

Article

Lane Line Detection Based on Object Feature Distillation

Malik Haris ^{1,*}  and Adam Glowacz ^{2,*} 

¹ School of Information Science and Technology, Xipu Campus, Southwest Jiaotong University, West Section, High Tech Zone, Chengdu 611756, China

² Department of Automatic Control and Robotics, Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Science and Technology, al. A. Mickiewicza 30, 30-059 Kraków, Poland

* Correspondence: malikharis@hotmail.com (M.H.); adglow@agh.edu.pl (A.G.)

Abstract: In order to meet the real-time requirements of the autonomous driving system, the existing method directly up-samples the encoder's output feature map to pixel-wise prediction, thus neglecting the importance of the decoder for the prediction of detail features. In order to solve this problem, this paper proposes a general lane detection framework based on object feature distillation. Firstly, a decoder with strong feature prediction ability is added to the network using direct up-sampling method. Then, in the network training stage, the prediction results generated by the decoder are regarded as soft targets through knowledge distillation technology, so that the directly up-samples branch can learn more detailed lane information and have a strong feature prediction ability for the decoder. Finally, in the stage of network inference, we only need to use the direct up-sampling branch instead of the forward calculation of the decoder, so compared with the existing model, it can improve the lane detection performance without additional cost. In order to verify the effectiveness of this framework, it is applied to many mainstream lane segmentation methods such as SCNN, DeepLabv1, ResNet, etc. Experimental results show that, under the condition of no additional complexity, the proposed method can obtain higher $F1_{Measure}$ on CuLane dataset.



Citation: Haris, M.; Glowacz, A. Lane Line Detection Based on Object Feature Distillation. *Electronics* **2021**, *10*, 1102. <https://doi.org/10.3390/electronics10091102>

Academic Editor: Valentina E. Balas

Received: 20 April 2021

Accepted: 4 May 2021

Published: 8 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: lane line detection; decoder; distillation; probability map

1. Introduction

With the increasing popularity of automobiles, the frequent occurrence of road traffic accidents and the aggravation of traffic congestion are becoming increasingly prominent. According to relevant statistics, about 1.35 million people died in traffic accidents worldwide in 2018, and tens of millions of people were injured or disabled in traffic accidents [1]. The occurrence of road traffic accidents not only causes huge economic losses to individuals, families, and even the whole country [2], but also seriously threatens people's safety. According to a statistical analysis of traffic accidents by the Ministry of Public Security, about 50% of significant traffic accidents are related to lane line departure. In this context, a lane-keeping system is favored by more and more drivers. In a lane-keeping system, lane line detection is closely related to the realization of lane-keeping function, which directly affects the robustness and real-time performance of the whole system.

In recent years, the advanced driver assistance system (ADAS) and autonomous driving are becoming more important to reducing traffic accidents. As a key technology for intelligent vehicles, lane detection has attracted widespread attention from plenty of institutes and automobile technology companies [3]. Generally, ADAS technology includes tasks such as lane changing (LC), object detection system (ODS), collision avoidance system (CAS), adaptive cruise control (ACC), lane departure warning system (LDWS) [4]. Among the research, vision-based lane detection has always been a hot topic in the field of lane line detection [5]. Lane markings are an essential traffic safety feature with the functions of separating road areas, stipulating the direction of travel, and providing guidance information for pedestrians. In addition, as one of the core elements of an auxiliary environment

awareness module, lane detection technology is crucial in the autopilot field [6] and is a powerful tool not only for building high-precision maps, but also for verifying their accuracy [7]. Researchers use different sensors such as radar, lidar, single camera, and stereo camera to obtain environmental data to identify the environment around the vehicle accurately. This paper will focus on vision-based methods because they are cheaper, but can correctly locate local locations in environmental perception for any subsequent lane departure detection and path/trajectory planning. Constantly monitoring the vehicle's position in the lane lines can be very helpful in avoiding collisions caused by accidental lane deviation, fatigue, or driving under the influence of controlled substances.

Lane line detection becomes very challenging and confusing due to the following factors: (a) The appearance of lane line marking variants, such as dashed or solid lines, white or yellow lines, and road boundaries with less knowledge of road geometry [8–10]. (b) The intrusion of nearby obstacles either concealed the lane line markings or was mistakenly regarded as the lane line-markings. (c) The existence and shadow effect of street signs or words are often considered lane lines feature points [11,12]. (d) The change in illumination, which will mutate the color and intensity.

The main contributions of this paper are as follows: (a) Considering the importance of the decoder in recovering the boundary information of the target pixel, the decoder branch is added based on the existing direct up-sampling lane line detection algorithm. It improves the classification and prediction ability of the lane line boundary pixel. (b) Considering the limitation of the decoder in the computational cost, the idea of knowledge distillation is adopted to use the classification probability graph generated by the decoder. It is a soft target to guide the direct up-sampling branch to learn the boundary classification ability of the lane line without increasing the time of the model in the inference stage. (c) The proposed method can be applied to all current lane line detection algorithms using a direct up-sampling structure. In this paper, the generality and effectiveness of the proposed method are verified on the most challenging CuLane dataset by conducting experiments on several mainstream lane line detection algorithms using a direct up-sampling structure.

The remaining part of the research is organized as follows:

- Section 2—Discusses the related work.
- Section 3—Describes the methodology and multi-tasks optimization.
- Section 4—Discusses the CuLane dataset.
- Section 5—Reports the experimental results and analysis.
- Section 6—Provides our conclusions.

2. Related Work

The automatic driving sensing module can obtain the environmental information around the vehicle. One of the most important components of the module is lane line detection, which is the premise to control the safe driving of the automatic driving vehicle within the lane line. Therefore, lane line detection has become a research hotspot in the field of autonomous driving perception. Existing in the lane line detection method can be roughly divided into three categories: traditional image processing method, image processing, and convolutional neural networks (CNN) with the neural network and the lane line detection method based on deep learning.

Among the traditional lane line detection methods, some works of literature [13–15] have made a comprehensive summary of it summarized that the traditional lane line detection methods [16] are usually divided into three steps: image pre-processing, local feature extraction, and lane line fitting. Among them, local feature extraction is to capture local lane line information by using edge [17], texture [18], and color [19] features on the region of interest (ROI) [20] in the image, which is a key step of traditional lane line detection methods. However, it needs to analyze the distribution characteristics of lane lines in the image, and manually design and combine feature extraction algorithm. The traditional lane line detection method was selected for specific road scenes. It has been

difficult to adapt to the complex scenes in the practical application of autonomous driving, and the design of the extraction algorithm requires a high professional level.

In the method combining traditional image processing and CNN, CNN relies on large-scale datasets [21], and its powerful nonlinear fitting ability, which overcomes the limitations of the traditional lane line detection method and improves the generalization ability of the algorithm. Early lane line detection methods [17,22] based on the convolutional neural network only use a neural network to replace the local feature extraction step in traditional ways, but they still need complex post-processing. For example, Kim et al. [17] combined neural network and RANSAC algorithm to detect lane lines and used CNN to learn lane line features in edge images. The RANSAC algorithm was used to remove outliers and fit the lane lines, but since CNN input is a lane line edge detection image, the accuracy of this method is directly affected by the edge detection algorithm. In literature [23], the top and front view of the lane lines are taken as input, and the dual-view convolutional neural network is used to detect the lane line. However, it requires complex pre-processing of the input image.

The lane line detection method based on deep learning is defined as a dense classification prediction problem. In order to better classification from the supervision of the sparse signal out of the lane line, Pan [24] proposed a spatial convolution neural network (SCNN). SCNN transmits messages between neurons from different directions in the space so that they can capture the spatial relationship between pixels, but it cannot recover the lane line boundary pixels well because of direct up-sampling. Therefore, most lane line segmentation networks [25,26] adopt the encoder–decoder network structure. For example, Kim et al. [27] redefined the point detection problem as a region segmentation problem. SegNet [28] was used to segment the lane lines on both sides in a sequential end-to-end transmission learning method without any post-processing. However, it relies heavily on the lane line segmentation graph as the supervision signal and the training process is very complicated. Zhang et al. [29] use the network structure of two decoders sharing the same encoder to segment the lane line and the driving area and use the link encoder to transfer the lane line information of the two decoders with complementary branches, while the decoder with multi branches will bring huge computing overhead. Literatures [30,31] also adopted the encoder–decoder structure and extended lane line detection to instance segmentation. They need to accurately predict pixel categories and assigned correct instance labels for each pixel. However, they used clustering or complex label allocation strategies, which increased the computational complexity of the model.

To solve the problem that the direct up-sampling algorithm method in the literature as mentioned above cannot recover the lane line features well, and the problem that the network using the encoder–decoder method increases the huge computational cost, this paper proposes an object feature distillation (OFD) framework. It uses a direct up-sampling structure network based on the addition of a decoder branch. It also needs to fuse the features of each stage of the encoder to supplement the detailed information of the target and better restore the classification prediction of the target. Meanwhile, considering the requirement of the autonomous driving system for the reasoning time of lane line detection algorithm, this paper is inspired by the knowledge distilling algorithm [32–34]. The classification probability graph generated by the decoder is used as the soft target of the distillation loss function. The classification probability graph of the up-sampling branch is directly used as input to learning the refined classification information of the target in the encoder branch. In addition, only the forward calculation is performed on the direct up-sampling branch in the model inference state. The decoder is not used, which will not bring any additional computational cost.

3. Methodology

Firstly, detail the SCNN architecture along with network details and then show how we fuse object feature distillation to get better results.

3.1. Spatial Convolution Neural Network (SCNN)

The target feature distillation method proposed in this paper is mainly applied to the method of directly up-sampling the feature of the image to the input original image size for pixel-to-pixel classification. The SCNN network is one of the representatives of this kind of lane line segmentation algorithm. As shown in Figure 1, it adds 4 SCNNs in the spatial direction after the top convolution layer (“FC7 layer”) of LargeFOV [35] to introduce spatial message propagation so that it can better capture the spatial relationship between pixels. Its backbone network is the modified VGGNet [36], which better preserves spatial information by removing the largest pooling layer in the 3rd and 4th stages, and the standard convolution is replaced by Dilated Convolution (DC) [37] to obtain a larger receiver field. Finally, the directly up-samples branch samples the feature image up by eight times through bilinear interpolation. The resolution is equal to the size of the input image to obtain a larger receptive field. Another existing prediction branch will pass the *softmax* operating to the feature image and input it to the average pooling layer and the fully connected layer (FC), the *sigmoid* function is used to predict whether there is a lane line.

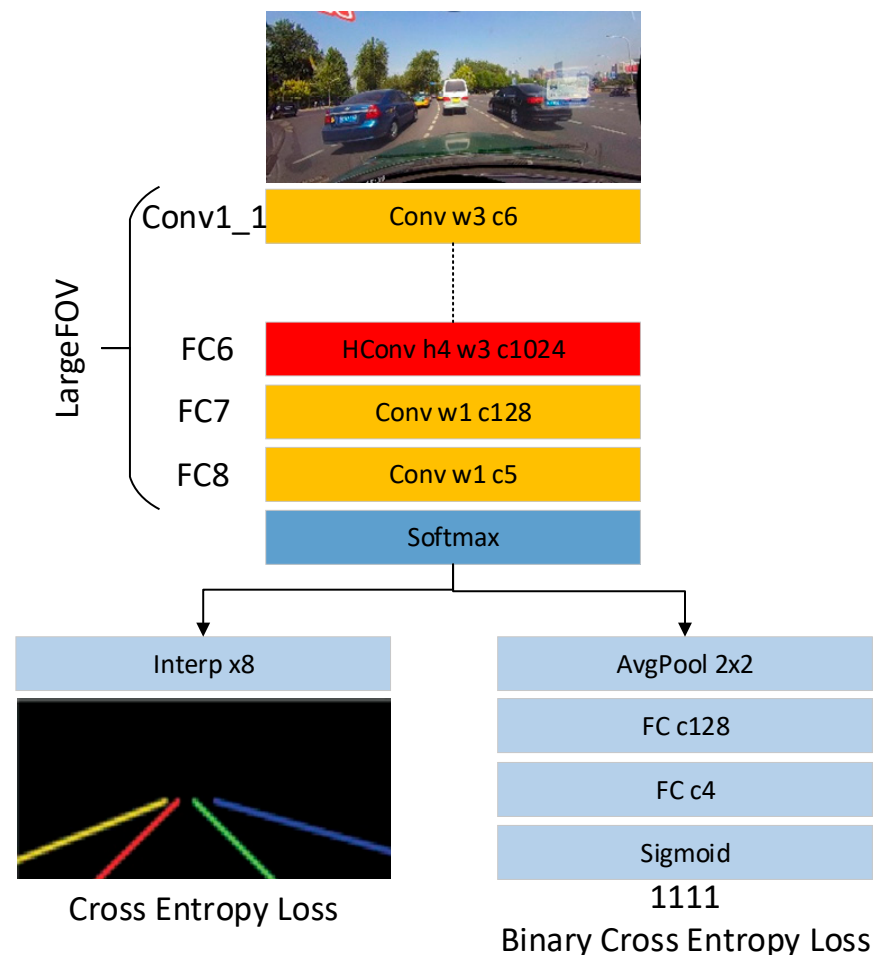


Figure 1. SCNN backbone network structure diagram.

3.2. Knowledge Distillation (KD) Method

Knowledge distillation (KD) is to extract knowledge from complex networks to guide compact network learning, which can significantly improve the performance of compact models and has been extended from image classification tasks to segmentation tasks [38–40]. Usually, in the knowledge distillation method, the compact student network takes the intermediate output of the teacher network as the soft target and supervises the student network to learn the knowledge extracted from the teacher network. In the

literature [39], the geometric information of target depth predicted by the teacher network is used to guide the semantic segmentation tasks. However, lane lines do not have similar target depth information in natural images. In lane line segmentation, knowledge distillation is used to use the more abstract feature images generated in the later stage as the soft target of the distillation [41], which enables the network to learn richer context information. However, it ignores the shallow local detail information, which is very important for lane line boundary restoration. The target feature distillation method proposed in this paper does not need to train the complex teacher network alone, and each branch shares the feature coding network. We only need to use the classification probability map generated by the encoder branch with lane line boundary thinning information like the soft target of distillation to guide the direct up-sampling branch. In addition, the method in this paper does not require additional annotation of the training dataset.

3.3. Methodology Follow in This Paper

This paper proposes a method of distilling lane line boundary refinement information from the decoder. It can be applied to various algorithms that directly up-samples the feature image to the input image size for lane line segmentation to learn more accurate lane line boundary information from the decoding branch and do not increase the reasoning time of the network. Next, the design details of the overall framework of the proposed method, the structure of the added decoder branch, and the implementation principle of distilling lane line feature information from the decoder are described in detail.

3.3.1. The Overall Structure Design of the Target Characteristic Distillation Network

In this paper, lane line detection is defined as a pixel-level classification problem; that is, lane line features are extracted from the input image through the backbone network. Then, the resolution of the feature image is gradually restored, and the lane line information in the feature image is refined in the decoder. Finally, distilling the classification results of each pixel predicted by the decoder to the direct up-sampling branch to enhance its prediction ability of lane line features. As shown in Figure 2, the whole network is composed of four parts, including 1 encoder and 3 output branches.

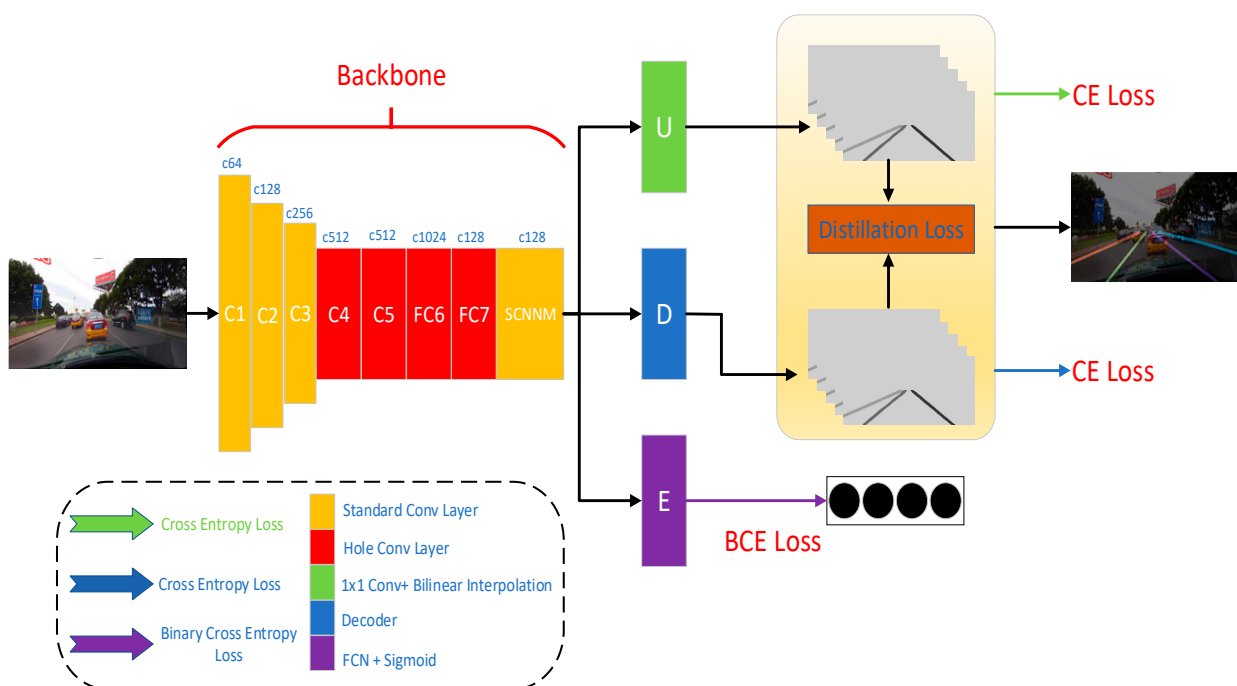


Figure 2. The overall block diagram of the target characteristic distillation network.

In the existing method, the encoder is usually a modified mainstream Convolutional Neural Network (CNN) [42,43]. For example, to preserve the spatial information of the target, VGGNet and ResNet will remove the maximum pooling layer of the last two stages in VGGNet and ResNet or modify the convolution step size to 1 and replace the standard convolution layer with hole convolution with a different void ratio, to obtain a larger receptive field. In addition, “FC6” and “FC7” fully connected convolution layer, as shown in Figure 1, are added. Their output with high-dimensional feature images enables the network to learn more abundant lane line features.

The direct up-sampling branch changes the channel of the output feature image of the backbone network through a 1×1 convolution layer. It then uses bilinear interpolation to directly restore the resolution of the feature image to the size of the input image. However, the resolution of the feature image is continuously reduced in the encoder, and the representation of the feature becomes more and more abstract. Therefore, the lane line detection network using the direct up-sampling method often fails to sufficiently recover the lane line boundary information. Therefore, a decoder branch is added in this paper, which can restore the resolution of the feature image and refine the lane line features through three times of two times up-sampling and convolution operations.

It is predicted that the lane lines have branches that will act on the backbone network. The design details are shown in Figure 1. Firstly, the feature graph output by the decoder is subjected to the *softmax* operation; then, it is stretched into a one-dimensional vector and sent to the average pooling layer and the two-layer fully connected layer. Finally, the output result of the full connection layer is used as a *sigmoid* operation input to the binary cross-entropy loss function. The lane line branch prediction results are used for the lane line evaluation stage, and only the segmentation image that predicts the presence of lane lines is obtained for lane line coordinates to reduce the network’s misdetection of non-existent lane lines.

3.3.2. Network Decoder Structure Design of Target Feature Distillation Network

The design of the encoder structure in the lane line detection network determines the ability of the network to extract the target features, and the design of the decoder structure affects the ability to restore and predict the details of the target features. In order to explore the influence of different decoder structures on the feature prediction of recovery lane lines, this paper added decoders to the SCNN network. The decoder is designed with three stages, and its structure is shown in Figure 3a. The output of the encoder is directly refined by three times of convolution layer and two times of bilinear interpolation so that the resolution of the feature image is restored to the original image size. The structure of the 5-stage decoder is shown in Figure 3b. In addition to the above operations, also fully consider the different characteristics of the target feature expression at a different stage of the encoder. The decoder features of the later stage are used to guide the feature learning of the previous stage to recover the prediction of target details better. This decoder design can be applied to various lane line segmentation algorithms using the direct up-sampling method. In addition, when using ResNet as the backbone network, only four stages of decoders are needed. In Figure 3, “ \times ” is used to define that how many convolutional layers we used, and “*” is used to define the convolutional kernel size we use in that specific layer.

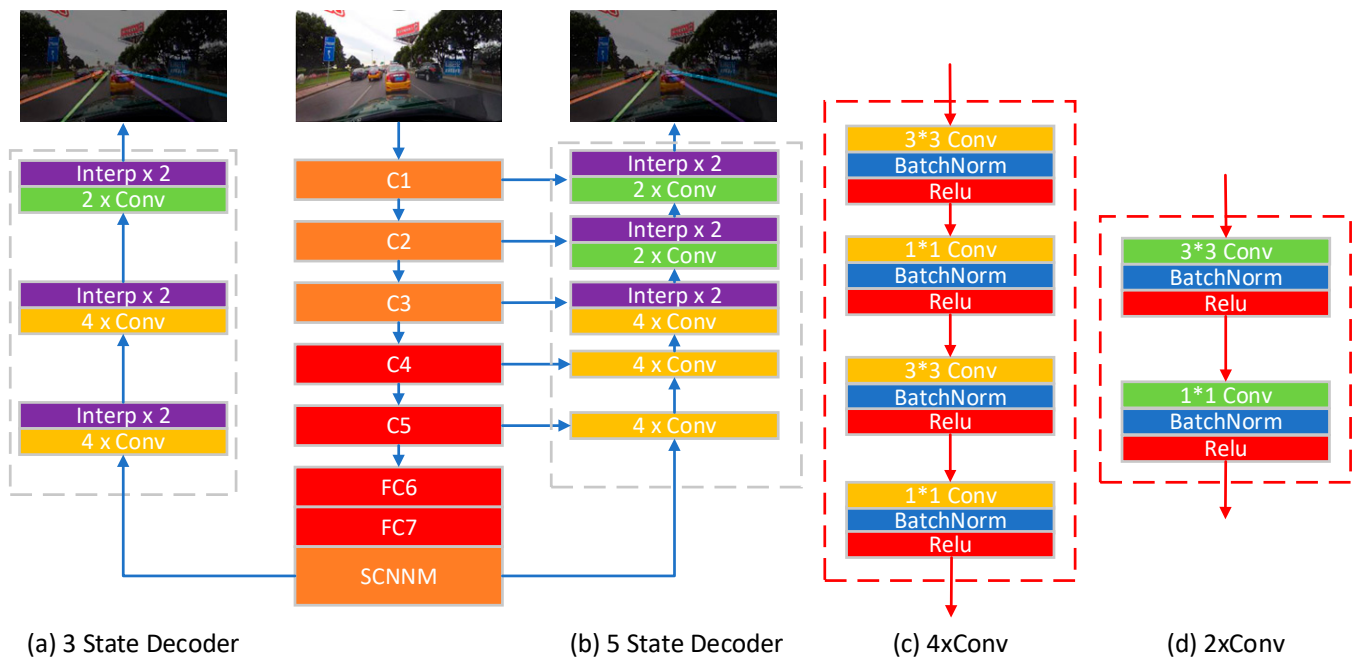


Figure 3. Block diagram of decoder structure of target feature distillation network.

3.3.3. Loss Function of the Target Feature Distillation Network

Inspired by using the classification probability map output by the teacher network as the soft target of the pixel-level distillation loss function in the existing research, the classification probability map generated by the decoder branch is distilled in this paper. The direct up-sampling branch learns the lane line boundary information to enhance the detection of lane line boundary. Therefore, the final convolution output of decoder branching is defined as $A \in R^{c \times h \times w}$, where c represents the number of channels of convolution output. h represents the height of the output feature graph; w represents the width of the output feature graph. Similarly, the directly up-samples branch convolution output is defined as $B \in R^{c \times h \times w}$. In order to enable the direct up-sampling branch to learn the pixel-level classification results of the decoder branch, the mean square error loss function is used as the distillation loss function of the two branches, and its definition is shown in Equation (1):

$$l_{dist}(A_{ij}, B_{ij}) = \frac{1}{w \times h} \sum_i^h \sum_j^w \|\varphi(A_{ij}) - \varphi(B_{ij})\|^2 \quad (1)$$

where $\varphi(\cdot)$ represents *softmax* operation. After the convolution output passes through *softmax*, the sum of all pixels in the channel dimension is 1, and any pixel value on the feature image represents the classification probability that the current position belongs to one of the lane lines or background; $\varphi(A_{ij})$ represents the target value of the decoder distillation loss function, and $\varphi(B_{ij})$ is the input value of the distillation loss function.

The distance between the classification probabilities of the two branch pixels is measured by Equation (1), and the network minimizes the distance between the input value and the target value through continuous iteration.

The definition of the total loss function of the method in this paper is shown in Equation (2):

$$l_{total} = l_{up}(B - \hat{B}) + l_{decoder}(A - \hat{A}) + \alpha l_{dist}(A, B) + \beta l_{exit}(E - \hat{E}) \quad (2)$$

The total loss function is composed of four parts, in which $l_{up}(\cdot)$ and $l_{decoder}(\cdot)$ represent the cross-entropy loss function of the direct up-sampling branch and the decoder

branch, respectively. \hat{A} and \hat{B} both represent the true annotation of the current input image; $l_{dist}(\cdot)$ represents the decoder information of distillation loss function in Equation (1), which means that the lane line has a branch loss function, using the binary cross loss function. E is the output of the *sigmoid* function; \hat{E} represents the true label of whether the lane line exists in the current input image. Finally, parameters α and β were used to balance the distillation task and the presence of lane lines to predict the influence of the task on the performance of the whole network.

4. Dataset

Despite the importance and complexity of traffic lane line detection, current datasets are either too limited or too easy to compare various approaches, and a large public annotated benchmark is needed [13]. KITTI [44] and CamVid [45] both have pixel-level annotations for lane markings, but only hundreds of images, making deep learning approaches ineffective. Caltech Lanes Dataset [16] and TuSimple Benchmark Dataset [46] include 1224 and 7000 photos with annotated lane markings, respectively, in a restricted scenario with light traffic and transparent lanes marking. Furthermore, none of these datasets annotates lane markings that are occluded or unseen due to abrasion. However, such lane markings can be inferred by humans and are extremely useful in real-world applications.

In this paper, we prefer to use the CuLane dataset [24]. Cameras were installed in six different vehicles driven by six different drivers and captured videos while driving in Beijing on various days. More than 55 h of video were collected, yielding 133,235 frames, more than 20 times the size of the TuSimple Dataset. The dataset was divided into 88,880 for preparation, 9675 for validation, and 34,680 for testing. These input images have a resolution of 1640×590 . We remove the sky and very close areas from the image and resize accordingly to the lane line network model. Our initial experiments found that traditional photometric and geometric recipes for data augmentation were not providing better driving results, so we do not use augmentation. Figure 4 depicts a variety of scenes, including urban, agricultural, and highway settings. As one of the world's largest and most congested cities, Beijing presents many daunting traffic scenarios for lane line detection. The dataset was divided into two categories: standard and difficult, which correspond to the nine examples in Figure 4. The proportion of each scenario is shown in Figure 5. The eight difficult scenarios account for the majority of the dataset (72.3%). We manually annotate the traffic lane lines with cubic splines for each frame. As previously mentioned, lane markers are often obscured by automobiles or are not visible. Even in these difficult situations, lane line detection algorithms must estimate lane locations from the background in real-world applications. As a result, we continue to annotate the lanes according to the context in these situations, as shown in Figure 4.

CuLane dataset is available at <https://drive.google.com/drive/folders/1mSLgwVTiaUMAb4AVOWwICD5jCWdrwpvu> (accessed on 1 May 2021).

Table 1 compares and prefers using the CuLane dataset [24] with existing lane marking datasets [47].

Table 1. Comparison among different lane line datasets.

Dataset	Training	Frame	Sequences	M. Weather	M. Time	Attributes
Caltech Lanes	-	1224	4	Yes	No	2
Road Marking	-	1443	29	Yes	Yes	10
KITTI Road	289	579	-	No	No	2
TuSimple	3626	7000	7000	No	No	2
VPGNet	14,783	21,097	-	Yes	Yes	17
BDD100K	70,000	100,000	100,000	Yes	Yes	11
CuLane	88,880	133,235	55 h Video	Yes	Yes	9



Figure 4. CuLane dataset example in different scenarios.

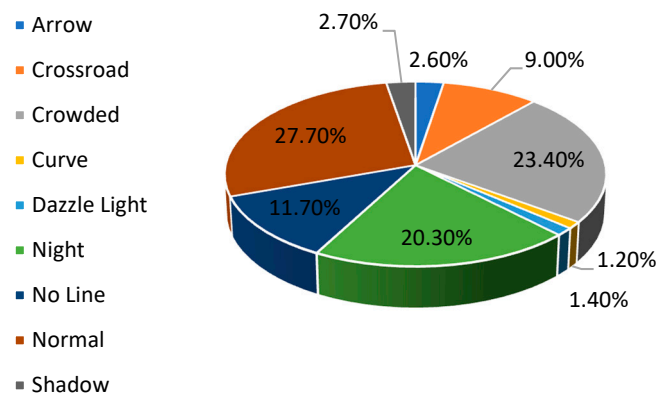


Figure 5. Percentage of each scenario.

5. Experimental Result and Analysis

The training, validation, and testing of the experimental model in this paper are all built by TensorFlow [48] framework and cuDNN [49] kernel used for calculation. Hardware equipment mainly includes a high-performance workstation host. The workstation is configured with an Intel@CoreTM i7-6800K CPU@3.40 GHz, NVIDIA 2080 Ti graphics card.

5.1. Evaluation Standard

To verify the effectiveness of the proposed method, the official evaluation terminology is fully used when evaluating the CuLane dataset. In the literature [25], to make the evaluation more reasonable, both the predicted and the real lane line are labeled as lines with a width of 30 pixels and calculated the intersection over union (IOU) between them. True-positive (TP) cases are selected by setting IOU threshold, which is usually set to 0.3 or 0.5. Then, the $F1_{Measure}$ is used to quantify the results of lane line detection, whose definition is shown in Equation (3):

$$F1_{Measure} = \left(1 + r^2\right) \frac{P \times R}{r^2 p + R} \quad (3)$$

Precision (P) is defined as

$$\text{Precision } (P) = \frac{TP}{TP + FP} \quad (4)$$

where FP represents false-positive cases, and Recall Rate (R) is defined as:

$$\text{Recall } (R) = \frac{TP}{TP + FN} \quad (5)$$

where FN denote false negative (FN), γ is usually set to 1. In addition, during the test, the coordinates of each lane line need to be determined from the probabilistic map predicted by the network, so the results with branches need to be used. If the existing value of the lane line in the predicted probability map is greater than the threshold value, then the pixel coordinate points of the corresponding lane line are obtained from the probability map for subsequent calculation of $F1_{Measure}$ which can effectively reduce the false detection rate.

5.2. Experimental Environment & Network Parameter Setting

This paper uses the standard Stochastic Gradient Descent (SGD) training model on the CuLane data set. The batch size is set as 12, and the basic learning rate is 0.01, the momentum is 0.9. The weight decay is 0.0001. The learning rate strategy adopts “Poly,” and the learning index and the iteration times are set as 0.9 and 60 K, respectively.

The backbone network used in the experiment is VGGNet and ResNet, both of which use the pre-trained weights of the ImageNet dataset and enhance the input data. First, random rotation of $[-2^\circ, 2^\circ]$ is performed on the batch data; then, random cropping is performed, and the cropped image size is equal to 0.05 times the original input image resolution. Finally, the image resolution is adjusted to 288×800 pixels. Through a large number of experiments, α in Equation (2) is set as 10, and β is set as 0.1. All experiments in this paper were carried out under the PyTorch framework, and two NVIDIA 2080 Ti GPUs were used. In addition, in training the network, the gradient of distillation loss during backpropagation does not update the decoder branch parameters; it only affects the weights of the parameters of the direct up-sampling branch.

5.3. Experimental Result and Comparative Analysis

5.3.1. Quantitative Analysis

To verify the effectiveness of the proposed method in this paper and calculate its $F1_{Measure}$ on the CuLane dataset, Table 2 shows the calculation results of the target feature distillation method applied on the SCNN [24]. In nine complex scenarios of the CuLane dataset, the $F1_{Measure}$ of the proposed method in six scenarios, and the whole dataset is ahead of the other algorithms. It verifies the effectiveness of the proposed method without increasing the additional computation cost. However, in-crowd and dazzle light scenario GCJ [29] method because $F1_{Measure}$ is higher than the other methods. This is because GCJ method uses the geometric prior information between the lane line and the driving area, which increases the context reasoning ability of the algorithm for occluding lane line. By adding geometric information, the segmentation effect of lane lines is improved, which is the next work of this paper.

Table 2. Model performance comparison.

Method	Scenario									
	Normal	Crowded	Dazzle Light	Shadow	No Line	Arrow	Curve	Crowded	Night	Total
SCNN [24]	90.6	69.7	58.5	66.9	43.4	84.1	64.4	69.7	66.1	71.6
GCJ [29]	89.7	76.5	67.4	65.5	35.1	82.2	63.2	76.5	68.7	73.1
SAD [41]	90.7	70.0	59.9	67.0	43.5	84.4	65.7	70.0	66.3	71.8
Our Smooth	92.1	72.6	61.6	71.7	46.0	86.9	68.9	72.6	69.4	74.1

As shown in columns 6–9 of Table 3, $F1_{Measure}$ of the target feature distillation method applied to DeepLabv1, ResNet50, ResNet101, and SCNN are shown, respectively [25]. The results show that the target feature distillation method proposed in this paper can improve the lane line detection performance in each sub-scene and the total dataset, especially in the shadow and dazzle light scene. This method brings significant improvement, which verifies its effectiveness and versatility. In addition, columns 2–5 in Table 3 are the evaluation results of literature [24] on the corresponding lane line detection algorithm. Since there is no lane line labeling in the crossroad sub-scene data, Table 3 only shows its FP measure.

Table 3. Experimental results of the application of target feature distillation method to the main stream lane segmentation method.

Scenario	Methods							
	DeepLabv1 [25]	ResNet50 [25]	ResNet101 [25]	SCNN [25]	DeepLabv1 + OFD	ResNet50 + OFD	ResNet101 + OFD	SCNN + OFD
Arrow	74.0	79.0	84.0	84.1	76.8	82.5	86.2	86.8
Cross Road	2060	2505	2183	1990	2445	2229	2004	2222
Crowded	61.0	64.1	68.2	69.7	62.8	67.1	69.8	71.5
Curve	61.0	59.8	65.5	64.4	61.8	64.5	68.5	67.0
Dazzle Light	49.9	54.1	59.8	58.5	55.3	58.6	63.3	64.3
Night	56.9	60.6	65.9	66.1	57.57	63.6	67.1	67.6
No Line	34.0	38.1	41.7	43.4	35.4	40.6	43.8	45.0
Normal	83.1	87.4	90.2	90.6	84.5	89.1	90.9	91.6
Shadow	54.7	60.7	64.6	66.9	64.7	67.6	68.3	73.0
Total	63.2	66.7	70.8	71.6	64.7	69.4	72.2	73.2

As shown in Figure 6, the improved $F1_{Measure}$ of the four groups of comparative experiments in Table 3 in each scene were calculated, and their mean and standard deviation were calculated. It can be seen from the figure that the average value of the improvement is the highest in the Resnet50 method, and it can be found that the network performance of the decoder with a relatively simple structure is improved more obviously, which indicates the importance of the decoder for the prediction of lane line feature recovery, and also proves the effectiveness of the target feature distillation method. Additionally, in the Resnet101 method, the performance of each sub-scene has the closest improvement range, so the standard deviation of its $F1_{Measure}$ improvement is the smallest, which may be due to its encoder's strong feature expression ability, which enables it to extract lane line features in each traffic scene effectively.

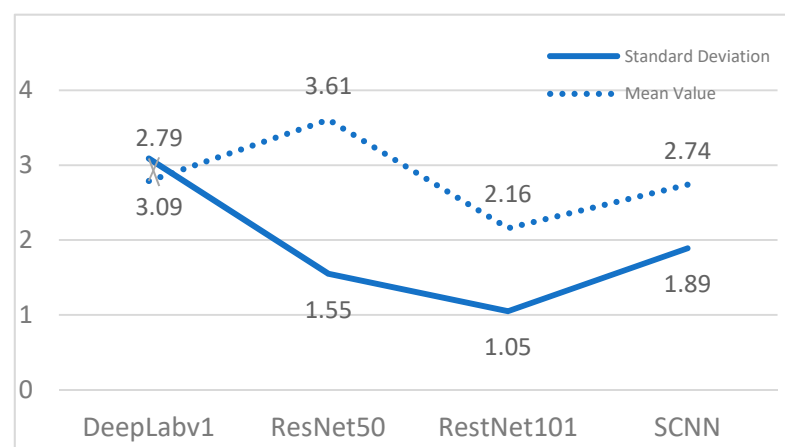


Figure 6. Mean and standard deviation of $F1_{Measure}$ improvement in each scene.

5.3.2. Qualitative Analysis

The target feature distillation method was applied to DeepLabv1 and SCNN methods, and lane line detection results before and after application were compared. As shown in Figure 7, column 1 represents the actual label of the input image. The images in column 2 represent the results without using the target feature distillation method. The image in column 3 represents the experimental results after applying the method. By comparing the lane line detection results at the red circle in column 2 image of the DeepLabv1 method, it can be seen that the target feature distillation method can more effectively recover the prediction of lane line pixels in congestion and night scenes. From the red circle in the lower right corner of the SCNN prediction result image, it can be seen that the method presented in this paper can effectively reduce the false detection rate of lane lines on the shadow road without clear lane line identification.

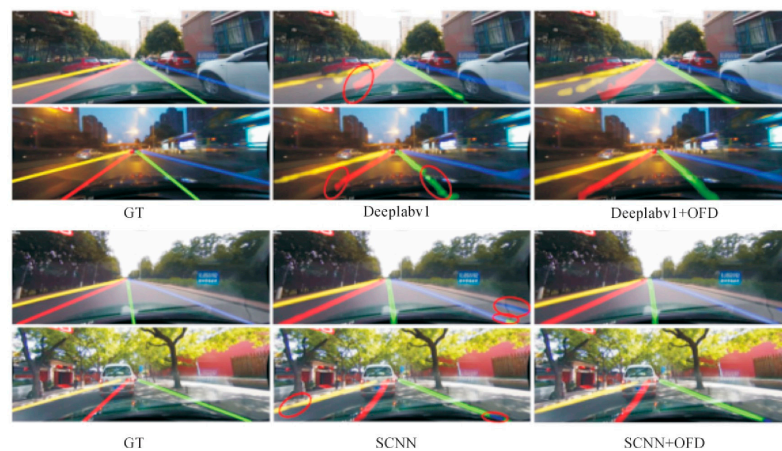


Figure 7. Lane line detection effect diagram before and after using the target feature distillation method.

Figure 8 shows lane line detection results after each algorithm applying the proposed method in different complex road scenarios of the CuLane dataset. Column 1 is the original image input, column 2 is the corresponding segmented image, and column 3–5 is the probability graph of lane line segmentation of each algorithm. To better see the probability of lane line classification, the probability diagram of the SCNN network in the fifth column is visualized by a heat map and displayed in the last column. The blue represents the minimum probability value, and the red represents the maximum. From left to right, as the algorithm complexity increases, the lane line boundary segmentation and lane line continuity in the lane line classification probability diagram get better and better. In particular, as the visualization result of line 6 in Figure 8 shows, with the enhancement of the network's ability to extract lane line features, the detection effect of the blue lane line on the right of the figure gets better and better. In addition, as shown in lines 4–6 in Figure 8, the method proposed in this paper can effectively detect lane lines under harsh conditions with no obvious illumination at night.

As different decoders have different degrees of recovery and refinement to the encoder feature classification prediction results. Various decoder networks are designed to add to the SCNN method, and the influence of lane boundary refinement information on lane detection performance in the distillation is explored. As shown in Table 4, row 2 shows the experimental results when the decoder with three stages is up-sampled, row 3 shows the results of using a five stage decoder network. Row 4 shows the results after using the smooth decoder network proposed in the literature [50]. Due to the highest complexity of decoder design and channel attention module, the attention weight vector can be learned and generated. By adjusting the channel weight, the classification probability of lane line pixel can be enhanced to improve the lane line detection performance and to distill the information of smooth decoder, the $F1_{Measure}$ of the new method is 74.1%,

which is 2.5% higher than that of the original SCNN. It also shows the effectiveness of the proposed method.

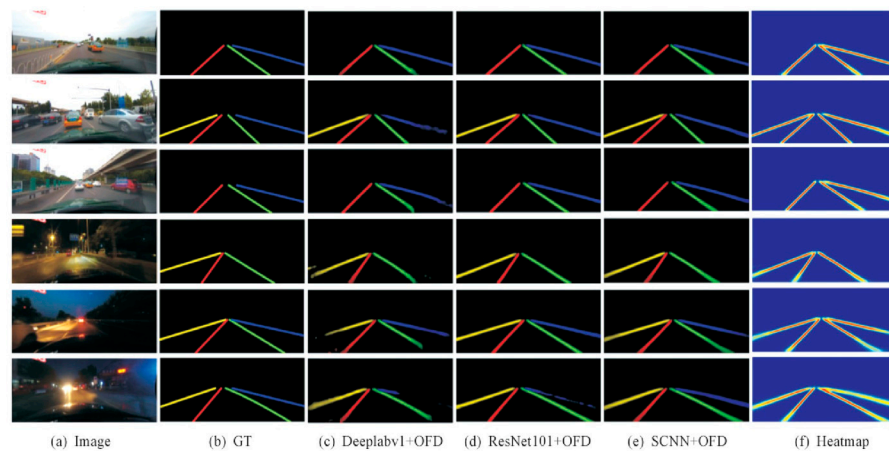


Figure 8. Visualize the lane line classification probability map generated by each algorithm.

Table 4. Influence of different decoder structures on lane line detection performance %.

Method	Total (%)
SCNN	71.6
SCNN (3 Stage)	72.4
SCNN (5 Stage)	73.2
SCNN (Smooth)	74.1

Since the final output result of the convolutional layer of the network is operated by *softmax*, the classification probability of the pixel belonging to each lane line and background in the channel dimension of the feature map satisfies the probability distribution. Therefore, in the design of distillation loss function and using the mean square error (MSE) to measure the distance of classification probability of each pixel directly. The KL divergence loss function can also measure the distance between the distribution of classification probability map of lane line output from direct up-sampling branch and decoder branch, which measures the distance between the output results of two branches from the perspective of a probability distribution. As shown in Table 5, row 2 is the experimental result of using the mean square error (MSE) loss function as the distillation loss function. Row 3 shows the experimental results using the KL divergence loss function as the distillation loss function, which improves the $F1_{Measure}$ on the dataset by 0.4% compared to the results using the mean square error (MSE) as the distillation loss function.

Table 5. Influence of different distillation loss functions on lane line detection performance %.

Method	Total (%)
SCNN	71.6
SCNN_5Stage_MSE	73.2
SCNN_5Stage_KL	73.6

6. Conclusions and Future Work

In this paper, lane detection is defined as the problem at a pixel-level classification. To solve the problem of lane line segmentation without using a decoder to recover the lane line boundary refinement information. By adding the decoder branch, the target of the lane line can be better recovered from the low-resolution feature image. The feature pixel classification prediction and the pixel-level distillation loss function are applied. The direct up-sampling branch learns the lane line segmentation probability map generated by

the decoder branch to make up for the lack of lane line boundary refinement information. Therefore, without increasing any computational cost, the network's ability to detect lane lines is improved. The experimental result shows that the target feature distillation method proposed in this paper improves the $F1_{Measure}$ and can better adapt the various types of road scenes. In straight lines, curves, backlit scenes, and vehicle occlusion scenes, the algorithm has good detection accuracy, strong robustness, and higher detection speed. In addition, the prior geometric information between lane lines can effectively improve the performance of lane detection.

In future work, the lane detection and classification results will fuse with a forward-collision warning strategy. It will be helpful for assisted driving in avoiding collision caused by accidental lane deviation, fatigue, or driving under the influence of controlled substances in complex or straight road environments structured.

Author Contributions: In this research activity, M.H. participated in the writing code, model generation, simulation, and analysis of results. A.G. reviews and monitors to improve the overall work and analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: CuLane dataset is available at <https://drive.google.com/drive/folders/1mSLgwVTiaUMAb4AVOWwlCD5JcWdrwpvu> (accessed on 1 May 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tran, N. *World Health Organization*; WHO: Geneva, Switzerland, 2018; pp. 5–11.
2. National Highway Traffic Safety Administration. Traffic Safety Facts 2017: A Compilation of Motor Vehicle Crash Data. DOT HS 812806. 2019. Available online: <https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812806> (accessed on 1 January 2021).
3. Song, W.; Yang, Y.; Fu, M.; Li, Y.; Wang, M. Lane Detection and Classification for Forward Collision Warning System Based on Stereo Vision. *IEEE Sens. J.* **2018**, *18*, 5151–5163. [[CrossRef](#)]
4. Haris, M.; Hou, J. Obstacle detection and safely navigate the autonomous vehicle from unexpected obstacles on the driving lane. *Sensors* **2020**, *20*, 4719. [[CrossRef](#)] [[PubMed](#)]
5. Khalifa, O.O.; Assidiq, A.A.M.; Hashim, A.H.A. Vision-based lane detection for autonomous artificial intelligent vehicles. In Proceedings of the ICSC 2009—2009 IEEE International Conference on Semantic Computing, Berkeley, CA, USA, 14–16 September 2009; pp. 636–641.
6. Zhang, X.; Huang, H.; Meng, W.; Luo, D. Improved lane detection method based on convolutional neural network using self-attention distillation. *Sens. Mater.* **2020**, *32*, 4505–4516. [[CrossRef](#)]
7. Homayounfar, N.; Ma, W.C.; Lakshmikanth, S.K.; Urtasun, R. Hierarchical Recurrent Attention Networks for Structured Online Maps. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
8. Sun, T.Y.; Tsai, S.J.; Chan, V. HSI color model based lane-marking detection. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), Toronto, ON, Canada, 17–20 September 2006; pp. 1168–1172.
9. Chiu, K.Y.; Lin, S.F. Lane detection using color-based segmentation. In Proceedings of the IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 6–8 June 2005; Volume 2005, pp. 706–711.
10. Li, Q.; Zheng, N.; Cheng, H. An adaptive approach to lane markings detection. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), Shanghai, China, 12–15 October 2003; Volume 1, pp. 510–514.
11. Bertozzi, M.; Broggi, A.; Fascioli, A. Vision-based intelligent vehicles: State of the art and perspectives. *Rob. Auton. Syst.* **2000**, *32*, 1–16. [[CrossRef](#)]
12. Jung, C.R.; Kelber, C.R. Lane following and lane departure using a linear-parabolic model. *Image Vis. Comput.* **2005**, *23*, 1192–1202. [[CrossRef](#)]
13. Bar Hillel, A.; Lerner, R.; Levi, D.; Raz, G. Recent progress in road and lane detection: A survey. *Mach. Vis. Appl.* **2014**, *25*, 727–745. [[CrossRef](#)]
14. Kumar, A.M.; Simon, P. Review of Lane Detection and Tracking Algorithms in Advanced Driver Assistance System. *Int. J. Comput. Sci. Inf. Technol.* **2015**, *7*, 65–78. [[CrossRef](#)]
15. Narote, S.P.; Bhujbal, P.N.; Narote, A.S.; Dhane, D.M. A review of recent advances in lane detection and departure warning system. *Pattern Recognit.* **2018**, *73*, 216–234. [[CrossRef](#)]
16. Aly, M. Real time detection of lane markers in urban streets. In Proceedings of the IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 7–12.

17. Kim, J.; Lee, M. Robust lane detection based on convolutional neural network and random sample consensus. In Proceedings of the International Conference on Neural Information Processing, Kuching, Malaysia, 3–6 November 2014; Volume 8834, pp. 454–461. [CrossRef]
18. Jung, H.; Min, J.; Kim, J. An efficient lane detection algorithm for lane departure detection. In Proceedings of the IEEE Intelligent Vehicles Symposium, Gold Coast, QLD, Australia, 23–26 June 2013; pp. 976–981.
19. Bottazzi, V.S.; Borges, P.V.K.; Stantic, B.; Jo, J. Adaptive regions of interest based on HSV histograms for lane marks detection. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 274, pp. 677–687.
20. Wu, P.C.; Chang, C.Y.; Lin, C.H. Lane-mark extraction for automobiles under complex conditions. *Pattern Recognit.* **2014**, *47*, 2756–2767. [CrossRef]
21. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
22. Huval, B.; Wang, T.; Tandon, S.; Kiske, J.; Song, W.; Pazhayampallil, J.; Andriluka, M.; Rajpurkar, P.; Migimatsu, T.; Cheng-Yue, R.; et al. An Empirical Evaluation of Deep Learning on Highway Driving. *arXiv* **2015**, arXiv:1504.01716.
23. He, B.; Ai, R.; Yan, Y.; Lang, X. Accurate and robust lane detection based on Dual-View Convolutional Neural Network. In Proceedings of the IEEE Intelligent Vehicles Symposium, Gothenburg, Sweden, 19–22 June 2016; pp. 1041–1046.
24. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial as deep: Spatial CNN for traffic scene understanding. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2 February 2018; pp. 7276–7283.
25. Wang, Z.; Ren, W.; Qiu, Q. LaneNet: Real-time lane detection networks for autonomous driving. *arXiv* **2018**, arXiv:1807.01726.
26. Ghafoorian, M.; Nugteren, C.; Baka, N.; Booi, O.; Hofmann, M. EL-GAN: Embedding loss driven generative adversarial networks for lane detection. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Munich, Germany, 8–14 September 2018; Volume 11129, pp. 256–272.
27. Kim, J.; Park, C. End-To-End Ego Lane Estimation Based on Sequential Transfer Learning for Self-Driving Cars. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; Volume 2017, pp. 1194–1202.
28. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
29. Zhang, J.; Xu, Y.; Ni, B.; Duan, Z. Geometric Constrained Joint Lane Segmentation and Lane Boundary Detection. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Munich, Germany, 8–14 September 2018; Volume 11205, pp. 502–518.
30. Hsu, Y.C.; Xu, Z.; Kira, Z.; Huang, J. Learning to cluster for proposal-free instance segmentation. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
31. Neven, D.; De Brabandere, B.; Georgoulis, S.; Proesmans, M.; Van Gool, L. Towards End-to-End Lane Detection: An Instance Segmentation Approach. In Proceedings of the IEEE Intelligent Vehicles Symposium, Rio de Janeiro, Brazil, 8–13 July 2018; Volume 2018, pp. 286–291.
32. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:150.00531.
33. Papamakarios, G. Distilling Model Knowledge. *arXiv* arXiv:1510.002437, 2015.
34. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for Thin Deep Nets. Available online: <https://arxiv.org/abs/1412.6550> (accessed on 1 January 2021).
35. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
36. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. Available online: <https://arxiv.org/abs/1409.1556> (accessed on 1 January 2021).
37. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2016**, arXiv:1511.07122.
38. Xie, J.; Shuai, B.; Hu, J.F.; Lin, J.; Zheng, W.S. Improving Fast Segmentation with Teacher-student Learning. *arXiv* **2018**, arXiv:1810.08476.
39. Jiao, J.; Wei, Y.; Jie, Z.; Shi, H.; Lau, R.; Huang, T.S. Geometry-aware distillation for indoor semantic segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; 2019, pp. 2864–2873.
40. Liu, Y.; Chen, K.; Liu, C.; Qin, Z.; Luo, Z.; Wang, J. Structured knowledge distillation for semantic segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; Volume 2019, pp. 2599–2608.
41. Hou, Y.; Ma, Z.; Liu, C.; Loy, C.C. Learning lightweight lane detection CNNs by self attention distillation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 1013–1021.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
43. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

44. Fritsch, J.; Kuhn, T.; Geiger, A. A new performance measure and evaluation benchmark for road detection algorithms. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), The Hague, The Netherlands, 6–9 October 2013; pp. 1693–1700.
45. Brostow, G.J.; Shotton, J.; Fauqueur, J.; Cipolla, R. Segmentation and recognition using structure from motion point clouds. In Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Marseille, France, 12–18 October 2008; Volume 5302, pp. 44–57.
46. Tusimple Benchmark. Available online: <https://github.com/%0ATuSimple/tusimple-benchmark> (accessed on 1 January 2021).
47. Shirke, S.; Udayakumar, R. Lane datasets for lane detection. In Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing (ICCSP), Dalian, China, 20–23 September 2019; pp. 792–796.
48. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **2016**, arXiv:1603.04467.
49. Chetlur, S.; Woolley, C.; Vandermersch, P.; Cohen, J.; Tran, J.; Catanzaro, B.; Shelhamer, E. cuDNN: Efficient Primitives for Deep Learning. *arXiv* **2014**, arXiv:1410.0759.
50. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Learning a Discriminative Feature Network for Semantic Segmentation. Available online: https://openaccess.thecvf.com/content_cvpr_2018/html/Yu_Learning_a_Discriminative_CVPR_2018_paper.html (accessed on 1 January 2021).