Lane Image Detection Based on Convolution Neural Network Multi-Task Learning

Junfeng Li, Dehai Zhang *, Yu Ma and Qing Liu

School of Software, Yunnan University, Kunming 650504, China; junfengli-ynu@foxmail.com (J.L.); mayu@mail.ynu.edu.cn (Y.M.); liuqing@ynu.edu.cn (Q.L.)
* Correspondence: dhzhang@ynu.edu.cn

Abstract: Based on deep neural network multi-task learning technology, lane image detection is studied to improve the application level of driverless technology, improve assisted driving technology and reduce traffic accidents. The lane line database published by Caltech and Tucson company is used to extract the ROI (Region of Interest), scale, and inverse perspective transformation as well as to preprocess the image, so as to enrich the data set and improve the efficiency of the algorithm. In this study, ZFNet is used to replace the basic networks of VPGNet, and their structures are changed to improve the detection efficiency. Multi-label classification, grid box regression and object mask are used as three task modules to build a multi-task learning network named ZF-VPGNet. Considering that neural networks will be combined with embedded systems in the future, the network will be compressed to CZF-VPGNet without excessively affecting the accuracy. Experimental results show that the vision system of driverless technology in this study achieved good test results. In the case of fuzzy lane line and missing lane line mark, the improved algorithm can still detect and obtain the correct results, and achieves high accuracy and robustness. CZF-VPGNet can achieve high real-time performance (26FPS), and a single forward pass takes about 36 ms or less.

Keywords: multi-task learning; driverless systems; lane image detection; convolution neural network

1. Introduction

The perception module is one of the most important modules in an autopilot system, which is to identify key information in the driving scene and understand the road environment. For example, driving area detection, lane line detection, vehicle detection, pedestrian and traffic light detection, real-time position and vehicle speed are all responsible for the perception module. At present, there are three schemes for the perception module of autonomous driving technology. One is based on light detection and ranging (LIDAR) sensors, such as driving radar: one kind is based on an infrared sensor, and another is based on visual perception, such as on-board cameras. Since the lidar solution and infrared sensor solution are more expensive than the vehicle camera solution, and the visual perception solution can also achieve very good results, the lane detection in this article uses a visual perception solution [1].

As a task of computer vision, vision-based lane detection can be divided into image classification, image semantic segmentation and target detection. In recent years, with the improvement in computer processing ability, the data-driven technology of deep learning has been successfully applied in various fields. Among them, convolutional neural network (CNN) has made a breakthrough in the task of computer vision [2], and target detection. Some scholars have begun to use deep learning to solve the problem of lane line detection. Seokju Lee et al. adopted a new grid method to label lane lines (a grid is a rectangular frame, and the lane line is a grid label composed of points on the line) and modeled the lane line detection problem as a regression problem [3]. The author designed a multi-task convolutional network structure (VPGNet) and used the “vanishing point” information to
further precisely limit the position of the lane line so that it can detect the lane line more accurately in real time. Pan et al. proposed a spatial convolutional neural network (SCNN), which explores the spatial correlation between row and column pixels in an image [4,5]. The authors also considered the rows and columns in the feature map as network layers, and thus designed a new network layer structure suitable for transmitting information in image rows and columns, realizing a convolutional network model suitable for detecting targets with slender continuous shapes (such as lane lines). In [6], a method that can directly predict the lane was proposed, which uses the differentiable property of the least squares method and combines the depth neural network to realize the end-to-end training of the lane detection network. Ref. [7] proposed a coefficient space convolutional neural network (SSCNN) based on visual deep learning. SSCNN has greatly improved the processing speed of lane line recognition compared with existing spatial CNN methods. Xiao et al. proposed an attention module (AMSC), which combines self attention and channel attention in parallel by using learnable coefficients, and applied it to the LargeFOV algorithm. An attention DNN (modified LargeFOV) for lane marking detection was proposed, which also has good performance in lane detection [8]. It can be seen that the lane detection algorithm based on deep learning and is in the ascendant [9]. However, part of the method based on deep learning is to collect specific data sets and design specific neural network structures for this purpose, so this method is generally not very generalizable [10].

The traditional lane line detection method is mainly based on image processing technology to compare the lane road with the surrounding environment, using threshold segmentation to extract the effective features [11]. However, this method requires that the lane line has obvious characteristics, compared with the surrounding environment, and it is easy to be affected by the damage of pavement occlusion, with low accuracy. Deep learning refers to the construction of a multi-layer hidden layer artificial network model, combined with massive data sets, to extract more essential features of the target so as to improve the accuracy of the classification [12]. In this study, the network structure of three task branches is used to complete lane image detection combined with deep learning theory. Compared with the traditional lane line detection method, the lane line detection method based on deep learning has improved the robustness and accuracy, but the demand for data is also higher [13].

2. Method

2.1. Image Preprocessing

Image preprocessing refers to the appropriate processing of the image in the data set. There is a lot of sky and other areas of the lane line in the data set, which will not only prolong the calculation time of the system detection, but also affect the accuracy of the lane line detection. In order to extract important features from the image for the classification decision, the image must be preprocessed [14] so that the processed image can meet the requirements of the lane line detection method of deep learning and save computing resources as much as possible. In this study, we will preprocess the image from the steps of image gray processing, target area extraction, image scaling, reverse perspective conversion and image flipping.

2.1.1. Image Grayscale

Since this study is not based on color information recognition, there is no need to use a color image. We grayscale the image, which not only greatly reduces the amount of calculation, but also makes the image easier to understand and analyze. All pixel information of the grayscale image is described by a quantized gray level, without color information. All pixel information of a color image (such as RGB) is composed of RGB, three primary colors. RGB is described by three different gray levels. The operation that we use to transform a color image into a grayscale image is called image grayscale. For an RGB image, when three components are the same, it is a grayscale image. The weighted average method is used to gray the color image. According to a certain weight, the values
of R, G and B are a weighted average, that is, \( R = G = B = \left( \frac{0.299R + 0.587G + 0.114B}{3} \right) \), where 0.299, 0.587 and 0.114 are the weights of R, G and B, respectively.

2.1.2. Target Area Extraction

ROI is used to capture the area of interest, i.e., the lane line area, to generate a new image. By intercepting the target area, the redundant information of the image is reduced, and the key area of the image is highlighted. As shown in Figure 1, the sky accounts for a large proportion, so using the image directly for neural network calculation is not conducive to the efficiency of the algorithm. Using ROI for image extraction is important to reduce false path detection and improve the computational efficiency [15].

![Figure 1. Comparison before and after image capture.](image)

2.1.3. Picture Zooming

If the ROI extracted image is still too large, resulting in a waste of system resources, the nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation can be used to scale the image, also known as image resampling. Among them, the nearest neighbor interpolation method has a small amount of calculation, but it may appear jagged in the changed place. The bilinear interpolation method overcomes the shortcomings of the nearest neighbor interpolation method, but it may also have blurred image contours, so the bicubic interpolation is a more appropriate algorithm for image scaling.

\[
B(x', y') = A(x \times (m / M), y \times (n / N)) = A(x / K, y / K) \tag{1}
\]

\[
P_{B(i',j')} = \sum_{i=0}^{3} \sum_{j=0}^{3} p_{ij} \times W(i) \times W(j) \tag{2}
\]

\[
W_{BiCubic}(d) = \begin{cases} 
(c + 2)|x|^3 - (c + 3)|x|^2 + 1, & \text{for } |d| \leq 1 \\
|c|x|^3 - 5c|x|^2 + 8c|x| - 4c, & \text{for } 1 < |d| < 2 \\
0, & \text{otherwise}
\end{cases} \tag{3}
\]

The basic principle is as follows, assuming that the size of the source image \( A \) is \( m \times n \), and the target image \( B \) after scaling \( K \) times is \( M \times N \), and the corresponding coordinate conversion is Equation (1). Each pixel of source image \( A \) is known, and the pixels of target image \( B \) are unknown. Each pixel value of target image \( B \) is obtained by weighted superposition of the nearest 16 pixel points \( p_{ij} \ (i, j = 0, 1, 2, 3) \) around the corresponding pixel point in source image \( A \), as shown in Equation (2). The weight is obtained by bicubic Equation (3), where the parameter \( d \) is the distance from the pixels around the target to the target pixels. \( c \) usually takes a value of \(-0.5\).

\[
\begin{align*}
\text{rFactor} &= \left(1 - \frac{2U}{M-1}\right) \times \tan(AlphaV) \\
\text{cFactor} &= \left(1 - \frac{2V}{N-1}\right) \times \tan(AlphaU)
\end{align*} \tag{4}
\]
\[
\begin{align*}
X_0(r,c) &= C_z \times \frac{1 + rFactor \times \tan(\theta)}{\tan(\theta) - rFactor} + C_x \\
Y_0(r,c) &= C_z \times \frac{cFactor \times \cos(\theta)}{\tan(\theta) - rFactor} + C_y \\
\end{align*}
\] (5)

\[
\begin{align*}
\text{Alpha}_U &= \arctan\left(\frac{W}{2 \times F}\right) \\
\text{Alpha}_V &= \arctan\left(\frac{H}{2 \times F}\right) \\
\end{align*}
\] (6)

\[
\begin{align*}
X(r,c) &= X_0(r,c) \times \cos(\gamma) + Y_0(r,c) \times \sin(\gamma) \\
Y(r,c) &= X_0(r,c) \times (-\sin(\gamma)) + Y_0(r,c) \times \cos(\gamma) \\
\end{align*}
\] (7)

2.1.4. Inverse Perspective Transformation

If the lane presents a convergence state, so that the original image is distorted by perspective, in order to reduce this distortion, an IPM (inverse perspective mapping) image is obtained by inverse perspective transformation, also known as the aerial view [16]. Compared with the original image, IPM presents the characteristics of the real world without any deformation. The collected lane lines are parallel and equal in width, which can greatly improve the accuracy of detection, as shown in Figure 2. The conversion principle of the original image to the bird’s-eye view is the image coordinate system to the world coordinate system. The conversion of each pixel is described by Equations (4) and (5); \((X, Y, Z)\) represents the world coordinate system; \((U, V)\) represents the image coordinate system; \(C = (C_x, C_y, C_z)\) is the position of the camera in the world coordinate system; \(\text{Alpha}_V\) and \(\text{Alpha}_U\) respectively represent the angle aperture of the camera in the vertical and horizontal directions, calculated by Equation (6); \(F\) is the focal length of the camera; and \(H \) and \(W\) are the height of the photosensitive element of the camera. If the road image has a certain turning angle, \(X\) and \(Y\) also need to be multiplied by the sine or cosine of the compensation angle.

![Figure 2. Image inverse perspective transformation.](image)

2.1.5. Picture Flip

According to specific requirements, we need a picture flipping operation; there are three ways, namely, horizontal flipping, vertical flipping and horizontal vertical flipping. Lane line images under different traffic rules are different, so lane line detection under different traffic rules can be realized by turning the images left and right, that is, horizontal flipping, and the lane line database can also be enriched. The horizontal flipping of the
image refers to the linear transformation of the image, with the image vertical coordinate being the center to map the image flipping.

2.2. Basic Network Selection

A convolution neural network is developed from the neural network, which not only classifies the image, but also solves the regression problem. The combination of the two can achieve the purpose of image detection. In this study, the last several layers of convolutional neural network are changed to realize multi task learning. Multi task learning is the sharing of basic network feature extraction, so the selection of the basic network is very important. In this study, ZFNet is used to replace the basic networks of VPGNet, and their structures are changed, so the efficiency is improved.

ZFNet Network

ZFNet was proposed by Zelier et al. in 2013. ZFNet slightly changes the structure of AlexNet. We make the first five layers of ZFNet replace the basic network of VPGNet. Zelier believed that the step size and convolution kernel of the first layer of AlexNet network are too large, so the step size is changed from $4 \times 4$ to $2 \times 2$, and the convolution kernel is changed from $11 \times 11$ to $7 \times 7$. In order to ensure the consistency of the data input and output, the network parameters of some layers of ZFNet are modified [17]. In addition, the visualization of the first layer of convolution kernel has a large impact, so the first layer of the convolution kernel is normalized. Because of the smaller step size and convolution kernel of the first layer of ZFNet and the lower sampling degree of the input data, the whole network needs to occupy more video memory in training. In order to utilize GPU resources reasonably, the batch processing of ZFNet network training is reduced. After modification, the classification effect of ZFNet is improved obviously.

3. Network Model Construction

3.1. Multi Task Structure

In reality, many learning tasks are related and have shared knowledge patterns. Multi-task learning is a machine learning framework that enables different learning tasks to share common knowledge and maintain their independence so as to improve the generalization performance, model learning efficiency and prediction accuracy of all tasks [18]. It adopts a FCN (fully convolutional network)-like network structure with three task branches and does not sample the picture completely. The first five layers sample the image 32 times down, and then use $1 \times 1$ convolution to sample the image 4 times up, so the processing advantage is faster than the calculation speed of full up-sampling, as shown in Table 1 and Figure 3. The network structure has three task modules: multi-label classification, grid box regression and object mask.

| Layer | Convolution Kernel | Pooling Layer | Additional Operation | Receptive Field |
|-------|--------------------|---------------|----------------------|-----------------|
| Conv1 | 11, 4, 0           | 3, 2          | LRN                  | 11              |
| Conv2 | 5, 1, 2            | 3, 2          | LRN                  | 51              |
| Conv3 | 3, 1, 1            | -             | -                    | 99              |
| Conv4 | 3, 1, 1            | -             | -                    | 131             |
| Conv5 | 3, 1, 1            | 3, 2          | -                    | 163             |
| Conv6 | 6, 1, 3            | -             | Dropout              | 355             |
| Conv7 | 1, 1, 0            | -             | Dropout, Branched    | 355             |
| Conv8 | 1, 1, 0            | -             | Branched             | 355             |
ZF-VPGNet performs three tasks: multi-label classification, object mask and grid box regression. The following are descriptions of these three tasks:

- **Multi-Label classification task module**
  The multi-label classification is used to classify the images, and the probability map with the size of $60 \times 80 \times$ channel is the output. The coordinates of each channel with a probability greater than 0.5 are taken as the output of different categories, and up to four different categories of lane lines can be output [5].

- **Object Mask task module**
  Object mask is a mask detector module, which makes the $4 \times 4$ window slide through the whole image. The window area is the center of the image. Object mask defines the minimum area that can be resolved by the neural network. Although some small objects may be ignored, it does not need to regress point by point, but regresses the $4 \times 4$ grid module, which can improve the operation speed of the system [19].

- **Grid box regression task module**
  Grid box regression is used to detect and locate the lane line [3]. There are many adjacent grid modules on each lane. The distance regression method is used to return the adjacent grid modules to a target.

### 3.2. Data Layer
In this study, a grid is used to transform the point annotation of the lane line into grid annotation so as to increase the feature information of the lane line. The traditional detection algorithm uses a rectangle frame to mark the target, and one frame represents one object. However, in this study, due to its special location and shape, the lane line is not suitable to be labeled with a single frame, so many $8 \times 8$ grid modules are used to label the lane line. The adjacent grid module returns to an object, that is, a lane line. Each grid module on the lane line is similar to the rectangular box marking of single object detection, which enables the network to locate other objects on the lane at the same time.

### 3.3. Network Model Compression
A driverless vehicle system needs a mobile embedded platform, so a lane detection algorithm also needs to run in the mobile embedded platform. Although the traditional network model has high precision, it requires a lot of memory and parameters, so it is
not suitable for the embedded platform. However, in practical application, it usually is necessary to make some trade-offs. If the algorithm is applied to the mobile platform, memory consumption is the primary optimization goal, and a certain precision can be sacrificed in exchange for a smaller network model.

There are four main network compression methods: one is parameter pruning and sharing. In this study, based on the VPGNet network, the channel pruning method is used to compress the network model. The second is compact convolution filtering. By using small convolution kernels and conceptual structures, such as GoogleNet, multiple small convolution kernels can be combined together to achieve the effect of approaching large convolution kernels, while reducing parameters and computation. There are also two methods, namely, low rank factor decomposition and knowledge distillation [20].

3.4. Multi-Task Loss Function

The conventional method of multi-task learning loss calculation is to simply add the loss of each task or set a uniform loss weight. Further, the weight adjustment may be performed manually. The above methods are inefficient because different tasks lose the scale difference, which may be so large that the overall loss is dominated by a certain task. In order to ensure that the loss function can correctly affect the learning process of the network sharing layer, we consider the homoscedastic uncertainty between each task to set the weight of the loss function of different tasks, as shown in Formula (7).

Let us assume that \( f_{i}^{W}(x) \) is the output of a neural network with a weight of \( W \) on the input \( x \), \( \sigma \) is the observed network model noise parameter, and \( L_{i} \) is the loss of different tasks, \( L(W, \sigma_{1}, \sigma_{2}, \sigma_{3}) \) is the multi-task loss function of the network model. The training loss of different tasks is shown in Figure 4.

\[
L(W, \sigma_{1}, \sigma_{2}, \sigma_{3}) = \frac{1}{2\sigma_{1}^2}L_{1}(W) + \frac{1}{2\sigma_{2}^2}L_{2}(W) + \frac{1}{2\sigma_{3}^2}L_{3}(W) + \log \sigma_{1} \sigma_{2} \sigma_{3}
\]

\[
L_{1}(W) = \|y_{1} - f_{i}^{W}(x)\|^2
\]

\[
\theta = \theta - \frac{lr}{\sqrt{L_{1} + \epsilon}} \frac{\partial L_{1}}{\partial \theta} - \frac{lr}{\sqrt{L_{2} + \epsilon}} \frac{\partial L_{2}}{\partial \theta} - \frac{lr}{\sqrt{L_{3} + \epsilon}} \frac{\partial L_{3}}{\partial \theta}
\]

![Figure 4. The training loss of different tasks.](image-url)
4. Experiments

4.1. Data Composition

A deep learning neural network needs a lot of data for training, and there are few open lane databases at present. In this study, 1225 pictures published by Caltech and 5599 pictures published by Tucson are used. Caltech’s database is the most suitable lane line database for this study at present, which collects images of urban roads in four scenarios, as shown in Table 2 [21]. The data set contains many complicated data, such as interference signs, intense lights, shadows, and a large number of vehicles. Compared with the expressway, urban roads have more kinds of lane lines, more kinds of shelters and more complex road conditions. Tucson's published rough labeled lane line data set adopts the scene pictures when the vehicle is driving. It collects 20 frames of images for each second of driving video, and only marks the last image of each second. The database contains a large number of complex driving scenarios, including lane line images under good and moderate weather conditions, road lane images with 2/3/4 or more lanes, and images under different traffic conditions; the data set also contains about 30% curvilinear lane lines, so it is very suitable for this study.

| Name     | Cordoval | Washington |
|----------|----------|------------|
| Number   | 250      | 406        | 336        | 232        |
| Traffic  | Interference signs, urban roads | Intense light | Shadow, large number of vehicles | Road markings |

4.2. Evaluation

We use the F1 score as an evaluation index to determine whether the lane mark is correctly detected. First, we calculate the intersection-over-union (IoU) between the ground truth and the prediction. Remember the true positive (TP) as the prediction that IoU is greater than a specific threshold, and vice versa as false positive (FP). At the same time, we record true negative (TN) as having no lane and predicting no lane, and false negative (FN) as having a lane but predicting no lane. Then, we employ \( F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \) as the final evaluation index, where \( \text{Precision} = \frac{TP}{TP + FP} \) and \( \text{Recall} = \frac{TP}{TP + FN} \). At the same time, we also use runtime (unit: ms) and frames per second (FPS) as the evaluation index of the comprehensive performance of the system.

4.3. Results and Discussion

We choose the ZFNet network as the basis of the network structure and customize and optimize the network to further reduce the complexity of the network. By observing the weights of the convolution kernel during the training of the network model, it is found that there are a large number of weights close to 0 in the 3rd to 8th convolutional layers of the network. Network sparsity brings a small amount of network performance improvement, but with it, there is an exponential increase in computation. After multiple training tests, we try to cut the number of input convolution kernels of the 3rd to 8th convolutional layers of the above two networks without excessively reducing performance, reducing the number of network weights close to 0 in the retrained network model, and we finally obtain a high-performance network model.

Ubuntu 16.04 is used as the running platform, training with GPU RTX2080Ti; caffe is the deep learning framework; the learning rate is the adaptive learning rate (\( \epsilon \) is a small positive number to prevent division by zero divergence, Equation (10)); and the Adam gradient descent algorithm is used for network training. The lane line detection results are obtained through multitask network output, and the network output results of the Caltech and Tucson data sets are shown in Figure 5. Red in the Caltech data set represents the white dotted line, yellow represents the white solid line and blue represents the yellow
solid line. In the Tucson data set, red represents the white solid line, green represents the white dotted line, and yellow represents the yellow solid line.

![Figure 5. Experiments prediction results.](image)

It can be seen from the results that the system is robust to the interference of ground obstacles, light changes, occlusion, shadow and so on. The fuzzy lane line and missing lane line mark are shown in Figure 5c. The improved algorithm can still be used to detect and obtain the correct results, which shows the feasibility of the algorithm. On the basis of the test results, the fitting operation can be carried out. However, it is difficult to fit when two different types of roads or prediction results are wrong, and further fitting by category is needed. Because the data in this study are not rich enough, there is an abnormal situation that there is no output, but there is a marker when there is no road, which requires more data sets and the classification of road markers to eliminate this impact.

In order to test the superiority of the algorithm, part of Cordoval1 and Washington1 and Tucson scenarios in the open database of Caltech are selected for testing, and the comprehensive index F1 score is used as the evaluation index. The test results are shown in Table 3 and Figure 6.

### Table 3. F1 score performance of different network models on Cordoval1 and Washington1.

| Network          | Cordoval1 | Washington1 |
|------------------|-----------|-------------|
| VPGNet           | 0.884     | 0.869       |
| three-VPGNet     | 0.835     | 0.849       |
| Compressed-Net   | 0.874     | 0.843       |
| Caltech          | 0.723     | 0.759       |
| DriveNet         | 0.866     | 0.848       |
| ZF-VPGNet        | 0.922     | 0.876       |
| CZF-VPGNet       | 0.874     | 0.843       |
It can be concluded from Table 3 that the performance of the lane line detection algorithm based on the improved network (ZF-VPGNet) is much higher than the traditional lane line detection algorithm. On Cordoval1, our ZF-VPGNet improved its F1 score by 3.8%, compared with other excellent network models; on Washington1, it has also improved to some extent, compared with other excellent network models.

It can be seen from the data set that the Washington1 scene is more complicated than the Cordoval1 scene and the experimental results are consistent; the complex road conditions and road images under shadow have higher requirements on the network model. As shown in Figure 6, although the compression of the network model leads to a decline in detection performance, the compressed network model CZF-VPGNet has higher real-time performance (26 fps) and takes less time (36 ms) for a single forwarding. The compressed network model has higher performance in terms of comprehensive operation speed and the proportion of memory.

5. Conclusions

By using the improved network labeling in this paper, the feature information of the lane line is added, and the network can locate other objects on the lane line. Using improved ZFNet and VGGNet networks instead of DriveNet and VPGNet networks, the experiments show that the improved network classification effect is better. At the same time, in order to transplant the neural network to embedded devices, the VPGNet network is compressed. We design a new convolution method to reduce the network parameters and calculations. The accuracy of the compressed network model does not change significantly, and the running speed is improved. Under the same conditions, the recognition rate of this algorithm is high, and at the same time, the balance between recognition rate and running time is basically achieved. In order to verify the feasibility of the improved network lane detection algorithm and the performance of the compressed network model, this research is only carried out under usual road conditions. In the future research, we will continue our research under more complex road conditions (such as bad weather, intersections, etc.), and we will conduct research on a random, untargeted, adversarial example [22] on the
basis of this research. This is because in autonomous vehicles, some adversarial examples are likely to cause fatal accidents.

**Author Contributions:** Conceptualization, J.L.; Methodology, J.L. and D.Z.; Investigation, J.L.; Visualization, Y.M.; Project administration, D.Z. and Q.L.; Writing—original draft preparation, J.L., D.Z. and Y.M.; Writing—review and editing, J.L., D.Z., Y.M. and Q.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been supported by (i) National Nature Science Foundation of China (NSFC) under Grant Nos. 61402397, 61263043, 61562093 and 61663046; (ii) Yunnan Provincial Young Academic and Technical Leaders Reserve Talents under Grant No. 2017HB005; (iii) Yunnan Provincial Innovation Team under Grant No. 2017HC012; (iv) Youth Talent Project of China Association for Science and Technology under Grant No. W8193209; (v) Science Foundation of Yunnan University under Grant No. 2017YDQN11; (vi) Yunnan Provincial Science Research Project of the Department of Education under the Grant No. 2018J5008; and (vii) Open Foundation of Key Laboratory in Soft-ware Engineering of Yunnan Province under Grant No. 2020YSE304.

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

**References**

1. Chen, W.; Wang, W.; Wang, K.; Li, Z.; Li, H.; Liu, S. Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: A review. *J. Traffic Transp. Eng.* **2020**, *7*, 748–774. [CrossRef]

2. Tang, J.; Li, S.; Liu, P. A review of lane detection methods based on deep learning. *Pattern Recognit.* **2020**, *111*, 107623. [CrossRef]

3. Lee, S.; Kim, J.; Yoon, J.S.; Shin, S.; Bailo, O.; Kim, N.; Lee, T.-H.; Hong, H.S.; Han, S.-H.; Kweon, S. VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 22–29 October 2017.

4. Gao, D.; Zhang, X. Saliency Detection Based on Spatial Convolutional Neural Network Model. *Comput. Eng.* **2018**, *4*, 240–245.

5. Pan, X.; Shi, J.; Luo, P.; Wang, X.; Tang, X. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, New Orleans, LA, USA, 2–7 February 2018.

6. Van Gansbeke, W.; de Brabandere, B.; Neven, D.; Proessmans, M.; van Gool, L. End-to-end Lane Detection through Differentiable Least-Squares Fitting. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* IEEE, Seoul, Korea, 27 October–2 November 2019.

7. Oh, M.; Cha, B.; Bae, I.; Choi, G.; Lim, G.C.A.Y. An Urban Autodriving Algorithm Based on a Sensor-Weighted Integration Field with Deep Learning. *Electronics* **2020**, *9*, 158. [CrossRef]

8. Xiao, D.; Yang, X.; Li, J.; Islam, M. Attention deep neural network for lane marking detection. *Knowl.-Based Syst.* **2020**, *194*, 105584. [CrossRef]

9. Li, W.; Qu, F.; Liu, J.; Sun, F.; Wang, Y. A lane detection network based on IBN and attention. *Multimed. Tools Appl.* **2019**, *79*, 16473–16486. [CrossRef]

10. Linjordet, T.; Balog, K. *Impact of Training Dataset Size on Neural Answer Selection Models*; Springer: Cham, Switzerland, 2019.

11. Chakkaravarthy, A.P.; Chandrasekar, A. An Automatic Threshold Segmentation and Mining Optimum Credential Features by Least-Squares Fitting. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* IEEE, Seoul, Korea, 27 October–2 November 2019.

12. Oh, M.; Cha, B.; Bae, I.; Choi, G.; Lim, G.C.A.Y. An Urban Autodriving Algorithm Based on a Sensor-Weighted Integration Field with Deep Learning. *Electronics* **2020**, *9*, 158. [CrossRef]

13. Ni, J.; Chen, Y.; Chen, Y.; Zhu, J.; Ali, D.; Cao, W. A Survey on Theories and Applications for Self-Driving Cars Based on Deep Learning Methods. *Appl. Sci.* **2020**, *10*, 2749. [CrossRef]

14. Ghani, H.A.; Besar, R.; Sani, Z.M.; Kamaruddin, M.N.; Syahali, S.; Daud, A.M.; Martin, A. Advances in lane marking detection algorithms for all-weather conditions. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 2088–8708.

15. Murl, M.J.; Aziz, N.H.A.; Ghani, H.A. A Review on Deep Learning and Nondeep Learning Approaches for Lane Detection System. In *Proceedings of the IEEE 8th Conference on Systems, Process and Control (ICSCPC)*, Melaka, Malaysia, 11–12 December 2020; pp. 128–144. [CrossRef]

16. Ghanei, S.; Kanungo, P.; Panda, G.; Parwekar, P. An improved and low-complexity neural network model for curved lane detection of autonomous driving system. *Soft. Comput.* **2021**, *1–12. [CrossRef]

17. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014.

18. Torres, J.; Bai, G.; Wang, J.; Zhao, L.; Vaca, C.; Abad, C. Sign-regularized Multi-task Learning. *arXiv* 2021, arXiv:2102.11191.

19. Han, J.; Ma, Y.; Zhou, B.; Fan, F.; Liang, K.; Fang, Y. A Robust Infrared Small Target Detection Algorithm Based on Human Visual System. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 2168–2172. [CrossRef]

20. Bejani, M.M.; Ghataye, M. A systematic review on overfitting control in shallow and deep neural networks. *Artif. Intell. Rev.* **2021**, *1–48. [CrossRef]
21. 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. [CrossRef]

22. Kwon, H.; Kim, Y.; Yoon, H.; Choi, D. Random Untargeted Adversarial Example on Deep Neural Network. *Symmetry* 2018, 10, 738. [CrossRef]