MLP-Based Efficient Convolutional Neural Network for Lane Detection

Xuedong Yao, Yandong Wang, Yanlan Wu, Guoxiong He, and Shuchang Luo

Abstract—**Lane detection is an important and fundamental task in autonomous driving. Modern convolutional neural network (CNN) methods have achieved high performance in lane detection; however, the intrinsic locality of convolution operations makes these methods limited in effectively modeling the long-range dependencies that are vital to capture global information of lanes. Additionally, numerous convolution operations result in considerable computational cost for high complexity. To overcome these difficulties, we propose an efficient lane detection method by combining CNN with a multilayer perceptron (MLP). First, an improved bottleneck-1D layer is used to replace the standard convolutional layer in overall network to reduce the computational cost and parameters while applying hybrid dilated convolution (HDC) to better capture multiscale lane information. Second, we construct a hybrid MLP block in the latent space to capture the long-range dependencies of lanes. The hybrid MLP projects tokenized convolutional features from spatial locations and channels, and then, they are fused together to obtain global representation, in which each output pixel is related to each input pixel. The introduction of MLP further decreases computational complexity and makes the proposed architecture more efficient for lane detection. Experimental results on two challenging datasets (CULane, Tusimple) demonstrate that our method can achieve a higher computational efficiency while maintaining a decent detection performance compared with other state-of-the-art methods. Furthermore, this study indicates that integrating the global representation capacity of an MLP with local prior information of convolution is an effective and potential perspective in lane detection.**

Index Terms—**Convolutional neural network (CNN), lane detection, long-range dependencies, multilayer perceptron (MLP).**

Manuscript received 13 November 2022; revised 21 February 2023 and 13 April 2023; accepted 9 May 2023. Date of publication 12 May 2023; date of current version 17 October 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 41271399, in part by the National Key Research Program of China under Grant 2016YFB0501403, and in part by the China Special Fund for Surveying, Mapping and Geoinformation Research in the Public Interest under Grant 201512015. The review of this article was coordinated by Dr. Chen Lv. *(Corresponding author: Yandong Wang.)*

Xuedong Yao, Guoxiong He, and Shuchang Luo are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan 430079, China (e-mail: yaoxuedong@whu.edu.cn; 2020206190019@ahu.edu.cn; 2020206190016@ahu.edu.cn).

Yandong Wang is with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan 430079, China, also with the Collaborative Innovation Center for Geospatial Information Technology, Wuhan 430079, China, and also with the Faculty of Geomatics, East China University of Technology, Nanchang 330013, China (e-mail: ydwang@whu.edu.cn).

Yanlan Wu is with the Information Materials and Intelligent Sensing Laboratory of Anhui Province, Anhui University, Hefei 230601, China, also with the School of Resources and Environmental Engineering, Anhui University, Hefei 230601, China, and also with the Anhui Engineering Research Center for Geographical Information Intelligent Technology, Hefei 230601, China (e-mail: wuyanlan@ahu.edu.cn).

Digital Object Identifier 10.1109/TVT.2023.3275571

I. Introduction

LANE detection, as an important component in the Advanced Driver Assistance System (ADAS), has been widely used for lane departure warning, navigation, traffic understanding, and so on [1], [2] which can assist drivers in safe driving. However, the thin and long shapes of lanes with few appearance clues and complex road environments including illumination changes and occlusions of vehicles or pedestrians, make lane detection challenging in the past few years.

Recently, lane detection methods based on convolutional neural networks (CNNs) have become dominant. Due to the powerful representation capabilities of CNNs, they further improve lane detection performance and far exceed traditional methods that use simple handcrafted features such as color-based features [3] and structural texture [4] to extract lane information in a limited scene. Lane detection is usually regarded as a semantic segmentation task to predict whether each pixel of the input image belongs to the lane marking. Generally, semantic segmentation obtains a larger receptive field and high-level semantic information through consecutive convolution and sampling operations or deepening the network to strengthen representation. Nevertheless, segmentation-based methods are flawed in addressing abovementioned challenges, especially in occlusion and extreme light condition scenes, which require more effective global and spatial information. To this end, several methods attempt to reinforce the structural feature of lanes by passing or aggregating spatial information within feature maps. [5] proposed a spatial convolution to pass information slice by slice along four directions. This sequential message passing operation prominently improves the lane detection performance, but it produces a high computational cost and is time-consuming, which goes against the demand of real-time detection. Inspired by this idea, [6] and [7] designed a similar structure to enrich the structural information of lanes and effectively reduce the computational cost. To better capture the global information while increasing the efficiency of lane detection, attention-based methods [8], [9] and transformer-based methods [10], [11] have been introduced to lane detection tasks. The attention mechanism which is popular in many visual tasks [12], [13], [14] includes spatial and channel attention and can be easily implemented in any network. Spatial attention can calculate the weighted sum of feature maps at all positions as the response at the current position to capture global information [13]. The channel attention aims to highlight important feature maps by calculating a weight for each channel [15]. Compared to the attention mechanism, the transformer initially designed for sequence-to-sequence prediction has been used as an alternative architecture in lane detection tasks because of the innate global self-attention mechanisms. It utilizes the multiheaded self-attention module to effectively aggregate the global information in every layer and improve lane representation. This architecture does not require a deeper network and can improve the efficiency of the model [16]. Therefore, other than the CNN architecture, transformer-based methods [10], [11] have gradually become popular in lane detection tasks in recent years. Although these methods can effectively capture global information to improve the lane detection performance, they still struggle to obtain a higher computational efficiency, where attention-based methods require extra cost to calculate attention maps and transformer-based methods introduce quadratic computational and more memory overhead during processing high-resolution images.

To address the aforementioned problems, in this paper, we develop an efficient lane detection method from a novel perspective. We combine CNN with a hybrid MLP block to build an effective network. To increase the efficiency of lane detection, we use an improved bottleneck-1D block with hybrid dilated convolution (HDC) [17] as a key component of CNN. This block can significantly reduce the computational cost and parameters while effectively strengthening lane information. Moreover, considering that CNN exhibits general limitations in modeling explicit long-range dependency, we construct a hybrid MLP block in latent space to replace aggressive convolution and pooling operations, which decreases the loss of local details while further simplifying the architecture. MLP is efficient while maintaining comparable performance, which has been applied to semantic segmentation tasks [18], [19], and it is better at modeling long-range representation but worse at capturing the local information [20]. Hence, we combine CNN with a hybrid MLP block in a sequential manner to effectively fuse the local prior information and global representation of lanes. Concretely, the hybrid MLP block consists of the spatial and channel MLP and extracts lane information from spatial locations and channels respectively, which can boost the global representation adequately.

Our method is validated in the challenging lane detection dataset (CULane, Tusimple). Experimental results demonstrate the effectiveness of our method. Compared with state-of-the-art lane detection methods, our method achieves a decent performance closer to that of them but a higher computational efficiency. The main contributions of this paper are summarized as follows:

- We provide a novel perspective that introduces the multilayer perceptron (MLP) to lane detection tasks. This paper proves the feasibility and effectiveness of MLP, which is worthy of further exploration in future research.
- We propose an efficient architecture by combining CNN with constructed hybrid MLP block. This architecture can fuse local information and long-range dependency to obtain stronger lane representation for lane detection while greatly reducing the computational cost.
- We validate our method on the CULane and Tusimple datasets to reveal the effectiveness and efficiency. Our method achieves good results in terms of the accuracy/cost trade-off and obtains a higher computational efficiency with minimal performance loss.

II. Related Works

A. Lane Detection Methods

Current lane detection methods are often classified into two classes: traditional methods and CNN-based methods. Most traditional methods are based on handcrafted features to conduct lane detection tasks. They mainly utilize these low-level visual characteristics of images, such as color information, texture structure, and gradient features. Yan et al. [21] and He et al. [22] used color information to extract lane markings. In [23], considering that color and edges are important features, the author fused color and edge information to detect lane markings and then proposed a line fitting model to compute the lane parameters. In [24], the authors used lane-mark colors to eliminate the influence of moving vehicles and lighting conditions. In addition, several methods applied the Hough transform [25], particle filtering [26] and Gabor filtering [27] to lane detection tasks. Song et al. [28] used a maximum likelihood angle to design a self-adaptive traffic lanes model in Hough Space. Li et al. [27] estimated vanishing points by extracting road texture features. Then, Gabor filter edge detection and Hough transformation were used to detect lanes with a constrained search by vanishing points, which is insensitive to variations in road conditions with clear texture features. In [26], the author proposed a robust real-time lane-detection algorithm based on RANSAC, which was combined with a particle-filtering algorithm by a probabilistic grouping framework. It obtained a promising result in different types of lanes while flawing in complex road scenes (i.e., shadow, occlusion). Although these traditional methods usually have a low computational cost and are fast yet simple, the detection results are always unsatisfactory due to the complex road environment, such as lighting conditions, occlusion and various kinds of lane types, making these methods have poor scalability.

CNN-based methods have been popular in lane detection with the development of deep learning in the computer vision field. These methods can be further divided into three categories: segmentation-based, classification-based and parameter-based, according to the usage of lane presentation. 1) Segmentation-based methods, which output pixel-level predictions are now mainstream in lane detection. Neven et al. [29] proposed treating the lane detection task as an instance segmentation problem and used a learned perspective transformation to parametrize the segmented lane instances. However, it is difficult to deal with broken and occluded lane markings because of limited global representation. In [30], the authors proposed a simple yet appealing network to exploit quick connections and gradient maps for effective learning of lane line features. It achieved a better performance on three datasets but was easily affected by the completely or partly occluded road surface and dim lights. To overcome lane occlusions, SCNN [5] proposed a message passing structure to capture the global context and enrich spatial information. However, this architecture requires considerable computational cost. Xiao et al. [6] proposed a recurrent slice convolution module (RSCM) to exploit the prior structural information of lane markings and achieved excellent computational efficiency while keeping decent detection quality. In addition, Liu et al. [31] designed a label-guided distillation method (LGAD) for lane segmentation and used a teacher network to reinforce the attention maps of the student network to capture long-range context. Similarly, Hou et al. [32] achieved a significant improvement without additional supervision or labeling by using the proposed self-attention distillation (SAD) module. Wang et al. [33] proposed a multitask method by integrating segmentation, handcrafted features, and fitting to improve the accuracy of location and convergence speed of networks. Ko et al. [34] proposed a traffic line detection method based on key points estimation and instance segmentation. 2) Some studies [35], [36], [37] cast lane detection as a classification task. Qin et al. [35] proposed a novel formulation with structural loss, which regarded lane detection as row-based classification using global features and achieved a high speed. In [36], the author also adopted row-wise classification to perform direct lane marker vertex prediction in an end-to-end manner without any postprocessing steps. Son et al. [38] used adaptive threshold and lane classification algorithm to construct a robust multi-lane detection method for challenging road conditions. 3) Different from the above methods, parameter-based methods consider lane detection as a lane curve model and directly regress these parameters by polynomial. Gansbeke et al. [39] used a deep neural network to predict the weight map of each lane line and proposed a differentiable least-squares fitting module to fit a curve. In [40], the author adopted deep polynomial regression to output polynomials representing each lane marking, which obtained a high efficiency but ignored the global information. LSTR [41] formulated a lane shape model based on road structures and camera poses and applied the transformer to learn richer structures and global information. These parameter-based methods have a fast inference speed but are sensitive to output parameters, and it is difficult to obtain a higher performance.

B. *Global Information Extraction*

Global context information is important to capture lane presentation from limited visual cues, especially in completely or partly occluded roads and extreme lighting conditions. In most lane detection methods, there are several ideas to accomplish this goal. First, enlarging the receptive field of feature maps is the fundamental and universal technique, which can be obtained by aggressive convolution and pooling operations or dilated convolution [42]. Li et al. [43] proposed a multiclass lane detection model based on DeepLabv3+ and achieved excellent performance in real traffic scenarios. This method has a higher computational cost and number of parameters with poor real-time performance. Second, many methods design specific modules to capture global information. In [44], [45], the author used the nonlocal block for capturing long-range dependencies by a self-attention mechanism. SCNN [5] proposed a slice-by-slice convolution and used message passing operations to obtain stronger spatial lane information. However, it suffers from expensive computation. The RESA [7] designed a similar convolution operation to make use of strong shape priors of lanes and captured spatial relationships of pixels to gather global information in vertical and horizontal directions. Although RESA is more efficient than the SCNN, it still has a high computational complexity. Apart from these abovementioned methods, some studies apply transformers to lane detection tasks. The transformer proposed by [46] is initially used for machine translation. Due to the excellent global self-attention capacity, it has been quickly applied to computer vision tasks [47], [48], [49] since the Vision Transformer (ViT) [50] was proposed. Recently, transformer-based lane detection methods have achieved a better performance in lane segmentation tasks [41], [51].

In this paper, we propose an efficient method from another perspective. We construct a hybrid MLP block in latent space to capture long-range representation, which is efficient and keeps a lower computational cost.

III. Methodology

This section mainly presents the proposed efficient network. The detailed description is as follows. We first introduce the overview of our method. Second, the Im-bottleneck-1D block is taken into consideration. Finally, the constructed hybrid MLP, loss function and detailed architecture of the lane detection network are described.

A. *Overview of Architecture*

Similar to most lane detection networks [52], [53], our proposed method is also composed of three components: the encoder, decoder and lane existence branch. Briefly, the encoder is used to decrease the resolution of feature maps while capturing multilevel information from top to bottom. Conversely, the decoder is applied to restore the size and output the final segmentation map. The lane existence branch aims to encode convolutional features into vectors for predicting lane existence. The overall network is displayed in Fig. 1.

We can see that given an input image with a spatial resolution of $H \times W$, the final goal is to obtain the corresponding pixelwise prediction map of lanes with the same size $H \times W$ and the probabilities of lane existence. The input image is first passed through 4 downsampling blocks followed by the hybrid MLP block, which includes the channel and spatial MLP. Then, the feature maps are sampled by 4 upsampling blocks, where the convolutional features are directly used as the input of the lane existence branch after the first upsampling block for the next operations. The MLP is good at modeling the long-range dependencies; hence, we place it behind convolution operations to encode information. Each sampling block makes the feature resolution decrease or increase 2 times. Moreover, we denote the number of channels in every stage as C1 to C6. Considering that the proposed network is not deep, these channels are set to 16, 64, 128, 256, 5 and 4. It is worth noting that there is only one skip connection layer between the encoder and encoder after the third downsampling block. The main idea behind this approach is that it can fuse convolutional features with long-range information from MLP more effectively, and the computational efficiency

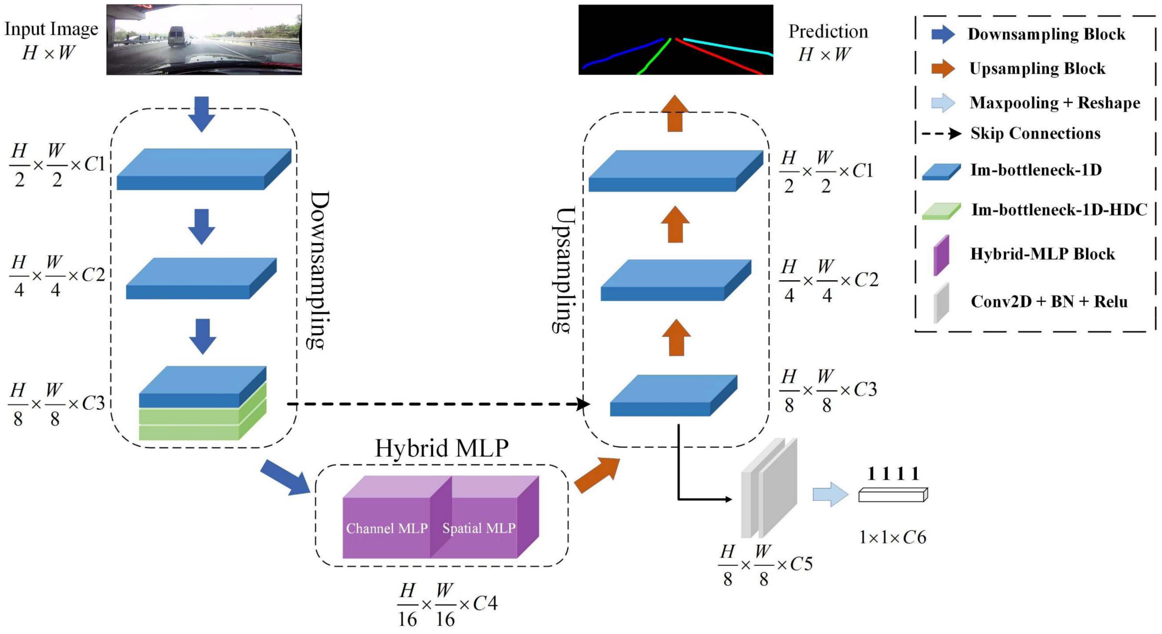
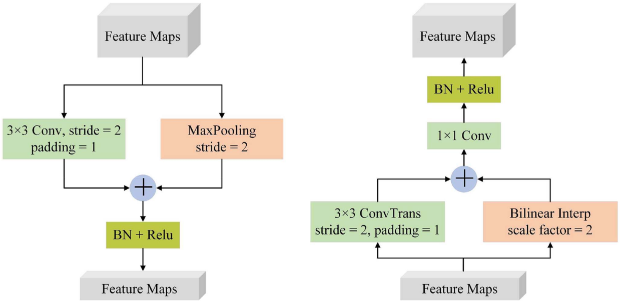

Input Image
$H \times W$
$\frac{H}{2} \times \frac{W}{2} \times C1$
$\frac{H}{4} \times \frac{W}{4} \times C2$
$\frac{H}{8} \times \frac{W}{8} \times C3$
Downsampling
Hybrid MLP
Channel MLP
Spatial MLP
$\frac{H}{16} \times \frac{W}{16} \times C4$
Upsampling
Prediction
$H \times W$
$\frac{H}{2} \times \frac{W}{2} \times C1$
$\frac{H}{4} \times \frac{W}{4} \times C2$
$\frac{H}{8} \times \frac{W}{8} \times C3$
$\frac{H}{8} \times \frac{W}{8} \times C5$
1 1 1 1
$1 \times 1 \times C6$
Downsampling Block
Upsampling Block
Maxpooling + Reshape
Skip Connections
Im-bottleneck-1D
Im-bottleneck-1D-HDC
Hybrid-MLP Block
Conv2D + BN + Relu

Fig. 1. Overview of proposed architecture for lane detection.



(a) (b)

Fig. 2. Structures of down-sampling block (left) and up-sampling block (right).

will be greatly improved due to the reduced number of skip connection layers.

Additionally, the semantic information used for consecutive lane segmentation and lane existence prediction is provided by the encoder network. In other words, the performance of the encoder network for feature extraction plays a significant role in the completion of lane detection. Generally, the pooling operation will lead to the loss of detailed information which is not conducive to subsequent tasks. Inspired by the classical ENet [54], we also adopt two strategies to sample feature maps in our downsampling blocks. One is the standard convolution operation with a stride of 2, and the other is the direct max-pooling operation. Their outputs are concatenated together to form new feature maps that can maintain rich detailed information well. Fig. 2(a) shows the detailed structure of downsampling block. Analogously, we also use the bilinear interpolation and transposed convolution operation in upsampling block. The former expands feature maps with an interpolated method that is simple and fast but will produce much coarse information. The latter can update learned weights like standard convolution operations to restore more information. Therefore, their combination can refine feature maps as much as possible during the upsampling process. Then, a 1 × 1 convolution operation is used to further reduce the parameters. The concrete structure of upsampling block is drawn in Fig. 2(b). In conclusion, the downsampling and upsampling blocks can make the proposed network more effective.

B. Improved Bottleneck-1D Block

A standard convolutional layer is indispensable for feature extraction in CNNs. Consecutive convolution operations can make full use of local prior information, but the parameters and computational complexity will increase obviously with the growing number of convolutional layers. To constrain the number of parameters in convolution operations, many intriguing techniques and structures have been proposed. The bottleneck structure [55], depthwise separable convolution [56], 1D convolution and difference convolution [57] are all representative works. In fact, we have verified that depthwise separable convolution has fewer parameters; however, the performance has a large drop, which is due to the decreased feature extraction ability. To this end, we combine the bottleneck structure with 1D convolution to propose an improved variant named the Im-bottleneck-1D block, which can balance the efficiency and performance to some degree.

Our Im-bottleneck-1D is similar to the Non-bottleneck-1D layer used in ERFNet [58]. The most obvious difference between them is the number of parameters. Fig. 3 shows the different variants of the residual block. It should be noted that the Im-bottleneck-1D block reduces the number of channels by 2 times in our experiment. Fig. 3(a) and (b) are the basic

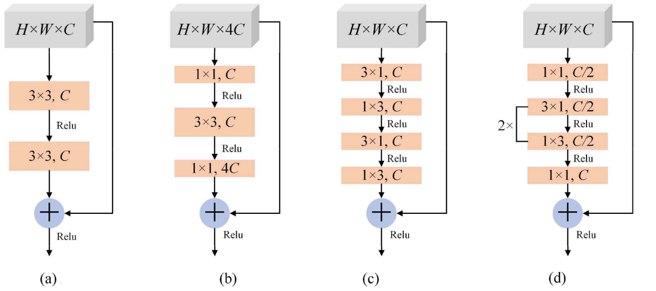


Fig. 3. Different variants of residual block. (a) Non-bottleneck. (b) Bottleneck. (c) Non-bottleneck-1D. (d) Im-bottleneck-1D.

TABLE I
COMPARISONS ON PARAMETERS OF DIFFERENT STRUCTURES IN ONE BLOCK. C IS THE NUMBER OF CHANNELS. THE BIAS TERM HERE IS NOT TAKEN INTO CONSIDERATION

Variants	Non-bot	Bot	Non-bot-1D	Im-bot-1D
Params	$18C^2$	$17C^2$	$12C^2$	$4C^2$

components of ResNet [55], and they have similar performance and number of parameters. According to the detailed structure, we can see that the parameters of the Non-bottleneck are $18C^2$ and the Bottleneck is $17C^2$. However, the bottleneck structure has a low computational cost. Compared to the former two blocks, the Non-bottleneck-1D divides a standard 3 × 3 convolution into 3 × 1 and 1 × 3 convolutions, and the parameters are directly reduced to $12C^2$. This structure receives a 33% reduction in parameters and can accelerate the execution time while maintaining a balanced performance. Therefore, the Non-bottleneck-1D block has been widely applied to many lane detection networks. Despite this, lane detection tasks still have a high demand on fewer parameters and computational cost. In this paper, we expand the Non-bottleneck-1D to Im-bottleneck-1D by introducing a bottleneck structure, as shown in Fig. 3(d). In the Im-bottleneck-1D block, feature maps first reduce the number ofpara channels by a 1 × 1 convolution, and then 3 × 1 and 1 × 3 convolution operations are conducted twice followed by a 1 × 1 convolution to restore the original channels. By using a bottleneck structure, we can obtain that parameters of the Im-bottleneck-1D block are further reduced to $4C^2$, which are 3 times smaller than those of the Non-bottleneck-1D. This indicates that the Im-bottleneck-1D block is more efficient than other structures while maintaining the advantages of the Non-bottleneck-1D block. Table I lists these parameters of different structures in one block. In a word, our proposed Im-bottleneck-1D block may make the performance of lane extraction a little drop but can produce as few parameters and computational costs as possible than other structures, promoting our method to be more efficient.

Furthermore, to mitigate the drawback of Im-bottleneck-1D and strengthen lane information, we apply hybrid dilated convolution (HDC) to the Im-bottleneck-1D block. The dilated convolution can effectively enlarge the receptive field, which is beneficial for capturing object information without introducing parameters. Many studies [59], [60] often adopt a series of dilated convolutions with large dilated rates in parallel or sequentially to extract multi-scale information. In ERFNet, for example, the dilation rates of some Non-bottleneck-1D layers are set to 2, 4, 8, and 16. However, when the receptive field becomes larger, it is difficult to capture the consecutive local information and leads to the loss of detailed features, called the "gridding effects". Here, we use HDC to tackle this issue. The HDC adopts a set of dilated rates with a serrated structure (i.e., r = [2, 3, 5]) that cannot have a common divisor greater than 1. By the complement of different receptive fields, this serrated structure can reduce the loss of local information. In our method, we embed a hybrid dilated convolution with dilated rates of [2, 3, 5] into several Im-bottleneck-1D layers to extract multiscale lane information. Generally, dilated convolution with a small dilation rate always focuses on local information, while the successive accumulation of a small receptive field will be equivalent to a larger one that can capture large-scale information. Hence, we take advantage of HDC to improve the performance of Im-bottleneck-1D. The detailed structure of the Im-bottleneck-1D-HDC layer is displayed in Fig. 4. In the next experimental section, we reveal the quantitative differences between Im-bottleneck-1D and Im-bottleneck-1D-HDC.

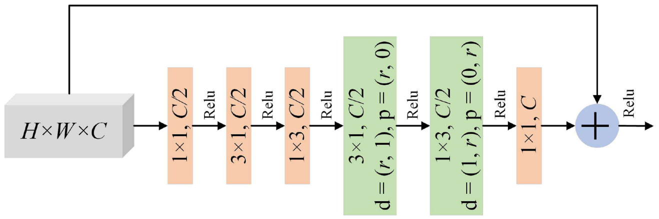


Fig. 4. Structure of Im-bottleneck-1D-HDC block.

C. Hybrid MLP Block

Inspired by [61] that an architecture based exclusively on MLPs is used to learn representations of images patches for image classification, in this paper, we construct a pixel-level hybrid MLP block in latent space to capture long-range representations from spatial location and channel dimension of deep feature maps.

An MLP block often takes a sequence of image patches as input. Generally, these patches are non-overlapping and will be projected to a hidden dimension C. Specifically, given an input image $x \in R^{\mathrm{H \times W \times 3}}$, the number of image patches can be defined as $N = HW/P^2$, where H, W and P denote the height, width of feature maps and the size of each patch, respectively. Then these patches are flattened and passed into a linear layer with output dimension C, obtaining the raw tokens $x_p \in R^{\mathrm{N \times C}}$. In our method, due to employing an MLP block in latent space, we assume that the feature map is $h \times w \times c$ and adopt 1 × 1 patches (i.e., $P = 1$) extracted from the feature map instead of from the input image. In other words, each pixel of the feature map is treated as a patch and each output pixel is determined by previous each input pixel, which can effectively capture long-range representation. In different tokenized dimensions, we finally construct the channel and spatial MLP, as shown in Fig. 5.

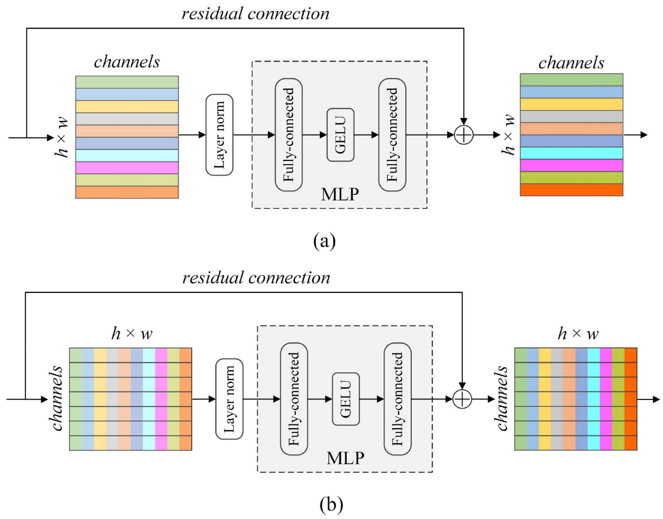


Fig. 5. Architectures of channel and spatial MLP block. (a) Channel-MLP. (b) Spatial-MLP.

We can observe that they have the same architecture: layer normalization, an MLP layer and residual connection. Here, layer normalization (LN) is preferred rather than batch normalization (BN) because LN can make more sense to normalize along the tokens instead of normalizing across the batch in the MLP block. The MLP layer consists of two fully-connected layers coupled with a GELU [62] activation layer, and the hidden dimension of fully-connected layer is a hyperparameter. After that, a skip connection layer is used to connect the output tokens with the original tokens. The channel-MLP is mainly used to fuse all channel information of each spatial location along the channel dimension, and the input tokens are $x_c \in R^{\mathrm{h\,w\,\times\,c}}$. Spatial-MLP aims to capture global spatial information of each channel along the spatial dimension, and input tokens are $x_s \in R^{\mathrm{c\,\times\,h\,w}}$ (i.e., $x_s = (x_c)^{\mathrm{T}}$). The concrete equation of channel and spatial MLP can be written as:

$$X^c = x^c + f_{c2}\alpha(f_{c1}LN(x^c)) \tag{1}$$

$$X^s = x^s + f_{s2}\alpha(f_{s1}LN(x^s)) \tag{2}$$

where *f* denotes the fully-connected operation in the MLP layer, α is the GELU activation function and LN denotes the layer normalization. Inferred by the calculation of the MLP, the hidden dimension of the channel MLP is irrelevant to the feature map size, so the computational complexity of the channel MLP is linear in spatial locations. Likewise, the hidden dimension of spatial MLP depends on the number of patches, and the complexity is linear in channels. Therefore, MLP blocks can effectively reduce the computational complexity.

The two MLP blocks focus on modeling long-range representation in spatial locations and channels. They have two hybrid manners. One is in a parallel manner, and the other is in a sequential manner. We denote that the input of the MLP block is *x*, and the output is *o*. The functions S_{MLP} and C_{MLP} represent the spatial and channel MLP, respectively. Therefore, parallel hybrid MLP can be summarized as follows (Fig. 6(a)):

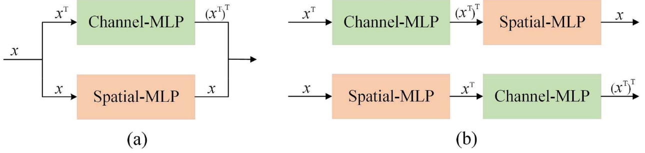


Fig. 6. Combination manners. (a) Is the parallel manner and (b) is the sequential manner.

$$o = S_{MLP}(x) + C_{MLP}^T(x^T) \tag{3}$$

Similarly, the sequential hybrid MLP can be written as (Fig. 6(b)):

$$o = S_{MLP}(C_{MLP}^T(x^T)) \tag{4}$$

$$o = C_{MLP}^T(S_{MLP}^T(x)) \tag{5}$$

Finally, the output tokens of the hybrid MLP block are reshaped to feature maps. In this paper, we find that the parallel and sequential methods have their own advantages, especially when the location of the channel MLP is before spatial MLP or when they are parallel. In this case, we prefer the upper sequential manner in Fig. 6(b) over the others. In fact, our experiment demonstrates that the upper sequential one can achieve a better performance than the lower one and has a slight improvement to the parallel manner. Section IV provides a quantitative comparison and detailed discussion.

D. Loss Function

In our method, there are two output branches in the lane detection network: the semantic segmentation branch and the lane existence branch as displayed in Fig. 1. Therefore, the final optimized loss contains segmentation loss and existence loss.

Segmentation Loss: Due to the thin and long shape, lanes account for a small ratio in the whole image, which is highly unbalanced compared to the background. To solve this problem, we use the weighted cross-entropy loss and this loss L_{ce} is formulated as:

$$L_{ce}(x) = -\sum_{H\times W}\sum_{c\in C} w_c\, y_c \log(p_c(x)) \tag{6}$$

Here *x* denotes the input image with a size of $H \times W$ and *C* is the output number of classes. w_c and $p_c(x)$ denote the weight and predicted probability map for class c, respectively. y_c represents the pixels that belong to class c in the corresponding ground truth label. Moreover, to further pay more attention to the lane information, we introduce the dice loss [63] to evaluate the similarity between predicted feature maps and labels. The dice loss can be written as:

$$L_{dice} = 1 - \sum_{c=1}^{C} \frac{2\sum_{i=1}^{H}\sum_{j=1}^{W} p_c(i,j)\, y_c(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W} p_c(i,j) + \sum_{i=1}^{H}\sum_{j=1}^{W} y_c(i,j)} \tag{7}$$

where $p(i,\ j) \in [0,\ 1]$ denotes the predicted probability of each pixel that belongs to lanes and $y(i,\ j) \in \{01\}$ represents

TABLE II
DETAILED ARCHITECTURE OF LANE SEGMENTATION BRANCH. THE *R* DENOTES DILATED RATE AND FUSION PART REPRESENTS THE ALTOGETHER OF HYBRID MLP BLOCK AND SKIP CONNECTION LAYER

Semantic segmentation branch		
Components	*Type*	*h* × *w* × *c*
Encoder part	Input image	208×976×3
	Down-sampling	104×488×16
	Down-sampling	52×244×64
	5 × Im-bot-1D	52×244×64
	Down-sampling	26×122×128
	Im-bot-1D	26×122×128
	Im-bot-1D-HDC (*r*=2)	26×122×128
	Im-bot-1D-HDC (*r*=3)	26×122×128
	Im-bot-1D-HDC (*r*=5)	26×122×128
	Im-bot-1D	26×122×128
	Im-bot-1D-HDC (*r*=2)	26×122×128
	Im-bot-1D-HDC (*r*=3)	26×122×128
	Im-bot-1D-HDC (*r*=5)	26×122×128
Fusion part	Down-sampling	13×61×256
	Hybrid MLP	13×61×256
	Up-sampling	26×122×128
Decoder part	Up-sampling	52×244×64
	2 ×Im-bot-1D	52×244×64
	Up-sampling	104×488×16
	2 ×Im-bot-1D	104×488×16
	Up-sampling	208×976×5

the corresponding label. Thus, the complete segmentation loss consists of two components and can be summarized as follows:

$$L_{seg} = L_{ce} + L_{dice} \tag{8}$$

Existence Loss: This existence loss is mainly used for the CULane datasets because lanes have been classified by their corresponding positions. It is a binary cross-entropy loss to determine the existence of each lane. The existence loss L_{exist} is formulated as:

$$L_{exist} = -\sum_{H\times W} y\log(p) + (1-y)\log(1-p) \tag{9}$$

Finally, we combine all the segmentation and existence losses to form the final objective function as follows:

$$L_{all} = \alpha L_{ce} + \beta L_{dice} + \gamma L_{exist} \tag{10}$$

The parameters α, β and γ are used to balance the cross-entropy loss, dice loss, and existence loss of the final objective function.

In our experiment, these parameters are set to 1, 0.5 and 0.1.

E. Details of Architecture

The detailed architecture of the lane segmentation branch is illustrated in Table II. We can see that the encoder has 13 Im-bottleneck-1D layers while the decoder only has 4 Im-bottleneck-1D layers. This asymmetric architecture can make the encoder network take full advantage of lane information for subsequent segmentation and classification tasks and accelerate the execution time by reducing the parameters and computational cost in the decoder network. In addition, the last upsampling block samples feature maps by only using the transposed convolution operation to output the final segmentation map. Moreover, considering that the lane existence branch and segmentation branch share the same encoder and fusion parts, we show the architecture of the lane existence branch in Table III individually and do not repeat the previous modules.

TABLE III
DETAILED ARCHITECTURE OF LANE EXISTENCE BRANCH. THE R AND P DENOTE DILATED RATE, PADDING VALUE, RESPECTIVELY

Lane existence branch	
Type	*h* × *w* × *c*
3×3 *Conv* (*r*=4, *p*=4)	26×122×32
BN + *RELU*	26×122×32
1×1 *Conv*	26×122×5
Max pool + *Flatten*	1×1×3965
*FC-layer*1 + *RELU*	1×1×128
*FC-layer*2 + *Sigmoid*	1×1×4

IV. EXPERIMENTS

This section demonstrates the effectiveness and efficiency of our proposed method by conducting experiments on two challenging datasets.

A. Setup and Evaluation

Dataset: To prove the generalization ability of our method, we conduct experiments on two widely used lane detection benchmark datasets: CULane and Tusimple [64].

CULane is a widely used lane detection dataset released by Pan et al. [5]. More than 55 hours of videos are collected by cameras on six different vehicles in urban and highway scenarios with various lighting conditions. After processing, it contains 133235 images with a resolution of 1640 × 590, where 88880 images are used for the training set, 9675 for the validation set and 34680 for the test set. These images consist of nine different road scenes, including normal, crowd, curve, dazzle night, night, no line, and arrow. The CULane dataset only focuses on the detection of four lane markings, which are given the most attention in real applications. Lane markings are instantiated based on the different locations seen in Fig. 7, and whether the lane marking exists or not is annotated as 0 or 1, in which 0 denotes inexistence, and 1 denotes existence. For example, there are four lane markings in an image, and the annotation is recorded as 1, 1, 1, 1, which is used as the ground truth label of the lane existence branch.

Tusimple is a small scale but also widely used dataset in lane detection, which is collected from camera video on highway scenarios with good or medium weather conditions. In this dataset, there are about 7000 one-second-long video clips of 20 frames each and the last frame of each clip is annotated. It contains 6408 images with the resolution of 1280 × 720 in total,

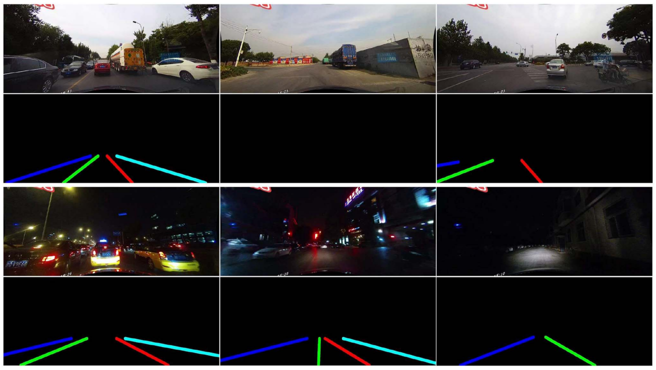


Fig. 7. Annotated samples of CULane dataset.

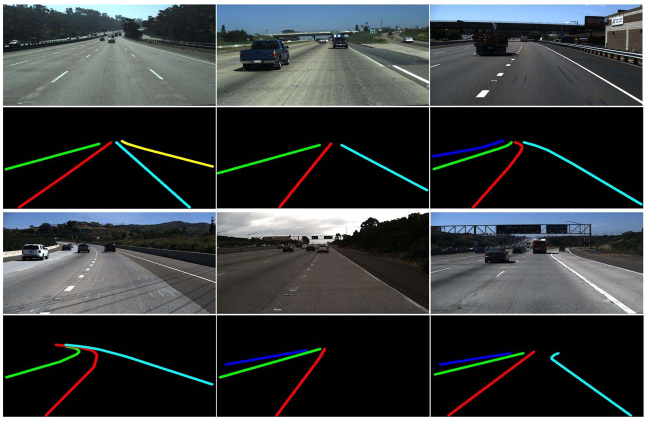


Fig. 8. Annotated samples of Tusimple dataset.

where 3626 images are used for training and 2782 for testing. Fig. 8 shows some samples with annotations.

Implementation: In our experiment, we use the PyTorch deep learning framework [65] to implement the proposed network on a single NVIDIA RTX3090 GPU with 24 GB memory. We use the stochastic gradient descent (SGD) [66] optimizer to train the network with an initial learning rate of 0.002, momentum of 0.9 and weight decay of 0.0001. We train this network without using any pretrained model. The learning rate decay rule obeys poly and the power is 0.9. Considering the imbalance between background and lanes, the segmentation loss of background is multiplied by 0.4. In the CULane dataset, the training epochs and batch size are set to 30 and 8, respectively. Finally, the total number of iterations is 333300 and the model file is outputted every epoch. Moreover, to better focus on lane information and reduce the interference of insignificant information, we discard the former 240 rows of input images and corresponding ground truth labels, because this part mainly contains sky information. Then, these images and labels are scaled to 976 × 208 as the input of the network. We evaluate the output model of each epoch on the validation set and save the corresponding training and validating losses to observe the training procedure dynamically. Fig. 9 shows the graph of all these losses. In the Tusimple dataset, the training epochs are 100 and batch size is also set to 8. The training iteration number is 45300. During training, we do not utilize any data augmentation methods and the input images and labels are resized to 512 × 256. In this experiment, the other operations are same as that on CULane dataset.

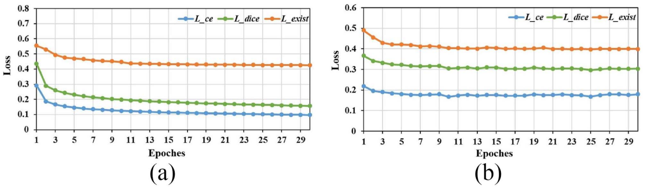


Fig. 9. Comparisons on train loss (a) and validation loss (b).

Evaluation Metrics: In the CULane dataset, the lane is described as a line with a width of 30 pixels, and then the intersection-over-union (IoU) between the predicted results and ground truth labels is calculated. When the IoU of the predicted lane is larger than 0.5, these lanes are regarded as true positive (TP). Similar to most studies in lane detection, in this experiment, we also adopt the F1-measure as the evaluation metric and define it as follows:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{11}$$

where the *Precision* and *Recall* are calculated as:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

In Tusimple dataset, the evaluation metrics are false positive (FP), false negative (FN) and accuracy. The accuracy is defined as follow:

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \tag{14}$$

where C_{clip} and S_{clip} are the number of correctly predicted lane points and the number of ground truth points, respectively.

B. Comparison With State-of-the-Art Methods

In this part, we compare our method with other state-of-the-art methods to demonstrate the effectiveness and efficiency of our method in two lane detection datasets.

Experiment on CULane Dataset: These methods are mainly from RSCM [6], which include the AMSC [8], SCNN [5], SAD [32], Res18-VP [67] and Res34-Ultra [35]. In addition, we also consider the lightweight version Res18-Ultra [35] and Res34-RESA [7], the result of [6] and our baseline network for comparison. The baseline network is only composed of the lm-bottleneck-1D blocks without any other components. All experimental results are listed in Table IV. From this table, we can observe that our method achieves better accuracy/cost trade-off results. Specifically, the proposed method improves the performance significantly compared to the baseline network with only growing a small number of the computational cost and parameters. This also indicates that the performance of lane detection is in a drop when merely increasing the computational efficiency. Compared with two efficient methods of

TABLE IV
COMPARISONS (%) WITH DIFFERENT STATE-OF-THE-ART LANE DETECTION METHODS ON CULANE DATASET. FOR CROSSROAD, ONLY FP ARE SHOWN. '–' DENOTES THE RESULT IS NOT AVAILABLE AND THE VALUES IN BOLD REPRESENT THE BEST RESULTS

Category	Baseline	Ours	RSCM	AMSC	R18-Ultra	R18-VP	SCNN	R101-SAD	R34-Ultra	R34-RESA
Normal	83.6	89.6	85.2	86.8	87.7	89.2	90.6	**90.7**	90.7	**91.9**
Crowded	62.3	67.1	63.8	64.5	66.0	67.9	69.7	70.0	70.2	**72.4**
Night	56.5	61.9	58.3	59.8	62.1	62.6	66.1	66.3	66.7	**69.8**
No line	35.2	41.8	43.9	38.4	40.2	41.7	43.4	43.5	44.4	**46.3**
Shadow	61.1	60.3	57.1	60.0	62.8	58.8	66.9	67.0	69.3	**72.0**
Arrow	74.8	83.0	77.8	77.4	81.0	81.6	84.1	84.4	85.7	**88.1**
Dazzle	50.4	59.9	56.6	55.7	58.4	59.3	58.5	59.9	59.5	**66.5**
Curve	58.0	61.4	57.0	63.4	57.9	60.8	64.4	65.7	**69.5**	68.6
Crossroad	2339	2071	2107	2358	**1743**	2919	1990	2052	2037	1896
Total	63.9	69.5	66.8	66.6	68.4	69.1	71.6	71.8	72.3	**74.5**
FLOPs(G)	5.38	**7.22**	23.6	-	8.4	-	328.4	162.2	25.1	41.0
Params(M)	1.34	**3.17**	3.21	20.7	-	16.07	20.72	52.53	57.9	-

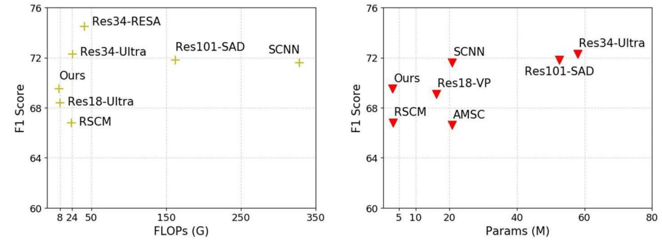


Fig. 10. Comparison charts. Y-axis denotes the F1 score. X-axis represents the FLOPs and number of parameters.

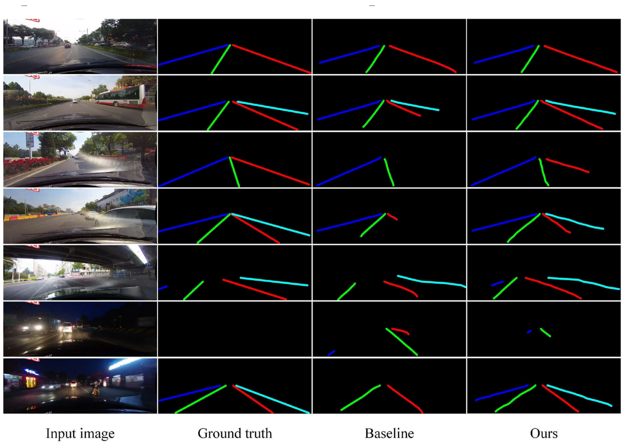


Fig. 11. Visual results on CULane dataset. The columns from left to right are the input image, ground truth label, the baseline model and our method, respectively.

RSCM and AMSC that respectively consider the prior structure information and dual-attention mechanism to strengthen lane presentation, our method outperforms them by 2.7% and 2.9% on accuracy, and the parameters and computational complexity are much smaller than those of AMSC and RSCM, respectively. This is because our method abandons redundant convolution operations and uses an efficient MLP block to model long-range representation effectively. This strategy can better balance performance and efficiency. Additionally, our method has a slight improvement over another efficient Res18-VP method, and the computational cost of our method is 5 times smaller than that of it. Compared to the lightweight Res18-Ultra method, our method is still superior in terms of segmentation results and computational cost. The performance of our method is inferior to the remaining methods, because they design many effective feature aggregation techniques. However, these aggregation techniques can generate higher computational complexity and more parameters than those of our method. Fig. 10 displays the comparison charts of F1 score vs. FLOPs and F1 score vs. number of parameters. Explicitly, comparisons between different methods demonstrate that our method has the potential to obtain a decent performance while retaining fewer computational complexity and parameters, which imply the advantage of our method in the accuracy/cost trade-off. Fig. 11 shows the visualizations of our method and baseline network.

Experiment on Tusimple Dataset: In this experiment, these comparison methods include RSCM, PointLaneNet [68], LaneNet [29], SCNN, PINet [34], Res34-RESA and CLRNet [69]. We also consider the baseline network for reference. The experimental results are shown in Table V. From this table, we see that our method also achieves better accuracy/cost trade-off results on Tusimple dataset. Our method improves the performance a lot than baseline network with only increasing a small amount of the computational cost and the parameters. Compared with the RSCM method, we achieve a closer performance while the computational cost of our method is lower than it. The small number of parameters of RSCM can be owed to the adjustment of channels. For the PointLaneNet and PINet, the accuracy of our method is lower than that of them, but the number of parameters is lower 2.63 M and 1.69 M than them. Similarly, the performance of remaining methods is obviously superior to our method in the accuracy, FP and FN. However, both the computational cost and the number of parameters of our method are much lower than them. From these experimental results, we can find that the further improvement of accuracy on Tusimple dataset is not easy, but the computational cost and parameters of network can be reduced sharply. This finding demonstrates the advantage

TABLE V
COMPARISONS (%) WITH DIFFERENT STATE-OF-THE-ART LANE DETECTION METHODS ON TUSIMPLE DATASET. '–' DENOTES THE RESULT IS NOT AVAILABLE AND THE VALUES IN BOLD REPRESENT THE BEST RESULTS

Method	Baseline	Ours	RSCM	PointLaneNet	LaneNet	SCNN	PINet	R34-RESA	CLRNet
Accuracy (%)	93.93	94.36	95.31	96.34	96.38	96.53	96.62	96.82	**96.84**
FP	0.076	0.055	0.084	0.047	0.078	0.061	0.031	0.036	**0.023**
FN	0.069	0.056	0.057	0.052	0.024	**0.018**	0.027	0.025	0.019
FLOPs(G)	3.49	**4.68**	7.33	-	74.64	111.24	-	-	-
Params(M)	1.16	2.70	**1.99**	5.33	15.98	20.66	4.39	-	-

TABLE VI
THE COMPARISON OF EFFECTIVENESS OF PROPOSED COMPONENTS ON CULANE DATASET. THE C_MLP AND S_MLP ARE THE ABBREVIATION FOR CHANNEL MLP BLOCK AND SPATIAL MLP BLOCK, RESPECTIVELY

Baseline	HDC	S_MLP	C_MLP	Runtime (ms)	FLOPs(G)	Params(M)	F1
√				13.5	5.38	1.34	63.9
√	√			13.5	5.38	1.34	67.2
√	√	√		14.2	6.80	2.64	67.8
√	√		√	14.3	7.01	2.35	67.8
√	√	√	√	14.5	7.22	3.17	69.5

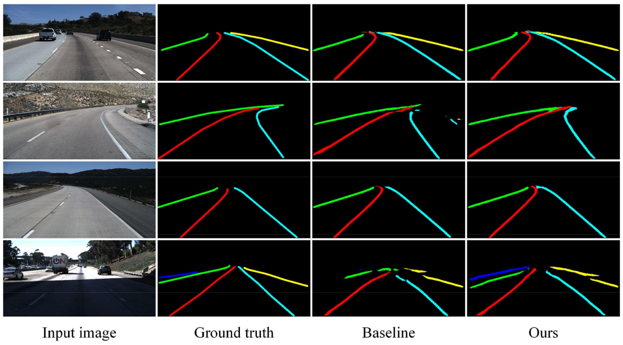


Fig. 12. Visual results on Tusimple dataset. The columns from left to right are the input image, ground truth label, the baseline model and our method, respectively.

of our method again that we can obtain a decent performance on lane detection while retaining fewer computational complexity and parameters. Fig. 12 displays the visualizations of our method and baseline network on Tusimple dataset.

C. Ablation Study

In Section III, we have discussed the Im-bottleneck-1D and hybrid MLP block and analyzed the characteristics of each block. In this section, we further perform detailed ablation studies on the CULane dataset to quantitatively prove their advantages.

Effectiveness of Each Component: We explore the effectiveness of each component in this part, including the HDC, spatial MLP block, channel MLP block and all of them. The baseline network, which is composed of stacked Im-bottleneck-1D layers, is used as the basic block. Then, each component is added into the basic block to verify the corresponding effectiveness. These results are shown in Table VI. We can see that the performance of the proposed network increases progressively with the introduction of each component. Specifically, we first add the HDC to the baseline network, and the performance improves by 3.3% without increasing the parameters and computational cost, which proves that the HDC can strengthen the lane information effectively. When the channel and spatial MLP blocks are taken into consideration, the result has a small boost compared to former components. The combination of channel and spatial MLP blocks makes the performance improve significantly with only a limited increase in parameters and computational cost. This indicates that the hybrid MLP block is effective in boosting the performance of our method. Moreover, we also present the average inference time on performing 10 images of resolution 976 × 208. Our method can achieve 70 fps, which shows efficient computation and meets the real-time applications.

Furthermore, to qualitatively describe the differences in components, we output the middle feature maps of the network coupled with each component. The visualizations of heatmaps are shown in Fig. 13. As we can see, Fig. 13(c) has more complete lane information than (b). This is because HDC can model local and large range representations to strengthen lane information compared with ordinary convolution operations. However, due to the lack of effective global representation, the feature maps of (b) and (c) are full of considerable noise, and the feature distribution is obviously different. With the introduction of MLP blocks, the feature distribution of lanes becomes smoother, as shown in (d) and (e), which effectively suppresses the noisy information. Compared to the previous components, the combination of two MLP blocks is more competitive. On the one hand, it has a smooth feature distribution with less noise as displayed in (f). On the other hand, it presents a powerful lane representation in feature maps. These visual results reveal that each component is beneficial for improving

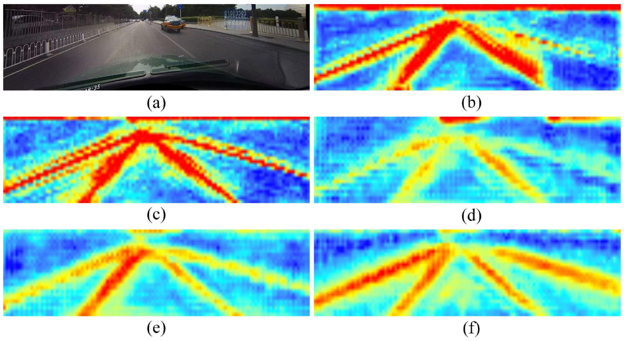


Fig. 13. Visualizations of heatmaps on each component. The (a–f) represent the input image, baseline, HDC, spatial MLP block, channel MLP block and combination of channel and spatial MLP.

TABLE VII
THE PERFORMANCE OF DIFFERENT COMBINATION MANNERS ON LANE DETECTION. THE VALUES IN BOLD REPRESENT THE BEST RESULTS

Category	S_C	C+S	C_S
Normal	89.1	89.3	**89.6**
Crowded	66.6	**67.5**	67.1
Night	60.1	61.6	**61.9**
No line	41.4	**42.9**	41.8
Shadow	60.1	55.8	**60.3**
Arrow	83.3	**84.3**	83.0
Dazzle light	59.0	**61.4**	59.9
Curve	61.0	**63.0**	61.4
Crossroad	3246	2432	**2071**
F1	68.3	69.4	**69.5**

lane representation and that all of them can produce an excellent performance, which again demonstrates the effectiveness of our method.

Combination Manner of the MLP Block: In this section, we quantitatively verify the importance of the combination manner for the lane detection task. As mentioned in Section III-C, there are two ways to place the channel and spatial MLP: the parallel and sequential methods. The sequential manner also includes two sequences. The detailed performance of each manner is listed in Table VII, where S_C and C_S denote the sequential manner, while C+S presents a parallel one. From this table, we can observe that the performance of the latter two hybrid manners is superior to the former one in most road scenes, which improves by 1.1% and 1.2% respectively. We have an intriguing finding that better lane detection performance can be obtained only when the placed position of the channel MLP block is prior to the spatial MLP block or they remain in a parallel manner. This reveals the importance of channel information and hints that channel MLP block can effectively model the spatial representation of different channels during the information fusion between the channel MLP and spatial MLP, which is good for reducing lane detection errors. For example, the false-positive rate of a crossroad scene decreases obviously with the change in the combination method. Comparing C+S with C_S, they have a similar performance, but the C_S manner still has a small advantage of 0.1%. Therefore, in our method, we use the sequential manner that the channel MLP block is in front of the spatial MLP block as our hybrid manner of MLP blocks. In fact, each combination is capable of improving the performance of lane detection, and we adopt the optimal one.

V. Conclusion

In this work, we present an efficient architecture based on a hybrid MLP to effectively enhance lane representation and improve the efficiency of lane detection. We construct an improved bottleneck-1D block with an HDC layer to reduce the computational complexity and parameters while capturing multiscale lane information and propose a hybrid MLP block to further learn the global lane representation by modeling long-range dependencies in channels and spatial locations. Experimental results on the challenging CULane and Tusimple datasets have demonstrated the effectiveness of our method. Compared with other state-of-the-art methods, our method achieves a higher computational efficiency with fewer parameters while maintaining a decent detection performance, indicating the feasibility of MLP. Furthermore, the combination between CNN and MLP in our method is an intuitive structure and there are still many potential improvements. In the future, we will construct the multilevel MLPs and try to extend a fixed input scale both in the training and inference stages to flexible input scales.

References

[1] H. Jung, J. Min, and J. Kim, “An efficient lane detection algorithm for lane departure detection,” in *Proc. IEEE Intell. Veh. Symp.*, 2013, pp. 976–981.

[2] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, “A review of recent advances in lane detection and departure warning system,” *Pattern Recognit.*, vol. 73, pp. 216–234, Jan. 2018.

[3] T. Sun, S. Tsai, and V. Chan, “HSI color model-based lane-marking detection,” in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 1168–1172.

[4] J. Niu, J. Lu, M. Xu, P. Lv, and X. Zhao, “Robust lane detection using two-stage feature extraction with curve fitting,” *Pattern Recognit.*, vol. 59, pp. 225–233, Nov. 2016.

[5] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial as deep: Spatial CNN for traffic scene understanding,” in *Proc. 32nd Assoc. Advance. Artif. Intell. Conf. Artif. Intell.*, 2018, pp. 7276–7283.

[6] D. Xiao, L. Zhuo, J. Li, and J. Li, “Structure-prior deep neural network for lane detection,” *J. Vis. Commun. Image Representation*, vol. 81, Nov. 2021, Art. no. 103373.

[7] T. Zheng et al., “RESA: Recurrent feature-shift aggregator for lane detection,” in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3547–3554.

[8] D. Xiao, X. Yang, J. Li, and M. Islam, “Attention deep neural network for lane marking detection,” *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105584.

[9] L. Tabelini, R. Berriel, T. M. Paix~ao, C. Badue, A. F. De Souza, and T. Olivera-Santos, “Keep your eyes on the lane: Attention-guided lane detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 294–302.

[10] J. Gao, J. Yi, and Y. L. Murphey, “A lane-changing detection model using span-based transformer,” in *Proc. 33rd Chin. Control Decis. Conf.*, 2021, pp. 2733–2738.

[11] J. Han et al., “Laneformer: Object-aware row-column transformers for lane detection,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 799–807.

[12] Z. Guo, Y. Huang, H. Wei, C. Zhang, B. Zhao, and Z. Shao, “DALaneNet: A dual attention instance segmentation network for real-time lane detection,” *IEEE Sensors J.*, vol. 21, no. 19, pp. 21730–21739, Oct. 2021.

[13] J. Fu et al., “Dual attention network for scene segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3141–3149.

[14] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 510–519.

[15] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7132–7141.

[16] Y. Zhang, H. Liu, and Q. Hu, "TransFuse: Fusing transformers and CNNs for medical image segmentation," in *Proc. Med. Image Comput. Comput. Assist. Intervention 24th Int. Conf.*, 2021, pp. 14–24.

[17] P. Wang et al., "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2018, pp. 1451–1460.

[18] J. M. J. Valanarasu and V. M. Patel, "UNeXt: MLP-based rapid medical image segmentation network," in *Proc. 25th Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, 2022, pp. 23–33.

[19] D. Lian, Z. Yu, X. Sun, and S. Gao, "AS-MLP: An axial shifted MLP architecture for vision," 2021, *arXiv:2107.08391*.

[20] X. Ding, C. Xia, Z. X., X. Chu, J. Han, and G. Ding, "RepMLP: Re-parameterizing convolutions into fully-connected layers for image recognition," 2022, *arXiv:2105.01883*.

[21] X. Yan and Y. Li, "A method of lane edge detection based on Canny algorithm," in *Proc. Chin. Automat. Congr.*, 2017, pp. 2120–2124.

[22] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 309–318, Dec. 2004.

[23] D. C. Hernández, D. Seo, and K. Jo, "Robust lane marking detection based on multi-feature fusion," in *Proc. 9th Int. Conf. Hum. Syst. Interact.*, 2016, pp. 423–428.

[24] H. Cheng, B. Jeng, P. Tseng, and K. Fan, "Lane detection with moving vehicles in the traffic scenes," *Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 571–582, Dec. 2006.

[25] Y. Lin, S. L. Pintea, and J. Gemert, "Semi-supervised lane detection with deep Hough transform," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 1514–1518.

[26] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, Mar. 2008.

[27] Z.-Q. Li, H.-M. Ma, and Z.-Y. Liu, "Road lane detection with Gabor filters," in *Proc. Int. Conf. Inf. Syst. Artif. Intell.*, 2016, pp. 436–440.

[28] W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang, "Lane detection and classification for forward collision warning system based on stereo vision," *IEEE Sensors J.*, vol. 18, no. 12, pp. 5151–5163, Jun. 2018.

[29] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 286–291.

[30] Y. Zhang, Z. Lu, D. Ma, J.-H. Xue, and Q. Liao, "Ripple-GAN: Lane line detection with ripple lane line detection network and Wasserstein GAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1532–1542, Mar. 2021.

[31] Z. Liu and L. Zhu, "Label-guided attention distillation for lane segmentation," *Neurocomputing*, vol. 438, pp. 312–322, May 2021.

[32] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection cnns by self-attention distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1013–1021.

[33] Q. Wang, T. Han, Z. Qin, J. Gao, and X. Li, "Multitask attention network for lane detection and fitting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1066–1078, Mar. 2022.

[34] Y. Ko, Y. Lee, S. Azam, F. Munir, M. Jeon, and W. Pedrycz, "Key points estimation and point instance segmentation approach for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8949–8958, Jul. 2022.

[35] Z. Qin, H. Wang, and X. Li, "Ultra-fast structure aware deep lane detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, vol. 12369, pp. 276–291.

[36] S. Yoo et al., "End-to-end lane marker detection via row-wise classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1006–1007.

[37] L. Liu, X. Chen, S. Zhu, and P. Tan, "CondLaneNet: A top-to-down lane detection framework based on conditional convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 3753–3762.

[38] Y. Son, E. S. Lee, and D. Kum, "Robust multi-lane detection and tracking using adaptive threshold and lane classification," *Mach. Vis. Appl.*, vol. 30, no. 1, pp. 111–124, 2019.

[39] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool, "End-to-end lane detection through differentiable least-squares fitting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 905–913.

[40] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. D. Souza, and T. Oliveira-Santos, "Polylanenet: Lane estimation via deep polynomial regression," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 6150–6156.

[41] R. Liu, Z. Yuan, T. Liu, and Z. Xiong, "End-to-end lane shape prediction with transformers," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 3694–3702.

[42] Y. Sun, L. Wang, Y. Chen, and M. Liu, "Accurate lane detection with Atrous convolution and spatial pyramid pooling for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 642–647.

[43] J. Li et al., "Lane-DeepLab: Lane semantic segmentation in automatic driving scenarios for high-definition maps," *Neurocomputing*, vol. 465, pp. 15–25, Nov. 2021.

[44] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[45] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 1971–1980.

[46] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6000–6010.

[47] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[48] H. Ma et al., "TransFusion: Cross-view fusion with transformer for 3D human pose estimation," 2021, *arXiv:2110.09554*.

[49] K. Kim et al., "Rethinking the self-attention in vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2021, pp. 3065–3069.

[50] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2021, *arXiv:2010.11929*.

[51] L. Chen et al., "PersFormer: 3D lane detection via perspective transformer and the OpenLane benchmark," in *Proc.17th Eur. Conf. Comput. Vis.*, 2022, pp. 550–567.

[52] T. Liu, Z. Chen, Y. Yang, Z. Wu, and H. Li, "Lane detection in low-light conditions using an efficient data enhancement: Light conditions style transfer," in *Proc. IEEE Intell. Veh. Symp.*, 2020, pp. 1394–1399.

[53] H. Zhang, J. Xie, J. Qian, and J. Yang, "Guided dual network based transfer with an embedded loss for lane detection in nighttime scene," in *Proc. IEEE 6th Int. Conf. Comput. Commun.*, 2020, pp. 1219–1223.

[54] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," 2016, *arXiv:1606.02147*.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[56] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[57] Z. Yu et al., "Multi-modal face anti-spoofing based on central difference networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 2766–2774.

[58] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[59] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[60] J. Yang and J. Jiang, "Dilated-CBAM: An efficient attention network with dilated convolution," in *Proc. IEEE Int. Conf. Unmanned Syst.*, 2021, pp. 11–15.

[61] I. O. Tolstikhin et al., "MLP-mixer: An all-MLP architecture for vision," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 24261–24272, 2021.

[62] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," 2016, *arXiv:1606.08415*.

[63] F. Milletari, N. Navab, and S. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

[64] TuSimple, "TuSimple benchmark," 2017. Accessed: Sep. 2020. [Online]. Available: https://github.com/TuSimple/tusimple-benchmark/

[65] A. Paszke et al., "Automatic differentiation in PyTorch," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1–4.

[66] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist. Paris*, 2010, pp. 177–186.

[67] Y. B. Liu, M. Zeng, and Q. H. Meng, "Heatmap-based vanishing point boosts lane detection," 2020, *arXiv:2007.15602*.

[68] Z. Chen, Q. Liu, and C. Lian, "PointLaneNet: Efficient end-to-end CNNs for accurate real-time lane detection," in *Proc. IEEE Intell. Veh. Symp.*, 2019, pp. 2563–2568.

[69] T. Zheng et al., "CLRNet: Cross layer refinement network for lane detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 888–897.

Xuedong Yao received the M.S. degree in environmental science from the School of Resources and Environmental Engineering, Anhui University, Hefei, China, in 2020. He is currently working toward the Ph.D. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. His research interests include lane detection, remote sensing image information extraction, and point clouds semantic segmentation.

Yandong Wang is currently a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. His research interests include indoor and outdoor 3D reconstruction, multi-source image information extraction, semantic segmentation, and spatio-temporal information mining and analysis. He is the Editor of *International Journal on Advances in Software*.

Yanlan Wu received the Ph.D. degree in cartography and geographic information system from Wuhan University, Wuhan, China, in 2004. She is currently a Professor with the School of Resources and Environmental Engineering, Anhui University, Hefei, China. Her research interests include deep learning, remote sensing image processing, and information extraction.

Guoxiong He is currently working toward the master's degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. His main research interests include crowd-source data mining and object detection and tracking.

Shuchang Luo is currently working toward the M.S. degree with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China. Her main research interests include lane detection and deep learning.