SLDNet: A Branched, Spatio-Temporal Convolution Neural Network for Detecting Solid Line Driving Violation in Intelligent Transportation Systems

Zhanke Zhou[1st], Ruiqi Li[1st], Yayu Gao✉, Chengwei Zhang and Xiaojun Hei
School of Electronic Information and Communications
Huazhong University of Science and Technology
Wuhan, China
{zhankezhou, rlee, yayugao, zhangcw, heixj}@hust.edu.cn

***Abstract*—Solid line driving is known as one of the major driving violations in China. In this paper, we propose a branched, spatio-temporal convolution neural network, named SLDNet, to recognize these violation acts from photographs captured by surveillance cameras and train it on Pingxiang solid-line-driving dataset. SLDNet can achieve 0.92 in accuracy and 0.91 in recall, which both out-perform the current human review. Our method will be implemented in intelligent transportation systems in Pingxiang city, Jiangxi Province in near future.**

Keywords-Intelligent transportation; Deep learning; Solid line driving

I. INTRODUCTION

A vast number of traffic accidents happened every year in China due to incautious driving. To maintain road traffic order and prevent these traffic accidents, the law of China on road traffic safety has specified a number of forbidden driving behaviors of vehicles, one of which is known as solid line driving. In the past few years, more and more surveillance cameras are set up in city transportation systems, to monitor road traffic status. These cameras capture images of vehicles with suspicious violation, and send these images to the traffic control center for policemen to review. Nevertheless, the large amounts of suspicious images lead to overwhelming workload. Therefore, it is desirable to design a method to automatically and efficiently recognize solid line driving violation from images information captured by camera.

In this paper, we propose a branched, spatio-temporal deep learning network integrating several different neural networks SLDNet to solve this question. The SLDNet uses LaneNet and Mask R-CNN as its front-end, in the meantime uses ResNet and sequence modeling methods to derive final result. Our network achieves 0.91 in accuracy with 0.92 in recall, and will be put into use in intelligent transportation systems of Pingxiang city in near future.

The remainder of this paper is organized as follows. Section II describes the solid line driving problem and Section III provides a literature review on the corresponding methods and computer vision models involved. Section IV presents the proposed pipeline of the solid line detection algorithm, which is evaluated by experimental results in Section V. Finally, concluding remarks are summarized in Section VI.

II. PROBLEM DESCRIPTION

Let us first state the current problem in the intelligent transportation systems. *Fig. 1* shows a typical positive labeled image in Pingxiang solid-line-driving dataset, captured by surveillance cameras of Pingxiang city, Jiangxi province. Each image in the dataset contains four photographs, which refer to as *sub-images* in the following content. The sub-image in bottom right is a zoom-in of suspicious vehicle, and the rest three sub-images form a time sequence to determine whether the targeted vehicle has transportation violation or not.

Fig. 1 Original image from Dataset

To address this issue, the main idea is to detect the location of vehicle and lane line, and combine the location information of both vehicle and lane line, based on what LaneNet, Mask R-CNN and other deep learning models have achieved. Also, to generate final estimation, ResNet and sequence modeling methods are used in SLDNet.

III. LITERATURE REVIEW

In this section, we will provide a detailed literature review on the traditional solutions for solid line driving problem and the recent development on deep learning technology for computer vision.

In 2013, a solid line driving detection system was proposed in [1] using feature points detection, CRF (Conditional Random Field) and other statistical learning method. However, due to the limitation of fundamental algorithms at that time, the system had restriction on generalization of usage scenario.

Since deep learning technology became dominating in computer vision area, a number of problems in intelligent transportation systems have been rethought. In the following, image classification and semantic segmentation problem for vehicle and lane line will be overviewed.

Image classification task is considered as one of the fundamental challenges in computer vision area, the goal of

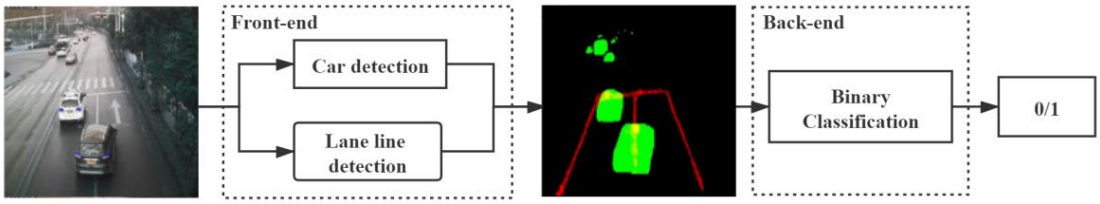

Fig. 2 Pipeline of the proposed SLDNet

which is to determine which category an image belongs to. Traditional pattern recognition methods solved the task generally by two major steps: feature extraction and classification. For example, [2] used SIFT (Scale-invariant feature transform) and LBP (Local Binary Pattern) for feature extracting and stochastic SVM for classification. [3] firstly introduced convolution neural network (CNN) in image classification task and obtain state-of-the-art result at that time. [4] expanded the depth of such CNN, and [5] solved problems of training such deep neural network to some extent.

Another elementary task in computer vision is the image segmentation task. Different from image classification, it demands the model to identify the category of every pixel. [6] built a fully convolutional neural network that takes input of arbitrary size and produce correspondingly-sized output, while showing that convolutional networks trained end-to-end, pixels-to-pixels can improve the result of semantic segmentation. [6] extended [8] by adding a branch for predicting an object mask and achieved significant result on instance level segmentation. In our work, the pipeline of [6] is adopted in generating the mask of vehicle.

Nowadays lane line detection is treated as a special case of semantic segmentation problem mainly for two reasons. Firstly, lane annotation has a long and narrow shape. Secondly, the lane pixels annotations are sparser comparing to general object semantic segmentation. Such form of data raise challenge for detection task. [9] tried to enhance the learning of such sparse label by importing transfer learning and self-attention method. [10] cast lane detection problem as an instance segmentation problem and achieved competitive results with an inference speed of 50 fps, therefore is adopted in the lane detection branch of our work.

IV. The Algorithms

In this section, we first present the overview of SLDNet and then show more details about each part of pipeline.

A. Pipeline Overview

We design a two-part, branched pipeline to determine whether an original image contains solid line driving behavior or not. As illustrated in *Fig. 2*, the front-end of pipeline consists of two branches, which are devised to detect lane lines and vehicles located in given input images and output segmentation maps of lane lines and vehicles, respectively. As to the back-end part of pipeline, it takes the segmentation map as input, and outputs the ultimate judging result, working as a binary classifier.

In a nutshell, the major idea is to eliminate noise and lower the complexity of traffic intersection images via keeping lane lines and vehicles and getting rid of the rest information, and then proceed binary classification on segmentation maps.

B. Pipeline Front-end

1) Lane Line Dection

Let us first introduce the lane line detection part in the front-end. A novel architecture proposed by Neven et al [10], which is called LaneNet, is adopted here in order to detect lane and output the binary lane segmentation (see the red part in *Fig. 2*). LaneNet is a branched, multi-task network, consisting of a lane segmentation branch and a lane embedding one. To speed up inference process, we simplify the architecture of LaneNet and only keep the lane segmentation branch, which is illustrated in *Fig. 2*. As the output of LaneNet, the binary lane segmentation is going to be stitched with vehicle segmentation, forming the final segmentation output of the front-end part in our detection pipeline.

In contrast to other traditional methods which are prone to robustness issues due to road scene variations, LaneNet handles lane changes and allows the inference of an arbitrary number of lanes, casting the lane detection problem as an instance segmentation task.

2) Vehicle Detection

For the vehicle detection part in the front-end, the two-stage object detection method Mask R-CNN [6] is adopted here. Mask R-CNN, a classic and widely used object detection and instance segmentation method, is able to output not only the bounding boxes of vehicles but also their accurate segmentation due to its multi-task network. The time cost of inference is acceptable for us and we do not need to build a real-time processing detection system of vehicle violation, hence we value more on the accuracy but inference speed.

C. Pipeline Back-end

Recall that each original image in our dataset contains 4 sub-images. To proceed binary classification, two methods are implemented in the back-end of pipeline for further evaluation.

1) Method 1. Single Image Classification + Voting

Firstly, we choose to apply mature schemes such as single image classification to conduct binary classification. We select 34-layer Resnet [5] as backbone and add dense layers to build classifier network. Each original image is split into four sub-images and marked correspondingly, based on

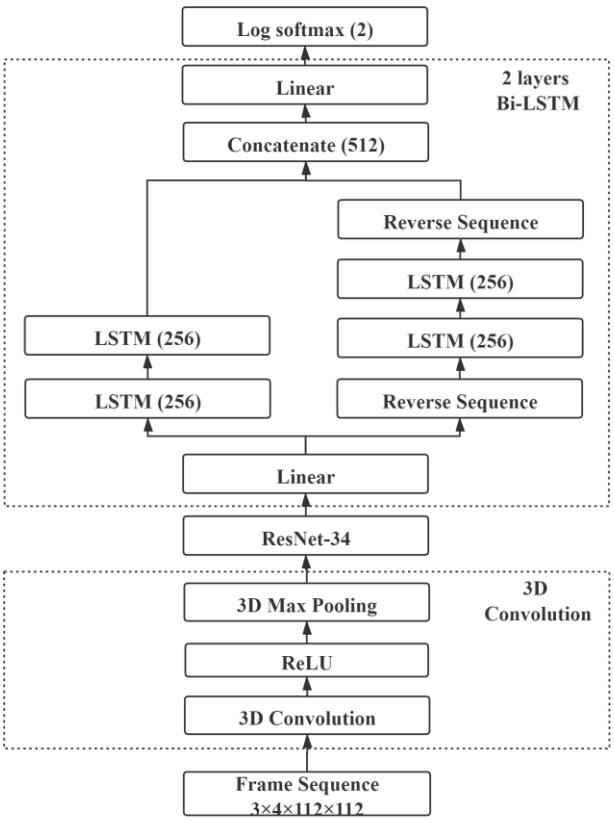

Fig. 3 Block-diagram of Sequence model network

which well-trained front-end of pipeline is used to generate the dataset for training binary classifier. A voting schema is then adopted to calculate the score of each original image. For example, if the binary classifier believes that two sub-images out of four in an original image contain offending vehicles, the score is given as two.

2) Method 2. Sequence Modeling

As the sub-images form a time sequence, we further adopt sequence modeling as the back-end of the pipeline. Two reasons for applying sequence modeling are: firstly, we are capable to obtain the features of vehicle trajectory through sequence modeling. Secondly, the effect of lane detection strikingly decreases when lane lines are covered by vehicles or other stuff especially in complicated intersections, hence causes performance degradation.

As shown in *Fig. 3*, the network architecture is mainly made of three parts. We employ 3D convolution as well as Bidirectional LSTM to extract temporal features, and 2D convolution to extract spatial features, forming a spatio-temporal convolution neural network. Compared with 2D convolution, 3D convolution has one more depth channel which may be a continuous frame on the video, or different slices in the stereo image. It can capture the temporal and spatio features of sequential images, and is widely used to action recognition as well as video classification.

V. EXPERIMENT

In this section, we describe Pingxiang solid-line-driving dataset first, then illustrate implementation details for each part of the proposed SLDNet pipeline, and finally present as well as discuss about the evaluation result.

A. Dataset

Pingxiang solid-line-driving dataset is made of 72235 images taken by surveillance cameras situated in traffic intersections. As shown in *Fig. 1*, each image consists of four sub-images, three of which are temporal sequential, and the other one is zoom-in of suspicious vehicle. In all 72235 images, 0 is the label for 56711 images which means the vehicles in these images do not have solid line driving behavior. Correspondingly, 1 is the label for 15524 images, meaning that at least one vehicle in each image is driven onto solid line and thus violate relevant traffic regulations.

B. Implementation Details

SLDNet is implemented in Pytorch framework and firstly train the front-end of pipeline. Each model of related branch in front-end of pipeline is trained separately on its relevant dataset. After that, we use well-trained front-end to do inference on pingxiang solid-line-driving dataset and generate the images of segmentation maps in order to train back-end.

1) Pipeline Front-end

a) Branch 1: Lane Line Detection

Trained on tusimple lane dataset [12], the LaneNet can achieve around a 50 fps which is similar to the description in the paper [10]. The images of tusimple lane dataset are re-scaled to 512×256 and the network is trained using Adam optimizer with a batch size of 8 and a learning rate 10^{-3} until convergence.

After training LaneNet on tusimple lane dataset, we test LaneNet model on Pingxiang solid-line-driving dataset and find strikingly decline of detection effect due to different shoot angles and road complexity. Thus, we annotated a small quantity of surveillance images keeping the same annotation format with tusimple lane dataset, hence fine-tuned pre-trained LaneNet model on small-scale manually annotated dataset and ultimately obtained extraordinary effectiveness in lane line detection on surveillance images, which lays a solid foundation for the latter binary classification. Transfer Learning technique is employed in this training process, and it focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. In practice, we initialize the model with weights gained from training on tusimple lane dataset, hence proceed fine-tuning on small amount of surveillance images.

b) Branch 2: Vehicle Detection

We apply Mask R-CNN implemented by MMDetection [11] which is an object detection toolbox that contains a rich set of object detection and instance segmentation methods as well as related components and modules. To determine the segmentation of vehicles, we filter out the outputs of Mask

TABLE I. EVALUATION RESULT

Method of Pipeline Back-end	Accuracy	Recall	Parame-ter	FPS
Single Image Classi-fication + Voting	0.81	0.70	22.4M	192.8
Sequence Modeling	0.91	0.92	19.5M	166.7

Voting threshold is set to two, which means an original image is considered to contain solid line driving behavior if its score if not less than two.

R-CNN and only keep those with label of car, truck or bus. A typical segmentation output of Mask R-CNN is shown in Fig. 2 with green color.

2) Pipeline Back-end

a) Method 1: Single Image Classification + Voting

For the first method adopt in the back-end of SLDNet, we resize segmentation images to $3\times224\times224$ and proceed image normalization to speed up training process, and the network is trained using Adam optimizer and cross entropy loss function with an auto adaptive learning rate which initial value is 10^{-4}. Tensorboard is employed here to visualize training process, drawing figures of accuracy and loss on train set as well as validation set. What's more, Attention mechanism on channel as well as spatial and data augmentation schemas such as horizontal and vertical flipping are also applied to improve model performance.

b) Method 2: Sequence Modeling

We split each origin image to 4 sub-images and resize each sub-image to $3\times112\times112$, making input shape $4\times3\times112\times112$. Because each sequence sample only contains 3 sequential images, we choose $3\times7\times7$ kernel size for 3D convolution with $1\times2\times2$ stride and $2\times3\times3$ padding. After finishing plenty of experiments on testing network backbone, we select ResNet-34 for its fewer parameters and higher accuracy. When it comes to Bi-LSTM, we choose 256 as dimension of hidden layers to simplify network and accelerate inference speed. It is worth raising that we use the following three steps training approach to train each part of the pipeline better. Initially, a temporal convolutional part is used instead of the Bi-LSTM. After convergence, the temporal convolutional part is removed and the Bi-LSTM is attached. The Bi-LSTM is trained for 5-10 epochs, keeping the weights of the 3D convolution and the ResNet fixed. Finally, the overall sequence model is trained end-to-end.

C. Evaluation

We randomly single out 4000 original images from dataset to make up validate set, and evaluate two kinds of back-end network on the same validate set. As shown in TABLE I. , method of sequence modeling surpasses method of single image classification + voting in both metrics of accuracy and recall.

Compared with single image classification method which processes each sub-image separately, sequence modeling method has better ability to complement the obscured lane lines and extract temporal features through taking four sub-images into computing at a time, hence makes it helpful to make final determination and bring improvement in performance. It is therefore the preferable option for practical system implementation.

VI. CONCLUSION

In this paper, we propose a branched, spatio-temporal convolution neural network (SLDNet) for detecting solid line driving behavior. In the front-end of network, lane lines and vehicles are detected, forming the segmentation image for each sub-image. After that, the back-end captures the general spatio-temporal dependencies through sequence modeling and output the final determination result. Experimental results on Pingxiang solid-line-driving dataset show that SLDNet can achieve better performance in both metrics of accuracy and recall, 0.91 and 0.92, respectively. The proposed pipeline will be put into use in intelligent transportation systems of Pingxiang city in near future.

REFERENCES

[1] H. Lee, S. Jeong and J. Lee, "Robust detection system of illegal lane changes based on tracking of feature points," *IET Intelligent Transport Systems*, 2013.

[2] Y. Lin et al., "Large-scale image classification: Fast feature extraction and SVM training," in *Proc. IEEE International Conference on Computer Vision*, Providence, RI, 2011.

[3] A. Krizhevsky, I. Sutskever, G. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. International Conference on Neural Information Processing Systems*, Red Hook, NY, 2012.

[4] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. International Conference on Learning Representations*, San Diego, 2015.

[5] K. He, X. Zhang, S. Ren, J. Sun. "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 2016.

[6] E. Shelhamer, J. Long and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640-651, April 2017.

[7] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *Proc. IEEE International Conference on Computer Vision*, Venice, 2017.

[8] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, June 2017.

[9] Y. Hou, Z. Ma, C. Liu, C. C. Loy, "Learning Lightweight Lane Detection CNNs by Self Attention Distillation," in *Proc. IEEE International Conference on Computer Vision*, Seoul, 2019.

[10] D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool, "Towards end-to-end lane detection: an instance segmentation approach," in *Proc. IEEE Intelligent Vehicles Symposium*, Changshu, 2018.

[11] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., et al., "MMDetection: Open mmlab detection toolbox and benchmark," arXiv:1906.07155, 2019.

[12] The tuSimple lane challange, http://benchmark.tusimple.ai