A Vision-based Lane Detection Technique using Deep Neural Networks and Temporal Information

Chun-Ke Chang[1] and Huei-Yung Lin[2][a]

[1]*Department of Electrical Engineering, National Chung Cheng University, Chiayi 621, Taiwan*

[2]*Department of Electrical Engineering and Advanced Institute of Manufacturing with High-tech Innovations, National Chung Cheng University, Chiayi 621, Taiwan*

Keywords: Advanced Driver Assistance System, Lane Detection, Convolutional Neural Network.

Abstract: With the advances of driver assistance technologies, more and more people begin to pay attentions on traffic safety. Among various vehicle subsystems, the lane detection module is one of the important parts of advanced driver assistance system (ADAS). Traditional lane detection techniques use machine vision algorithms to find straight lines in road scene images. However, it is difficult to identify straight or curve lane markings in complex environments. This paper presents a lane detection technique based on the deep neural network. It utilizes the 3D convolutional network with the incorporation of temporal information to the network structure. Two well-known lane detection network structures, PINet and PolyLaneNet, are improved by integrating 3D ResNet50. In the experiments, the accuracy is greatly improved for the applications to a variety of different complex scenes.

1 INTRODUCTION

With the recent of advances of deep learning technology, many related techniques have been developed for various real world applications. Among them, the autonomous driving or advanced driver assistance system (ADAS) have attracted much more attentions for the researchers and practitioners in automotive industries. The lane departure warning system is one of the most important modules for current driver assistance functions. It is used to automatically notify the driver when the vehicle deviates from the center of the original lane (Lin et al., 2020). Thus, some possible traffic accidents can be avoided effectively.

As illustrated in Figure 1, the lane departure warning system requires to detect the front lane markings on the road, presumably using an on-board camera installed inside the vehicle. It can be used to determine the distance between the lane markings and the vehicle. When the driver inadvertently deviates from the lane, the system will send out a timely warning signal. The lane deviation information can also be adopted by autonomous vehicles as the feedback signals for driving control. Consequently, it can be considered as the first stage towards the safety issues related to the lane keeping for unmanned driving.

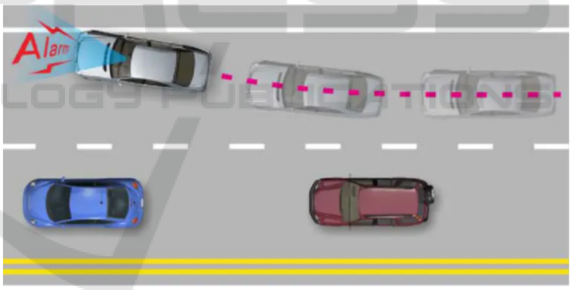


Figure 1: An illustration of the lane departure warning for driver assistance.

The detection of lane markings is essential for lane keeping or departure warning. In the past decades, the traditional lane detection methods were implemented based on the Hough transform, with a post-processing step for lane marking identification. In the early work, Aly performed an inverse perspective mapping (IPM) on the input image to detect straight lines, and utilized the RANSAC algorithm to evaluate the lane markings (Aly, 2008). Although this approach is fairly efficient, it highly depends on the real scene environment. The lane detection results will be seriously affected when there are occlusions by the vehicles.

The recent advances on deep learning approaches have improved the accuracy of lane detection results,

[a] https://orcid.org/0000-0002-6476-6625

Chang, C. and Lin, H.
A Vision-based Lane Detection Technique using Deep Neural Networks and Temporal Information.
DOI: 10.5220/0010973200003191
In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2022)*, pages 172-179
ISBN: 978-989-758-573-9; ISSN: 2184-495X

compared to the traditional methods. There exist several important network structures, including LaneNet (Neven et al., 2018), SCNN (Pan et al., 2018), PointLaneNet (Chen et al., 2019) and Line-CNN (Li et al., 2019), for the deep learning based lane detection techniques. Most of the existing methods use a single image for lane marking detection. This is, however, not stable enough due to the continuity nature of the sequential image input. To take this temporal information into consideration, SegnetConvLSTM (Zou et al., 2019) utilized the semantic segmentation from Segnet (Badrinarayanan et al., 2017) and added ConvLSTM (Shi et al., 2015) to detect lanes. Although the detection of road markings are relatively stable compared to the single image processing, there are some limitations on semantic segmentation. It requires the lane markings or pre-processing of image pixels for training. This generally takes more computational time for real-time processing and is relatively expensive.

2 RELATED WORK

Most recent works on lane detection are based on the semantic segmentation of input images. The method aims to divide the input images into different regions with their own categories. It is also capable of the expression of complex-shaped curves according to various generative models. In the existing literature, (Pan et al., 2018; Hou et al., 2019; Lo et al., 2019; Ghafoorian et al., 2018) have demonstrated the application of semantic segmentation in the lane detection tasks. In addition, there are also some algorithms using multiple categories to distinguish real instances. Although instance segmentation is able to deal with the multi-classification problem, multiple categories can only be classified for fixed instances in the approaches.

Neven *et al.* proposed a LaneNet network model which attempts to use instance segmentation to solve the problem of different lane categories (Neven et al., 2018). It contains a shared encoder with two types of decoders. There are two branches in the architecture, one performing the semantic division of the input images and the other predicting the embedding features for instance segmentation. The lane segmentation result is then formed by combining the last two branch outputs.

Pan *et al.* proposed a spatial CNN (convolutional neural network), SCNN, for traffic scene understanding (Pan et al., 2018). It utilized a CNN-like scheme to provide an effective information propagation in the spatial level. One important characteristic of SCNN is its capability to preserve the continuity of long thin structures. The network is also able to work with the

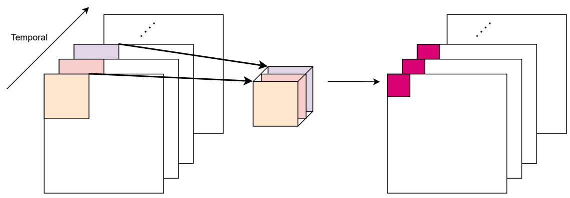


Figure 2: The 3D ConvNets (or 3D CNN) used to extract the time and space information from an image sequence.

LargeFOV model (Chen et al., 2018) to achieve a significant improvement. In Line-CNN (Li et al., 2019), the main component is the RPN (region proposal network) from Faster R-CNN. By adapting to a line proposal unit (LPU), the network can predict the starting points of the lane markings. The LPU will then draw the lines based on the fixed x-axis with the horizontal offset of the y-axis at the end point.

To consider the temporal information, the recurrent neural network (RNN) is a classic network architecture which can transfer the output calculated in the previous layer to itself as an input and add to the next sum. Thus, the network model is very suitable for the data with sequential correlation. One major drawback is that the information fed in at the beginning of the input sequence will be gradually forgotten after a long training time. Shi *et al.* proposed an LSTM network architecture with convolution operation, ConvLSTM, which was able to process image sequences with temporal information and widely used in video analysis. For semantic segmentation of images, the time series input can also be processed effectively.

Zou *et al.* proposed a hybrid neural network combining CNN and RNN for lane detection (Zou et al., 2019). It took multiple continuous frames as an input, and semantic segmentation of the lane was carried out by ConvLSTM. In the proposed framework, a CNN encoder was first used to extract the features of each input frame, followed by processing the sequentially encoded features of all input frames by a ConvLSTM. The encoder-decoder structure was adopted for information reconstruction and lane prediction by incorporating time and space dimensions. In addition to lane detection by semantic segmentation, ConvLSTM can also be adopted for various scene understanding tasks such as anomaly detection and passenger demand prediction, etc.

Another approach to incorporate time information to CNN is the 3D convolutional neural network with an additional time dimension. 3D ConvNets proposed by Tran *et al.* provides better spatiotemporal feature learning compared to the conventional 2D CNNs. As illustrated in Figure 2, 3D ConvNets (or 3D CNN) is able to extract the time and space information in the video more effectively by adding the frame by frame time domain information. In (Yuan et al., 2018), a lane keeping technique was proposed using a multi-state model. A 3DCNN-LSTM end-to-end model was trained for going straight and turning left/right decision. Nevertheless, they did not explicitly identify the lane markings for vehicle localization.

3 OUR APPROACH

The proposed technique for lane marking detection is based on PINet and PolyLaneNet, with the temporal information embedded for improvement.

PolyLaneNet and PINet

PolyLaneNet (Tabelini et al., 2021) is an end-to-end lane detection network. It is based on the deep polynomial regression output to represent the lane markings in the image using polynomial curves. In addition to generating the curve fitting for each lane, it also provides the confidence level. In PolyLaneNet, a preset maximum number of lanes needs to be defined, and the lane markings are determined by the start and end points of a lane, with a confidence value provided. The end-to-end lane detection network is not only fast in terms of the computational speed, but also not required to perform additional post-processing.

To detect the important features in the image, the key point estimation approach is commonly adopted. It is widely used in deep neural networks, such as for human pose estimation. In this paper, the key point estimation is used for hourglass network for lane detection. An hourglass block is a network architecture that can transmit various scales of information to each layer. It can then obtain the global and local characteristics of the entire network. For feature extraction, corner detection or the center of a regional object are used. Since the hourglass network can be stacked to make the network deeper, this characteristic is used to incorporate the key point estimation to generate a new frame for lane detection.

PINet (Ko et al., 2020) is a structure that utilizes two hourglass networks to synthesize, and combine key point estimation and point cloud segmentation to detect lane markings. The network adopts three output branches, namely confidence branch, lane offset branch and embedding branch. The confidence and lane offset branches are mainly used to predict the location of each lane, and the embedding branch is used to generate the characteristics of each predicted point. Different lane markings are distinguished according to their characteristics. PINet aims to work in general

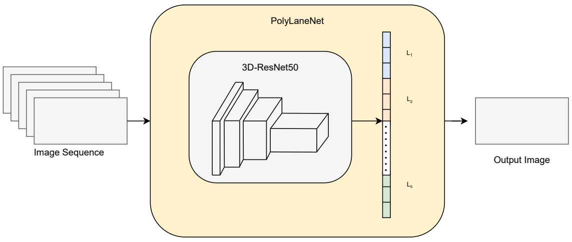


Figure 3: PolyLaneNet is improved with 3D ResNet by incorporating the temporal information.

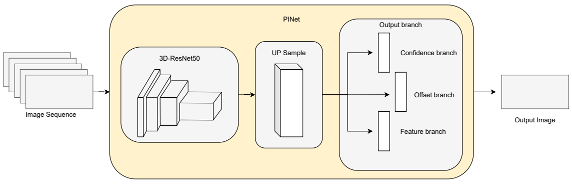


Figure 4: 3D ResNet is integrated with PINet for feature extraction.

scenes with any number of lanes. The model is lightweight and the output lane markings are presented in dots.

Improvement on PolyLaneNet and PINet

In the original PolyLaneNet, ResNet (He et al., 2016) was used for feature extraction. Based on the latest works, it is necessary to use video sequence as input instead of the image frames for lane detection. In this paper, we utilize the features in PolyLaneNet. As illustrated in Figure 3, 3D ResNet50 is adopted for feature extraction, with 5 consecutive image frames taken as the input for processing.

In the original PINet, two hourglass networks are adopted for feature extraction. To improve the recognition rate and stability of lane detection, the network backbone is changed to 3D ResNet50, as depicted in Figure 4. The image information along the time axis is included for robust lane detection. In our 3D ResNet implementation, 5 consecutive image frames are taken as input for feature extraction.

Post-processing

Different from the general representation of lanes using curves adopted by most algorithms, the outputs of PINet are represented using points. Since there might be noisy for the lane marking detection result, in this work we further adopt RANSAC for curve fitting and outlier removal. As the flowchart illustrated in Figure 5, it consists of four major steps:

1. The output points of the network are fitted with the second order RANSAC.

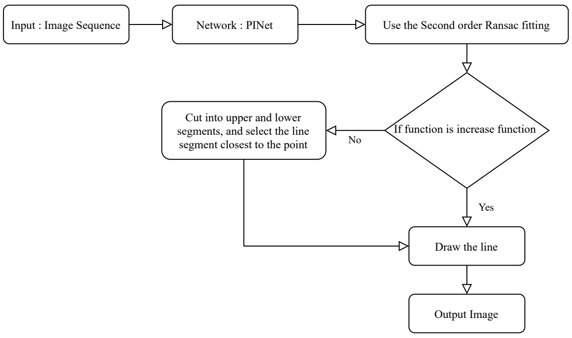


Figure 5: The system flowchart of our proposed lane marking detection algorithm.

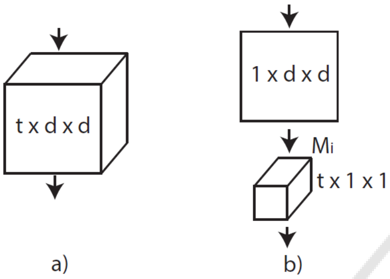


Figure 6: The original architecture of 3D CNN is split into spatial and time convolutions.

2. Check if the fitting results with quadratic equation are appropriate.
3. Separate the curve into upper and lower segments if the fitting is not smooth enough.
4. Take the line segments to approximate the points.

Improvement on 3D ResNet

In the past few years, the one-dimensional convolution has been applied to many convolutional architectures. It can be used to transform different channels, as well as increase or reduce channel dimensions. Recently, Tran *et al.* proposed a time-space split convolution architecture (Tran et al., 2018). They designed a R(2+1)D architecture, which changed the original 3D CNN to a 2D space convolution and a 1D time convolution. That is, to separate the time dimension from the space dimension in the 3D CNN. As illustrated in Figure 6, the original architecture of 3D CNN convolves time and space together with $t \times d \times d$. It can be split into the spatial convolution $1 \times d \times d$ and the time convolution product $t \times 1 \times 1$. Since the split R(2+1)D has a remarkable improvement compared to the original 3D CNN, it is adopted in this work for 3D ResNet50 architecture.

The Leaky ReLU is an improved function based on ReLU proposed by Xu *et al.* (Xu et al., 2015). In the original ReLU, a linear rectification function with the value set to 0 for $x < 0$ but remained unchanged for $x > 0$. The problem of using ReLU is that it tends to cause over-fitting. On the contrary, Leaky ReLU is defined with a non-zero slope when $x < 0$. Thus, it is adopted in this work for our ResNet implementation to avoid over-fitting.

4 EXPERIMENTS

One important objective of this work is considering the computational speed during the lane detection. It is also focused on the improvement of PolyLaneNet and PINet after incorporating 3D ResNet, including the accuracy and stability of lane marking detection in the urban areas. The algorithms are executed on a PC running Linux-Ubuntu-16.04, Python3.6, Pytorch 1.6.0, and Nvidia GTX 1080 8G. The training parameters are as follows: learning rate = 0.0003, batch size = 10.

Datasets

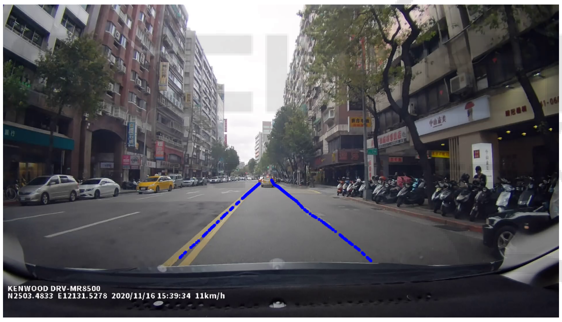
TuSimple is currently one of the most commonly used datasets for lane detection (Kai, 2017). The dataset is by far the easiest one for training since the images are mainly captured from highway driving. It consists of 3,626 video clips, with 20 image frames for each. In this work, we also collect image datasets from Taiwan road scenes by ourselves. The images are mainly acquired during daytime with normal weather conditions. In the urban areas, the road scenes are more complicated compared to the highway traffic. Our image recording contains totally 43 videos of 1-minute footage. The images are then taken for every 5 frames and result in 351 for each video. Moreover, we have also removed the traffic scenes with unclear lane markings such as the images captured near the crossroads and stopped for too long. Finally, there are 8,465 images in our dataset, and a sample image is shown in Figure 7.

After annotating all the lanes in our own dataset, it is found that some lanes have too few marking points. This might cause unsatisfactory results for lane detection due to the inadequate lane marking fitting for training. In terms of perspective distortion, the locations closer to the vanishing point require more lane markings and vice versa. Thus, the following procedures are carried out to provide more lane marking points for training data, and the comparison is shown in Figure 8.

- The originally marked lane points are connected

Figure 7: An image from our own road scene image dataset.

(a) The result using original training data.



(b) The results with additional lane marking points.

Figure 8: The comparison of lane marking detection results with different training data annotation.

to create a grayscale graphical image, and the connected lines are stored in the grayscale map.

- Set a threshold in the image vertical direction to store all markings above and random access a fixed number of markings below.
- Keep the lane marking points being accessed, so the number of marking points can be increased by 3 to 5 times.

Results

There contain 3,626 images in the TuSimple dataset, so it is split into 3,268 samples for training and 358 samples for testing. The accuracy of our lane marking detection is evaluated based on the x-axis value of the vertical 48 points in the testing data.

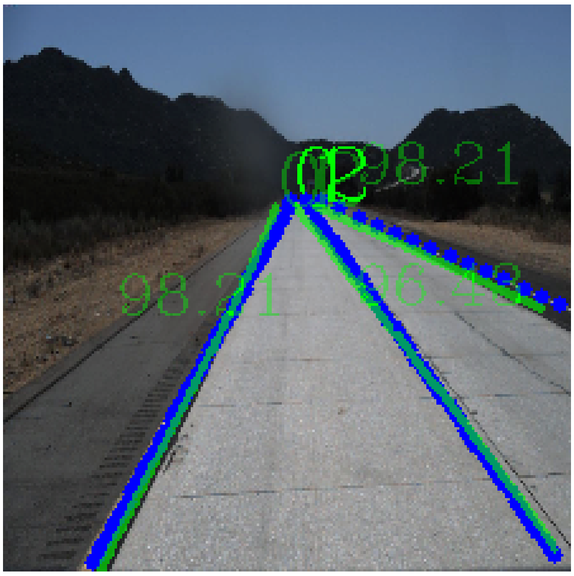


Figure 9: The result from PolyLaneNet 3D-ResNet 2P1D with the image size of 224×224.

For the improvement over PolyLaneNet, the time-space split convolution (2P1D) is added to the original 3D ResNet50. The original activation function is changed from ReLU to LeakyReLU, while the image input size is maintained at 224×224. The input images are taken at 10 frames at a time. Table 1 tabulates the comparison of parameters and results of the original and modified PolyLaneNet. The results show that when the size of images and the image sequence remain unchanged, the time dimension information can be captured more effectively during training due to the the time and space split convolution. Consequently, the overall accuracy can be improved significantly.

In Table 2, we tabulates the testing results of the PolyLaneNet-3D-ResNet under different image sizes and different numbers of input frame numbers. It also includes our improvement over 3D ResNet. When the images become smaller, we can find that the accuracy is decreased as the number of image frames is increased. Also, the improvement on the number of input frames for 10 images is much greater than using 5 images. By adding the convolution of space and time to the original 3D ResNet and changing ReLU to LeakyReLU, a significant improvement in terms of accuracy can be obtained.

Next, we evaluate the integration of PINet with the general 3D ResNet50 network architecture, as well as the general ResNet50 network architecture for comparison. The difference between the time and space split convolution with the LeakyReLU added to the 3D ResNet50 network architecture to show the effect

Table 1: The comparison of parameters and results of the original and modified PolyLaneNet.

	PolyLaneNet	PolyLaneNet-2P1D
Dataset	TuSimple	TuSimple
Train	3,268	3,268
Model	3D-ResNet50	3D-ResNet50-2P1D
Validation	358	358
Image size	224×224	224×224
Image sequence	10	10
Batch-Size	8	8
Epoch	2,656	2,588
Accuracy(%)	65%	70%

Table 2: The testing results of the PolyLaneNet-3D-ResNet under different image sizes and different numbers of input frame numbers.

Model	Image size	Image sequence	Accuracy
PolyLaneNet-3D-ResNet	360×640	5	67%
PolyLaneNet-3D-ResNet	224×224	10	65%
PolyLaneNet-3D-ResNet-2p1d-LeakyReLU	224×224	10	70%

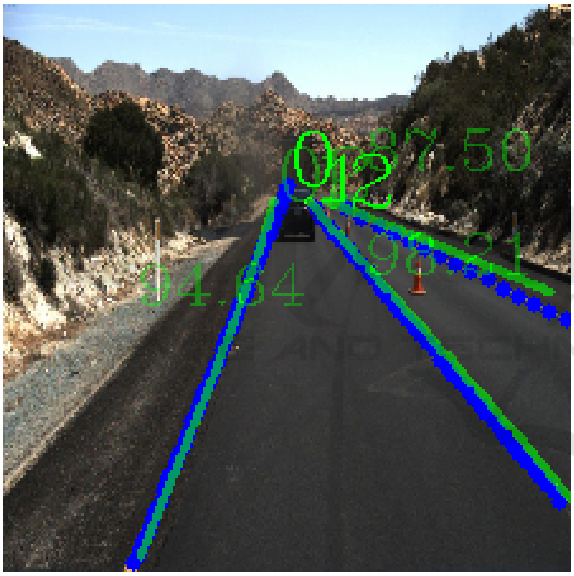
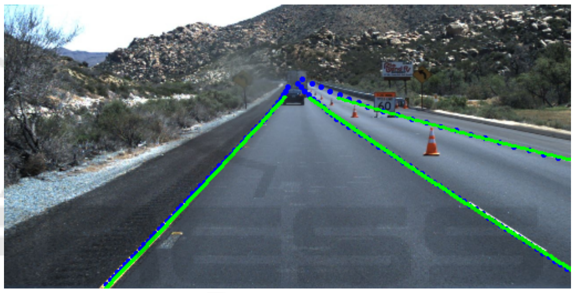


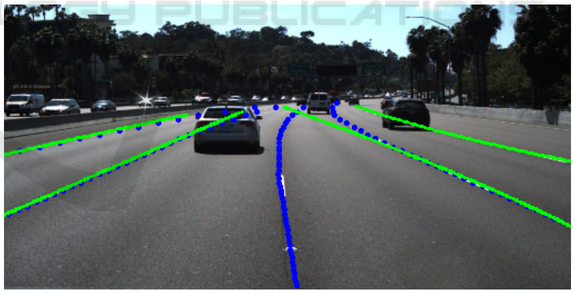
Figure 10: The result from PolyLaneNet 3D-ResNet 2P1D with the image size of 224×224.

of ResNet50 on our own dataset. The original PINet utilizes 2 hourglass networks for feature extraction. To compare with 3D ResNet50, the feature extraction part in PINet is changed to ResNet50 network architecture. As shown in Table 3, although the accuracy is slightly dropped, it is more reasonable to incorporate an image sequence as the network input for practical applications.

Figure 11(a) shows a result of PINet-ResNet50. The blue dots in the figure represent the actual lane markings of the TuSimple dataset, and the green lines represent the straight line generated by our detection network and RANSAC. The result demonstrates the effectiveness of straight lane detection using the



(a) The result of lane marking detection.



(b) The lane marking not detected in the middle is due to the vehicle's lane change.

Figure 11: The results from PINet ResNet50.

proposed model. The advantage of using image sequences as the inputs to the network is to avoid mis-detection due to the lane markings blocked by vehicles. Figure 11(b) shows the output when ResNet 50 is adopted as the network model. The lane marking not detected in the middle is due to the vehicle's lane change.

Finally, the comparison between ResNet50 and 3D ResNet50 is provided, as well as the results of the original 3D ResNet50 and the time-space split convo-

Table 3: The comparison of PINet-ResNet50 and PINet-3D-ResNet50.

	PINet-ResNet50	PINet-3D-ResNet50
Dataset	TuSimple	TuSimple
Train	3,268	3,268
Validation	358	358
Image size	512×256	512×256
Image sequence	1	5
Batch-Size	10	5
Epoch	273	272
Accuracy(%)	91%	89%

Table 4: The comparison of PINet-3D-ResNet50 and PINet-3D-ResNet50-2P1D-LeakyReLU.

	PINet-3D-ResNet50	PINet-3D-ResNet50-2P1D-LeakyReLU
Dataset	TuSimple	TuSimple
Train	3,268	3,268
Validation	358	358
Model	3D-ResNet50	3D-ResNet50-2P1D-LeakyReLU
Image size	512×256	512×256
Image sequence	1	5
Batch-Size	10	5
Epoch	273	272
Accuracy(%)	89%	91%

Table 5: The testing results of PINet.

Model	Accuracy	FPS
PINet-ResNet	91.3%	60
PINet-3D-ResNet	89.1%	16
PINet-3D-ResNet-2P1D-LeakyReLU	91.3%	12

Table 6: The comparison with different algorithms on TuSimple dataset.

Method	Acc	FP	FN
Line-CNN (Li et al., 2019)	96.87%	0.0442	0.0197
SCNN (Pan et al., 2018)	96.53%	0.0617	0.0180
PolyLaneNet (Tabelini et al., 2021)	93.36%	0.0942	0.0933
PINet (Ko et al., 2020)	93.36%	0.0467	0.0254
Ours (PolyLaneNet-3D-ResNet50-LeakyReLU)	70.06%	0.6081	0.5852
Ours (PINet-3D-ResNet50-LeakyReLU)	91.34%	0.1138	0.1101

lution with ReLU changed to LeakyReLU. As shown in Table 4, adding time and space split convolution and LeakyReLU have significantly improved the accuracy. The detection result of PINet-3D-ResNet50-2P1D network architecture is shown in Figure 12. The testing results for PINet is shown in Table 5, with different feature extraction networks. It is found that the execution speed of the PINet network under ResNet50 can achieve 60 fps, which is superior to the original network. Although the accuracy of 3D ResNet50 has slightly dropped, the main improvement of the network lies in its stability. Nevertheless, the accuracy is improved after adding time and space split convolution and LeakyReLU.

At present, most of the lane detection network architectures in the literature use a single image as input. Although the evaluation in Table 6 does not show clear accuracy improvement over other network models. However, this work presents an innovative network architecture which incorporates the temporal information for lane detection.

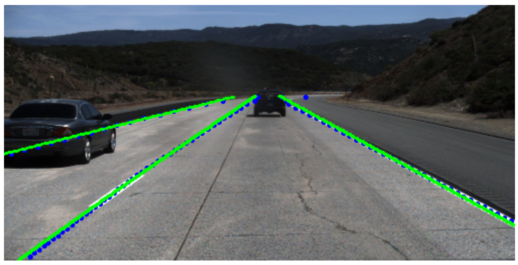


Figure 12: The lane detection result from PINet 3D-ResNet50-2P1D.

5 CONCLUSION

This paper presents a lane detection technique based on deep learning models with the use of temporal information. We improve the convolutional methods for the neural network architecture 3D ResNet50. The main contribution of this work consists of two parts, the first is incorporating the time axis with PINet and PolyLaneNet, and the other is the improvement on the 3D ResNet50 network model. In the experiments, the accuracy is greatly improved for the applications to a variety of different complex scenes.

ACKNOWLEDGMENTS

This work was financially/partially supported by the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, the Ministry of Science and Technology of Taiwan under Grant MOST 106-2221-E-194-004 and Create Electronic Optical Co., LTD, Taiwan.

REFERENCES

Aly, M. (2008). Real time detection of lane markers in urban streets. In *2008 IEEE Intelligent Vehicles Symposium*, pages 7–12. IEEE.

Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848.

Chen, Z., Liu, Q., and Lian, C. (2019). Pointlanenet: Efficient end-to-end cnns for accurate real-time lane detection. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2563–2568. IEEE.

Ghafoorian, M., Nugteren, C., Baka, N., Booij, O., and Hofmann, M. (2018). El-gan: Embedding loss driven generative adversarial networks for lane detection. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hou, Y., Ma, Z., Liu, C., and Loy, C. C. (2019). Learning lightweight lane detection cnns by self attention distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1013–1021.

Kai, Z. (2017). Tusimple datasets. *https://github.com/TuSimple/tusimple-benchmark*.

Ko, Y., Jun, J., Ko, D., and Jeon, M. (2020). Key points estimation and point instance segmentation approach for lane detection. *arXiv preprint arXiv:2002.06604*.

Li, X., Li, J., Hu, X., and Yang, J. (2019). Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):248–258.

Lin, H. Y., Dai, J. M., Wu, L. T., and Chen, L. Q. (2020). A vision based driver assistance system with forward collision and overtaking detection. *Sensors*, 20(18):100–109.

Lo, S.-Y., Hang, H.-M., Chan, S.-W., and Lin, J.-J. (2019). Multi-class lane semantic segmentation using efficient convolutional networks. In *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE.

Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., and Van Gool, L. (2018). Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 286–291. IEEE.

Pan, X., Shi, J., Luo, P., Wang, X., and Tang, X. (2018). Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. *arXiv preprint arXiv:1506.04214*.

Tabelini, L., Berriel, R., Paixao, T. M., Badue, C., De Souza, A. F., and Oliveira-Santos, T. (2021). Polylanenet: Lane estimation via deep polynomial regression. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6150–6156. IEEE.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459.

Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Yuan, W., Yang, M., Li, H., Wang, C., and Wang, B. (2018). End-to-end learning for high-precision lane keeping via multi-state model. *CAAI Transactions on Intelligence Technology*, 3(4):185–190.

Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., and Wang, Q. (2019). Robust lane detection from continuous driving scenes using deep neural networks. *IEEE transactions on vehicular technology*, 69(1):41–54.