Pattern Recognition 111 (2021) 107623

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Rapid and Brief Communication

A review of lane detection methods based on deep learning

Jigang Tang[a,b], Songbin Li[a,b,*], Peng Liu[a,b]

[a] Institute of Acoustics, Chinese Academy of Sciences, 100190, China
[b] University of Chinese Academy of Sciences, Beijing, 100049, China

ARTICLE INFO

Article history:
Received 11 December 2019
Revised 13 June 2020
Accepted 29 August 2020
Available online 15 September 2020

Keywords:
Lane detection
Deep learning
Semantic segmentation
Instance segmentation

ABSTRACT

Lane detection is an application of environmental perception, which aims to detect lane areas or lane lines by camera or lidar. In recent years, gratifying progress has been made in detection accuracy. To the best of our knowledge, this paper is the first attempt to make a comprehensive review of vision-based lane detection methods. First, we introduce the background of lane detection, including traditional lane detection methods and related deep learning methods. Second, we group the existing lane detection methods into two categories: two-step and one-step methods. Around the above summary, we introduce lane detection methods from the following two perspectives: (1) network architectures, including classification and object detection-based methods, end-to-end image-segmentation based methods, and some optimization strategies; (2) related loss functions. For each method, its contributions and weaknesses are introduced. Then, a brief comparison of representative methods is presented. Finally, we conclude this survey with some current challenges, such as expensive computation and the lack of generalization. And we point out some directions to be further explored in the future, that is, semi-supervised learning, meta-learning and neural architecture search, etc.

1. Introduction

With the development of intelligent transportation, environment perception, as an essential task for autonomous driving, has become a research hotspot. Lane detection is an important part of environmental perception. Many efforts have been done during the last decades. However, it is still a challenge to develop a robust detector under unlimited conditions. Because there are too many variables, such as fog, rain, illumination variation, and partial occlusion. They may have effects on the final results.

Pre-processing steps play an important role in heuristic recognition-based lane detection methods. To remove unwanted noise, many filters are used, including mean, median [1], Gaussian [2], and FIR [3] filters. To deal with illumination variation, the general solutions employ threshold segmentation [4] algorithm, including Otsu [5], and PLSF [6], etc. The region of interest (ROI) is usually used to reduce redundant information. Fixed-size ROI [7], vanishing point-based ROI [2] and adaptive ROI [8] have been widely explored. Color is another information for pre-processing. Color space conversion between RGB and YCbCr or HLS is generally used to enhance the quality of lane mark.

Feature extraction and lane modeling are critical for obtaining mathematical description of lanes. Many algorithms including Sobel [7], Canny [9], FIR filter [10], and Hough transform [11] are applied to extract feature. Many algorithms model lanes as straight lines. For modeling curves, parabolic [12], Catmull-Rom spline [13], cubic B-spline [2], and clothoid curve [14] are used. In complex conditions, inverse perspective transformation [15], image enhancement [16], stereo camera [17], and wavelet analysis [18] are used.

In 2012, convolutional neural networks (CNN) AlexNet [19] won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). Since then, deep learning algorithms became a promising tool. Over the past several years, via multi-layer nonlinear transforms, deep learning has achieved promising results in many fields. A variety of deep learning methods have been applied to tackle the lane detection task. Range from early CNN-based method (e.g., [20,21]) to end-to-end segmentation-based methods (e.g., GCN [22], SCNN [23]), GAN-based method (e.g., EL-GAN [24]), et al. In addition, knowledge distillation [25], attention map [26] have brought new ideas for lane detection (e.g., SAD [27]). How to understand the structure of lane lines from the perspective of directed acyclic graphica, DAGMapper [28] gives a good explanation. Although promising results have been achieved, the lack of generalization ability is still a main challenge of existing methods. A

* Corresponding author.
E-mail address: lisongbin@mail.ioa.ac.cn (S. Li).

https://doi.org/10.1016/j.patcog.2020.107623
0031-3203/

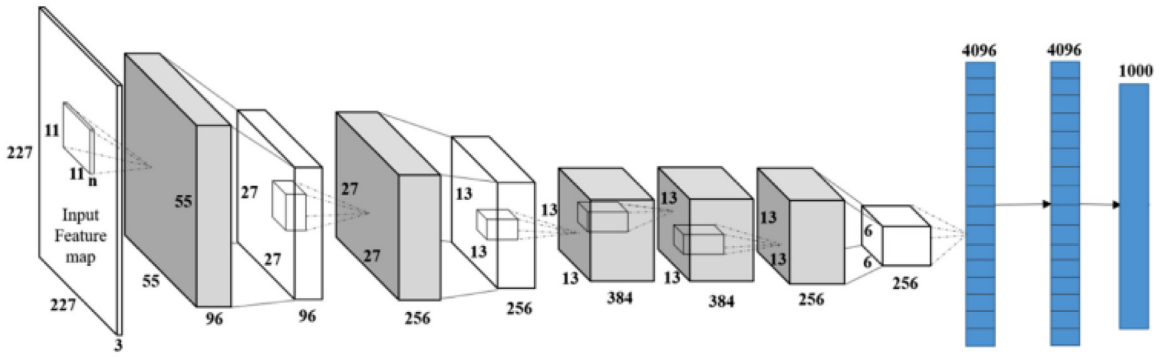

Fig. 1. Architecture of a classfication CNN. It consists of eight convolution layers and three fully connected layers. The output is 1000 categories probability. Picture from [36].

CNN trained in one scenario may perform less accurate in other, especially at night.

To the best of our knowledge, this paper is the first article to make a comprehensive review of recent deep learning-based lane detection algorithms. The rest of the survey is arranged as follows. Section 2 describes the background of relative CNN. Section 3 describes CNN architectures, loss functions, pre-processing, and post-processing of deep learning-based lane detection algorithms. In Section 4, we do some experiments to demonstrate four state-of-the-art and representative algorithms. The conclusion and future work are discussed in Section 5.

2. Background of related convolutional neural networks

Vision-based lane detection task, as an application of computer vision, can be defined as image classification, object detection or semantic segmentation. CNN has revealed its powerful effects in a wide range of computer vision tasks. Due to the close link between lane detection and fundamental vision task, it is necessary to introduce the background of related convolutional neural networks.

2.1. Image classification

In 2012, AlexNet won ILSVRC with five convolution layers and three fully connected layers, which essentially extends the depth of LeNet [29] and applies some techniques such as ReLU [30] and Dropout [31]. The structure of AlexNet is relatively simple but demonstrated the remarkable success of CNN.

It is proved that the solution space of CNN can be expanded by increasing its depth or its width [14]. Following AlexNet [19], GoogLeNet [32], VGG [33] got higher accuracy with deeper and wider architectures in ILSVRC2014. VGG increased the depth to 16–19 layers, GoogLeNet increased both depth (22 layers) and width. GoogLeNet is also named Inception-v1. It has evolved from Inception-v1 to Inception-v4 [34] with different optimizations. The generalization performance of VGG is better, and it is often used to extract image features in many fields. Deeper CNN may cause the problem of exploding or vanishing gradient. The short-cut connections from ResNet [35] make the training possible. The architecture of a single pipeline CNN can be seen in Fig. 1.

2.2. Object detection

We can group the existing deep learning-based detectors into two categories: two-stage and one-stage methods. Two-stage methods including R-CNN [37], Fast R-CNN [38], Faster R-CNN [39], CoupleNet [40], and Light-Head R-CNN [41], etc., which first generate candidate regions by CNN or traditional methods, then classify them into a category. One-stage methods including YOLO [42], G-CNN [43], SSD [44], DSDD [45], and RON [46], etc. They can directly generate the category probability and position coordinate value without region proposal stage. Major methods of deep learning-based object detection are shown in Fig. 2.

2.3. Semantic segmentation

Semantic segmentation is another fundamental task of computer vision, which aims to classify every pixel into a category. In 2015, Jonathan Long et al. proposed Fully Convolutional Networks (FCN) [48]. Following FCN, encoder-to-decoder architecture is widely used to address image segmentation issues. An encoder-decoder CNN architecture is shown in Fig. 3. To fuse different contextual information, GCN [49] used large convolution kernels, PSP-Net [50] proposed a pyramid pooling module, Deeplab series [51] adopt dilated spatial pyramid pooling, EncNet [52] introduced a channel attention method. For real-time segmentation, ENet [53] proposed bottleneck module, ERFNet [54] used residual connections and factorized convolutions, EDANet [55] designed efficient dense modules and discarded deconvolution layers in order to remain efficient while retaining remarkable accuracy. In addition, PSANet [56] captures pixel-wise relation by a convolution layer. Readers can refer to survey [57] for more comprehensive review.

3. Deep learning for lane detection

We can group the existing lane detection methods into two categories: two-step and one-step methods. Two-step methods are composed of feature extracting step and post-processing step. To be specific, feature extracting includes heuristic recognition-based feature extracting and deep learning-based feature extracting. Post-processing mainly contains clustering and fitting. One-step methods can get the detection and clustering results directly from the input image. Hence, there is no need for clustering and we summarize those methods as one-step methods.

Around the above summary, we will discuss the existing deep learning-based lane detection algorithms from two perspectives: network architectures and loss functions. Post-processing is another critical part of two-step algorithm, which will be introduced in Section 3.3 together with pre-processing.

3.1. Network architecture

As a specific application, there are many strategies for lane detection. From the perspective of how to define lane detection task, it can be summarized as the following three categories:

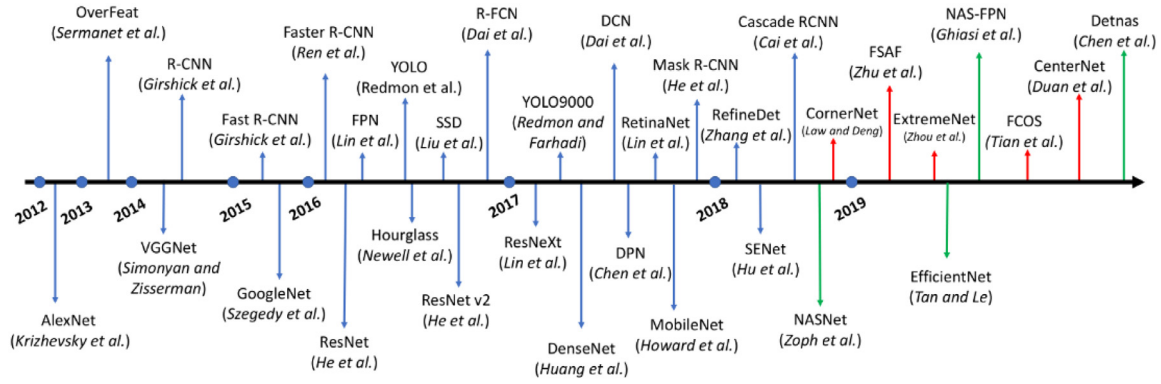

Fig. 2. Major methods of deep learning-based object detection. Anchor-free (in red) and AutoML (in green) techniques have becoming two important research directions. Picture from [47].

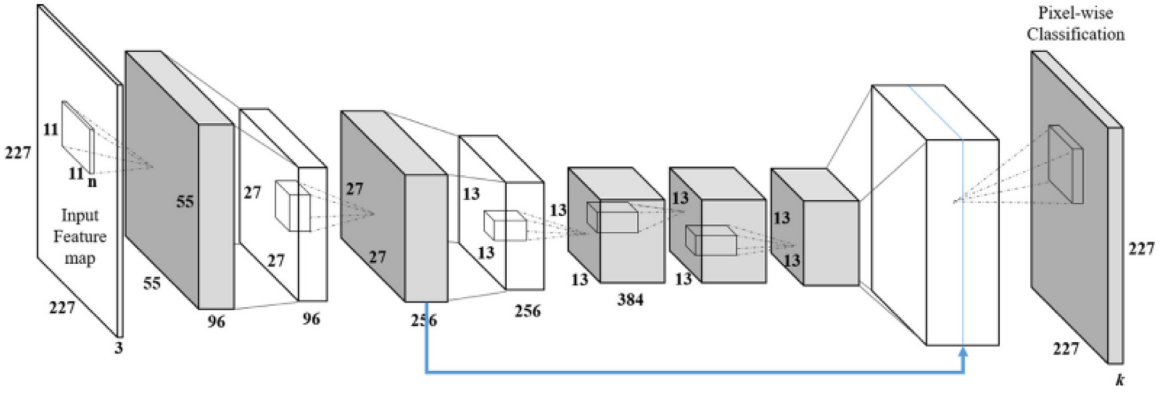

Fig. 3. Sketch of an encoder-decoder CNN. Convolution with pooling constitute encoder section, transposed convolution consitute decoder section. Skip connection between encoder and decoder is usually used for construct-preserving. Picture from [36].

1) classification-based methods, which combine some prior information to determine lanes position; 2) object detection-based method. By labeling regression bounding boxes or feature points for each lane segment, lanes can be detected by coordinate regression; 3) segmentation-based method. Lanes and background pixels are labeled as different classes. And the detection results can be obtained in the form of pixel-level classification (semantic segmentation/instance segmentation). From the perspective of model structure, it can be summarized as the following two types: 1) single-task model. Only lane detection is considered, and other road signs are not involved; 2) multi-task model, which combine lane detection with other tasks, such as drivable area detection, road marking recognition, road type, or lane type classification. In practice, many available structures and ideas can be extracted from existing CNN, such as the feature extractor VGG, ResNet, and the end-to-end architecture of FCN, etc.

3.1.1. Benchmark of CNN based method for lane detection

In 2014, Jiun Kim and Minho Lee [20] proposed a detector, where CNN was first used to extract land features and random sample consensus (RANSAC) was used to clustering. The CNN architecture is shown in Fig. 4, which consist of 8 layers with 3 convolutional, 2 subsampling, and 3 fully-connected layers. The training dataset is composed of images after ROI selection and edge detection. The last fully-connected layer output the predicted image (10,015), where predicted pixels of lanes are denoted as white.

Though the CNN sketch of [20] is relatively simple, it can be seen as an approximation of the complex mapping between the input and output spaces. Despite the advancement of this method compared to traditional ones, the following problems exist: (1) the pre-processed input of this method leads to complicated data processing. Therefore, can we cancel the pre-processing steps and train CNN in a more concise way? (2) this method has an eight-layer architecture, thence, can more complicated CNN architectures (with different depths, widths, and topologies) achieve better results?

3.1.2. Classification and object detection based methods

A *Classification based lane detection methods*

The application of image classification generally aims to discriminate what object is contained in the input. To all appearances, this way cannot obtain the location of the lanes. Some tricks need to be used between classification and lane detection. We assume that some location-dependent prior knowledge is known, which is denoted as $pk(p)$. CNN, as a mapping function $f(x)$, can be combined with $pk(p)$ to form a new formulation $o = f(x, pk(p))$. DeepLane [21] is a method based on this idea. The overall CNN architecture is shown in Fig. 5. In detail, the training dataset consisted of images (resolution: 240 × 360) from laterally-mounted down-facing cameras. To obtain a probability distribution of lane positions, softmax function is applied to the output of the last fully connected layer (317 outputs: 316 possible classes for lane positions and one class for the absence of lane marker). Hence, the CNN output a vector $Y_i = (y_0, ..., y_{316})$. To locate the lane marking, the estimated position e_i is defined as Eq. (3.1).

$$e_i = argmax(y_i), 0 \le i \le 316 \tag{3.1}$$

DeepLane got a better result by a more complex network (normalization layer, dropout layer) than [20]. However, the prior position setting limits its application scenario. A more general ap-

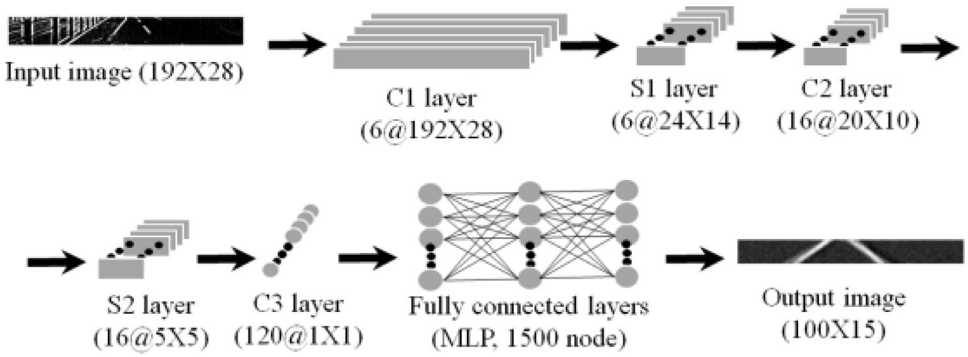

Fig. 4. CNN architecture in [20]. The detector consists of three convolution, two subsampling and three fully connected layers. The output 1500 nodes of last fully connected layer are reshaped to 100 × 15, which is considered as the predicted map.

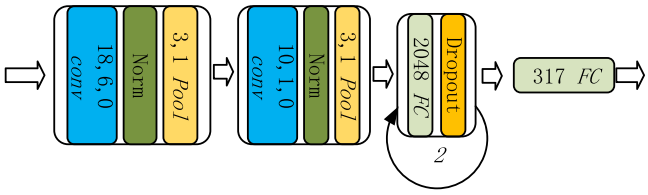

Fig. 5. CNN sketch of Deeplane [21]. The last fully connected layer output 317 numbers that represent the position probability of lanes. Similar to Fig. 4, the backbone consists of convolution, pooling and three fully connected layers.

proach is needed. Furthermore, classification, as a high-level task, does not match well with lane detection. Regressing lane coordinates in the form of object detection is also a feasible idea to achieve lane detection.

A *Object detection based lane detection methods*

Autonomous driving requires a variety of visual perception, including lane detection, vehicle recognition, and road marking recognition, etc. We believe that it is necessary to emphasize the importance of multi-supervised training in detection. In 2013, Sermanet et al. proposed OverFeat [58], which indicated that in some cases, training a network to do classification, location, and detection simultaneously can improve the accuracy of all three tasks.

For object detection, two main goals were pursued: (1) predicting categories of objects in an image; (2) determining the position of detected objects. In 2015, Brody Huval et al. made an empirical evaluation of deep learning on highway driving (EELane) [59]. In EELane the same strategy of OverFeat was used for lane and vehicle detection. The regression for vehicle class predict a five-dimension value (four for the bounding box and one for depth estimation). The lane regression predicts a six-dimension vector. The first four dimensions indicate two endpoints of a local line segment, and the remaining two dimensions indicate the depth of the endpoints with respect to the camera. The idea of estimating structural or geometric information by CNN to assist the main task has been applied in many domains, such as inpainting and edge-detection. Readers can refer to [60] for further understanding.

VPGNet [61] was proposed by Seokju Lee et al. based on the idea of VPD [62]. It is another method to estimate geometric characteristics by CNN, and consist of four branches. In VPGNet, a modified vanishing point (VP) location approach was proposed. The main improvement of VPGNet is that the VP can guide lane detection and road marking recognition. However, complicated post-processing needs expensive computation. For lane detection, point sampling, clustering, and lane regression were used during post-processing. Its network architecture is shown in Fig. 6.

The effectiveness of multiple branch strategies in VPGNet and EELane implies that branches for the different tasks can share much intermediate representation and prior knowledge can guide lane detection. To explore this idea further, in 2018, Yuhao Huang et al. proposed STLNet (Spatial-Temporal Lane detection Network) [63], which is constitutive of three steps: pre-processing, CNN based classification and regression, lane fitting. The architecture is shown in Fig. 7. In STLNet, CNN was used to classify the boundary type and regress the lane boundaries position. In terms of prior knowledge, the main difference between STLNet and VPGNet/EELane is that STLNet explored temporal constraints features. Because of the inverse perspective transformation (IPM) process during post-processing, its robustness is conditioned.

3.1.3. Image segmentation based lane detection algorithms

Shriyash Chougule et al. [64] indicated that semantic segmentation-based approaches are an inefficient way of detecting lanes. The segmentation paradigm is inherently too strict, and the emphasis is on obtaining accurate classification per pixel rather than specifying the shape. Despite these potential drawbacks, segmentation-based lane detection algorithms have achieved promising results. Many strategies have been proposed to solve the above problems. In 2005, Chiu Kuo-Yupaper et al. [65] considered lane detection as an image segmentation task. However, the traditional image segmentation algorithms (before deep learning) did not go far enough.

A *Application of existing end-to-end segmentation networks*

Bigger convolution kernels provide long-ranged information. Wenhui Zhang and Tejas Mahale [22] used GCN [49] to recognize lane areas. Carla Simulator[1] was used for generating datasets. Riera Luis et al. [66] designed a lane departure detection system by training Mask-RCNN [67] for lane detection and used a Kalman filter for lane tracking. For real-time detection, ten layers of CNN were designed by Shriyash Chougule et al. [68]. For more comprehensive recognition, the type of road or lanes is noteworthy. In [69], Pizzati Fabio et al. modified ERFNet to detect the drivable areas and the road class. DBSCAN was adopted to group the pixel-wise free space points to a set of polygons.

When applying DCNN to semantic segmentation, some inherent flaws appear, such as the design of pooling layer. Up-sampling and pooling layer (e.g. max-pooling) are not learnable. If there are five pooling layers (2 × 2 max-pooling), any object less than 32 pixels can not be reconstructed theoretically. That is, in order to obtain a large receptive field, a large amount of information is lost. The dilated convolution [70] is designed to optimize those problems. For

[1] https://carla.readthedocs.io/en/latest/.

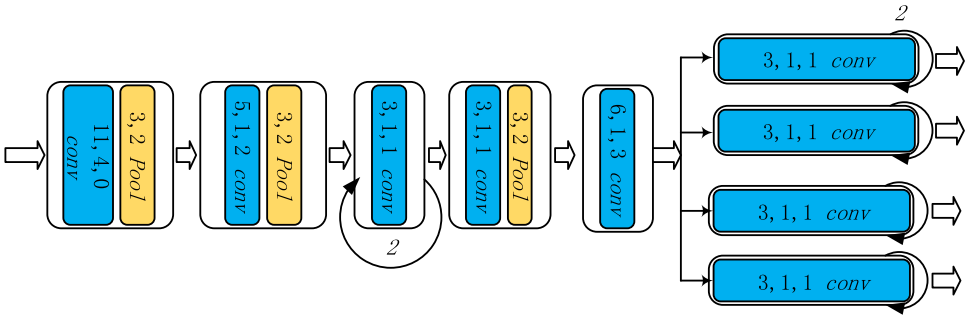

Fig. 6. Sketch of VPGNet [61]. Four branches CNN is designed for grid regression, object detection, multi-label classfication and vanishing point prediction.

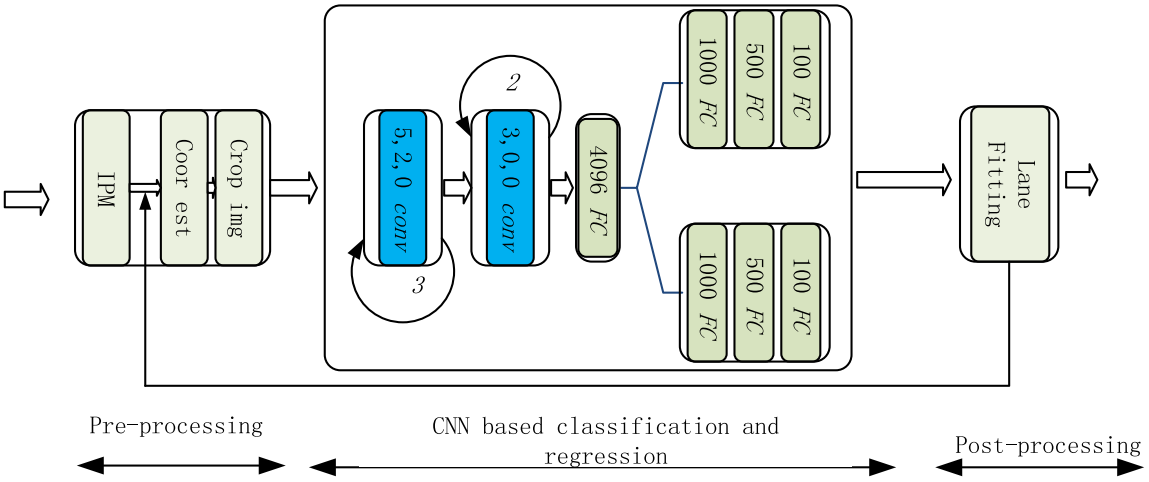

Fig. 7. Sketch of STLane.

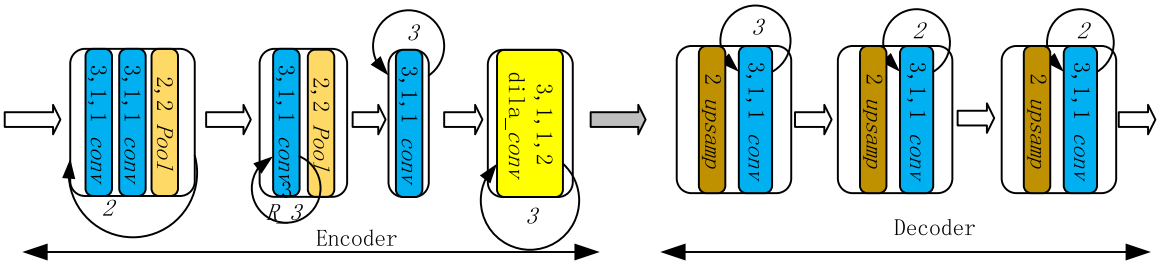

Fig. 8. Sketch of LMD [61]. Three dilated convolution layers are added between encoder and decoder.

a more detailed illustration of dilated convolution, readers can refer to [71]. The advantages of dilated convolution have been found. But how to design effective CNN structures based on dilated convolution is a new problem.

Based on VGG, Shao-Yuan Lo et al. proposed LMD (Lane Marking Detection) [72]. Three dilated convolution layers are embedded between the encoder and decoder. Its architecture can be seen in Fig. 8. The output is a predicted binary segmentation image. The predicted lane pixels are denoted as 1, and the predicted background pixels denoted as 0. In 2019, based on EDANet, Shao-Yuan Lo et al. [73] proposed another embedded dilated convolution lane detection CNN. By rethinking the relationship between downsampling operations and spatial information, FSS (Feature Size Selection) and DDB (Degressive Dilation Block) were proposed.

How to effectively obtain associated long-range information is another problem. Inspired by the non-local means [74] in classic computer vision, Wang Xiaolong et al. proposed a learnable non-local [75] operations to capture long-range dependencies. The learnable non-local long-range dependencies can be used to establish the connection between two pixels with an uncertain distance in an image, to establish the connection between two frames in a video, and to establish the connection between different words in a paragraph, etc. Non-local operation meets the needs of lane detection. In 2019, Li Wenhui et al. added non-local in IANet (Instance batch normalization and Attention Network) [76] to force CNN to focus on lane regions. Experimental results show that the mechanism is suitable for two-class segmentation scenes.

A *Combining prior knowledge with segmentation*

In terms of road geometric features, GCLNet (Geometric Constrained Network) [77] went further than VPGNet. In GCLNet, Zhang Jie et al. proposed a multiple-task framework with mutually interlinked sub-structures between lane segmentation and lane boundary detection to improve overall performance. As shown in Fig. 9, each decoder is connected to a link encoder to transfer complementary information between two tasks and thus the features of two decoders could be reciprocally refined. The sufficient exploration of the relationship between lane areas and lane boundary provide ideas for future researchers trying to adopt a multi-task strategy. Vijay John et al. proposed PsiNet [78] for detecting

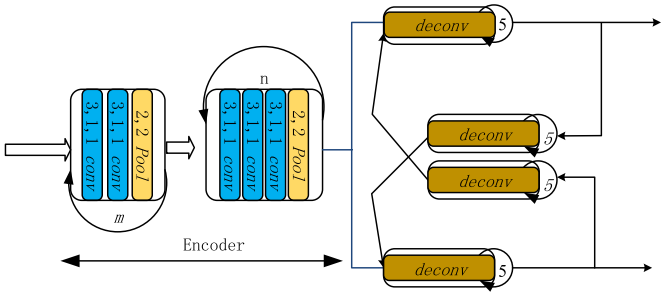

Fig. 9. Sketch of GLCNet [77]. The *m*, *n* and 5 in encoder and decoder does not indicate specific. details of network. The GLCNet emphasizes the strategy rather than feature exactor.

the free space, visible lane marker, and road scene label, which is a similar idea with GCLNet.

As mentioned in Section 3.1.2.A, in addition to spatial or geometric features, temporal correlation is another prior knowledge. During the running of the vehicle, the lanes are continuous linear structures in the captured video. Hence, lanes that cannot be accurately detected in a single frame may be inferred by combining information from previous frames. Long short-term memory (LSTM) has the ability to extract time information. The general LSTM cell at time t can be formulated as (3.2), where $\mathcal{X}_i$, $\mathcal{H}_i$, $\mathcal{C}_i$ are the in-out of lstm. i_t, o_t and f_t denote output of input gate, output gate and forget gate, respectively. W_{ij} is the weight matrix of the corresponding feature. b_i is the bias of corresponding weight matrix. The subscript t and $t-1$ denote time. '$*$' and '$\circ$' is convolution operation and Hadamard product, respectively.

$$\begin{aligned}
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c\right) \\
f_t &= \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{W}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f\right) \\
o_t &= \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{W}_{t-1} + W_{co} \circ \mathcal{C}_{t-1} + b_o\right) \\
i_t &= \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{W}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i\right) \\
\mathcal{H}_t &= o_t \circ \tanh\left(\mathcal{C}_t\right)
\end{aligned} \tag{3.2}$$

In 2019, CNN-LSTM [79] was proposed. Zou Qin et al. added two layers of LSTM between the encoder and decoder, which were shown to achieve improved performance under the occlusion scene. As shown in Fig. 10, the input of encoder is a sequence of 5 frames, and the output feature maps of encoder are used as the input of LSTM layer, where the temporal information is fused. The combination of CNN and LSTM is resource consuming. However, the benefits of this algorithm are obvious, which can get better performance under adverse weather conditions.

A *How to simplify the post-processing step*

We can summarize most of the above-mentioned methods into feature extraction algorithms, which focus on how to extract lane features more effectively, and do not consider the optimization of post-processing. Its output lane features are indistinguishable without post-processing. For all practical purposes, how to design effective strategies to get good performance through a concise network has always been the research direction. In this section, we pay more attention to the design of strategies rather than specific CNN structure.

Let us rethink the lane detection task. The fundamental purpose is to detect the position of each lane. Therefore, there are two possible types of algorithm output: points and lines. The question is how to assign different categories to different lanes without post-processing. We can group the existing solutions into three types: 1) multi-branch CNN structure. Each branch detects one lane line; 2) multi-class semantic segmentation. The output of each lane line is labeled as a different class; 3) instance segmentation. Each lane line is considered as a separate instance.

CooNet (Coordinate Network) [64] is the algorithm of multi-branch structure, proposed by Shriyash Chougule et al., which define lane detection as a coordinate regression task. The output 4 30 1 vectors denote coordinates of four-lane lines. As shown in Fig. 11, this strategy does not require any clustering process.

Xingang Pan, Jianping Shi et al. proposed Spatial CNN (SCNN) [23], which is an algorithm of multi-class semantic segmentation. The lane lines are labeled into four classes (labelled as 1, 2, 3, 4). Fig. 12 shows the architecture of SCNN, where a lane line discrimination branch is designed.

There are two ways to achieve instance segmentation: top-to-down, object detection-based methods, such as Mask R-CNN. And down-to-top, semantic segmentation-based methods, such as Deep Clustering [80].

LaneNet [81] is an instance segmentation-based lane detection method, which is an application of algorithm [80]. Sketch of LaneNet is shown in Fig. 13. Readers can refer to [53] for detailed description of ENet. There are two decoders. The top one is used for segmentation and the bottom one is used for instance embedding. The learnable clustering algorithm is the greatest contribution of [80], which allows LaneNet to cluster each lane line during training rather than post-processing. A shortcoming of SCNN and CooNet is only up to 4 lane lines can be detected. But LaneNet has no such problem.

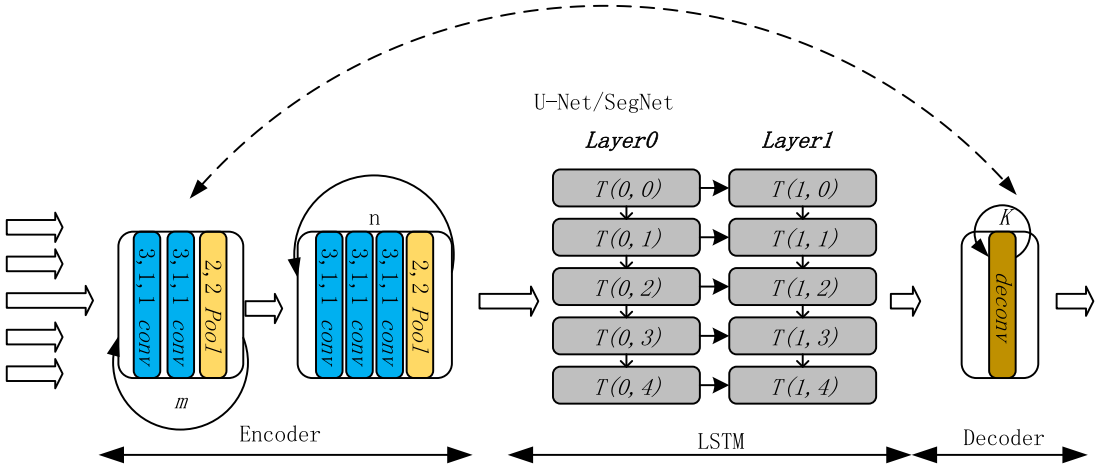

Fig. 10. Sketch of CNN-LSTM [79]. The Figure does not show the details of the convolution layer of encoder and decoder, which can be U-Net or SegNet.

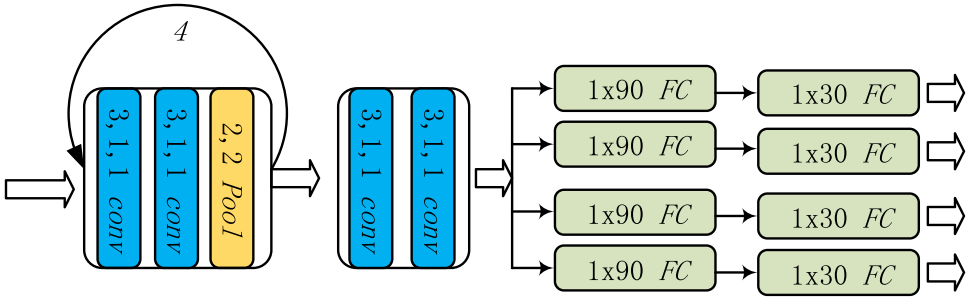

Fig. 11. Sketch of CooNet [64]. The maximum of four lane lines can be detected. The 1 × 30 output vector denote coordinates of 15 points.

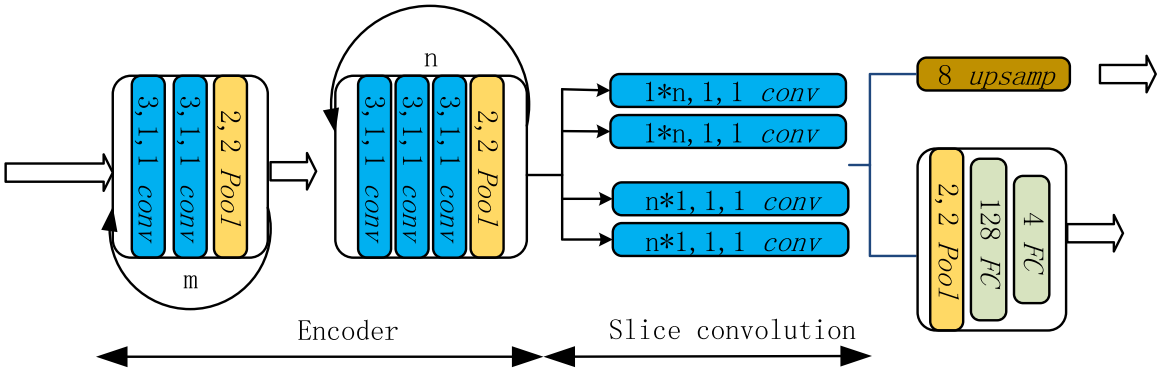

Fig. 12. Sketch of SCNN [23]. The encoder can be VGG or other effevtive feature exactor. We can consider the slice-to-slice convolution as four 1 × n and n × 1 convolutions.

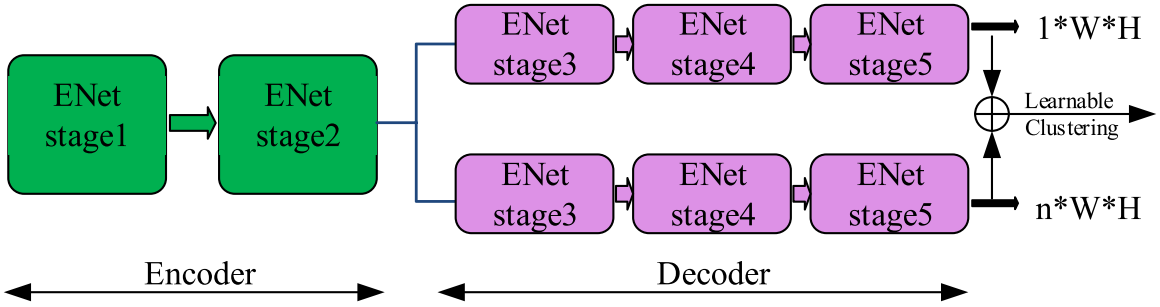

Fig. 13. Sketch of LaneNet [81]. This method is based on [80], the two branches are same with ENet.

3.1.4. Some optimization strategies of deep learning-based lane detection methods

How to remove the post-processing steps? How to get good performance with a small dataset? Despite what has been achieved, there are still many problems that need to be solved. These problems also exists in the application of deep learning in computer vision.

A *Good performance with a small dataset*

The effectiveness of the pre-training strategy implies that models can share many intermediate representations in different datasets. How to use a trained model to help the training of new models? We summarize the feasible methods used in lane detection into the following two types: transfer learning and knowledge distillation methods.

We can divide the dataset for transfer learning into two categories: source dataset and target dataset. The source dataset refers to additional data and is not directly related to the task. The target dataset is directly related to the task. When both target data and source data are labeled, fine-tuning is a common method to deal with this problem. In order to achieve a good transfer performance, there is a wide discussion on which layers to fix and which layers to train [82]. In 2015, knowledge distillation (KD) [25] was proposed by Hinton et al., which used a well-performance network (teacher network) to guide the training of small parameter networks (student network) for improving the performance of student network. Furthermore, the studies [26,83] expanded KD to attention distillation.

How to apply the above methods to lane detection? In 2017, Jiman Kim, Chanjong Park proposed TLELane (Transfer Learning for Ego Lane detection) [84], based on two transfer learning steps. The first step changes the representation domain of the network from the general scene to the road scene, the second step reduces the target from general road objects to the left and right ego lanes. In a trained segmentation-based lane detection network, the attention maps from different layers will capture rich contextual information, which informs the lanes location and rough outline. Hence, it is a feasible way to utilize the preceding block to mimic the attention maps of a deeper block. In [27], Yuenan Hou et al. proposed a self-attention distillation algorithm. Different from attention distillation, the network learns from itself (e.g., block 3 mimic block 4 and block 2 mimic block 3). Therefore, it is named self-attention-distillation based method.

A *To remove post-processing*

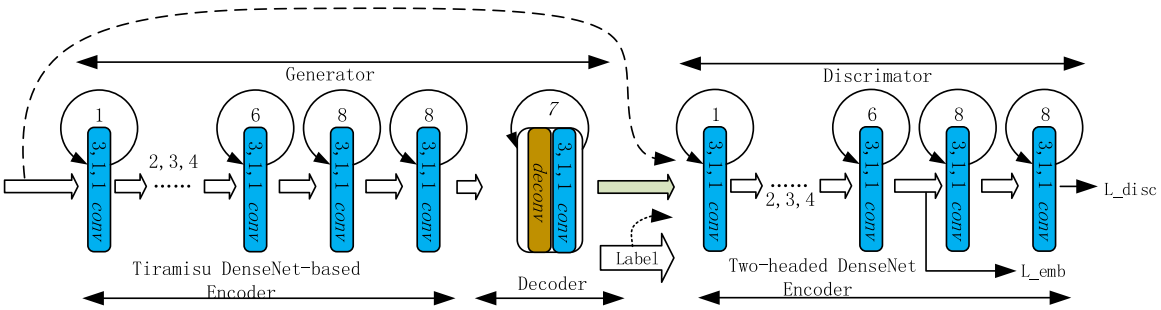

Fig. 14. Sketch of EL-GAN. The generator and discrimator are Tiramisu DenseNet-based and Two-headed DebseNet based, respectively. The L_{emb} denotes the loss of feature map, which can be considered as a perception loss.

In Section 3.1.3.C, we discussed how to simplify the post-processing, which mainly focuses on how to remove the clustering. In this section, we will discuss how to remove the post-processing, including the clustering step and fitting step. That is, the output of CNN contains the predicted lanes and the parameterized description of each detected lane.

RLaneNet (Real-time Lane Network) [85] is another attempt to fuse CNN and LSTM, where the LSTM serves as a solution to the uncertain number of lanes and a decoder to decode the parameters of each lane. The predict values of RLaneNet are the three X coordination values of the points which lies on the intersections of the lane with three horizontal lines ($Y = 0, Y = h/2,\ Y = h$). There is an assumption that lanes can be described by a three-point coordinate and quadratic function. h is the image height after IPM.

DLFNet (Differentiable Least-squares Fitting Network) [86] is a more general strategy, which estimating lane curvature parameters by solving a weighted least-squares problem in-network. The weights are generated by a deep network conditioned on the input image. A geometric loss function IS used to train the network to minimize the area between the predicted lane line and the ground-truth. The weighted least-squares fitting problem can be considered as Eq. (3.3), where ω is the pixels weighted map and X, Y is the coordinate matrix. Therefore, the parameters β of the best-fitting curve through the weighted pixel coordinates can be obtained from Eq. (3.3).

$$\omega \mathbf{X}\beta = \omega \mathbf{Y} \tag{3.3}$$

A *Better output structure-preserving*

As mentioned in Section 3.1.3.A, the emphasis of semantic segmentation is to obtain accurate classification per pixel rather than specifying the shape. In 2014, Ian J [87] proposed the generative adversarial networks (GAN) architecture. The basic principle of GAN is a game between generation network and discriminant network. The generator generates synthetic data from a given noise (generally referred to as a uniform distribution or a normal distribution). And the discriminator discriminates the output of the generator and real data. The former attempts to produce data that is closer to reality. Accordingly, the latter attempts to accurately distinguish whether the input is real or generated. Readers can refer to [88–90] for further discussions of GAN.

Based on the principle of GAN, Mohsen Ghafoorian et al. proposed EL-GAN (Embedding loss GAN) [24] for structure-preserving of lane detection. DenseNet [91,92] was used in generator and discriminator, as shown in Fig. 14. As a matter of fact, we can regard the embedding loss as perceptual loss [93] and the EL-GAN is a combination of CGAN and perceptual loss.

3.2. A summary of deep learning-based representative lane detection algorithm

In this section, we give a summary of methods, advantages and limitations of existing representative deep learning-based lane detection algorithms in Table 1. We divide those algorithms into two categories: two-step algorithms and one-step algorithms.

3.3. Loss function

As a key point of deep learning, loss functions are used to calculate the inconsistency between predicted output and ground truth and guide the model optimization. Different loss functions focus on different tasks. Therefore, a variety of loss functions are adopted to guide the lane detection task. In this section, we take a look at loss functions that are used in the lane detection field.

3.3.1. For classification

L_1 loss, L_2 loss, or Cross-Entropy loss are widely used in many tasks, such as lane line type classification and pixel-level classification. The equation of L_1 and L_2 are shown in (3.4), (3.5), where h, w, c denote the height, width, and channel numbers of an image, respectively. $\hat{I}$ and I are the predicted output and input. As shown in Eq. (3.6), the inter-class competition mechanism is adopted in Cross-Entropy loss. When $C = 2$, the Cross-Entropy loss can be defined as Eq. (3.7), where s_i denoted predicted probability of class t_i. In Eq. (3.8), different weights are given to each category. It is an effective solution for the problem of sample imbalance. For example, in lane detection, the unbalanced ratio of lane line areas and background is rather big. In [79] and [23], the weights of lane lines and background were setted as 1.0 and 0.4, respectively.

$$L_1(\hat{I}, I) = \frac{1}{hwc}\sum_{i,j,k}\left|\hat{I}_{i,j,k} - I_{i,j,k}\right| \tag{3.4}$$

$$L_2(\hat{I}, I) = \frac{1}{hwc}\sum_{i,j,k}\left(\hat{I}_{i,j,k} - I_{i,j,k}\right)^2 \tag{3.5}$$

$$L_{ce} = -\sum_{i}^{C} t_i \log(s_i) \tag{3.6}$$

$$L_{bce} = -\sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1)\log(1 - s_1) \tag{3.7}$$

$$L_{wce} = -\sum_{i}^{C} w_i t_i \log(s_i) \tag{3.8}$$

Comparing with L_1 loss, the L_2 loss is more sensitive to large errors but less sensitive to small errors. Assuming that the simplified L_2 loss function is shown in Eq. (3.9) and $\hat{y}_i = (Wx_i + b)$. Its partial derivative is shown in Eq. (3.10), where

$\sigma'(Wx_i+b)=\sigma(Wx_i+b)(1-\sigma(Wx_i+b))$ and σ is *sigmoid*. Thus, we can see that when $\sigma(Wx_i+b)$ is close to 0 or 1, $\frac{dJ}{dW}$ will be close to 0, which leads to slow convergence at the beginning of training. For Cross-Entropy loss function, its partial derivative is shown in Eq. (3.11). Therefore, in samentic segmentation or segmentation-based lane detection algorithm, Cross-Entropy loss or weighted Cross-Entropy loss is more applicable.

$$J=\frac{1}{2}\left(y_i-\hat{y}_i\right)^2 \tag{3.9}$$

$$\frac{dJ}{dW}=\left(y_i-\hat{y}_i\right)\sigma'(Wx_i+b)x_i \tag{3.10}$$

$$\frac{dL_{ce}}{dW}=[\sigma(s_i)-y_i]\cdot x_i \tag{3.11}$$

However, due to the inter-class competition mechanism, Cross-Entropy loss only cares about the accuracy of prediction probability of the correct label and ignoring the difference of other incorrect labels. This makes the features learned scattered. To overcome the drawback, from the perspective of activation function, there are L-Softmax [94] and A-Softmax [95] etc. From the perspective of loss function, [77] proposed an IoU loss for lane detection. We named it as $L_{IoU-soft}$ because it represents the relationship between predicted probability and the ground-truth. As shown in Eq. (3.12), $\mathcal{I}$ denotes the set of image pixels, y_p is the output probability of pixel p, $g=\{0,1\}^{M\times N}$ denotes the set of pixels ground-truth and ' $\times$ ' denotes pixel-wise multiplication. To increasing the intersection-over-union between the predicted lane pixels and ground-truth lane pixels, another IoU loss was proposed in [27]. We named it as $L_{IoU-hard}$ because it represents the relationship between predicted results and the ground-truth. $L_{IoU-hard}$ is defined by Eq. (3.13), where N_p is the number of predicted lane pixels, N_g is the number of ground-truth lane pixels and N_o is the number of lane pixels in the overlapped areas between predicted lane areas and ground-truth lane areas.

$$L_{IoU-soft}=1-\frac{\sum_{p\in\mathcal{I}}(y_p\times g_p)}{\sum_{p\in\mathcal{I}}(y_p+g_p-y_p\times g_p)} \tag{3.12}$$

$$L_{IoU-hard}=1-N_p/(N_p+N_g-N_o) \tag{3.13}$$

3.3.2. For regression

In [59,61] and [64], coordinate regression and grid regression were used. The mainly distance measurement method are L_1, L_2 or variations of L_1 and L_2. For example, in [64], L_{cood} is defined as Eq. (3.14), where x_{pi} and y_{pi} denote predicted coordinate of *xi* and *yi*, x_{gi} and y_{gi} are the corresponding ground truth.

$$L_{coor}=\sum_{i=1}^{15}|xp_i-xg_i|+\sum_{i=1}^{15}|yp_i-yg_i| \tag{3.14}$$

The grid regression loss is measured by L_1 or L_2, too. We take YOLO as an illustration, which is formulated as Eq. (3.15), where x_i, y_i, w_i, h_i denote the center coordinates and the width, height of ground-truth grid box. $\hat{x}_i,\hat{y}_i,\hat{w}_i,\hat{h}_i$ denote the corresponding prediction.

$$\begin{aligned}L_{coor}=&\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\right]\\&+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]\end{aligned} \tag{3.15}$$

3.3.3. For adversarial training

GAN has been introduced to a variety of computer vision tasks, which is a game between generator and discriminator. Loss functions are defined as Eq. (3.16)-(3.18), where x is ground truth data, z is input noise data. It is a variation of Cross-Entropy loss.

$$\min_G\max_D V(D,G)=E_{x\sim P(x)}[\log D(x)]+E_{z\sim Pz(z)}[\log(1-D(G(z)))] \tag{3.16}$$

$$\max_D V(D,G)=E_{x\sim Pdata(x)}[\log D(x)]+E_{z\sim Pz(z)}[\log(1-D(G(z)))] \tag{3.17}$$

$$\min_C V(D,G)=E_{z\sim Pz(z)}[\log(1-D(G(z)))] \tag{3.18}$$

In EL-GAN, the loss functions are constituted by adversarial loss L_{adv}, Cross-Entropy loss L_{cce}, and L_2 loss L_{emb}, as shown in Eq. (3.19)-(3.22). The L_{emb} loss can be seen as a perceptual loss, which compares the feature obtained by the convolution of real images with the feature obtained by the convolution of generated images, so as to make the high-level information (content and global structure) close to each other. The perceptual loss was widely used in the super-resolution field [93] for better structure-preserving.

$$L_{fit}=L_{fit}(G(x;\theta_{gen}),y)=L_{cce}(G(x;\theta_{gen}),y) \tag{3.19}$$

$$L_{cce}(\dot{y},y)=\frac{1}{wh}\sum_{i}^{wh}\sum_{j}^{c}y_{i,j}\ln\left(\dot{y}_{i,j}\right) \tag{3.20}$$

$$L_{adv}=E_{x\sim P(x)}[\log(1-D(G(x)))] \tag{3.21}$$

$$L_{emb}(\dot{y},y;x,\theta_{disc})=\|D_e(y;x,\theta_{disc})-D_e(\dot{y};x,\theta_{disc})\|_2 \tag{3.22}$$

In this section, we intriduce various loss funtions widely used in lane detection field. For a multi-branch detector, a weighted average of multiple loss functions is frequently used. There are many other loss functions in the lane detection field, however, we can consider it as some combinations of mentioned functions in this section.

3.4. Pre-processing and post-processing

3.4.1. Pre-processing

To reduce the workload of the computer and accelerate the running speed, ROI cropping was widely used in traditional lane detection algorithms and part of the deep learning-based methods. The clipped area is usually focused on the sky portion, as shown in 15. The 720 1280 3 image is cropped to 500 1280 3, which reduces computation by 30.5%.

Robustness is an important factor in applying tasks from research to practice. There existing two feasible solutions on how to improve the generalization of CNN: 1) some innovative training strategies, such as meta-learning [96]; 2) increase the diversity of the training sets. Data augmentation (including rotation, and brightness adjustment etc.) is widely used to ensure the diversity of the dataset. As shown in Fig. 15, rotating, brightness enhancement or mirroring are widely used to simulate the diversity of road.

3.4.2. Post-processing

The post-processing algorithm is necessary to transform detected points into mathematical descriptions. In the lane detection field, we can summarize post-processing algorithms into two steps: clustering and curve-fitting.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a widely used algorithm for lanes clustering. The idea is that the points in one cluster should be close to each other.

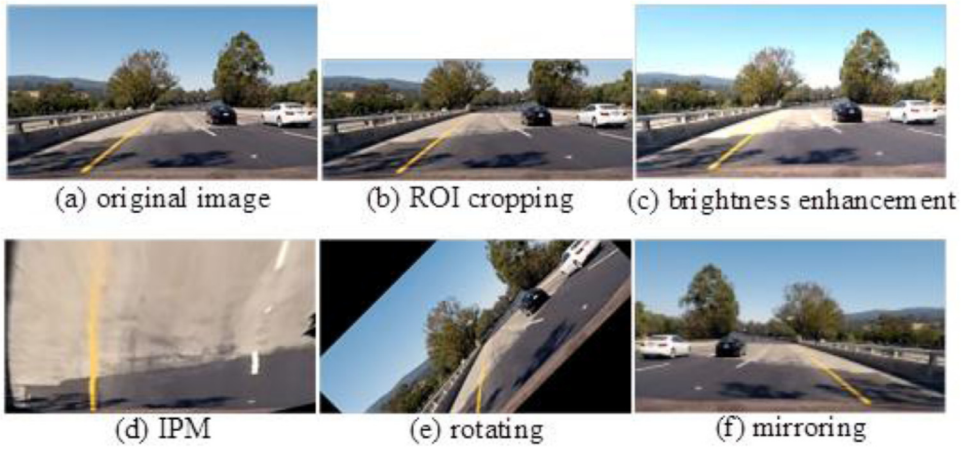

Fig. 15. pre-processing.

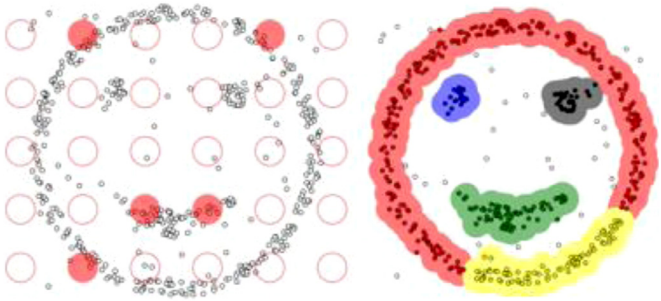

Fig. 16. An illustration of DBSCAN algorithm in a smile dataset. The dataset is clustered into five clusters when min-points = 4, epsilon = 1.

There are two key parameters in DBSCAN: min-points and epsilon. Picking an arbitrary point in our dataset as a center. If there are more than min-points within a distance of epsilon from the center (including the original point), all of those points should be considered as the same cluster. The starting point is marked as visited. Then all the points in the cluster that are not marked as visited are processed. By this way, the cluster can be extended recursively. A test on smile dataset is illustrated in Fig. 16, where min-points = 4, epsilon = 1. Readers can get more instances on the website.[2]

Clustering algorithms make lanes into different clusters. The modeling curves of each cluster of lanes are another critical step for mathematical description. As mentioned in Section 1, many functions including parabolic, Catmull-Rom spline, cubic B-spline, clothoid curve, et al. were used. An illustration of Catmull-Rom, B-spline, and polyfit is shown in Fig. 17, where $x = [1–16]$, $y = [4.00, 6.40, 8.00, 8.80, 9.22, 9.50, 9.70, 9.86, 10.00, 10.20, 10.32, 10.42, 10.50, 10.55, 10.58, 10.60]$. We can see that B-spline got the best performance.

4. Discussion and analysis

For a more intuitive comparison, we select four representative segmentation-based algorithms [23,79,81,86] for demonstration in this section. The four methods have been detailedly explained in Section 3. In addition, before 2018, TuSimple[3] was the largest lane detection dataset and many algorithms have been tested on it. Hence, the deep learning-based algorithms also tested on the TuSimple dataset in this section. In term of evaluation metrics,

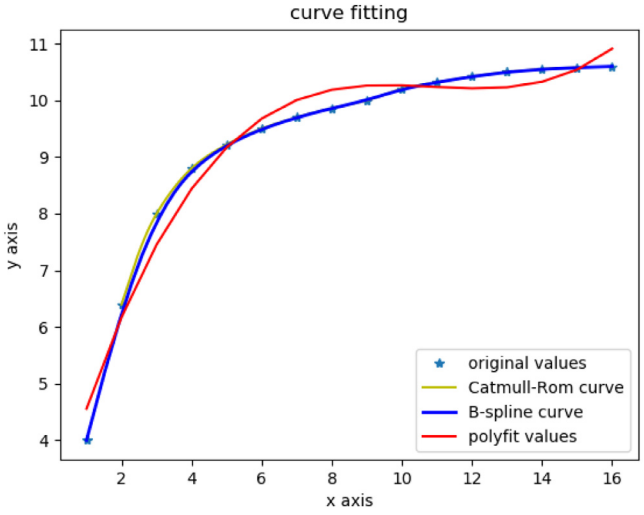

Fig. 17. An fitting results illustration of Catmull-Rom, B-spline, and polyfit.

TP (true positive), *TN* (true negative), *FP* (false positive), *FN* (false negative) are the common evaluation metrics in image processing domain. Furthermore, *FNR* (false negative rate), *FPR* (false positive rate), *ACC* (accuracy) can be calculated by Eq. (4.1). *N* represents the total pixel number of an image. The basic information of the TuSimple dataset is shown in Table 2. Three samples of the TuSimple dataset are shown in Fig. 18.

$$\begin{aligned} ACC &= (TP + TN)/N \\ FNR &= FN/(TP + FN) \\ FPR &= FP/(TN + FP) \end{aligned} \tag{4.1}$$

The general process of traditional lane detection algorithm is shown in Fig. 19. For example, the detection steps of the famous project CarND are: (1) inverse perspective transformation; (2) convert RGB image to HLS color space and extract S channel features; (3) binarize the RGB and HLS space image and fuse the two binarized image; (4) use the histogram algorithm to locate the range of lane lines, and use sliding window method combine with curve equation to fit lane lines. As shown in Fig. 20, it is obvious that when illumination changes, the traditional algorithm is difficult to get good performance .

Table 3 shows a comparison of some experimental results of segmnetation-based lane detection methods. We trained the ERFNet-DLSF [86] for ego-lane (two-class) detection. ResNet18-

[2] https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/.
[3] http://benchmark.tusimple.ai/#/t/1.

Table 1
Summary of various deep learning-based lane detection algorithms.

Authors	Method	Advantages	limitations
Two step			
Jiun Kim et al. (2014) [20]	Classification: A: 8 layers CNN. B: RANSAC.	1: Compared with the traditional method, which got a good performance.	1: The network structure is not efficient enough.
Brody Huval et al. (2015) [59]	Object detection: A: Vehicle detection and lane detection. B: Depth estimation	1:More satisfied with autopilot requirement. 2:Robustness against occlusion to some extent. 3:Used CNN to estimate geometric information.	1: Complicated data collection and annotation. 2: Repeated detection and coordinate regression.
Seokju Lee et al. (2017) [61]	Object detection: A: lane and road marking recognition. B: Vanishing point estimation.	1: Improved robustness under various conditions.	1: Post-processing take high computational complexity.
Huang et al. (2018) [63]	Object detection: A: IPM. B: Coordinate regression. C: Sub-image Extraction.	1: Used temporal and spatial constraints to reduce the range of searching area.	1: Complicated data flow and architecture. 2: When initial assumptions are not met, pre-process cannot provide valid results.
Chen, Ping-Rong et al. (2018) [72]	Segmentation: A: end-to-end. B: dilated convolution.	1: The dilated convolution isemployed to enlarge the receptive fields.	1: This is just an application of dilated convolution, the performance is not achieved SOTA.
Zhang Jie et al. (2018) [77]	Segmentation: A: end-to-end. B: multiple-task framework.	1: This work pay more attention on interlinked relationship of sub-structures.	1: Complex loss function and network structure bring many difficulties in training.
Zou Qin et al. (2019) [79]	Sementation: A: Combining CNN and LSTM. B: The input of encoder is 5 consecutive frames.	1: The exploration in temporal improve its performance under occlusion scene.	1: High computational complexity. 2: When the input images does not change much, the improved performance is conditioned.
Mohsen Ghafoorian et al. (2018) [24]	Segmentation: A: CGAN network.	1: The Embedding loss in discriminator can effectively control the output boundary closer to the label.	1: Large amount of parameters.
One step			
Gurghian et al. (2016) [21]	Classification: A: Combine prior position and classification result to estimate lane position.	1: Fast detection. 2: Simple network structure.	1: Limited application scenarios. 2: Fixed camera parameters.
Li Wenhui et al. (2019) [76]	Segmentation: A: end-to-end. B: Non-local attention. C: Instance batch normalization.	1: Verified the attention is suitable for two-class semantics segmentation task with only lane and background.	1: Non-local adds large computation.
Shriyash Chougule et al. (2018) [64]	Regression: A: multi-branch. B: Coordinate regression. C: Data augmentation.	1: This strategy does not require cluster step. 2: Lightweight networks.	1: Fixed number of lane lines can be detected.
Xingang Pan et al. (2018) [23]	Segmentation: A: end-to-end. B: slice-by-slice convolution.	1: The slice-by-slice convolution is suitable for long continuous shape structure.	1: High computational complexity.
Neven Davy et al. (2018) [81]	Segmentation: A: Instance segmentation.	1: Proposed a H–Net to estimate IPM transformation matrix. 2: Do not fixed number of lanes	1: The H–Net is not very effective.
Jiman Kim et al. (2017) [84]	Segmentation: 1: Two steps transfer learning.	1: Overcome weakpoints of small dataset.	1: Only ego lanes can be detected.
Yuenan Hou et al. (2019) [27]	Segmentation: A: Self-attention distillation.	1: The self-attention distillation strategy is efficient.	1: The complex training strategies and loss functions will make the hyperparameter adjustment difficulty.
Wang Ze et al. (2018) [85]	Regression: A: Edge proposal. B: Parameters regression.	1: The LSTM serves as a solution to the uncertain number. 2: Do not need any post-processing.	1: The ordinate of the three points to be detected is predefined.
Van Gansbeke et al. (2019) [86]	Segmentation: A: Generating coordinate weight map. B: A differentiable least-squares fitting module	1: This is a more general strategy without any predefine condition.	1: Fixed number of lane lines can be detected. 2: When added the number of weight map, the performance will be degraded

Fig. 18. Three sample images of TuSimple dataset.

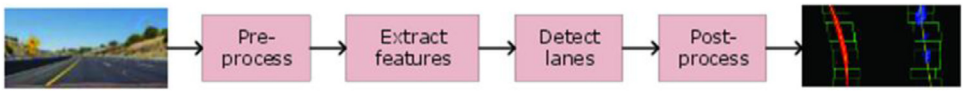

Fig. 19. Flow diagram of lane detection algorithm.

Fig. 20. Experiments of CarND under different circumstances. Line 1: ideal environment. line 2: complex conditions. From left to right: input, after IPM, after Binarization, after fited, histogram map.

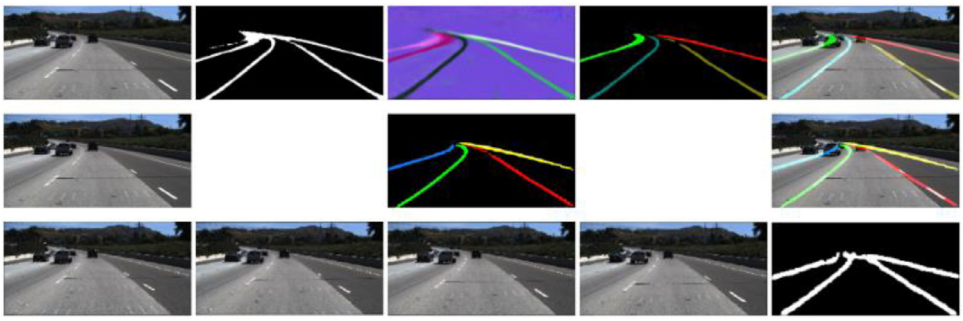

Fig. 21. Three results under curved lanes. Line 1: result of LaneNet [81]. Line 2: result of SCNN [23]. Line 3: result of ERFNet-DLSF [79].

Table 2
Basic information of TuSimple dataset.

Name	Frame	Train	Validation	Test	Resolution
TuSimple	6408	3268	358	2782	1280 × 720

Table 3
Comparisons among some representative deep models.

Paper	Method	Dataset	ACC	FPR	FNR
[35]	ResNet18-based	TuSimple	92.69	0.0948	0.0822
[35]	ResNet34-based	TuSimple	92.84	0.0918	0.0796
[53]	ENet	TuSimple	93.02	0.0886	0.0734
[86]	ERFNet-DLSF	TuSimple	93.38	0.1064	0.0983
[27]	ENet-SAD	TuSimple	96.64	0.0602	0.0205
[81]	LaneNet	TuSimple	96.38	0.0442	0.0197
[23]	S-CNN	TuSimple	96.53	0.0617	0.018
[24]	EL-GAN	TuSimple	96.39	0.0412	0.0336
[79]	CNN-LSTM (SegNet+)	TuSimple	97.30	0.0416	0.0186
[79]	CNN-LSTM (UNet+)	TuSimple	97.20	0.0424	0.0184

based and ResNet34-based methods used spatial upsampling as the decoder rather than deconvolution. The overall architecture of four selected algorithms [23,79,81,86] have been shown in Section 3. As shown in Fig. 21, Fig. 22, the experiments are conducted under curve lines and shade situation.

Middle layers feature maps reflect what the CNN extracted. For encoder-decoder structure CNN, the encoder encoded input image to multi-dimension and low-resolution feature maps. Fig. 23 shows several feature maps of [23,79,86] encoder output. We can see that SCNN extract more obvious linearity features.

5. Conclusion and future work

In this paper, we present an overall review of recent deep learning-based lane detection algorithms. There are three main contributions. First, this paper is the first overall review of recent deep learning-based lane detection algorithms, which will facilitate readers to understand how to apply deep learning to lane detection task. Second, we introduce those algorithms from CNN architectures and loss functions. This will facilitate researchers to design their own detector. Third, four representative state-of-the-art algorithms are highlighted and conducted experiments on TuSimple dataset, which will help readers to understand the best performance and optimization directions. We will clean up and open all the test code used in this work, which will further facilitate the readers from theory to application.

Thanks to the improvement of hardware computing power and GPUs. In recent years, a variety of vision-based assisted driving systems have been deployed on vehicle platforms, such as Lane Departure System and Lane Change Assist System, etc. The detection accuracy of lane detection has increased to 97% [79] on TuSimple dataset. However, there are still many underlying challenges not been overcome. There are two main challenges: (1) the lack of generalization ability. [23,27] proposed two modules that can be easily transplanted into other CNN to get a better performance. However, the supervised learning method cannot make appropriate adjustments to the situation that has not appeared in the training set; (2) it is difficult to be deployed on mobile devices. The complicated and excellent performance CNN usually accompanied by millions of parameters, this is a challenge for real-time computation on mobile devices.

Along with those deficiencies, several directions may be further explored in the future. First, semantic segmentation remains

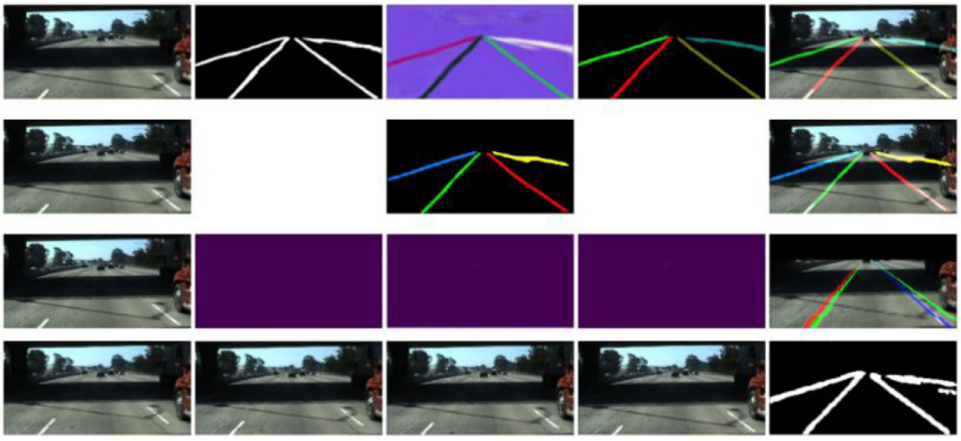

Fig. 22. Four results under shadows shade condition. Line 1: results of LaneNet [81], from left to right, semantic segmentation map, embedding map, instance segmentation map and fused image. Line 2: results of SCNN [23]. From left to right: instance segmentation map, fused image. Line 3: results of ERFNet-DLSF [86]. From left to right: weight map of left lane line, weight map of right lane line, fused weight map of the above two, predicted result. Line 4: result of CNN-LSTM [79]. From left to right: four continuous frames of input, result of semantic segmentation.

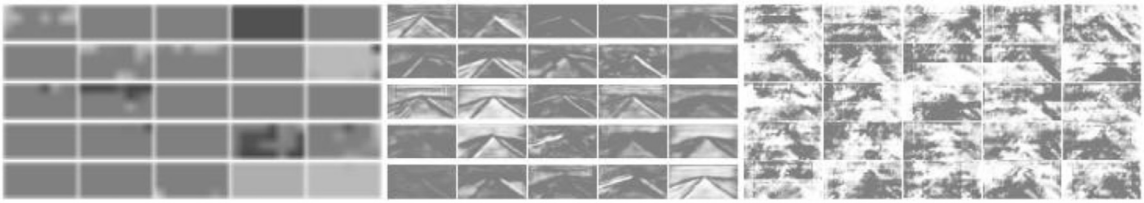

Fig. 23. From left to right: features maps of [79,23,86] encoder output, respectively.

a computationally intensive algorithm for embedded deployment. More efficient CNN architectures should be explored. Second, supervised learning requires a large amount of annotated data, and labeling data is a boring and costly task. Semi-supervised or weakly-supervised algorithms have been developed to make semi-supervised semantic segmentation possible. In addition, how to make accurate predictions under a variable environment is critical. Meta-Learning should be a viable exploration. Then, existing segmentation architectures rely on experience design, auto-machine learning may provide new ideas for more efficient feature extractor.

Declaration of Competing Interest

We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the Important Science and Technology Project of Hainan Province under Grant ZDKJ201807, in part by the Hainan Provincial Natural Science Foundation of China under Grant 618QN309, in part by the Scientific Research Foundation Project of Haikou Laboratory, Institute of Acoustics, Chinese Academy of Sciences, in part by the IACAS Young Elite Researcher Project under Grant QNYC201829 and Grant QNYC20174

References

[1] J.-.G. Wang, C.-.J. Lin, S.-.M. Chen, Applying fuzzy method to vision-based lane detection and departure warning system, Expert Syst Appl 37 (1) (2010) 113–126.
[2] P.-.Y. Hsiao, C.-.W. Yeh, S.-.S. Huang, L.-.C. Fu, A portable vision-based real-time lane departure warning system: day and night, IEEE Transactions on Vehicular Technology 58 (4) (2008) 2089–2094.
[3] X. Wang, Y. Wang, C. Wen, Robust lane detection based on gradient-pairs constraint, in: Proceedings of the 30th Chinese Control Conference, IEEE, 2011, pp. 3181–3185.
[4] J. Duan, Y. Zhang, B. Zheng, Lane line recognition algorithm based on threshold segmentation and continuity of lane line, in: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), IEEE, 2016, pp. 680–684.
[5] Y. Chai, S.J. Wei, X.C. Li, The multi-scale hough transform lane detection method based on the algorithm of otsu and canny, in: Advanced Materials Research, Vol. 1042, Trans Tech Publ, 2014, pp. 126–130.
[6] V. Gaikwad, S. Lokhande, Lane departure identification for advanced driver assistance, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2014) 910–918.
[7] C. Mu, X. Ma, Lane detection based on object segmentation and piecewise fitting, TELKOMNIKA Indones, J. Electr. Eng. TELKOMNIKA 12 (5) (2014) 3491–3500.
[8] D. Ding, C. Lee, K.-y. Lee, An adaptive road roi determination algorithm for lane detection, in: 2013 IEEE International Conference of IEEE Region 10 (TENCON 2013), IEEE, 2013, pp. 1–4.
[9] P.-.C. Wu, C.-.Y. Chang, C.H. Lin, Lane-mark extraction for automobiles under complex conditions, Pattern Recognit 47 (8) (2014) 2756–2767 29.
[10] T. Aung, M.H. Zaw, Video based lane departure warning system using hough transform, in: International Conference on Advances in Engineering and Technology, 2014, pp. 29–30.
[11] J. Niu, J. Lu, M. Xu, P. Lv, X. Zhao, Robust lane detection using two-stage feature extraction with curve fitting, Pattern Recognit 59 (2016) 225–233.
[12] J.C. McCall, M.M. Trivedi, Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation, IEEE Trans. Intelligent Transportation Systems (2006) 20–37.
[13] Y. Wang, D. Shen, E.K. Teoh, Lane detection using spline model, Pattern Recognit Lett 21 (8) (2000) 677–689.
[14] G.F. Montufar, R. Pascanu, K. Cho, Y. Bengio, On the number of linear regions of deep neural networks, in: Advances in neural information processing systems, 2014, pp. 2924–2932.
[15] M. Fu, X. Wang, H. Ma, Y. Yang, M. Wang, Multi-lanes detection based on panoramic camera, in: 11th IEEE International Conference on Control & Automation (ICCA), 520, IEEE, 2014, pp. 655–660.
[16] Y. Li, L. Chen, H. Huang, X. Li, W. Xu, L. Zheng, J. Huang, Nighttime lane markings recognition based on canny detection and hough transform, in: 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), IEEE, 2016, pp. 411–415.
[17] J.-.G. Kim, J.-.H. Yoo, J.-.C. Koo, Road and lane detection using stereo camera, in: 2018 IEEE 525 International Conference on Big Data and Smart Computing (BigComp), IEEE, 2018, pp. 649–652.

[18] D. Mingfang, W. Junzheng, L. Nan, L. Duoyang, Shadow lane robust detection by image signal local reconstruction, International Journal of Signal Processing, Image Processing and Pattern Recognition 9 (3) (2016) 89–102.
[19] I.S. Krizhevsky, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.
[20] J. Kim, M. Lee, Robust lane detection based on convolutional neural network and random sample consensus, in: International conference on neural information processing, Springer, 2014, pp. 454–461. 30.
[21] T.K. Gurghian, S.V. Bailur, K.J. Carey, V.N. Murali, Deeplanes: end-to-end lane position estimation using deep neural networksa, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 38–45.
[22] W. Zhang, T. Mahale, End to end video segmentation for driving: lane detection for autonomous car, arXiv:1812.05914, 2018.
[23] X. Pan, J. Shi, P. Luo, X. Wang, X. Tang, Spatial as deep: spatial cnn for traffic scene understanding, Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[24] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, M. Hofmann, El-gan: embedding loss driven generative adversarial networks for lane detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018 0-0.
[25] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531, 2015.
[26] S. Zagoruyko, N. Komodakis, Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer, arXiv:1612.03928, 2016.
[27] Y. Hou, Z. Ma, C. Liu, C.C. Loy, Learning lightweight lane detection cnns by self-attention distillation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1013–1021.
[28] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, R. Urtasun, Dagmapper: learning to map by discovering lane topology, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2911–2920.
[29] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.
[30] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814. 560.
[31] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv:1207.0580. 31, 2012.
[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[33] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computer Science (2014).
[34] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[35] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[36] Z. Hu, J. Tang, Z. Wang, K. Zhang, L. Zhang, Q. Sun, Deep learning for image-based cancer detection and diagnosis a survey, Pattern Recognit 83 (2018) 134–149.
[37] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object 575 detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.
[38] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
[39] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.
[40] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, Couplenet: coupling global structure with local parts for object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4126–4134.
[41] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun, Light-head r-cnn: in defense of two-stage object detector, arXiv:1711.07264, 2017.
[42] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788. 32.
[43] M. Najibi, M. Rastegari, L.S. Davis, G-cnn: an iterative grid based object detector, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2369–2377.
[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.
[45] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: deconvolutional single shot detector, arXiv:1701.06659, 2017.
[46] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, Y. Chen, Ron: reverse connection with objectness prior networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5936–5944.
[47] X. Wu, D. Sahoo, S.C. Hoi, Recent advances in deep learning for object detection, Neurocomputing (2020).
[48] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
[49] C. Peng, X. Zhang, G. Yu, G. Luo, J. Sun, Large kernel matters-improve semantic segmentation by global convolutional network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4353–4361.
[50] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.
[51] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587, 2017
[52] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, A. Agrawal, Context encoding for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7151–7160.
[53] Paszke, A.C., S. Kim, E. Culurciello, Enet: a deep neural network architecture for real-time semantic segmentation, arXiv:1606.02147. 33, 2016.
[54] E. Romera, J.M. Alvarez, L.M. Bergasa, R. Arroyo, Erfnet: efficient residual factorized convnet for real-time semantic segmentation, IEEE Transactions on Intelligent Transportation Systems 19 (1) (2017) 263–272.
[55] S.-Y. Lo, H.-M. Hang, S.-W. Chan, J.-J. Lin, Efficient dense modules of asymmetric convolution for real-time semantic segmentation, in: Proceedings of the ACM Multimedia Asia on ZZZ, 2019, pp. 1–6.
[56] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, J. Jia, Psanet: point-wise spatial attention network for scene parsing, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 267–283.
[57] Y. Guo, Y. Liu, T. Georgiou, M.S. Lew, A review of semantic segmentation using deep neural networks, Int J Multimed Inf Retr 7 (2) (2018) 87–93.
[58] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: integrated recognition, localization and detection using convolutional networks, international conference on learning representations., 2013.
[59] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, et al., An empirical evaluation of deep learning on highway driving, CoRR. (2015).
[60] O. Elharrouss, N. Almaadeed, S. Almaadeed, Y. Akbari, Image inpainting: a review, Neural Processing Letters (2019) 1–22.
[61] S. Lee, J. Kim, J. Shin Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. Seok Hong, S.-H. Han, I. So Kweon, Vpgnet: vanishing point guided network for lane and road marking detection and recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1947–1955.
[62] Borji, Vanishing point detection with convolutional neural networks, ArXiv-abs/1609.00967, 2016.
[63] Y. Huang, S. Chen, Y. Chen, Z. Jian, N. Zheng, Spatial-temproal based lane detection using deep learning, in: IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer, 2018, pp. 143–154. 34.
[64] S. Chougule, N. Koznek, A. Ismail, G. Adam, V. Narayan, M. Schulze, Reliable multilane 645 detection and classification by utilizing cnn as a regression network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018 0-0.
[65] K.-Y. Chiu, S.-F. Lin, Lane detection using color-based segmentation, in: IEEE Proceedings. Intelligent Vehicles Symposium, 2005., IEEE, 2005, pp. 706–711.
[66] L. Riera, K. Ozcan, J. Merickel, M. Rizzo, S. Sarkar, A. Sharma, Driver behavior analysis using lane departure detection under challenging conditions, arXiv:1906.00093, 2019.
[67] K. He, G. Gkioxari, P. Dollar, R. Girshick, Mask r-cnn., IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017 991-1.
[68] S. Chougule, A. Ismail, A. Soni, N. Kozonek, V. Narayan, M. Schulze, An efficient encoderdecoder cnn architecture for reliable multilane detection in real time, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 1444–1451.
[69] F. Pizzati, F. Garc´ıa, Enhanced free space detection in multiple lanes based on single cnn with scene identification, in: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2019, pp. 2536–2541.
[70] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, international conference on learning representations.,2015.
[71] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, G. Cottrell, Understanding convolution for semantic segmentation, in: 2018 IEEE winter conference on applications of computer vision (WACV), IEEE, 2018, pp. 1451–1460.
[72] P.-R. Chen, S.-Y. Lo, H.-M. Hang, S.-W. Chan, J.-J. Lin, Efficient road lane marking detection with deep learning, in: 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), IEEE, 2018, pp. 1–5.
[73] S.-Y. Lo, H.-M. Hang, S.-W. Chan, J.-J. Lin, Multi-class lane semantic segmentation using efficient convolutional networks, in: 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), IEEE, 2019, pp. 1–6. 35.
[74] B.Coll Buades, J.-M. Morel, A non-local algorithm for image denoising, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, IEEE, 2005, pp. 60–65.
[75] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7794–7803.
[76] W. Li, F. Qu, J. Liu, F. Sun, Y. Wang, A lane detection network based on ibn and attention, Multimed Tools Appl (2019) 1–14.
[77] J. Zhang, Y. Xu, B. Ni, Z. Duan, Geometric constrained joint lane segmentation and lane boundary detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 486–502.
[78] V. John, N.M. Karunakaran, C. Guo, K. Kidono, S. Mita, Free space, visible and missing lane marker estimation using the psinet and extra trees regression, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 189–194.

[79] Zou Q., Jiang H., Dai Q., Yue Y., Chen L., Wang Q., Robust lane detection from continuous driving scenes using deep neural networks, IEEE Transactions on Vehicular Technology., 2019.
[80] J.R. Hershey, Z. Chen, J. Le Roux, S. Watanabe, Deep clustering: discriminative embeddings for segmentation and separation, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016, pp. 31–35.
[81] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, L. Van Gool, Towards end-toend lane detection: an instance segmentation approach, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 286–291.
[82] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, R. Feris, Spottune: transfer learning through adaptive fine-tuning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 4805–4814.
[83] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156–3164. 36.
[84] J. Kim, C. Park, End-to-end ego lane estimation based on sequential transfer learning for self-driving cars, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 30–38.
[85] Z. Wang, W. Ren, Q. Qiu, Lanenet: real-time lane detection networks for autonomous driving, arXiv preprint arXiv:1807.01726, 2018.
[86] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, L. Van Gool, End-to-end lane detection through differentiable least-squares fitting, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019 0-0.
[87] I Goodfellow, J Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
[88] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, 2017, pp. 214–223.
[89] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, in: Advances in neural information processing systems, 2017, pp. 5767–5777.
[90] X. Mao, Q. Li, H. Xie, R.Y. Lau, Z. Wang, S. Paul Smolley, Least squares generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2794–2802.
[91] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[92] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, Y. Bengio, The one hundred layers tiramisu: fully convolutional densenets for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 11–19.
[93] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and superresolution, in: European conference on computer vision, Springer, 2016, pp. 694–711. 37.
[94] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, in: ICML, 2, 2016, p. 7.
[95] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Sphereface: deep hypersphere embedding for face recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 212–220.
[96] M. Andrychowicz, M. Denil, S. Gomez, M.W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, N. De Freitas, Learning to learn by gradient descent by gradient descent, in: Advances in neural information processing systems, 2016, pp. 3981–3989.

JIGANG TANG received the B.S. degree in control technology and instruments from China university of petroleum (Beijing), Beijing, China, in 2018. He is currently pursuing the Master's degree in Institute of acoustics, Chinese Academy of Sciences, Beijing, China. His-current research interests include computer vision and deep learning.

SONGBIN LI received the Ph.D. degree from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2010. He was a Postdoctoral Fellow and a Visiting Professor with Tsinghua University and the University of Southern California, respectively. He has been a Professor with the Institute of Acoustics, Chinese Academy of Sciences, since 2018. He has been the Principle Investigator on several projects of the National Natural Science Foundation of China. His-current research interests include machine learning, multimedia signal processing, and information forensics.

PENG LIU received the B.S. degree in communication engineering from Hainan University in 2011 and the Ph.D. degree from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2016. He has been an Associate Professor with the Chinese Academy of Sciences since 2018. His-current research interests include computer vision, multimedia signal processing, and information forensics.