2020 IEEE Intelligent Vehicles Symposium (IV)
October 20-23, 2020. Las Vegas, USA

Deep Transfer Learning Enable End-to-End Steering Angles Prediction for Self-driving Car

Huatao Jiang*, Lin Chang, Qing Li, Dapeng Chen

Abstract — **Autonomous driving has developed rapidly over the last few years. Predicting the steering angle for self-driving car according to different road conditions is very important. There are some endeavors for this topic, including lane detection, object detection on roads, 3-D reconstruction etc., but in our work we focus on a vision based model that directly maps raw input images to steering angles using deep networks and this model don't depend on specifying the features to learn. In this paper, we propose an end-to-end steering angle prediction model based on deep transfer learning and it can accurately predicts steering angles based on input image sequences which are from on-board camera. This prediction model combine two deep learning models including the convolution neural network (CNN) and the long short-term memory (LSTM). The CNN model we use is VGG16 which is based on transfer learning techniques, and pre-trained on Imagenet with good performance. This network is used to extract spatial features of the input image sequences. And the LSTM network is used to capture the temporal information of the provided images. The model we proposed fully considers spatial-temporal information, and fit the nonlinear relationship well between the input images and the steering angles. In order to validate the proposed model, the experimental study is conducted using the real-world dataset which is provided by Udacity. Experimental results show that the proposed model in this paper can efficiently predict the steering angles and clone humans' driving behaviors, and our model has a better performance, higher accuracy, and less training time.**

I. Introduction

Autonomous driving techniques which is one of the most important technologies for intelligent transportation systems (ITS) has been widely studied in recent years and has made great progress. For self-driving cars, steering angle prediction is a critical task. Accuracy of predicting steering angles based on real-time road conditions directly affects the safety and performance of the self-driving cars. As a result, establishing a robust model to predict steering angles has become a hot topic in the research of automatic driving.

The self-driving car steering angle decision module calculates the decision value according to the input amount of the system to ensure that the self-driving car runs safely and stably. The traditional self-driving car steering angles decision module uses the lane line detection, vehicle detection, obstacle detection etc. to decide steering angles . These methods fully depend on designing and specifying the features to be learned in advance which does not guarantee that the entire system obtains the optimal solution. Although many researchers have deeply studied the topic, most of these existing methods are paying attention to designing mathematical models which are limited to some particularly predefined rules, such as road boundaries , lane and traffic rules. The robustness of these methods is poor. And a reliable steer angle prediction model should make different decisions according to various scenes and learn from new situations.

Recently, machine learning techniques has achieved great breakthroughs due to the development of deep learning. Combining deep learning with end-to-end learning has become a new method for learning driving policies for self-driving cars. Unlike traditional methods that calculating lane and obstacle information, End-to-End methods often directly learn the mapping relationship from raw image sequences to vehicle actuation. For example, NVIDIA released a paper [1] regarding End-to-End learning. In this paper, the authors used a basic CNN architecture to extract features from the input images and directly predicting steering angles. The End-to-End methods have the advantage of direct optimization without specified rules and features which need to be learned.

However, for steering angle prediction, most of the existing End-to-End models use a single deep convolution neural network to map from perception to control. These methods can achieve good performance in a single environment but has some drawbacks. First, single neural network model based on CNN doesn't consider any temporal information which is critical in self-driving. Second, although some spatial-temporal models (such as 3D CNN) are proposed, the calculations are heavy and require more training time. Third, existing models only considers current perception information for decision making and lack of understanding of temporal context features in a scenario. In other words, these models lack memory.

Inspired by these drawbacks of the existing models, in this paper, we propose a end-to-end steering angles prediction model which combine VGG16 model with LSTM model. Our model gives full consideration to

Huatao Jiang is with the Institute of Microelectronics of Chinese Academy of Sciences and Wuxi Internet of Things Innovation Center Co., LTD, CO 100029 China (jianghuatao@ime.ac.cn).

Lin Chang is with the Institute of Microelectronics of Chinese Academy of Sciences and Wuxi Internet of Things Innovation Center Co., LTD, CO 100029 China (changlin@ime.ac.cn).

Qing Li is with the Institute of Microelectronics of Chinese Academy of Sciences (liqing@ime.ac.cn).

Dapeng Chen is with the Institute of Microelectronics of Chinese Academy of Sciences and Wuxi Internet of Things Innovation Center Co., LTD, CO 100029 China (dpchen@ime.ac.cn).

978-1-7281-6673-5/20/$31.00 ©2020 IEEE

spatial-temporal information. For this mixed model we propose, the VGG16 model is used to extract spatial features of the input image sequences and the LSTM model is used to capture the temporal information. In order to reduce the training time, inspired by the idea of transfer learning, the pre-trained VGG16 model is used which is trained on the Imagenet dataset and have good performance on it. Comparing with training the network by starting from the beginning, our model based on transfer learning is more time-saving.

II. RELATED WORK

For steering angles prediction module, traditional models divide it into two separate parts, perception and control which contain task such as road lanes detection, path planning and control [2]. For these traditional methods, object detection is regarded as key techniques for autonomous driving, including vehicle / pedestrian detection [3], [4], [5], [6] and lane detection [7], [8]. After accurately identifying these objects, the self-driving car can obtain an accurate and comprehensive understanding of its immediate surroundings. Then, according to the perceived environmental information, the steering angles can be reasonably predicted. These traditional methods require quantifying the features to be learned in advance. The robustness of them is poor and only achieve better performance in specific environments.

The End-to-End prediction model eliminates the module of calculating intermediate variables and directly learn the mapping relationship from raw input information to vehicle actuation. This model does not need to define specific features manually and has good robustness. End-to-end models have been widely used in autonomous driving such as lane-keeping [9] or obstacle avoidance[10].

Deep neural networks have been proven to be very successful in dealing with nonlinear mapping problems and are widely used in autonomous driving. Combining deep learning with end-to-end model is a potential technique to learn driving policies from complex environments. Combining a neural network with End-to-End method for autonomous vehicle steering angle prediction was pioneered by Pomerleau (1989) [11] who built the ALVINN. In this paper, the authors used a neural network to directly map front-view camera images to steering angle. Although this model is was simple, it demonstrated the potential of neural networks for end-to-end autonomous driving. Inspired by ALVINN, NVIDIA released a paper in 2016 [1], and extended that model proposed in paper [11] with deep neural networks. In this paper, the authors used a deep CNN architecture to extract features from the input images, and this framework performs well in relatively simple real-world scenarios. Other existing examples of learning End-to-End control of self-driving vehicles include [9, 12]. Most of these existing models using a single deep network to learn driving policy and can't fully learn the features information (such as temporal information)of the input images. What's more, these models emphasize training the complete model from the beginning which will consume a lot of time.

In order to make the model be rich in spatial-temporal information, LSTM model is introduced in some papers. LSTM is typical representative of recurrent neural networks (RNN) and is great at capturing long-term temporal dependencies. In paper [13], the FCN-LSTM model was proposed to derive a generic driving model. In paper [14], the authors combined LSTM module with optical flow to extract temporal information. Because of calculating optical flow, this model require more training time. In paper [15], the authors combined CNN with more different kinds of RNN model and compared them, what should be mentioned is the training time is very long. What we need to emphasis is that temporal information has not been fully utilized in autonomous driving. For example, PilotNet [16] learned to control the cars by solely looking into current video frame. In this proposed model, the authors pre-defined specific rules to predict steering angle and the robustness was poor.

For autonomous driving, the training time and response time of the steering angle prediction model are very important because of the real-time requirement. In paper [17, 18] , the proposed models captured spatial-temporal information but the training time was very long. These models are very demanding for hardware platforms and it is necessary to consider both the training time and accuracy of the model. Transfer learning is a way of using high quality models that were trained on existing large datasets. We can transfer that features learned in the lower layers of the existing models to driving images collected by ourselves. These learned features on other dataset would be useful in the new dataset. The benefits of using a pre-trained model is faster convergence time and better performance. The model based on transfer learning is more time-saving than training a complete model from beginning.

Contributions of this paper are as follow. Firstly, based on End-to End method and transfer learning, we build a deep neural network model which combines CNN and LSTM to predict steering angle. The CNN layer is used to extract spatial information of the input images, and these spatial features are fed into the LSTM network for extracting temporal features. The output of this network is a real-time steering angle. Secondly, the VGG16 network of extracting spatial features is based on transfer learning which is pre-trained in existing dataset. Our model is more time-saving than some existing models. Third, we validate our steering angle prediction model on real-world dataset which is provide by Udacity and the experimental results show our model can predict steering angles accurately in different environments. What's more, we compare three different prediction models: Nvidia, VGG16 and our models. And the results show that VGG16+LSTM which is we designed is more time-saving and accurate than the other two models. In a word, the model proposed in this paper achieves faster convergence and higher accuracy.

III. Dataset and Data Augmentation

A. Dataset Description

The dataset we used in this paper is provided by Udacity, is collected by NVDIA Dave-2 [1] system by driving in various traffic and lighting conditions. Three cameras -Left, Right and Center, installed behind the wind shield of an automobile is used to collect image frames and corresponding steering angles at 20 FPS. Time-stamped video from the cameras is captured simultaneously with the steering angle applied by the human driver. This steering command is obtained by tapping into the vehicle's Controller Area Network (CAN) bus. The car is driven for a total of 28.23 minutes, whose split up can be seen below. Training images have been extracted from 5 different videos recorded by the automobile:

• Clip1: 221 seconds, direct sunlight, many lighting changes. Good turns in beginning, discontinuous shoulder lines, ends in lane merge, divided highway.

•Clip2: Discontinuous shoulder lines, ends in lane merge, divided highway 791 seconds, two lane road, shadows are prevalent, traffic signal (green), very tight turns where center camera can't see much of the road, direct sunlight, fast elevation changes leading to steep gains/losses
over summit. Turns into divided highway around 350s, quickly returns to 2 lanes.

•Clip3: 99 seconds, divided highway segment of return trip over the summit.

• Clip4: 212 seconds, guardrail and two lane road, shadows in beginning may make training difficult, mostly normalizes towards the end.

•Clip5: 371 seconds, divided multi-lane highway with a fair amount of traffic.

The steering angles are modelled as 1/r , where r is the turning radius to make the system independent of car geometry. The data was extracted from ROSbag files using a docker interface.

Training dataset contains 101397 frames and corresponding labels including steering angle, torque and speed. This training dataset contains 33746 frames. And the validation dataset we used is also provided by Udacity which contains 5615 frames. The original resolution of these image is 640x480.

B. Preprocessing and Data Augmentation

Preprocessing

Histogram of steering angles and the curves of steering angles over the five videos have been shown in Figure 1 and 2 respectively. It can be observed that a large number of labels were neutral angles ((± 0.050)) . This affects the CNN prediction since the model is biased towards neutral angles and cannot be fed into the networks for training. To overcome this, we removed frames within (± 0.010) with 40% probability.

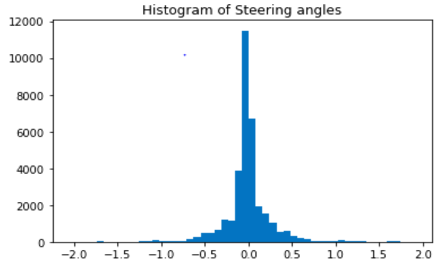

Figure 1. Histogram of steering angles

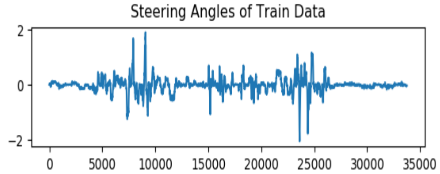

Figure 2. Curves of steering angles

Data Augmentation

Data augmentation is necessary to teach the car to recover from edges of the roads and avoid over-fitting. The augmentations used in this paper are taken from the paper [1], mainly include brightness augmentation, random shadows and crop and resize.

Brightness Augmentation: The change of brightness can seriously affect the direction of neural network judgment and adjusting the image brightness can improve the robustness of the network to different environments. Brightness is randomly changed to simulate different light conditions. The brightness of the images have been changed in HSV domain.

Random Shadows: Some part of the images are darkened to create a shadow effect, so that the model becomes robust to darkness. The main purpose of random shadows is the model can predict accurately the correct steering angle although the camera has been shadowed (maybe by rainfall or dust).

Crop and Resize: For each image, we normalize the pixel values from [0,255] to [0,1] by dividing the image matrix by 255. What's more, we further resize the image to a 224x224x3 square image for transfer learning model.

IV. Methods

To emulate human driving patterns in an autonomous vehicle and predict steering angles in many different situations in real time, we proposed a steering angles prediction model using deep transfer learning techniques. The structure of the proposed model is shown in the Figure 3. This model learns features from the input images sequences and directly predicting steering angles. The input images of this model are captured from a front-facing camera. We divide this prediction model into two parts: spatial features extracting sub-network and temporal features extracting sub-network. For spatial features extraction layer, the model we used is VGG16 which is based on transfer learning. Then, the spatial features are fed into temporal features extraction layer which is based on LSTM to further extract temporal information of the input images sequences. Finally, the Dense layer outputs the predicted steering angle.

The proposed steering angle prediction model can be describe by the following equation (1),

$$\mathcal{F} = \mathcal{N}(\mathcal{W}, \mathcal{V}) \qquad (1)$$

In this expression, $\mathcal{F}$ is the output of the model. $\mathcal{N}$ is the mapping relationship from raw image sequences to steering angle which is learned by the neural networks model. $\mathcal{W}$ is the trained weight parameter and $\mathcal{V}$ is the input images sequences. Loss function is an important basis to reflect the advantages and disadvantages of the model. The loss function we design in this paper is using the difference between the predicted steering angle and the true value. In training stage, we use this loss function to modify the weights in the VGG16 and LSTM sub-network through back-propagation algorithm and optimize the deep neural network.

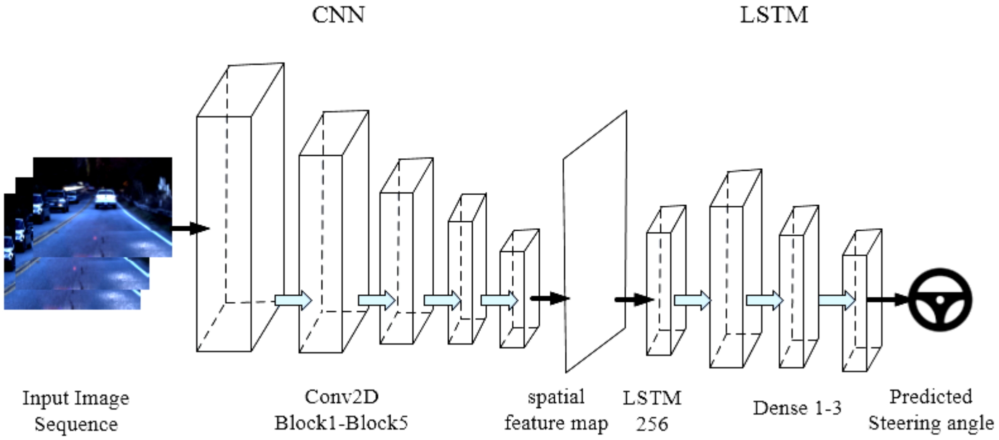

Figure 3. Structure of the proposed model

A. Spatial Features Extracting sub-network

CNN have been proved to be quite accurate in handling object recognition problems in recent years [19]. And NVIDIA has designed a CNN architecture to predict the steering angle and achieved better performance [1]. The weakness of NVIDIA's model is ignoring temporal features of the input images. For spatial features extraction, we design a CNN network which is based on VGG16.VGG16 as the basic architecture to design spatial features extracting sub-network. As we all know, the training of deep neural network models takes a lot of time. In paper [15], the proposed model has a good performance but the training time requires about 4-5 days over a single GPU. Obviously, too much training time does not meet the actual needs. For VGG16 model, we used the idea of transfer learning to save training time. Transfer learning is a method of using existing models which are pre-trained on existing large datasets. And The pre-trained weighted parameters can be directly transferred to the new dataset. This idea avoids training models from scratch which will be time-saving in training stage.

The pre-trained deep neural network models are available, such as VGG16, ResNet50, InceptionV3 and they are pre-trained on ImageNet. The reason we choose VGG16 is that some papers verify that VGG16 is good at spatial feature extraction [21]. What's more, computational complexity and response time is very important for a real-time steering angle prediction system and VGG16 contains fewer layer than other models so that it runs faster. So, VGG16 based on transfer learning is a basic of the proposed model. VGG16 has a good performance to recognize approximately 1000 different kinds of objects on ImageNet.

Inspired by the idea of transfer learning, we need fine tune the pre-trained VGG16 model to make it focus on Udacity's dataset. Because of VGG16's task on ImageNet is classification and our designed model's goal is extracting spatial features of the input images. So, we prune the original VGG16 network to make sure this model meet our needs. The modified architecture of VGG16 is shown in Figure 4. The modified model still 16 layers in the network, and the weights of the reserved 13 layers were blocked from updating on training stage. The input of this architecture is input images with three channels. The original resolution of the input images is 640x480 and we resize the resolution to 224x224 for the transfer learning model.

In general, full connection layers (FC) in CNN contain too many weights. The FC layers of original VGG16 are regarded as a classifier and the needs of the model we proposed is extracting spatial information. So we replace the last three full connection layers with three convolutional layers with the size 3×3. And these three replaced layers contain 1024, 2048 and 4096 kernels respectively. The last pooling layer's output of reserved layers is a feature map with the size 7×7×512. And this feature map feeds into the replaced convolutional layers. The final output of this spatial features extracting sub-network is spatial feature map with the size 1×1×4096. And this spatial feature map will be fed into temporal features extracting sub-network directly which is based on LSTM.

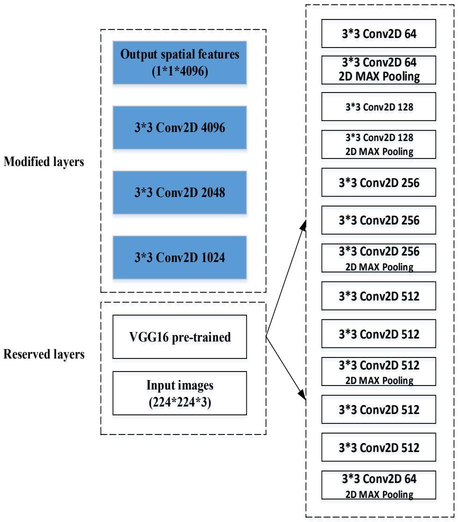

Figure 4. Structure of the modified VGG16

B. *Temporal Features Extracting and steering angle prediction sub-network*

After extracting spatial feature of the input images from the VGG16 model, a temporal features extracting sub-network based on LSTM is used to process temporal information of the images. The input of this sub-network is spatial feature maps which from VGG16. The architecture of this sub-network is shown in Figure 5. And the designed specific internal structure of LSTM is shown in Figure 6.

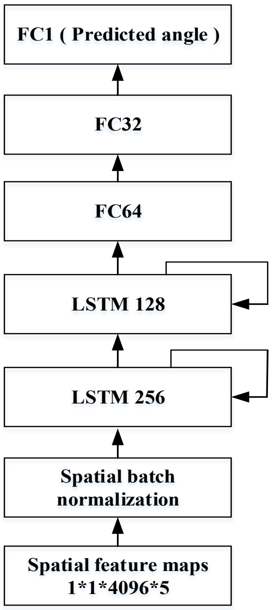

Figure 5. Structure of the Temporal Features Extracting sub-network

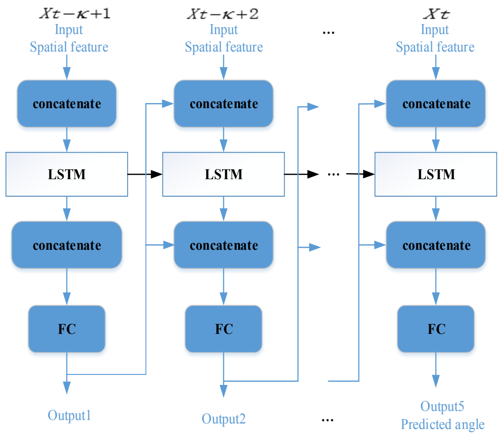

Figure 6. Internal structure of LSTM

The LSTM model is used to processes the context temporal feature vector γ in a sliding window of size κ. The outputs of LSTM model are temporal features. But final output of this sub-network is steering angles, so we design full connection layers to connect LSTM with steering angles. The result of steering angle prediction is dependent on κ past input observations $X_{t-\kappa+1} \sim X_t$. The output of this network is predicted steering angles at time step t. And the input observations are spatial features vectors extract from VGG16. In our experiments, we input the spatial features extracted from κ previous continuous images. By changing the value of κ, we can choose how long the system takes to predict angles. What should be emphasized is κ is a experience value. Small κ leads to short-term dependences and increases the deviation of prediction. Larger κ leads to more accurate predictions but requires longer training time. In this paper, the value of κ we choose is 5.

When the spatial feature vectors feed into LSTM, we add the layer of spatial batch normalization between two sub-networks. Spatial batch normalization not only normalizes among different samples but also among the spatial axis of images.

C. Loss Function and Optimization

To measure the predictive performance of the proposed model, loss functions need to be defined. In this prediction model, the steering angle is a continuous variable and can come down to a regression issue. So, we choose MSE (see Equation 2) as loss function of the proposed model. This function is the mean of the sum of the squared differences between the actual angles $\mathcal{Y}_i$and predicted angles $\hat{\mathcal{Y}}_i$.

$$\mathrm{MSE}(\mathcal{Y}_i, \hat{\mathcal{Y}}_i) = \frac{1}{n}\sum(\mathcal{Y}_i - \hat{\mathcal{Y}}_i)^2 \quad (2)$$

The proposed prediction model is trained to get optimal weight vector $\mathcal{W}$ by minimizing the loss function, which can be expressed by Equation (3).

$$\mathcal{W} \leftarrow \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathrm{MSE}(\mathcal{N}(\mathcal{W}, \mathcal{V}), \hat{\mathcal{Y}}_i) \quad (3)$$

In the process of finding the minimum of MSE, Adam optimizer is used [18]. This optimizers is often the good choice for deep learning networks and performs better than other stochastic gradient descent methods. Adam computes an adaptive learning rate through its formula. Because of the software environment is Keras, we use the default values of the Adam optimizer in Keras (learning rate of 1e-3, β1 = 0.9, β2 = 0.999, ϵ = 1e − 8, and decay=learning rate/batch size)

V. EVALUATION AND RESULTS

In our experiment, the software environment of training and validating the networks includes Ubuntu 16.04 and Keras. And the hardware environment includes Intel (R) Xeon(R) Silver 4116 CPU, 64G RAM, 2.1Ghz Frequency, and NVIDIA TITAN Xp GPU. In the experiment, the number of training samples is 33746 and the number of validating samples is 5246 which are from Udacity's dataset. In training stage, the model has been trained for 40 epochs at a learning rate of 0.001 and a batch size of 250. And the model training time is 34 minutes based on these parameters which is more time-saving than other models.

A. Feature Visualization

In order to examine what our model capture relevant in an image, we can visualize the output of each layer by using saliency maps. Saliency maps can be used to analyze which variables are important to the model and it can be also called feature maps. A method is used in a recent NVIDIA paper[1].

Figure 9 shows the visualizations of convolution layers in block 2 of modified VGG16 and Figure 10 shows modified LSTM's learned feature maps of the first layer. These pictures are part of the proposed model's feature maps. It can be observed that the edges , the lane, and the nearby cars are highlighted, which means that the model is learning to drive inside the boundaries of the lane and keep away from surrounding vehicles. What should be emphasized is that the learned features of our model are not pre-defined in advance. This also demonstrates the superiority of the End-to-End methods.

B. Results and Analysis

The model we proposed in this paper can effectively predict steering angles in different driving conditions. The prediction results of the model we designed on the training and validation dataset are shown in the Figure 7 and Figure 8. And these two figures also show the results of our model (the blue curve) compared with other existing models. The comparison models we used are VGG16 (the green curve) and Nvidia model (the orange curves) which is proposed in [1]. From these figures we can observe that the prediction results of our model are better than others. The prediction curves of our model fit the actual steering angle curves (the red curves) well. What should be emphasize is that the prediction results of VGG16 and Nvidia are non-smooth, and considering temporal information in our model is very important to smooth driving experience.

In this paper, we use Mean Square Error (MSE) to describe prediction errors of the models. Table I shows the values of MSE in training set and validation set. Clearly, the MSE of our model are all less than 0.03, which confirms that the proposed model can make different prediction according to real-time surroundings. The values of MSE also indicate that our model superior to other models. What's more, this table list the values of FPS which indicate the times of prediction per second.

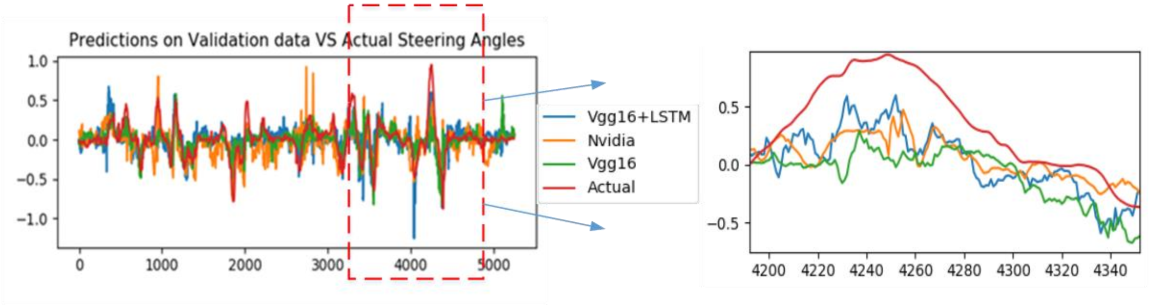

Figure 7. Prediction results on validation set

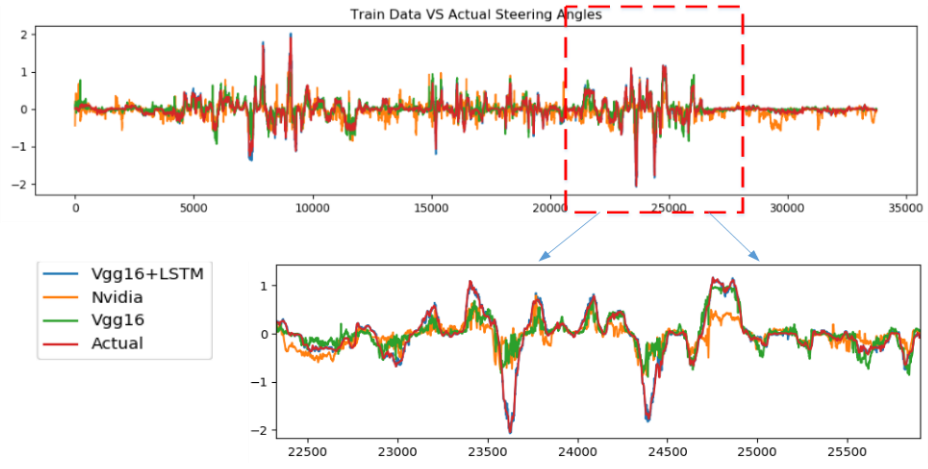

Figure 8. Prediction results on training set

The FPS of three models is all more 20 and human takes 150ms to predict steering angle [20]. So, the model we designed can be used in real-time autonomous driving systems.

Table I

Model	MSE (training)	MSE (validation)	FPS
Nvidia	0.038636	0.047976	26
Vgg16	0.041317	0.025500	24
Ours	0.029517	0.000759	23

The proposed model in this paper can handle different road conditions, such as shadows, up hill, strong light and sharp turning. And we visualization the prediction results of our model in these circumstances. The visualization results are shown in the Figure 11. Blue pointer is actual steering angle and yellow pointer is prediction results . From the figure we can see that our model achieved good performance and has good robustness.

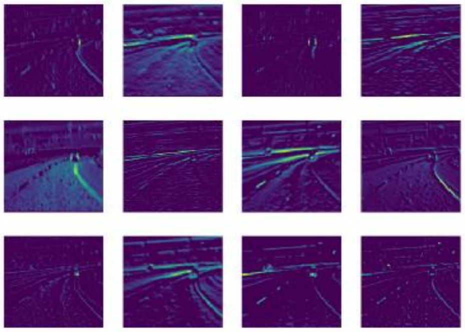

Figure 9 Visualization features of modified VGG16

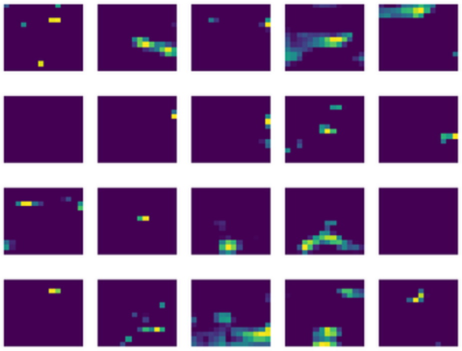

Figure 10 Visualization features of LSTM

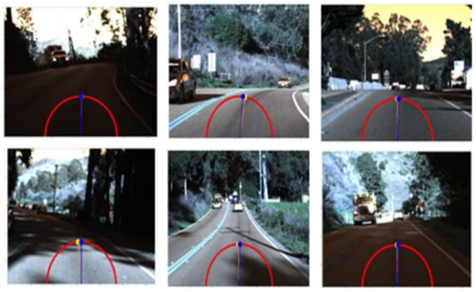

Figure 11 Visualization of prediction results

VI. Conclusion

In this paper, a End-to-End steering angle prediction model which combines modified VGG16 with LSTM is proposed. This model not only extracts the spatial features of the input image sequence by modified VGG16 network but also captures temporal information by LSTM network. This designed mixed network fully considers spatial-temporal information of the input images and predicts steering angles effectively. The convolutional layer's visualization results (Fig) of these two sub-networks show that the key features of the roads are learned although we don’t pre-defined these features and this is also an advantage of the End-to-End model. Because of the modified VGG16 is based on transfer learning, training the proposed model is more time-saving. Finally, we train and validate our model on Udacity’s dataset. The results indicate that our model can predict the steering angles accurately and has good robustness to different environments. The shortcomings of this paper is that this model isn’t tested in the real world (also in special test dataset) and we will do it in the future work. But because of the experimental results show good robustness of the designed model, we believe our model can be used in real-time autonomous driving.

References

[1] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316, 2016.

[2] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi. Towards a viable autonomous driving research platform. In Intelligent Vehicles Symposium (IV), 2013 IEEE, pages 763–770. IEEE, 2013.

[3] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in CVPR, 2005, pp. 886–893.

[4] Q. Zhu, M. Yeh, K. Cheng, and S. Avidan, “Fast human detection using a cascade of histograms of oriented gradients,” in CVPR, 2006.

[5] O. Tuzel, F. Porikli, and P. Meer, “Human detection via classification on riemannian manifolds,” in CVPR, 2007.

[6] Y. Yang and D. Ramanan, “Articulated human detection with flexible mixtures of parts,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 12, pp. 2878–2890, 2013.

[7] M. Aly, “Real time detection of lane markers in urban streets,” CoRR , vol. abs/1411.7113, 2014.

[8] A. Gurghian, T. Koduri, S. V. Bailur, K. J. Carey, and V. N. Murali, “Deeplanes: End-to-end lane position estimation using deep neural networks,” in CVPR Workshops, 2016.

[9] Z. Chen and X. Huang. End-to-end learning for lane keeping of self-driving cars. In Intelligent Vehicles Symposium (IV), 2017 IEEE, pages 1856–1860. IEEE, 2017. 1, 2.

[10] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. L. Cun. Offroad obstacle avoidance through end-to-end learning. In Advances in neural information processing systems, pages 739– 746, 2006.

[11] D. A. Pomerleau. Alvinn, an autonomous land vehicle in a neural network. Technical report, Carnegie Mellon University, Computer Science Department, 1989.

[12] V. Rausch, A. Hansen, E. Solowjow, C. Liu, E. Kreuzer, and J. K. Hedrick. Learning a deep neural net policy for end-toend control of autonomous vehicles. In American Control Conference (ACC), 2017, pages 4914–4919. IEEE, 2017.

[13] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[14] Chen Y, Praveen P, Priyantha M, et al. Learning on-road visual control for self-driving vehicles with auxiliary tasks[C]//2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019: 331-338.

[15] Chi L, Mu Y. Deep steering: Learning end-to-end driving model from spatial and temporal visual cues[J]. arXiv preprint arXiv:1708.03798, 2017

[16] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. D. Jackel, and U. Muller, “Explaining how a deep neural network trained with end-to-end learning steers a car,” CoRR, vol. abs/1704.07911, 2017.

[17] Song S, Hu X, Yu J, et al. Learning a Deep Motion Planning Model for Autonomous Driving[C]//2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018: 1137-1142.

[18] Du S, Guo H, Simpson A. Self-driving car steering angle prediction based on image recognition[J]. arXiv preprint arXiv:1912.05440, 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[20] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," nature, vol. 381, no. 6582, p. 520, 1996.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.