2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Learning a Deep Motion Planning Model for Autonomous Driving

Sheng Song, Xuemin Hu, Jin Yu, Liyun Bai and Long Chen

***Abstract*—To deal with the issue of computational complexity and robustness of traditional motion planning methods for autonomous driving, an end-to-end motion planning model based on a deep cascaded neural network is proposed in this paper. The model can directly predict the driving parameters from the input sequence images. We combine two classical deep learning models including the convolution neural network (CNN) and the long short-term memory (LSTM) which are used to extract spatial and temporary features of the input images, respectively. The proposed model can fit the nonlinear relationship between the input sequence images and the output motion parameters for making the end-to-end planning. The experiments are conducted using the data collected from a driving simulator. Experimental results show that the proposed method can efficiently learn humans' driving behaviors, adapt to different roads, and has a better robustness performance than some existing methods.**

***Index Terms*—autonomous driving, deep motion planning, cascaded neural network, CNN, LSTM**

I. Introduction

Autonomous driving which is one of the most important technologies for intelligent transportation systems (ITS) plays a crucial role in reducing traffic accidents and improving traffic efficiency. Motion planning, as a significant part of autonomous driving, aims to search safe paths or motion parameters in order to make the ego-vehicle navigate from the initial state to the destination state considering the vehicle dynamics, obstacles on roads, as well as road boundaries and traffic rules [1].

Existing motion planning algorithms such as rapidly-exploring random tree (RRT) [2], A-Star algorithm and its variables[3], numerical optimization approaches [4], and interpolating curve planners [5] have achieved great success in the fields of autonomous driving and intelligent robots. Although many researchers have deeply studied the issue, most of the methods are paying attention to designing mathematical models which are limited to some particularly predefined rules. Therefore, it is powerless to handle the situations beyond the rules. However, well-trained human drivers are able to make different decisions according to various scenes and learn from new situations. Furthermore, most of rule-based algorithms cannot directly handle the data from the sensors, while they usually require more time to represent the map from the environment. The time-consuming data process delays the system and increases the reaction time, which leads to higher risk of accidents. Thus, a good planning algorithm should have the capability to learn from the new environment to adapt various situations instead of reacting according to pre-defined rules.

Recently, machine learning has achieved revolutionary breakthroughs due to the development of deep learning [6-10]. Moreover, convolutional neural networks, as well as long short-term memory, are the most widely applied model. CNNs which have an outstanding capacity of image recognition have achieved great successes in obstacle avoidance [11], lane detection [12] and semantic segmentation, etc. [13]. CNNs make the end-to-end motion planning possible, and related algorithms have been applied to mobile robots [14]. The LSTM is great at capturing long-term temporal dependencies [15], so it is widely used in constructing sequence models such as handwriting recognition [16], language translation [17], and action recognition [18], etc. For human action recognition, The LSTM shows high accuracy in a large-scale dataset, which implies the model is robust and able to adapt to other tasks such as motion planning.

In this paper, we propose a deep motion planning model for autonomous driving by combining an improved VGG-net and a LSTM to a deep cascaded neural network. This model applies deep learning methods to solve the motion planning problem. The sequence images collected by the vehicle-mounted camera are fed to the CNN layer to extract spatial features, then the spacial sequential features are input into the LSTM layer to extract temporal features. The cascaded neural network outputs control commands for autonomous vehicles to complete the motion planning task.

There are mainly two contributions in this paper. Firstly, we developed a novel deep cascaded neural network which consists of an improved VGG-net and a LSTM for imitating human drivers' behaviors. Secondly, we collected driving videos for about eight hours from a simulated driving environment and tested our method in this dataset, which we will publicly release in the future. The experimental results show our method outperforms some state-of-the-art methods. Compared with existing datasets, our dataset contains a variety of different roads and supports online test.

II. Deep Motion Planning Model

The proposed motion planning model based on the deep cascaded neural network is shown in Fig. 1. The model are fed by the images captured from a front-facing camera. In order to provide the information behind the ego-vehicle, the images from the left rear and right rear view mirrors are embedded in

*Research supported by the Hubei Provincial Natural Science Foundation of China under Grant 2018CFB158.

Sheng Song, Xuemin Hu, Jin Yu and Liyun Bai are with School of Computer Science and Information Engineering, Hubei University, Wuhan, Hubei, 430062, China (Xuemin Hu, corresponding author to provide e-mail: huxuemin2012@ hubu.edu.cn).

Long Chen is with School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong, 510275, China.

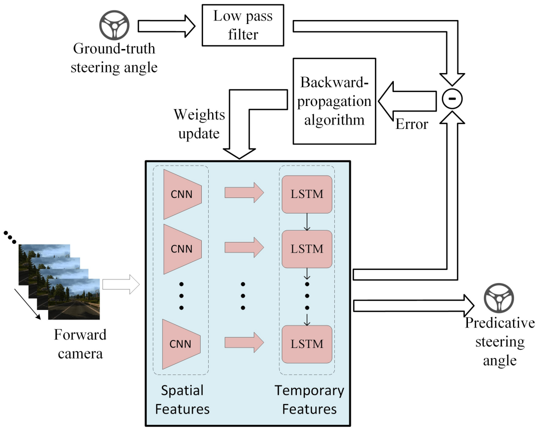


Fig. 1 A simplified block diagram of the proposed model.

the upper left and upper right of the images from the front-facing camera, respectively, and input to the proposed model together. An example of the image is shown in Fig.5 (b). Through inputting the sequence images, the current steering angle is predicted by the deep cascaded neural network, where the CNN layer extracts spatial features in each image. Then the spatial features are input to the LSTM layer to get the temporal features of the continuous sequence images. The output layer of the network outputs the predicted steering angle. The process of making planning decisions with the well-trained model can be described by (1).

$$s = N(\boldsymbol{\theta}, \mathbf{v}), \quad (1)$$

where N represents the deep cascaded neural networks model, and $\boldsymbol{\theta}$ is the weight parameter vector. Vector $\mathbf{v}$ represents the input sequence images, and s is the predicted steering angle.

In the training stage, we design the loss function using the difference between the predicted steering angle and the ground truth from human drivers. Because of the shakes from human drivers' hands and the noise from sensors, the original steering angles are not directly suitable for training. The data collected from the sensors first pass through a low-pass filter to get more stable data. Then we use the loss function to update the weights in the CNN and LSTM layers through backpropagation algorithm and optimize the deep cascaded neural network.

A. Architecture of the CNN layer

CNNs have been proved to be quite accurate and efficient in handling object recognition problems in recent years [19]. Because of using local connection and weight sharing, CNNs can effectively extract both local and global features including texture, shape and topological structure, etc. in an image.

Motion planning algorithms for autonomous vehicles are required to process massive data from sensors, particularly the images from cameras. In this paper, we choose Visual Geometry Group Net (VGG-Net) as the basic architecture to design our CNN layer. Many researchers have proved that VGG-Net is good at spatial feature extraction and image classification [20]. Compared with the other deeper networks, such as Residual Networks (ResNet), the VGG-Net contains fewer layers so that it runs faster, which is very important for a real-time motion planning system. Considering accuracy and computational complexity, the VGG-Net is the most suitable CNN model for designing a motion planning model. Moreover, the VGG-Net is pre-trained on the Salient Object Subitizing dataset and has outstanding capability to recognize approximately 1000 different kinds of objects [21]. Inspired by the idea of transfer learning, we fine tune the pre-trained VGG-16 network to make it focus on driving images collected by us, which is more time-saving than training the network by starting from the beginning.

The original VGG-16 network is trained to classify objects belonging to the about 1000 different kinds. However, the goal of the CNN layer in the proposed model is to extract spatial features from the driving images instead of classification, so the original VGG-16 network is not suitable for our solution. In this case, the original VGG-16 network is modified for improvement. The improved VGG-Net is shown in Fig. 2. There are still 16 layers in the network, where the driving images with three channels are fed as the input. The size of the input images is 224×224, which is inherited from the original VGG-Net and contains enough information for driving.

In general, full connection layers (FC) in CNNs contain too many weights. Deeper networks have better performance for extracting features in a neural network. The FC layer in the original VGG-16 network is usually regarded as a classifier, but the CNN in the proposed network is used to extract spatial features for driving images rather than classify the images. Therefore, we remove the last three full connection layers and replace them with three convolutional layers with the size 3×3. The three convolutional layers contain 1024, 2048 and 4096

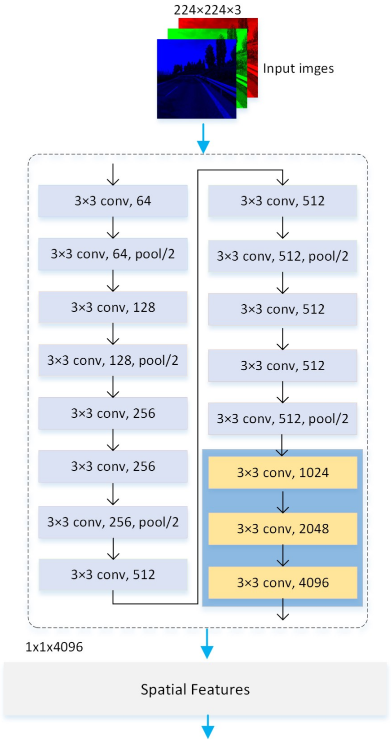


Fig. 2 Improved VGG-Net

kernels, respectively. Thus, for every image fed into the CNN layer, the last pooling layer outputs a feature map with the size 7×7×512, which is fed into the three designed convolutional layers. A vector with the size 1×1×4096 which is regarded as the spatial feature is output from the improved VGG-Net and fed into the LSTM layer.

B. Architecture of the LSTM layer

The model of long short-term memory is inherited from recurrent neural networks. A LSTM unit mainly contains a memory cell that can preserve its state during time, and several non-linear gates that are treated as regulators of the information flow of going through the cells [22]. The LSTM has been developed for decades, and there are many variants of the original LSTM. As shown in Fig. 3, the LSTM architecture applied in the proposed model contains a memory cell, an input gate, an output gate, and a forget gate [18]. c_{t-1} and c_t represent the last and current cell information, respectively. F_t is the input spatial features. h_t and h_{t-1} are the output of the LSTM unit which represents the temporary feature of the last and current units, respectively. f_t is the control information from the forget gate. i_t and g_t are representative of input information through input gate and input modulation gate, respectively. o_t is the output of the output gate. The memory cell stores a value for a certain period, which is realized by using a certain activation function for the cell. Due to the gating mechanism, the LSTM has the capability to extract temporary features and overcome the drawbacks of simple recurrent nets, such as the gradient exploding and vanishing problems of training a deep network.

The process of extracting temporary features from continuous driving images is shown in Fig. 4. The LSTM layer takes the spatial features vectors from the improved VGG-Net and outputs the temporary features, which are connected with steering angles through a full connection layer.The prediction is based on the several spatial feature vectors extracted from ous images by the improved VGG-Net. For each input spatial feature, the LSTM layer and full connection layer share the weights. In our experiments, we input the spatial features extracted from *k* previous continuous images, which represented by $F_{r\text{-}k+1}$ to F_r in Fig.4. The predicted steering angle from $F_{r\text{-}k+1}$ to F_r is treated as the motion planning decision at time step *r*. *k*, as the continuous image number of inputting into the network, is an experience value. A small value of *k* means the model only learns short-term dependences and increases the deviation of prediction. On the contrary, a larger value of *k* represents more stable predictions of steering angle but leads to a higher possibility to learn wrong long-term dependences. In this paper, *k*=11.

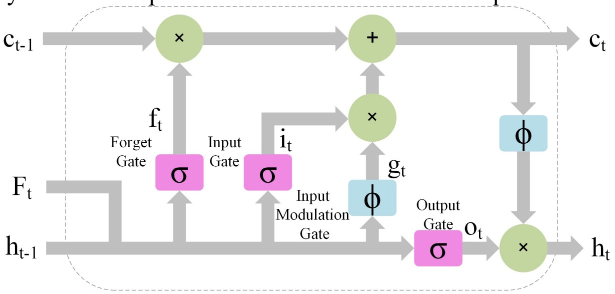


Fig. 3 LSTM unit used in the proposed network

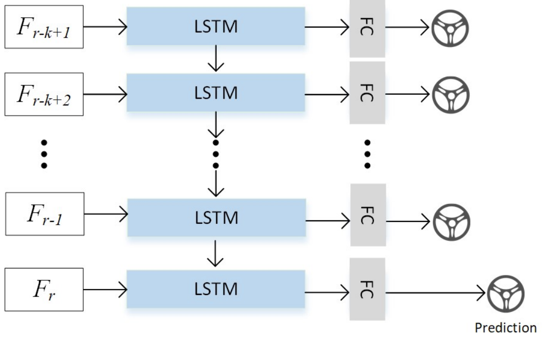


Fig. 4 LSTM layer

C. Output and Loss Function

Deep neural networks are widely applied for classification problems, so the outputs of deep networks are usually connected with various numbers of output nodes. However, the steering angle is a continuous variable, so motion planning is a regression issue instead of classification. Since only one motion command, the steering angle, is predicted, one output node is desinged in the proposed network. The square error between the output value *s* and the ground-truth s_g is used to design the loss function *L*, which is shown in (2).

$$L(s,s_g)=\|s-s_g\|_2 \tag{2}$$

The network is trained to get the optimal weight vector **θ** by minimizing the loss function, which can be expressed by (3). Stochastic gradient descent is used during the training process.

$$\boldsymbol{\theta} \leftarrow \arg\min_{\boldsymbol{\theta}} \; L(N(\boldsymbol{\theta},\mathbf{v}),s_g) \tag{3}$$

III. Experimental Results

In this section, we first introduce the dataset used in the paper and the method of evaluating experimental results. Then we present the details of the results and analysis.

A. Dataset

We create our dataset using a driving simulator, where we can collect images from the view of a human driver as well as the images from the left rear and right rear view mirrors, and record motion commands by input devices which are operated by human drivers at the same time. The Betop BTP-3189 which is show in Fig.5 (a) is used as the input device to collect the synchronized steering angles for the sequence images. The range of the steering wheel is from turning left about 180 degrees to turning right about 180 degrees.

In this paper, we select European Truck Simulator 2 (ETS 2) to create the simulated driving environment. Compared with other driving simulator such as Open Racing Car Simulator (TORCS) [23], ETS 2 has more realistic graphics and more types of roads. Our dataset contains about eight hours of driving data including country roads, freeways and mountain roads, and we record images and motion commands at 30

(a) Input device (b) Sample Image

Fig. 5 Data collection

frames per second (FPS). Compared with exiting datasets where driving data are collected from real roads, the data in our dataset are collected from the virtual environment with realistic graphics, which includes more kinds of roads and weather conditions. Besides, in the virtual environment, we can collect driving data of some emergency events, while other datasets contain few emergencies. Three scenes are used to test, and the rest of the data are used to train the network in our experiments.

B. Evaluation Metrics

The average system prediction error can be described by the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). In terms of predicting steering angles, large errors are unexpected, so we select RMSE to display the errors. Large errors pay relative high contributions to the results. The calculation of RMSE is expressed by (4).

$$RMSE=\sqrt{\frac{1}{T}\sum_{i=1}^{T}(GT_i-P_i)^2}\ , \tag{4}$$

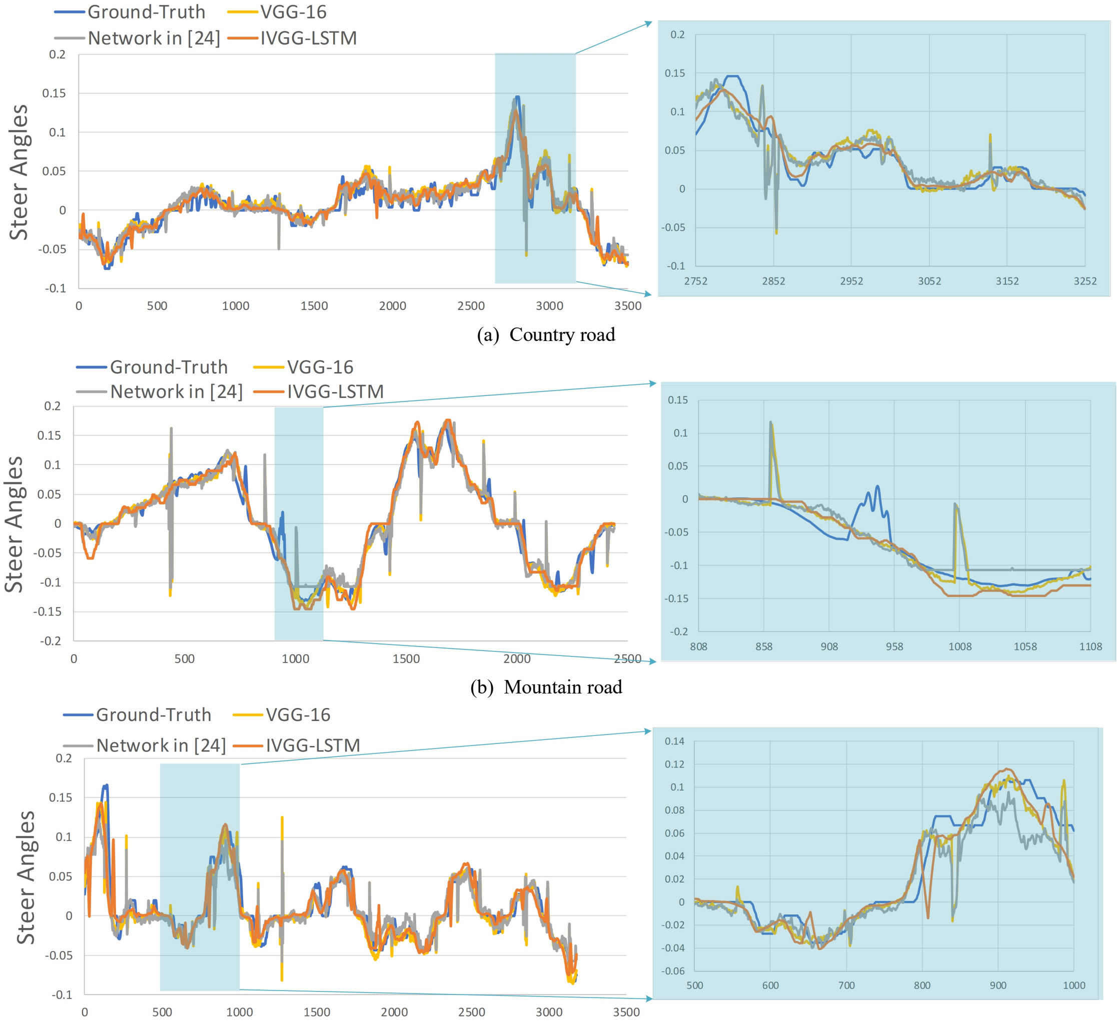
where T is the length of test data. GT_i is the ground truth, and P_i is the prediction for the ith testing sample.

C. Results and Analysis

The software environment of training and testing the networks includes Ubuntu 16.04 and Caffe. The hardware

TABLE I. RMSE VALUES OF THE TESTING RESULTS

Road type	Network in [24]	VGG-16	**IVGG-LSTM**
Country road	0.012657	0.012316	**0.009553**
Mountain road	0.020413	0.018267	**0.016486**
Freeway	0.016054	0.014761	**0.014680**

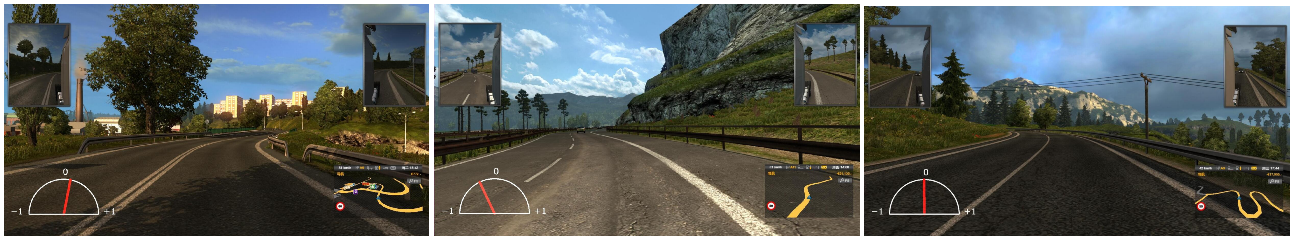


(a) Country road

(b) Mountain road

(c) Freeway

Fig. 6 Experimental results



(a) Country road (b) Mountain road (c) Freeway

Fig.7 Testing examples of the three roads. The white semicircle shows the steering angle range which is from far left to far right. The red line in semicircles shows the current steering angle. -1 and +1 represent the far left and the far right of the steering wheel, respectively. 0 means the vehicle is navigating in a straight line.

includes Intel Core i7-7700K (Quad-core 4.2 GHz) CPU, 32GB RAM, and NVIDIA GTX 1080Ti GPU. We test the model in three classical road conditions including a country road, a freeway, and a mountain road. In order to compare the proposed model consisting of an improved VGG network and a LSTM network (IVGG-LSTM) with other motion planning models including the network consisting of only a original VGG-16 network and the network proposed in [24] are used as the comparative methods in our experiments. Fig. 6 shows the experimental results of the three methods for the testing roads. The RMSE values are shown in Table I. Some testing examples of the three roads are shown in Fig. 7.

The model can effectively predict steering angles of driving in different types of roads. From Fig. 6, we can observe the prediction curve by the proposed method and the ground truth curve are very similar. In Table I, the values of RMSE by the proposed method in the three scenes are all less than 0.017, which confirms that the proposed model has the ability to recognize different scenes and make rational predictions of the steering angle for autonomous driving.

The proposed model can handle roads with sharp and long turn. The magnifying view in Fig.6 (a) shows the details of a sharp curve on the country road. In Fig. 6 (b), we provide the details of a long curve on the mountain road. The curves of the prediction and the ground truth are similar. Moreover, the example images in Fig.7 (a) and (b) show two sharp turns, and the proposed method can make effective prediction of steering angles.

In the Fig 6 and Table I, the proposed IVGG-LSTM network performs more stable than the network proposed in [24], and the RMSE values are also smaller, because the proposed network is deeper than the the network proposed in [24]. Driving scenes are so complicated as to need a deeper network to express. Thus, the proposed network performs much better on imitating humans’ driving behaviors.

With the LSTM layer, the proposed network has a better ability to deal with dynamic objects such as other vehicles, pedestrians, and bicycles, etc. than the original VGG-16 network without the LSTM layer. Since there are driving behaviors such as overtaking cars and trucks in the testing roads, the planning model should make rational predictions of steering angles when overtaking or avoiding moving obstacles. Additionally, we improved the VGG-16 network and combined the LSTM layer with the improved VGG network to generate the deep cascaded network, so the proposed IVGG-LSTM network is more appropriate to extract the spatial and temporal features from continuous driving images. As shown in Fig.6 and Table I, the proposed network outperforms than the original VGG network.

Due to the application of CUDA Deep Neural Network library, the process of computing predictions is significantly accelerated. In our experiments, the proposed method can make 20 times prediction per second, while visual processing for humans roughly takes 150 ms [25]. Thus, the proposed method can be applied in real-time autonomous driving systems.

IV. Conclusions

A motion planning model based on a deep cascaded neural network for autonomous driving is proposed in this paper. An improved VGG network and a LSTM network which are used to extract spatial and temporal features in autonomous driving scenes, respectively, are designed to compose the deep cascaded neural network. For testing the proposed model, we built a dataset which includes about eight hours driving videos using a driving simulator. The experimental results show that the proposed model can effectively predict steering angles and make rational motion planning for autonomous vehicles according to the continuous driving images. Meanwhile, the real-time performance of the proposed planning model is good enough for autonomous driving systems. The future work will focus on testing the planning model on real traffic roads.

References

[1] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical Systems and Signal Processing,* vol. 100, pp. 482-500, 2018.

[2] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[3] S. A. Stouffer, A. A. Lumsdaine, M. H. Lumsdaine, R. M. Williams Jr, M. B. Smith, I. L. Janis, S. A. Star, and L. S. Cottrell Jr, "The American soldier: Combat and its aftermath.(Studies in social psychology in World War II), Vol. 2," 1949.

[4] L. B. Cremean, T. B. Foote, J. H. Gillula, G. H. Hines, D. Kogan, K. L. Kriechbaum, J. C. Lamb, J. Leibs, L. Lindzey, and C. E. Rasmussen, "Alice: An information ‐ rich autonomous vehicle for high ‐ speed desert navigation," *Journal of Field Robotics,* vol. 23, no. 9, pp. 777-810, 2006.

[5] H. Fuji, J. Xiang, Y. Tazaki, B. Levedahl, and T. Suzuki, "Trajectory planning for automated parking using multi-resolution state roadmap considering non-holonomic constraints," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, 2014, pp. 407-413: IEEE.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE*

conference on computer vision and pattern recognition, 2016, pp. 779-788.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, 2016, pp. 21-37: Springer.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks,* vol. 61, pp. 85-117, 2015.

[9] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature,* vol. 521, no. 7553, p. 436, 2015.

[11] L. Tai, S. Li, and M. Liu, "A deep-network solution towards model-less obstacle avoidance," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2016, pp. 2759-2764: IEEE.

[12] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE transactions on neural networks and learning systems,* vol. 28, no. 3, pp. 690-703, 2017.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.

[14] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research,* vol. 17, no. 1, pp. 1334-1373, 2016.

[15] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans Neural Netw Learn Syst,* vol. 28, no. 10, pp. 2222-2232, Oct 2017.

[16] P. Doetsch, M. Kozielski, and H. Ney, "Fast and robust training of recurrent neural networks for offline handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, 2014, pp. 279-284: IEEE.

[17] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba, "Addressing the rare word problem in neural machine translation," *arXiv preprint arXiv:1410.8206,* 2014.

[18] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625-2634.

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

[21] J. Zhang, S. Ma, M. Sameki, S. Sclaroff, M. Betke, Z. Lin, X. Shen, B. Price, and R. Mech, "Salient object subitizing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4045-4054.

[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[23] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net,* vol. 4, 2000.

[24] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, and J. Zhang, "End to End Learning for Self-Driving Cars," 2016.

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