existence of vehicle lines in the row anchor. Thus, the total classification computation is $h * (w+1)$. We take multi-scale feature maps X^l as input to categorize each grid, and the following equation is exploited to detect lane lines row by row:

$$P_{i,j:} = f^{ij}\left(X^1\right), \ s.t. \ i \in [1, C], j \in [1, h] \tag{6}$$

where C is the number of lanes. h is the pre-defined row anchor; and, f^{ij} denotes the classifier that selects the lane position on the i *th* lane and j *th* row anchor. In this paper, full connectivity is utilized as the classifier. $P_{i,j:}$ is a $(w+1)$-dimensional vector representing the selection of the *th* lane, as well as the anchor on the *th* row. The lane points are extracted by selecting the grid cell for each line with the highest probability of the presence of lanes in each row of anchors. The loss function is as follows:

$$L_{ck} = \sum_{i=1}^{c} \sum_{j=1}^{h} L_{CE}\left(P_{i,j:}, T_{i,j:}\right) \tag{7}$$

where L_{CE} is the cross-entropy loss and $T_{i,j:}$ is the ground truth after one-hot encoding.

3.4 High-order Deep Infomax

Lane line detection requires strong perception to locate lane locations, however, local features cannot effectively perceive the global structure of the image. To enhance the lane line information on the roadway, we maximize the global and local features of the lane lines while adding the location encoding, which thus deeply exploits the higher-order mutual information to mine the correlation between global, local, and lane locations for the purpose. In addition, the location encoding consists of randomly generated marker information for the features during feature extraction.

Inspired by the literature [40, 41], we extend its idea to three variables. For the number of variables $N \geq 3$, given a set of random variables, $X_1, \cdots, X_N$

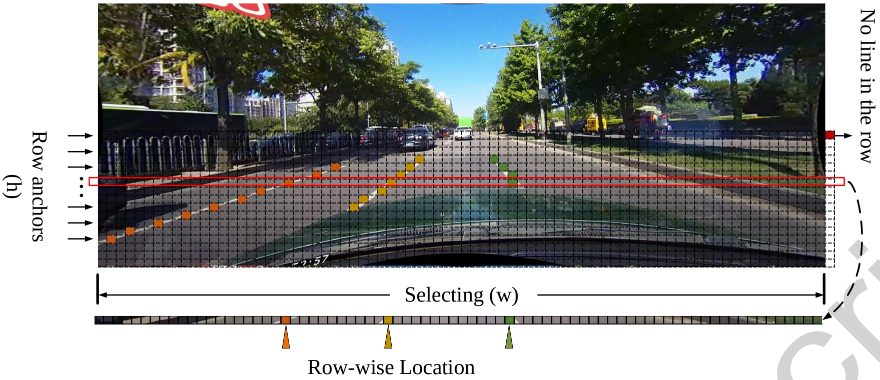


Fig. 5 Schematic diagram of dividing the grid in row classification

the higher order mutual information is defined as follows:

$$I\left(X_1, \cdots, X_N\right)=\sum_{n=1}^{N}(-1)^{n+1} \sum_{i,<\cdots,d_n} H\left(Xi_i, \cdots, X_{i_n}\right) \tag{8}$$

where $H\left(Xi_i, \cdots, X_{i_n}\right)$ is the cross-entropy of $Xi_i, \cdots, X_{i_n}$. The three random variables of local features L, global features G, and position encoding P are described in this study. Their higher order mutual information equations are as follows:

$$\begin{aligned} & I(L;\ G;\ P)=H(L)+H(G)-H(L,G) \\ & +H(L)+H(P)-H(L,P) \\ & -H(L)-H(G,P)+H(L,G,P) \\ & =I(L,G)+I(L,P)-I(L,G,P) \end{aligned} \tag{9}$$

Where $I(L;\ G;\ P)$ denotes the mutual information between L with the joint distribution of G and P . Specifically, $I(L,G)$ denotes the correlation between global and local information, $I(L,P)$ denotes the correlation between local information and location coding, and then $I(L,G,P)$ denotes the correlation between the three random variables of global, local, and location coding. Maximize the mutual information between the three with the following equation:

$$\begin{aligned} & maxI(L;\ G;\ P) \\ & =max(I(L,G)+I(L,P)-I(L,G,P)) \\ & =max(I(L,G))+max(I(L,P))-min(I(L,G,P)) \\ & =max(I(L,G))+max(I(L,P))+max(I(L,G,P)) \end{aligned} \tag{10}$$

In this paper, different coefficients are used for different mutual information. The final objective function is.

$$L_{mutual} = \lambda_g I(L, G) + \lambda_p I(L, P) + \lambda_a I(L, G, P) \tag{11}$$

where λ_g , λ_p and λ_a are variable parameters.

The maximization of global and local mutual information can be achieved by maximizing the following objective function.

$$L_a = E\left[\log D_l(L, G)\right] + E\left[\log\left(1 - D_l(\tilde{L}, G)\right)\right] \tag{12}$$

where L and $\tilde{L}$ denote the positive samples with local information and the negative samples with local information transformed. In this paper, we utilize the features after random permutation as negative samples. Similarly, for the three variables proposed in this paper, we construct negative samples of the location information to capture the correlation between the three, by considering that the information between global and local, and global and location has been calculated previously. The objective function equation is as follows:

$$L_a = E\left[\log D_a(L, G, P)\right] + E\left[\log\left(1 - D_a(L, G, \tilde{P})\right)\right] \tag{13}$$

where D_a denotes the discriminator of mutual information among the three. Therefore, The final maximization equation thus becomes the following:

$$L_{mutal} = \lambda'_g L_l + \lambda'_p L_p + \lambda'_a L_a \tag{14}$$

where λ'_g , λ'_p and λ'_a are variable parameters.

For discriminators D_l and D_p , we use a convolutional layer. Moreover, for discriminator C , we use a nonlinear function. The equation of the nonlinear function is as follows:

$$ZP = \sigma\left(W_p P\right) \tag{15}$$

$$ZG = \sigma\left(W_L G\right) \tag{16}$$

$$Z = \sigma\left(W_Z[ZP;\ ZG]\right) \tag{17}$$

$$D_a = \sigma\left(L^T M_a Z\right) \tag{18}$$

where W_p , W_L, W_z and M_a are parameters, σ is the sigmoid activation function, and [;] is the connection operation.

3.5 Loss function

Our overall loss of the model may be separated into three categories based on the information above: segmentation loss, classification loss, and higher order mutual information maximization loss. As a result, the total mode's loss function is as follows:

$$L_{total} = \alpha L_{seg} + \beta L_{cls} + \lambda_g L_l + \lambda_p L_p + \lambda_a L_a \tag{19}$$

The segmentation and classification losses are denoted by L_{seg} and L_{cls} , respectively. The global and local, local and line location, and mutual information maximization loss among the three are denoted by L_l , L_p and L_a , respectively.

In this research, the segmentation branch and higher-order mutual information maximization are only involved in the training stage to increase real-time performance during model inference. As a result, even though the divided half of the model in this research contains more parameters, it does not affect on the entire model.

The parameters of the model are optimized by minimizing the error between the model prediction and the ground truth. Since the output of the network is a binary value (1 for foreground and 0 for background), the loss employs the Softmax Cross-Entropy Loss loss function, abbreviated as:

$$L = -\frac{1}{MN}\sum_{M}\sum_{N} G_{MN} \log\left(\frac{e^{Y_{MN}}}{\sum e^{Y_{MN}}}\right) \tag{20}$$

where G_{MN} is ground truth; Y_{MN} is the output; M and N are the output size, which is the same as the input image size and will vary with the input size.

Batch normalization is applied in the encoder and decoder for each convolutional layer to speed up model training. The activation function is Rectified Linear Units (ReLU). The model is trained and tested via PyTorch. The SGD network performs the training for the input training samples to maximize the updating of the network parameters.

4 Experiment

4.1 Experimental setting

4.1.1 Datasets

We evaluate the proposed model on two publicly accessible datasets, i.e. CULane[6] and TuSimple[42]. Culane is a frequently used large lane detection dataset that retrieved 133235 frames from over 55 hours of video. Normal, Crowded, Dazzle, Shadow, No line, Arrow, Curve, Cross, and Night are among the nine various settings available for model training. Complex scenarios including Crowded, Shadow, and Curve give the foundation for customizing the model to various training conditions. Another extensively utilized dataset for autonomous driving scenarios is TuSimple. The TuSimple dataset, released by autonomous driving company Tucson, is the first dataset to provide a benchmark for lane line detection. It consists of 3626 training images and 2782 test images for straight, curved, well-lit, damaged, disturbed, and shadow-obscured roads, including road images taken at different times of the day.

Details of these two datasets are shown in Table 1. In addition, in order to further verify the performance of the DTHDI model, we also use real scene data sets to evaluate the effectiveness of the proposed method. The dataset

Table 1 Details of TuSimple and Culane datasets

Dataset	Train	Val	Test	Road type
Culane	88.9K	9.7K	34.7K	Urban&Highway
Tusimple	3.3K	0.4K	2.8K	Highway

used is the urban road scene collected in Changchun of China. This data set has 1500 images taken during the vehicle traveling. The image test effect is shown in Figure 6.

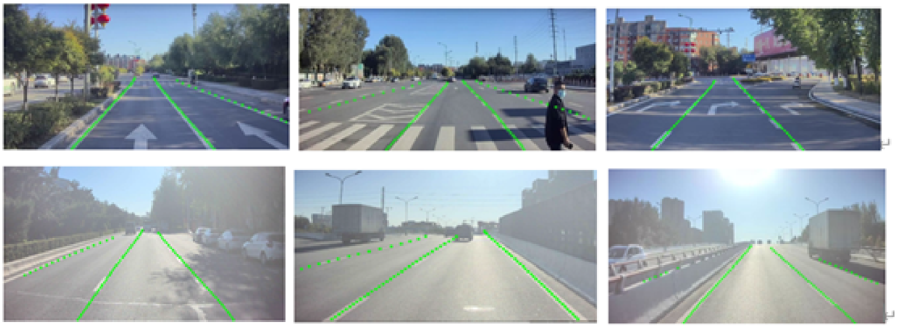


Fig. 6 Detection results of the DTHDI model on real datasets.

4.1.2 Evaluation metrics

The official assessment measures for the two datasets differ, with Accuracy and F1 being used in each case. The major assessment statistic for the TuSimple dataset is Accuracy. According to[43], the main evaluation statistics of the TuSimple dataset are accuracy. The Tusimple disclosed the following accuracy formula when specifying the dataset. Therefore, we still choose to use this precision formula when using the Tusimple dataset, to achieve more reliable results compared with other methods under the same standard. For the TuSimple dataset, there are three official indicators: false-positive rate (FPR), false-negative rate (FNR), and accuracy. The following is the formula for calculating it:

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \tag{21}$$

where C_{clip} is the number of correctly predicted lane points, and S_{clip} is the total number of ground truth in each segment. Lane with accuracy greater than 85% is considered as a true-positive otherwise false positive or false negative. Besides, the F1 score is also reported.

Each lane is evaluated as a 30-pixel wide line in the CULane assessment measure. It is computed the intersection over union (IoU) between the ground truth and the anticipated outcomes. True positives are defined as forecasts with an IoU greater than 0.5. With the following equation, the F1 readings can be used as assessment indicators:

$$Fl - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{22}$$

Precision + Recall where $Precision = \frac{TP}{TP+FP}$; $Recall = \frac{TP}{TP+FN}$; TP is a true positive; FP is a false positive and FN is a false negative.

4.1.3 Implementation details

For model training, both input images are transformed to an image size of 800*280, and the conversion is repeated at the image output to return to the original size. This work designs the Adam optimizer with step learning rate decay and an initial learning rate of 1e-4 in the optimization process. The experiments are trained on CULane and TuSimple with 50 and 100 epochs, respectively. The batch size is 16. The CULane and TuSimple datasets have 200 and 100 grid divisions, respectively. The global and local, local and position encoding, and the initialization weight parameters among the three are 1, 1 and 0.001. The number of lane lines of the model is set to 4. All experiments are run on a PC with an Nvidia Tesla V100 graphics card.

4.2 Overall Performance Comparison

we utilize Resnet18 and Resnet34 as the base networks for the feature extraction section to illustrate the efficiency of the DTHDI. The authors' source code and models with default configurations are directly available in this work to maintain the fairness of the comparative tests. The authors' significance detection results are directly obtained using various data sets. TuSimple: We compare our method based on the TuSimple dataset to eight state-of-the-art lane detection methods that have become popular in recent years, including Res18-Seg [19], Res34-Seg [19], SCNN [6], FastDraw [44], SAD [45], Resa[8],

Table 2 Comparison with other methods on TuSimple dataset

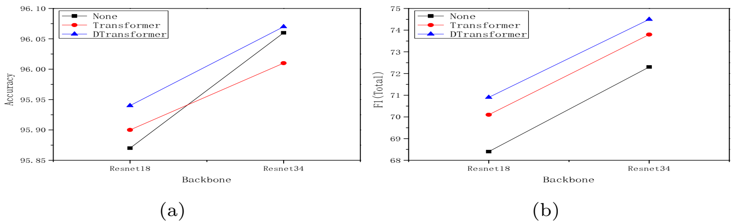
Method	Accuracy	FPS	FP	FN
Res18-Seg	92.69	39.52	0.0948	0.0822
Res34-Seg	92.84	19.80	0.0918	0.0796
SCNN	96.53	7.49	0.0617	0.0180
FastDraw	95.20	90	0.0760	0.0450
SAD	96.64	294.11	0.0602	0.0205
Resa	96.82	36	0.0363	0.0248
Res18-UFast	95.87	312.5	0.1905	0.0391
Res34-UFast	96.06	169.49	0.1906	0.0392
Res18-Ours	96.08	322	0.0589	0.0599
Res34-Ours	96.77	165	0.0297	0.0385

Table 3 Comparison with other methods on TuSimple dataset

Category	Seg	SCNN	FastDraw	SAD	Res18-UFast	Res34-UFast	SwiftLane	RESA-34	Res18-DTHDI	Res34-DTHDI
Normal	87.4	90.6	85.9	90.1	87.7	90.7	90.46	91.9	90.6	92
Crowded	64.1	69.7	63.6	68.8	66	70.2	71.07	72.4	68.9	73.3
Night	60.6	66.1	57.8	66	62.1	66.7	68.77	69.9	65.6	70.1
No-line	38.1	43.4	40.6	41.6	40.2	44.4	46.17	47.7	42.3	48.1
Shadow	60.7	66.9	59.9	65.9	62.8	69.3	73.69	72	65.4	75.2
Arrow	79	84.1	79.4	84	81	85.7	85	88.1	85.2	88
Dazzlelight	54.1	58.5	57	60.2	58.4	59.5	62.51	66.5	59.9	63.4
Curve	59.8	64.4	65.2	65.7	57.9	69.5	64.92	68.6	61.2	69.8
Crossroad	2505	1990	7013	1998	1743	2037	1096	1896	1678	2003
Total	66.7	71.6	-	70.8	68.4	72.3	74.03	74.2	70.9	74.5
FPS	-	7.49	90.3	74.62	322.58	175.43	411	36	301	157
GELOPS	-	328.4	-	-	16.56	16.56	-	-	17.92	17.92

Res18-UFast [30], and Res34-UFast [30]. We use ResNet-18/34 as the backbone and they are labeled as RES-18/34. The results are shown in Table 2. RES34-Ours achieves an accuracy of 96.77, which is only slightly lower than Resa, thus ranking 2nd. We also analyze the FP and FN of each method. Lower FP and FN mean fewer false predictions, and our method ranks first in the FP metric. This is due to the fact that the TuSimple dataset is a highway dataset with more homogenous scenes, in which the lane lines have more regular forms. It shows the efficiency of Deformable Transformer in this paper compared with the above methods. Moreover, this paper draws on the idea of mutual information maximization for the related purpose of further aggregation based on various features. In addition, our method has faster FPS than almost all results on TuSimple dataset, which illustrates the excellence of the row-based classification strategy proposed in this paper, enabling our method to achieve real-time efficiency and guarantee high efficiency while ensuring high accuracy. Meanwhile, we take the idea of mutual information maximization to achieve further aggregation based on various relevant features. In addition, our method has faster FPS than all baselines on TuSimple dataset, which illustrates the excellence of the row-based classification strategy proposed in this paper, enabling our method to attain real-time efficiency and guarantee high efficiency while ensuring high accuracy.

Culane: We selected eight state-of-the-art lane detection methods for comparison with our approach to the Culane dataset. They are Seg [19], SCNN [6], FastDraw [45], SAD [8], Res18-UFast [30], Res34-UFast [30], SwiftLane [31], and Resa [44]. As shown in Table 3, our method achieves 301 FPS, which still guarantees an excellent balance between high efficiency and high real-time performance. DTHDI performs better in Culane compared to the TuSimple dataset and outperforms all baselines. The reason for this is that the Culane dataset is larger and contains more images of different scenes, which can provide better conditions for model training. The fact that the Culane dataset contains more scenes while DTHDI is constructed to accommodate diverse scene detection makes it more successful on fuzzy feature scenes. In particular, the experimental results on Normal, Night, Wireless, and Shadow scenarios all illustrate the necessity of adding a Deformable Transformer for solving the sparse feature problem in fuzzy scenarios, and the importance of aggregating different features based on the idea of maximizing mutual information.



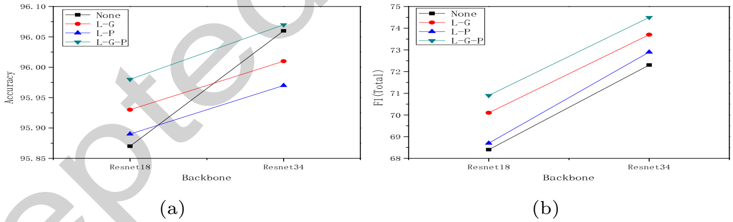
(a) (b)

Fig. 7 Comparison of different transformers based on Tusimple (a) and Culane (b) datasets, respectively

4.3 Ablation Analysis

4.3.1 Comparative Analysis of Transformer Structures

In this section, we investigate the effect of transformers and different transformer variants. None denotes no transformer structure, Transformer denotes regular transformer structure, and Dtransformer denotes deformable transformer structure. Figure 6 depicts the results of the experiment. First, DTHDI outperforms the baseline model on both datasets. This is because the Transformer structure of feature processing focuses more on location information, while the Deformable Transformer structure is used to obtain location information between lane line pixels. Furthermore, the Dtransformer still performs best when using the Culane dataset. This is because the deformable Transformer structure incorporates the idea of adaptivity, thus adaptively learning more multi-scale feature information in complex environments.



(a) (b)

Fig. 8 Comparison of different mutual information based on Tusimple (a) and Culane (b) datasets, respectively

4.3.2 Comparative analysis of mutual information

In this section, we investigate the effect of mutual information. None represents that no mutual information is added. L-G represents that local and global mutual information is added to the model. L-P represents that mutual information between local and positional codes is added to the model. L-G-P represents that higher-order mutual information between local, global, and positional codes is added to the model. We summarize the performance of each module in Table 4. First, the incorporation of mutual information improves the model's resilience and accuracy for lane line detection. Second, in terms of mutual information between variables, the correlation between global and local variables significantly improves the model's performance.

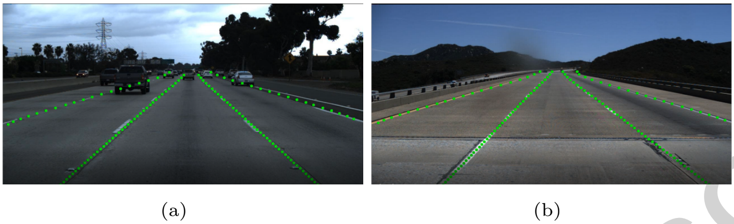
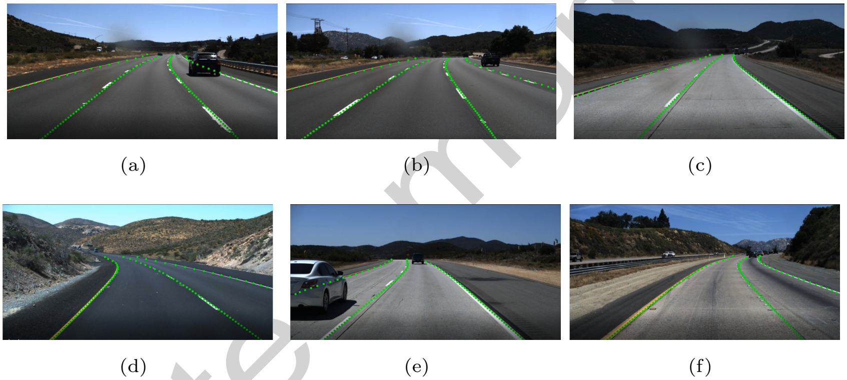

(a) (b)

Fig. 9 Detection results of the DTHDI model on conventional roads.


(a) (b) (c)

(d) (e) (f)

Fig. 10 Detection results of the DTHDI model on conventional roads.

4.4 Visualization of test results

We first make a visualization of the different scenarios as follows. From Fig.8, it can be seen that the proposed DTHDI model detects regular and complete lane lines on conventional roads, which demonstrates the ability of the Deformable Transformer to extract global information of lane lines. Fig.9 illustrates that the model also gets good results for curves, and shows strong adaptability for different angles of curves. It is verified that the proposed higher-order mutual information maximization for the bent part of the local information and the global information is well intermingled. From the detection results of Fig.10, it can be concluded that when the lane lines are obscured by vehicles on the road, the proposed model can still detect the lane lines on the road surface completely and has a good prediction for the shape of the lane lines. In Fig.11, it can be seen that the model can detect the shape of the lane lines on the road surface even when the off light extremely affects the detection of the lines.

At present, the public dataset of lane lines is limited to extreme cases such as low illumination, blocked lanes, and shadow lanes. The public datasets we use, Tusimple and Culane, contain some extreme images. The lane line detection effect is shown in Figure 13. However, on the real dataset we tested, the Changchun, China street dataset, the percentage of extreme scene images exceeded 70%, as shown in Figure 14. The extreme scenes mainly include congestion, shadow occlusion, lane line blurring, and backlighting. Under the dataset with high extreme scene density, our proposed DTHDI model still showed good detection performance, as shown in Fig.14. In our experiment, different datasets are compared with other methods under a unified standard, which ensures the credibility of the data and the authenticity and effectiveness of the experiment.

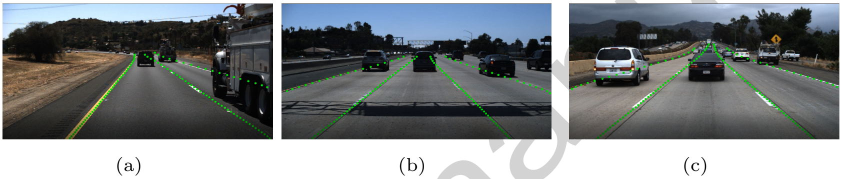

(a) (b) (c)

Fig. 11 Detection results of the DTHDI model when it is obscured.

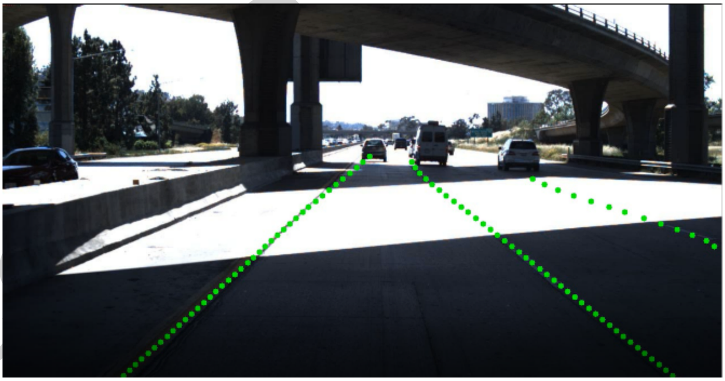


Fig. 12 Detection results of the DTHDI model under extreme lighting.

5 Conclusion

In this paper, we propose a novel end-to-end lane detection method, named DTHDI. Firstly, we utilize the encoder-decoder structure and design a deformation transformer to extract multi-scale features with contextual information

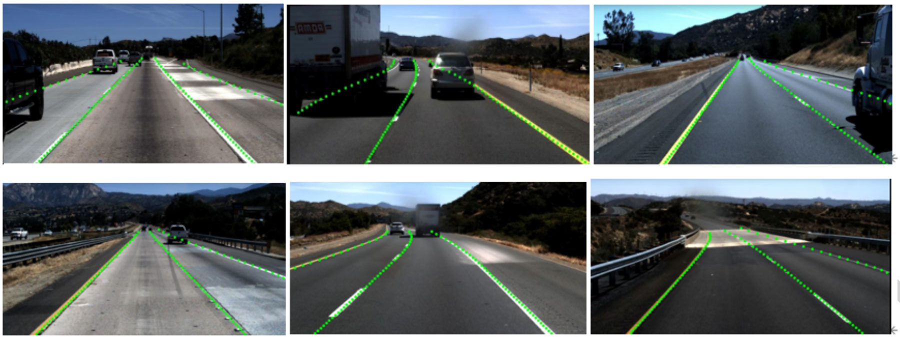


Fig. 13 Public dataset detection image in real world.

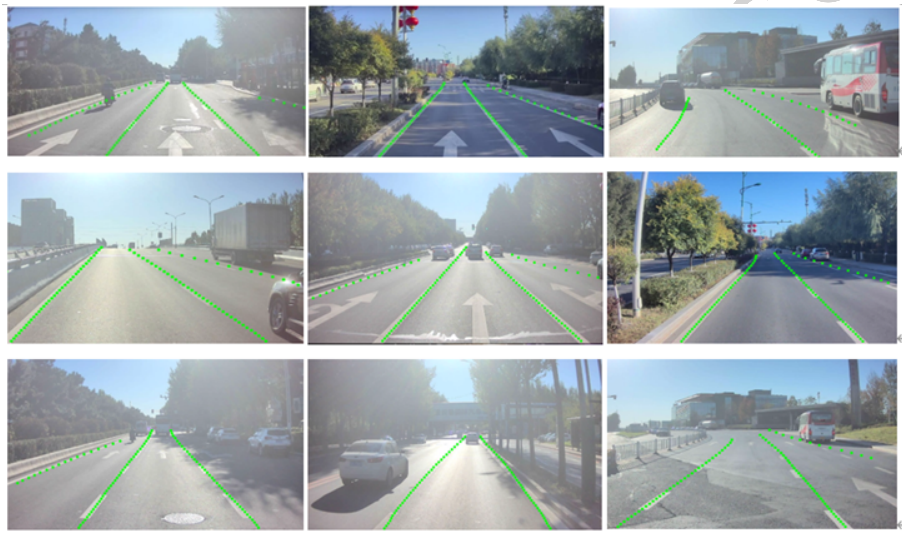


Fig. 14 Real dataset detection image in Changchun of China.

for effectively capturing long-distance visual cues when lane features in blurred scenes are sparse. Secondly, to enhance the discriminability of features, we maximized the higher-order mutual information among the global shape, local shape, and lane line position of the lane lines. In addition, to detect accurately while ensuring real-time efficiency, we build an end-to-end row-wise classification to generate anchor lines over the image with a strong shape prior to deducing lane lines. In this way, DTHDI greatly improves the upper bound of accuracy without speed delay. We also evaluate the generation of our method in various scenarios, the excellent performance demonstrates the powerful detection capability of DTHDI.

6 Declarations

Ethical Approval
Not applicable.
Competing interests
Not applicable.
Funding

This work described in this paper was supported by the Open Foundation of State Key Laboratory for Novel Software Technology at Nanjing University of P. R. China (no. KFKT2021B12). This work is supported in part by the Future Network Scientific Research Fund Project (FNSRFP-2021-YB-54), the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (17KJB520028), Tongda College of Nanjing University of Posts and Telecommunications (XK203XZ21001), Major Science and Technology Project of Jilin Province, China(20210301030GX) and Key Research and Development Program of Hubei Province, China (2021BAA179 and 2022BAA079). The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

Availability of data and materials
All of our datasets come from public datasets.You can go to the corresponding official website to download.

References

[1] V. John, Z. Liu, S. Mita, and et al., Real-time road surface and semantic lane estimation using deep features. Signal, Image and Video Processing, 12, 1133-1140(2018).

[2] A. Borkar, M. Hayes, M. Smith, A novel lane detection system with efficient ground truth generation. IEEE Transactions on Intelligent Transportation Systems, 13, 365-374(2012)

[3] P. Wu, C. Chang, C. Lin, Lane-mark extraction for automobiles under complex conditions. Pattern Recognition, 47, 2756-2767(2014).

[4] A. Hillel, R. Lerner, D. Levi, and et al., Recent progress in road and lane detection: a survey. Machine Vision and Applications, 25, 727–745(2014).

[5] T. Lin, P. Doll, R. Girshick, and et al., Feature pyramid networks for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2117–2125(2017).

[6] X. Pan, J. Shi, P. Luo, and et al., Spatial as deep: Spatial cnn for traffific scene understanding. Proceeding of the 32nd AAAI Conference on Artificial Intelligence, 7276-7283(2018).

[7] Z. Qin, H. Wang, X Li, and et al., Ultra-fast structure aware deep lane detection. Proceedings of European Conference on Computer Vision, 276–291(2020).

[8] T. Zheng, H. Fang, Y. Zhang, and et al., Resa: Recurrent feature-shift aggregator for lane detection. Proceeding of the 35th AAAI Conference on Artificial Intelligence, 3547– 3554(2021).

[9] J. Niu, J. Lu, M. Xu, and et al., Robust lane detection using two-stage feature extraction with curvefitting. Pattern Recognition, 59, 225-233(2016).

[10] S. Narote, P. Bhujbal, A. Narote, and et al., A review of recent advances in lane detection and departure warning system. Pattern Recognition,73, 216–234(2018).

[11] M. Lee, J. Lee, D. Lee, and et al., Robust lane detection via expanded self-attention. (2021), arXiv:2102.07037.

[12] H. Xu, S. Wang, X. Cai, and et al., Curve lane-NAS: Unifying lane-sensitive architecture search and adaptive point blending. (2020), arXiv:2007.12147.

[13] Z. Chen, Q. Liu, and C. Lian, Point LaneNet: Efficient end-to-end CNNs for accurate real-time lane detection. IEEE Intelligent Vehicles Symposium, 2563–2568(2019).

[14] L. Tabelini, R. Berriel, T. Paixao, and et al., Keep your eyes on the lane: Attention-guided lane detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 294-302(2021).

[15] R. Liu, Z. Yuan, T. Liu, and et al., End-to-end lane shape prediction with transformers. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 3694–3702(2021).

[16] Q. Zhan, J. Huan, Z. Yang, and et al., Focus on local: Detecting lane marker from bottom up via key point. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14122– 14130(2021).

[17] Z. Yao, X. Wu, P. Wang, and et al., DevNet: Deviation aware network for lane detection. IEEE Transactions on Intelligent Transportation Systems, 23, (17584 -17593)2022.

[18] J. Wang, Y. Ma, S. Huang, and et al., A key point-based global association network for lane detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 1392-1401(2022).

[19] N. Carion, F. Massa, G. Synnaeve, and et al., End-to-end object detection with transformers. Proceedings of the European Conference on Computer Vision, 213–229(2020).

[20] J. Gonzalez, U. Ozguner, Lane detection using histogram-based segmentation and decision trees. Proceedings of the IEEE Intelligent Transportation Systems, 346–351(2000).

[21] P. Wu, C. Chang, C. Lin, Lane-mark extraction for automobiles under complex conditions. Pattern Recognition, 47, 2756–2767(2014).

[22] J. Hur, S. Kang, S. Seo, Multi-lane detection in urban driving environments using conditional random fields. Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, 1297–1302(2013).

[23] D. Neven, B. Brabandere, S. Georgoulis, and et al., Towards end-to-end lane detection: an instance segmentation approach. In IEEE Intelligent Vehicles Symposium, 286–291(2018).

[24] L. Chen, G. Papandreou, I. Kokkinos, and et al., Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 40, 834–848(2017).

[25] S. Lo, H. Hang, S. Chan, and et al., Multi-Class lane semantic segmentation using efficient convolutional networks. Proceedings of the IEEE 21st International Workshop on Multimedia Signal Processing, 1–6(2019).

[26] J. Zhang, T. Deng, F. Yan, and et al., Lane detection model based on spatiotemporal network with double convolutional gated recurrent units. IEEE Transactions on Intelligent Transportation Systems, 1–13(2021).

[27] J. Su, C. Chen, K. Zhang, and et al., Structure guided lane detection. (2021), arXiv:2105.05403.

[28] H. Xu, S.Wang, X.Cai, and et al., Curve lane-NAs: Unifying lane-sensitive architecture search and adaptive point blending. Proceedings of the European Conference on Computer Vision, 689–704(2020).

[29] M. Lee, J. Lee, D. Lee, and et al., Robust lane detection via expanded self-attention. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 533–542(2022).

[30] O. Jayasinghe, D. Anhettigama, S. Hemachandra, and et al., Swiftlane: Towards fast and efficient lane detection. (2021), arXiv:2110.11779.

[31] S. Yoo, H. Lee, H. Myeong, and et al., End-to-end lane marker detection via row-wise classification. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 1006–1007(2020).

[32] A. Vaswani, N. Shazeer, N. Parmar, and et al., Attention is all you need. Annual Conference on Neural Information Processing Systems, 5998–6008(2017).

[33] A. Dosovitskiy, L. Beyer, A. Kolesnikov, and et al., An image is worth 16x16 words: Transformers for image recognition at scale. (2020), arXiv:2010.11929.

[34] W. Wang, E. Xie, X. Li, and et al., Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. (2021), arXiv:2102.12122.

[35] R. Liu, Z. Yuan, T. Liu, and et al., End-to-end lane shape prediction with transformers. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 3694–3702(2021).

[36] R. Hjelm, A. Fedorov, S. Lavoie-Marchildon, and et al., Learning deep representations by mutual information estimation and maximization. (2019), arXiv:1808.06670.

[37] C. Dongyue, B. Haozhe, T. Chunren, and et al., Multi-information constraint learning for unsupervised domain adaptive person re-identification. Neural Processing Letters, 1-19(2022).

[38] S. Mukherjee, H. Asnani, S. Kannan, CCMI: Classifier based conditional mutual information estimation. Proceedings of the 35th Uncertainty in Artificial Intelligence Conference, 1083–1093(2020).

[39] P. Bachman, R. Hjelm, W. Buchwalter, Learning representations by maximizing mutual information across views. Proceedings of the 33rd International Conference on Neural Information Processing Systems, 15535–15545(2019).

[40] J. Xu, A. Vedaldi, J. Henriques, Invariant information clustering for unsupervised image classification and segmentation. 2019 International Conference on Computer Vision, 9865-9874(2019).

[41] T. Chen, S. Kornblith, M. Norouzi, and et al., A simple framework for contrastive learning of visual representations. (2020), arXiv:2002.05709.

[42] Tusimple, Tusimple lane detection benchmark (2017). https://github.com/TuSimple/tusimple-benchmark

[43] Tusimple, Tusimple benchmark (2019). https://github.com/TuSimple/tusimple-benchmark

[44] J. Philion, Fastdraw: Addressing the long tail of lane detection by adapting a sequential prediction network. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11582–11591(2019).

[45] Y. Hou, Z. Ma, C. Liu, and et al., Learning lightweight lane detection cnns by self-attention distillation. Proceedings of the IEEE/CVF International Conference on Computer Vision, 1013–1021(2019).