

# Brain Inspired Cognitive Model with Attention for Self-Driving Cars

Shitao Chen, Songyi Zhang, Jinghao Shang, Badong Chen, Nanning Zheng\*, *Fellow, IEEE*

**Abstract**—Perception-driven approach and end-to-end system are two major vision-based frameworks for self-driving cars. However, it is difficult to introduce attention and historical information of autonomous driving process, which are the essential factors for achieving human-like driving into these two methods. In this paper, we propose a novel model for self-driving cars named brain-inspired cognitive model with attention (CMA). This model consists of three parts: a convolutional neural network for simulating human visual cortex, a cognitive map built to describe relationships between objects in complex traffic scene and a recurrent neural network that combines with the real-time updated cognitive map to implement attention mechanism and long-short term memory. The benefit of our model is that can accurately solve three tasks simultaneously: i) detection of the free space and boundaries of the current and adjacent lanes. ii) estimation of obstacle distance and vehicle attitude, and iii) learning of driving behavior and decision making from human driver. More significantly, the proposed model could accept external navigating instructions during an end-to-end driving process. For evaluation, we build a large-scale road-vehicle dataset which contains more than forty thousand labeled road images captured by three cameras on our self-driving car. Moreover, human driving activities and vehicle states are recorded in the meanwhile.

**Index Terms**—autonomous mental development, cognitive robotics, end-to-end learning, path planning, vehicle driving.

## I. INTRODUCTION

**A**UTOMATICALLY scene understanding is the core technology for self-driving cars, as well as a primary pursuit of the computer vision. During the recent decades, considerable progress and development have been achieved in the field of vision-based self-driving cars. It is well known that most of related information required for self-driving cars can be obtained by cameras, which is originally inspired by humans' driving behaviors. Besides, the mechanism of attention can help people choose the effective data in memory to determine the objects existing in the current image and their relationships, so as to form correct decisions in current moment. Therefore, it is significant to develop a self-driving car only based on vision [1], [2], [3], where the mechanism of attention is felicitously implemented.

Nowadays, there are two popular vision-based paradigms for self-driving cars: the perception-driven method and the end-to-end method. For the **perception-driven method** [4], it is required to establish a detailed representation of the real world.

By fusing multi-sensor data, a typical world representation usually contains descriptions of the prime objects in various traffic scene, including (but not limited to) lane boundaries, free space, pedestrians, cars, traffic lights and traffic signs. With such descriptions, the path planning module and the control module are used to determine the actual movement of the vehicle. In the path planning process, besides accurate perception results and high-precision map information, it is often necessary to design some assistant rules manually. Such a motion trajectory needs to be adjusted and updated in real-time according to the state of the vehicle at each moment, taking the temporal dependences into account, so as to form a correct trajectory sequence. With the motion trajectory calculated by planning module, the vehicle is steered to track each task point on the planned path under the guidance of high-precision positioning information.

As for the **end-to-end method** [5], based on the breakthrough of convolutional neural networks (CNN) [6] and GPU technology, a deep neural network is able to learn the entire processing pipeline needed for controlling a vehicle, direct from the human driving behaviors. Instead of using hand-crafted features as in perception-driven method, we enable the CNN to learn the most valuable image features automatically and directly map them to the control of the steering angle. Since the actual control of the car only relates to the velocity and steering angle, such method of directly mapping images to the control of direction, is more efficient and effective in some scenarios.

The perception-based method was the most widely used one in the past decades. It can be applied to most challenging tasks, but the disadvantage is that all features and task plans are manually designed, and the entire system lacks the self-learning ability. In recent years, the end-to-end learning strategy for self-driving gradually boomed with the success of deep learning [7]. End-to-end strategy merely requires some visual information, and is capable of learning from human driving behaviors. However, the disadvantage is, when the system structure is simple, the external information is unable to be introduced in to control the behavior of the self-driving system. Therefore, while the system is running, we have no way to know where the vehicle is going, neither can we control the system as well. In the meanwhile, temporal information has never been considered in this end-to-end process.

In our point of view, it is believed that a highly effective and reasonable autonomous system should be inspired by the cognitive process of human brain. First of all, it is able to perceive the environment as rationally as the visual cortex, and then to process the perception results in a proper way. After

S. Chen, S. Zhang, J. Shang, B. Chen and N. Zheng are with the Department of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R. China. E-mail: chenshitao, zhang-songyi@stu.xjtu.edu.cn; chenbd, nnzheng@mail.xjtu.edu.cn.

\* Correspondence: nnzheng@mail.xjtu.edu.cn

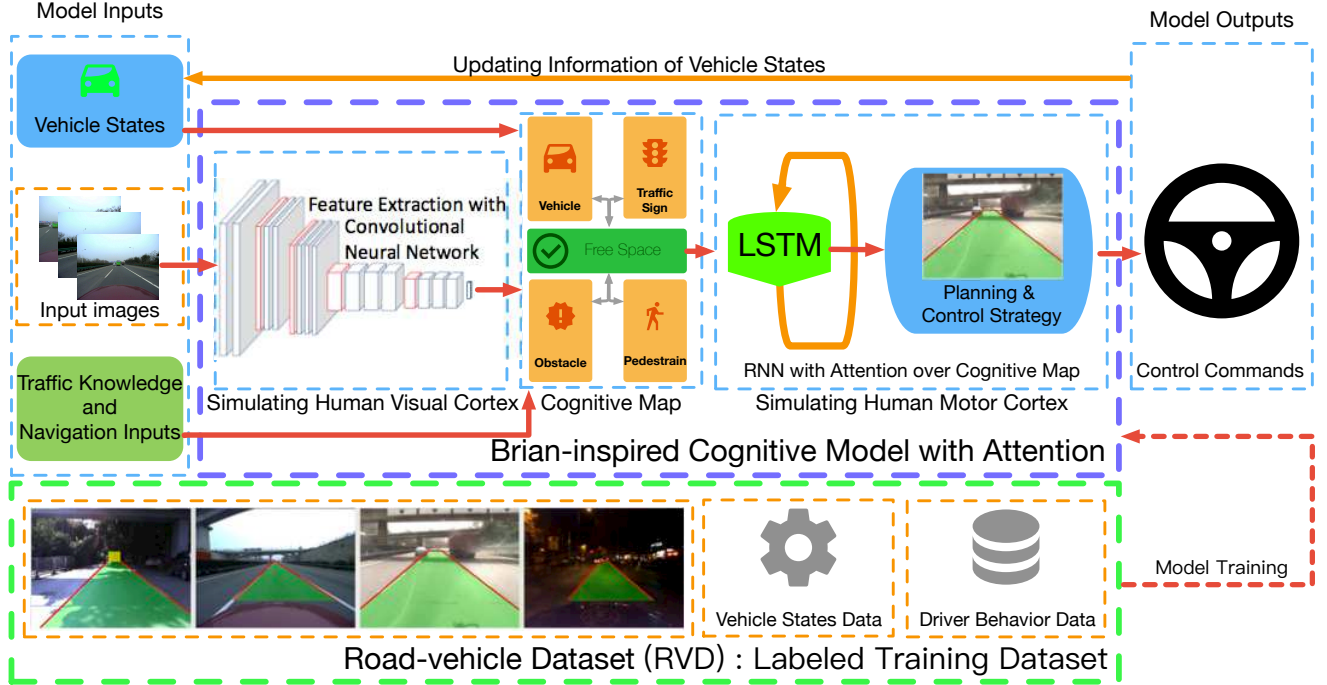


Fig. 1. Framework of our cognitive model with attention (CMA). The road images are first processed by multiple convolutional neural networks to simulate the function of human visual cortex. A cognitive map consists of vehicle states, navigation inputs and perception results. Based on the information of cognitive map, recurrent neural network models the attention mechanism by historical states and scene data in time to form driving decision in each computing step.

that, the system plays a role of the motor cortex to plan and control the driving behaviors. And in the whole process, the concept of human-computer collaborative hybrid-augmented intelligence [8] is well referred, so that the self-driving system can learn smartly from human driving behaviors.

In this paper, we aim to build a brain-inspired cognitive model with attention. When a person view a scene, the message flows through LGN to V1, onward to V2, then to V4 and IT [9], which occurs within the first 100ms of a glance to objects. This process is proved to be highly similar to the operating principle of the convolutional neural network. Thus in our model, we adhere to apply CNNs for the processing of the visual information, which is a simulation of the visual cortex to process information. Similarly, as in *On Intelligence*, Jeff Hawkins argues [10] that time holds the vital place in brain when solving a problem. We believe that brain has to deal with spatial and temporal information simultaneously, since spatial patterns need to be coincident with temporal patterns. Therefore, we need to simulate the functions of motor cortex, which means, in dealing with planning and control problems, a long-term memory must be considered to form the optimal driving strategy for the current. With this motivation, it is necessary to introduce the attention mechanism into the cognitive computing model for self-driving cars, which allows the model to choose reasonable information from a large set of long-term memory data at each computing step.

Moreover, Mountcastle *et al.* [11] points out that the functional areas in the cerebral cortex have similarities and consistency. He believes the regions of cortex that control muscles are similar to the regions which handle auditory or optical inputs in structure and function. Inspired by this, we argue that

the recurrent neural network (RNN), which performs well in processing sequential data and has been successfully applied in video sequence classification and natural language processing tasks, is also capable to solve planning and control problems simultaneously as human motor cortex. The discussion above is an important motivation for us to implement planning and control decision with RNN.

In order to introduce attention mechanism into the proposed cognitive model, and to solve the problem that general end-to-end models cannot introduce external information to guide, we define the concept of cognitive map in real traffic. The term of **cognitive map** was first coined by Edward Tolman [12] as a type of mental representation of the layout of one's physical environment. Thereafter, this concept was widely used in the fields of neuroscience [13] and psychology. The research results on cognitive map in these areas provide an important inspiration for us to construct a new model of autonomous driving. To apply this concept to the field of self-driving, combining with our work, **cognitive map for traffic scene** is built to describe the relationship between objects in complex traffic scene. It is a comprehensive representation of the local traffic scene, including lane boundary, free space, pedestrian, automobile, traffic lights and other objects, as well as the relationships between them, such as direction, distance, etc. Furthermore, the prior knowledge of traffic rules and the temporal information are also taken into consideration. The cognitive map defined in this paper is essentially a structured description of vehicle state and scene data in the past. This description forms the memory of a longer period of time. The proposed cognitive model, in which the cognitive map combines long-short term memory, mimics the human driving

ability to understand about traffic scene and to make decisions of driving.

In precise, our framework first extracts valid information from the traffic scene of each moment by a convolutional neural network to form the cognitive map, which contains the temporal information and a long-term memory. On this basis, we add external control information to some descriptions of the cognitive map, e.g., the guidance information from the navigation map. And finally, we utilize a recurrent neural network to model attention mechanism based on historical states and scene data in time, so as to perform path planning and control to the vehicle.

With all above, in our model, a novel self-driving framework combined with attention mechanism has come to form, which is inspired by human brain. The framework is named brain-inspired cognitive model with attention (CMA). It is able to handle the spatial-temporal relationships, so as to implement the basic self-driving missions. In this paper, we realized a self-driving system with only vision sensors. It performs well in making path planning and producing control commands for vehicles with attention mechanism. Fig. 1 shows the main scheme of our CMA method. The remainder of the paper is organized as follows: in section II, we review some previous studies of self-driving cars; in section III, we describe our approach in detail; in section IV and V, we present a large-scale labeled self-driving dataset and the evaluation of our method; finally section VI concludes the work.

## II. RELATED WORK

In the past decades, remarkable achievements [14], [1], [4] have been reached with perception-driven method in the filed of self-driving cars. Several detection methods for car and lane boundary have been proposed to build a description of the local environment.

Many lane detection methods in [15], [16], [17], [18] have been developed to locate the lane position with canny edge detection or hough transformation. The defect of these methods is that they lack some geometric constraints to locate the arbitrary lane boundary. Therefore, Nan *et al.* [19] presented a spatial-temporal knowledge model to fit the line segments, which finally outputs the lane boundaries. Huval *et al.* [20] introduced a deep learning model to perform lane detection at a high frame rate. Different from the traditional lane boundary detection approach whose output is the pixel location of the lane boundary, the work [14] represented a novel idea which uses convolutional neural network to map an input image directly to a deviation between vehicle and lane boundary. With this method, the output of the neural network can be directly used in controlling the vehicle, without coordinate transformation. The limitation of this model is that per-training for a specific vehicle is needed.

For object detection task, researches [21], [22] adopt the method of generating a bounding box to describe the location of the object. However, in the self-driving task, it is not necessary to get a precise location of the bounding box. We only need to know if there is a obstacle in our lane and how far the obstacle is. Thus it is a more convenient and efficient

way to represent the obstacle as a point instead of a bounding box.

The concept of end-to-end learning method was originally inspired by Pomerleau *et al.* [23], and it was further developed in the works [24], [5], [25]. Pomerleau *et al.* [23] attempted to use a neural network to navigate an autonomous land vehicle. With breakthrough of deep learning, DAVE-2 in [5] learned the criterion to steer a vehicle automatically. Similarly, Xu *et al.* presented a FCN-LSTM architecture in [25], which can predict egomotion of the vehicle by its previous state. All the works above lack the ability to supervise the action of the vehicle, which means we have no way to know where the vehicle is going, although the vehicle may safely drive on road.

Several control strategies using deep learning approach to control robot have been proposed in many papers. A vision-based reinforcement learning method and evolve neural network as a controller in TORCS game have been reported in [26], [27]. Reinforcement learning approach in [28], [29], [30] has been successfully used to train the artificial agent which has an capability to play several games. Although the combination of convolutional neural network and reinforcement learning has shown a good performance in some strategic games [31]. This is because the decision-making in such games usually relies on a short-term of time information or the current image information. However, for complex tasks such as self-driving cars, planning and control decisions must be made with a long-term information, so as to form the optimal driving strategy for current in real traffic scene. In [14], a direct perception approach is used to manipulate a virtual car in TORCS game. The controller in this work is a hand-crafted linear function which directly uses vehicle's position and pose. This approach may preform well in game environment, but the action generated by this function is different from human's behavior and it can not be applied in real traffic as well. Other path planning and control methods for self-driving car in [32] commonly require real time GPS information to form a real trajectory. Nevertheless, as known that a human being can drive a car only by visual information, it is a promising way to develop a model which can handle planning and control simultaneously based only on vision.

## III. BRAIN-INSPIRED COGNITIVE MODEL WITH ATTENTION

In driving, human visual cortex and motor cortex play the leading roles. On the one hand, the visual cortex contributes to perceiving environment and form a cognitive map of the road scene by combining the memory of the traffic knowledge with external information, such as map navigation information. On the other hand, planning and control are determined by the motor cortex. With the information from the cognitive map in a long memory, mechanism of attention will help people discover the most significant information in time to form planning and control strategy. In one word, the entire driving behavior consisting of sensing, planning and control are guided and inferred mainly by the above two cortexes in brain.

Similarly, a brain-inspired model based on the perception, memory and attention mechanisms of human can be constructed. In this paper, it is believed that the most primary

perception relies on a single frame of the road scene. However, as for planning and controlling processes, multiple frames and many historical states of vehicle are required to form long memory or short memory to actually manipulate the self-driving car.

Fig.1 shows our CMA method for self-driving cars. Our ultimate goal with this network is to build a cognitive model with attention mechanism, which can handle sensing, planning and control at the same time. And differing from other deep learning or end-to-end learning methods which just map input images to uncontrollable driving decisions, our network can not only make a car run on a road, but also accept external control inputs to guide the actions of the car. To achieve this goal, the road images are first processed by multiple convolutional neural networks to simulate the function of human visual cortex and form a basic cognitive map similar to human brain's, which is a structured description of the road scene. And this description of road contains both human-defined features and latent variables learned by the network. A more explicit cognitive map can be constructed based on the contents of the basic cognitive map above and combined with prior knowledge of traffic, states of vehicle and external traffic guidance information. Therefore, the cognitive map is built by the description of the road scene, the current states of vehicle and the driving strategy of the near future. Through the recurrent neural network(RNN), the cognitive map formed in each frame is modeled to give a temporal dependency in motion control, as well as the long-term and short-term memory of past motion states, imitating human motor cortex. Finally, the real motion sequences with consideration of planning and control commands of self-driving cars can be generated.

#### A. Perception Simulating Human Visual Cortex

The purpose of the CMA framework is to solve the defect that the conventional end-to-end learning methods can not incorporate external control signals, which causes that the vehicle can only produce an action based on the input image, but does not know where it will go. By constructing a cognitive map, additional control information can be put into the end-to-end self-driving framework. The establishment of a cognitive map primarily relies on the perception of the environment. It is well known that the perception of environment is the focus and challenge of self-driving missions. Thus, in the CMA framework, inspired by the architecture of human visual cortex, we use a state-of-art convolutional neural network to learn and generate basic descriptions of road scene. We fixed three cameras in our self-driving car to capture the scene in current lane and the lanes on both sides. Different from the conventional method, the scene representations in our approach are learned by convolutional neural network but not hand-crafted. We apply several convolutional layers to process images captured by on-board cameras on vehicles. In our approach, we launch multiple convolutional networks to extract different road information from different camera views. Instead of directly using the network as a classifier, we utilize it as a regressor to directly map an input image to several key pixel points that will be used in path planning and control.

With the pixel points extracted by multiple CNNs, one can calculate and construct a basic cognitive map to describe the local environment surrounding the vehicle, as shown in Fig. 2.

We define the input images as  $\mathbf{I}^t = \{I_m^t, I_l^t, I_r^t\}$ , which are captured by the middle, left and right cameras, respectively. Based on the input images, a self-driving car needs to know the accurate geometry of the lane and position of the obstacles. So the output vector  $X^t$  of the convolutional neural network for each camera is composed with five different point identities, that is

$$X^t = [p_{l\_t}, p_{l\_b}, p_{r\_t}, p_{r\_b}, p_o], \quad (1)$$

where  $p_{l\_t}$  and  $p_{l\_b}$  represent the x-coordinates of the intersections of the left lane boundary's extended line with the top and the bottom edges of image plane,  $p_{r\_t}$ ,  $p_{r\_b}$  denote the x-coordinates corresponding to the points on the right lane, and the  $p_o$  stands for the y-coordinate of the obstacle point in corresponding lane.

To achieve a high performance in real-time, the architecture of our convolutional neural network is very simple and shallow. Five convolutional layers are utilized to extract spatial features from each image  $I^t$ . The configurations of the five convolutional layers are the same as [5]. The first three layers are stride convolutional layers with a  $2 \times 2$  stride and a  $5 \times 5$  kernel. While the last two convolutional layers have a  $3 \times 3$  kernel and with no stride. Since we use the network to locate feature points rather than classify, the pooling layer which makes the representation become invariant to small translations of the input is unnecessary in our network. The convolution operation without pooling layer is expressed as

$$Z_{i,j,k} = \sum_{l,m,n} [V_{l,(j-1) \times s + m, (k-1) \times s + n} K_{i,l,m,n}], \quad (2)$$

where  $Z$  and  $V$  are the output feature maps with  $i$  channels and input feature with  $l$  channels,  $s$  denotes the number of stride,  $j$  and  $k$  are the indexes of row and column. A rectified linear units (ReLU) is used for each hidden neurons of the convolutional layers. Following the five convolutional layers, three fully connected layers are utilized to map the representations extracted by the convolutional layers, to the output vector  $X^t$ .

According to the five descriptors in  $X^t$ , we can calculate the physical quantities as shown in Fig. 2. Suppose  $D_{m\_left}$  and  $D_{m\_right}$  are the lateral distances to the vehicle respectively from the left and the right lane boundary in the view of the middle camera.  $D_{l\_left}$ ,  $D_{l\_right}$ ,  $D_{r\_left}$  and  $D_{r\_right}$  are respectively the two distances in the view of the left and the right camera, similar to the middle one. We define the angle between vehicle and road as  $V_a$ , and obstacle distances in each lane as  $O_{C\_line}$ ,  $O_{L\_line}$  and  $O_{R\_line}$ . With the obstacle distances, driving intention  $D_i^t$  could be derived by Algorithm 1. For calculating these physical distances described above, we define that any two pixel points in a lane boundary are  $(l_{xm}, l_{ym})$  and  $(l_{xb}, l_{yb})$ , and y-coordinate of the obstacle is  $o_y$ . With the optical center  $(u_0, v_0)$ , and the height of camera  $H$ , the positions of two points  $(X_m, Z_m)$ ,  $(X_b, Z_b)$  in vehicle coordinate system can be presented as

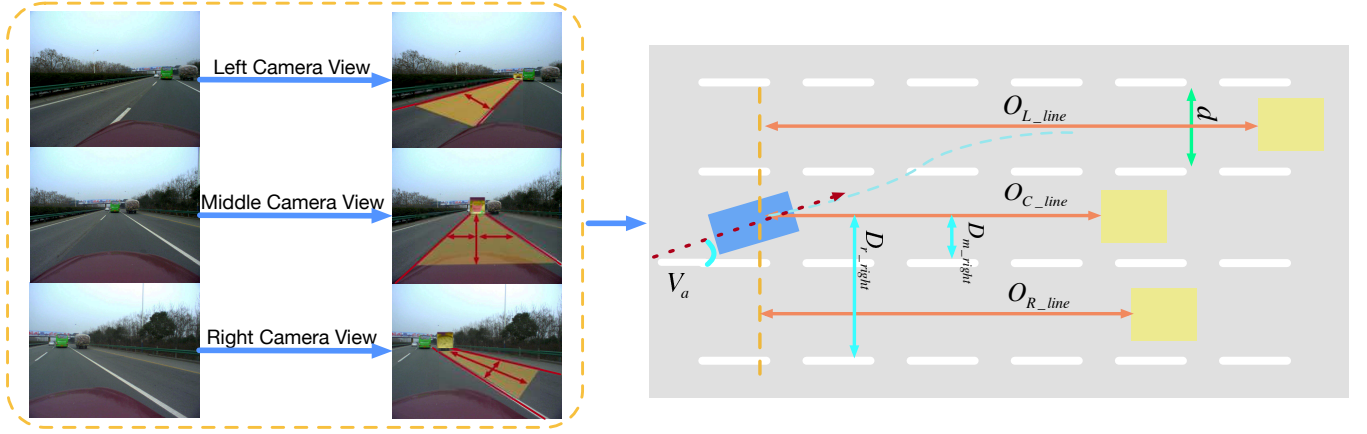


Fig. 2. Illustration of constructing a basic cognitive map. The convolutional neural network will output the pixel location of lane boundary and obstacle vehicles. And based on the output, a description of local environment can be generated.

$$(X_m, Z_m) = \left( \frac{(l_{xm} - v_0) \cdot H}{(l_{ym} - u_0)}, \frac{f \cdot H}{l_{ym} - u_0} \right), \quad (3)$$

$$(X_b, Z_b) = \left( \frac{(l_{xb} - v_0) \cdot H}{(l_{yb} - u_0)}, \frac{f \cdot H}{l_{yb} - u_0} \right). \quad (4)$$

The distance  $D$  from vehicle to the lane boundary can be obtained by

$$D = \frac{|X_m Z_b - X_b Z_m|}{\sqrt{(X_b - X_m)^2 + (Z_b - Z_m)^2}}. \quad (5)$$

Meanwhile, the angle between the vehicle and the lane boundary  $V_a$  is

$$V_a = \arctan \frac{X_b - X_m}{Z_b - Z_m}. \quad (6)$$

Similarly, obstacle distance in each lane is presented as

$$O = \frac{f \cdot H}{o_y - u_0}. \quad (7)$$

By Eq. 3, 4, 5, 6 and 7, we can obtain the perception results  $\{X_m^t, X_l^t, X_r^t\}$  from the views of three cameras.

### B. Planning and Control Simulating Human Motor Cortex

The structured description of cognitive map  $C^t$  with the vehicle states  $V_{States}$ , formed as

$$C^t = [X_m^t \ X_l^t \ X_r^t \ D_i^t \ V_{states}], \quad (8)$$

is extracted from each frame in three cameras. With these representations, the CMA method will model a temporal dynamical dependency of planning and control. In an ordinary self-driving framework, path planning and vehicle control are two separate tasks. Different from traditional method, the new approach contains memories of the past states and generates control commands with recurrent neural network, so the two tasks are driven simultaneously.

Long Short-Term Memory (LSTM) is a basic unit of recurrent neural network and it's well known in processing sequential data and molding temporal dependencies. LSTMs have many varieties, a simple one is used in our CMA framework. One cell in a LSTM unit is controlled by three

### Algorithm 1 Generating Driving Intentions with Basic Cognitive Map in Real Traffic

**Input:**

$G_{navi}$ : guidance information from the navigation

**Output:**

$D_i^t$ : drive intention based on obstacle distances and navigation signal;

- 1: **if**  $G_{navi} = \text{stay in line} \wedge O_{C\_line} \geq \text{safety distance}$  **then**
- 2:     **return**  $D_i = \text{stay in line}$
- 3: **else if**  $G_{navi} = \text{stay in line} \wedge O_{C\_line} \leq \text{safety distance}$  **then**
- 4:     **if**  $O_{L\_line} \geq \text{safety distance}$  **then**
- 5:         **return**  $D_i^t = \text{change to left}$
- 6:     **else if**  $O_{R\_line} \geq \text{safety distance}$  **then**
- 7:         **return**  $D_i^t = \text{change to right}$
- 8:     **else**
- 9:         **return**  $D_i^t = \text{break and stay in line}$
- 10:    **end if**
- 11: **else if**  $G_{navi} = \text{change to left} \wedge O_{L\_line} \geq \text{safety distance}$  **then**
- 12:     **return**  $D_i^t = \text{change to left}$
- 13: **else if**  $G_{navi} = \text{change to right} \wedge O_{R\_line} \geq \text{safety distance}$  **then**
- 14:     **return**  $D_i^t = \text{change to right}$
- 15: **else**
- 16:     **return**  $D_i^t = \text{break and stay in lane}$
- 17: **end if**

gates (input gate, output gate and forget gate). Forget gate and input gate use a sigmoid function, while output and cell state are transformed with  $\tanh$ . With these gates, LSTM network is able to learn long-term dependencies in sequential data and model the attention mechanism in time. We employ the outstanding characteristics of LSTM in our CMA framework, so that it can learn human's behaviors during a long-term driving process. The memory cell is used to store information in time, such as historical vehicle states in its vectors, which can be chosen with attention mechanism in the network. Meanwhile, the dimension of the hidden states should be

chosen according to the input representation  $C^t$ .

Based on the driving intention  $D_i$  in cognitive map  $C^t$ , a lateral distance  $D_o$  from the current point to the objective lane is calculated by Algorithm 2. Then a more complicated cognitive map  $C^t$  is built as

$$C^t = [X_m^t, X_l^t, X_r^t, D_i^t, V_{states}^t, D_o^t]. \quad (9)$$

Before importing the representation of cognitive map  $C^t$  to the LSTM block, a fully connected layer is utilized to organize the various information in  $C^t$ . The representation  $R^t$  can be presented by the fully connected layer with a weight matrix  $W_f^C$  and a bias vector  $b_f^C$  as

$$R^t = W_f^C C^t + b_f^C. \quad (10)$$

The descriptor  $R^t$  contains much latent information organized by the fully connected layer, which means a description of traffic scene and driving intentions. The effectiveness of this descriptor can be improved by an appropriate training process with human driving behaviors.

In the proposed CMA framework, we adopt three LSTM layers, making the network learn higher-level temporal representations. The first two LSTM blocks return their full output sequences, but the last one only returns the last step in its output sequence, so as to map the long input sequence into a single control vector to vehicles. The mapping  $\Phi^R$  contains three LSTM layers with parameters  $\Theta^R$  to explore temporal clues in the representation set

$$\{R^t\}, t = 1, 2, \dots, n, \quad (11)$$

which contains multiple cognitive results in different time steps. The hidden state  $h_3^t$  is the  $t$ th output of the third layer and it shows the result of a temporal model which processes the tasks of path planning and control. The hidden state  $h_3^t$  is presented as

$$h_3^t = \Phi^R(\Theta^R, \{h^t\}, \{R^t\}), \quad (12)$$

where  $\{h^t\}, t = 1, 2, \dots, n$ , is the hidden states set of each LSTM layer. Then, a fully connected layer defined by weights  $W^R$  and bias  $b^R$  will map the hidden state  $h_3^t$  to a driving decision used to control self-driving cars. In a word, with the memory and predictive abilities in an LSTM block, we consider planning and controlling process as a regression problem.

In automatic driving mode, steering command and velocity command are used to control the vehicles. In our self-driving framework, we generate these two commands separately. For the velocity command, it will be determined by traffic knowledge and traffic rules. The real velocity of the vehicle will be treat as a part of vehicle state  $V_{states}^t$  to form cognitive map of current time step. According to a long-term cognitive map  $\{C^t\}$ , steering angle  $S_a$  is generated by

$$S_a = W^R \cdot \Phi^R(\Theta^R, \{h^t\}, \{W_f^C C^t + b_f^C\}) + b^R, \quad (13)$$

which is described above in detail.

Suppose  $D_{m\_left}$  and  $D_{m\_right}$  are respectively the lateral distance from the left and the right lane boundary to vehicle in the view of middle camera. And  $D_{l\_left}$ ,  $D_{l\_right}$ ,  $D_{r\_left}$  and  $D_{r\_right}$  are respectively the two distances in the left and

right camera views, same as the middle one. We define the angle between vehicle and road as  $V_a$ . And we use  $V_{states}$  as a representation of the states of vehicle which can be obtained through OBD port in vehicle.

With these notations, the entire workflow of CMA framework is summarized in Algorithm 2. In a self-driving procedure, cognitive map is first constructed with the multiple convolutional neural networks. Subsequently, based on the cognitive map and vehicle states in a period of time, a final control command will be generated by recurrent neural network.

---

**Algorithm 2** Planning and Control Processes by Cognitive Map and LSTM

---

```

1: while in self-driving mode do
2:   generate cognitive map  $C^t$  with multiple CNNs
3:   if  $D_i = \text{stay in line}$  then
4:      $D_o = \frac{D_{m\_left} + D_{m\_right}}{2}$ 
5:     generate steering angle by RNN
6:     generate pedal commands by desired speed
7:   else if  $D_i = \text{change to right}$  then
8:      $d = D_{r\_right} - D_{r\_left}$ 
9:     while changing lanes do
10:      if  $D_{m\_right} < D_{r\_right}$  then
11:        the car still in current lane
12:         $D_o = D_{r\_right} - \frac{d}{2}$ 
13:        generate steering angle by RNN
14:        generate pedal commands by desired speed
15:      else if  $D_{m\_right} == D_{r\_right} \vee$ 
the car is on the boundary of lane then
16:        if  $(D_{m\_right} - D_{m\_left}) \gg d$  then
17:          the car is on the boundary of lane
18:           $D_o = D_{m\_right} - \frac{d}{2}$ 
19:          generate steering angle by RNN
20:          generate pedal commands by desired speed
21:        else
22:          the car has changed lane
23:           $D_i = \text{stay in line}$ 
24:          break
25:        end if
26:      end if
27:    end while
28:   else if  $D_i = \text{change to left}$  then
29:     while changing lanes do
30:       similar to change to right lane
31:     end while
32:   end if
33: end while

```

---

#### IV. DATA COLLECTION AND DATASET

For exploring a new framework of self-driving cars and evaluate the proposed method, we construct and publish a novel dataset: Road-vehicle Dataset (RVD), for training and testing our model.<sup>1</sup> The platform of data collection is shown

<sup>1</sup>The Road-vehicle Dataset is available at [air.xjtu.edu.cn/xszy/RVD.htm](http://air.xjtu.edu.cn/xszy/RVD.htm)



in Fig. 3a. The data is collected on a wide variety of roads in different lighting and weather conditions. In total, we recorded more than 10 hours of traffic scenarios using different sensors, such as color cameras, high-precision inertial navigation system and differential GPS system. Three cameras are used to record images and the data of driver's behaviors, which are reflected through the steering angle and pedals' states, are recorded by the CAN bus of OBD interface. The GPS/IMU inertial navigation system is used to record accurate attitude and position of our vehicle. Meanwhile, in order to generate images at different views, we use viewpoint transformation method to augment our dataset.



Fig. 3. *Challenger* self-driving car. (a) *Challenger* is based on a *CHERY TIGGO*. The *TIGGO* has four-wheel drive system (4WD), and a 4-speed auto transmission. (b) Computing system and GPS/IMU inertial navigation system in the vehicle's trunk. (c) Three cameras on the top of the vehicle. Each camera monitors a corresponding lane. (d) A mechanism with a DC motor installed on the steering is used to control the steering electronically.

#### A. Sensors and Calibration

The sensor setup is shown in Fig. 3b, Fig. 3c, and Fig. 3d:

- 3 × IDS UI-5240CP color cameras, 1.31 Megapixels, 1/1.8" e2v CMOS, global shutter
- 3 × KOWA LM3NCM megapixel lens, 3.5mm, horizontal angle of view 89.00°, vertical angle of view 73.8°
- 1 × NovAtel ProPak6 Triple-Frequency GNSS Receiver, TERRASTAR-C accuracy 4cm
- 1 × OXTS RT2000 inertial and GNSS navigation system, 6 axis, 100Hz

For calibrating the extrinsic and intrinsic parameters of the three vehicle mounted cameras, we use trilinear method [33] to calibrate extrinsic parameters and method proposed by Zhang et.al [34] to calibrate intrinsic parameters. The position matrix is given as  $T$ , pitch angle matrix is given by  $R$ , and internal parameters matrix is given by  $I$ . And these parameters will be used later in the experiment part.

#### B. RVD Dataset

In the data acquisition process, drivers were asked to drive in a diverse set of weather and road conditions at different times of the day. Furthermore, to collect abundant data of driving behaviors, we required the drivers to do lane changing and turning operations in suitable cases. For the mission of self-driving, it is a key problem to make the vehicle recover from error states. Therefore, with the method of viewpoint transformation, we combined the data from three cameras to simulate the visual data of the error-state vehicle, and generated additional road images in a variety of viewpoints.

In precise, our dataset covers:

1) *Diverse Visual data*: Visual data including kinds of roads, such as the urban roadways and the highways of single-lane, double-lane and multi-lane. The data was collected in different weather conditions, such as day, night, sunny, cloudy, and foggy, by three viewpoint changeable cameras, which extends our data scale up to 146,980 images.

2) *Vehicle States Data*: We recorded the data of real-time vehicle states while collecting the road video, where more than 100 kinds of internal and external vehicle information, such as speed, attitude and acceleration, was included.

3) *Driver behavior Data*: We collected real-time behavior (operation to the vehicle) of the driver in each moment, including steering angle, control to the accelerator pedal and the brake pedal.

4) *Artificial Tagged Data*: In our collected video, we manually tagged the 43,621 pieces of road image data, where the lane position, obstacle location, etc. were marked.

Our dataset is innovative in two aspects: i) covers most of the visual data in scene of self-driving, all data are collected by three points of view simultaneously, and the dataset is expanded in the later stage by means of viewpoint transformation; ii) contains abundant records of vehicle states data and human drivers' driving behaviors, which provide better exploration and training for the end-to-end frameworks of self-driving cars.

## V. EXPERIMENTS

In this section, experiments are presented to show the effectiveness of the proposed CMA model. Comprehensive experiments are carried out to evaluate the cognitive map (free space, lane boundary, obstacle, etc.) formed by CNNs based on the data of real traffic scene videos. We also evaluate the path planning and vehicle control performance of the RNN that integrates the cognitive map with the long-short term memory. In addition, a simulation environment is set up, in which some experiments hard to operate in reality can be carried out. All experiments are based on the *challenger* self-driving car, as shown in Fig. 3, a tuning vehicle on *CHERY TIGGO*.

### A. Constructing Cognitive Maps with Multiple Convolutional Neural Networks

1) *Visual Data and Visual Data Augmentation*: Training a convolutional neural network to generate the descriptions of road scenarios needs a lot of image data. Although we've already got plenty of images taken by the vehicle-mounted camera, it's still hard to cover all possible situations a self-driving vehicle may encounter. Only recording images from the driver's point of view is inadequate, so our method should have an ability to adjust the vehicle to recover from an error state. For example, the samples taken from a human-driven vehicle can not cover the situation where its yaw angle is very large, since a human driver does not allow the car to deviate too much from the middle of the road. Thus, the collected data may be unbalanced, and it is hard to train a network that can deal with various road situations. In order to augment our training data and simulate a variety of attitudes of vehicle driving on the lane, we propose a simple viewpoint transformation method to generate images with different rotations from the direction of the lane. We record images from three different viewpoints by three cameras mounted on the vehicle, and then simulate other viewpoints by transforming the images captured by the nearest camera. Viewpoint transformation requires the precise depth of each pixel which we cannot acquire. However in our research, we only care about the lanes on the road. We assume that every point on the road is on a horizontal plane. So one can accurately calculate the depth of each pixel on the road area by the height of camera.

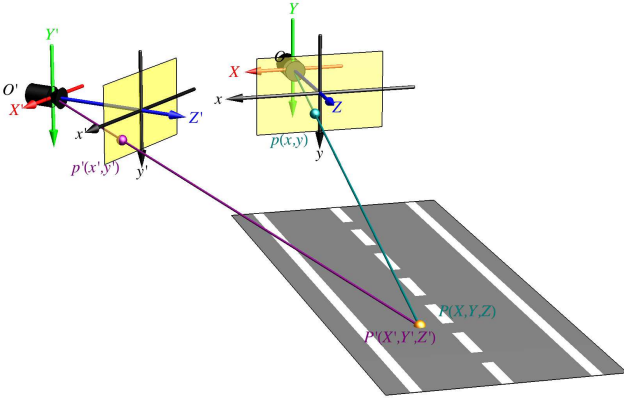


Fig. 4. Principle of viewpoint transformation.

Suppose a point  $p(x, y)$  in the image coordinate system is known, the pitch angle of the camera is approximate to 0. With the basic camera model illustrated in Fig. 4, one can obtain the position of the point  $P(X, Y, Z)$  in the camera coordinate system as

$$Z = Y \times \frac{f}{y}, \quad (14)$$

$$X = Z \times \frac{x}{f}, \quad (15)$$

where  $Y$  equals the height  $h$  of the camera, and the focal length of the camera is  $f$ .

A point  $P'(X', Y', Z')$  in the simulated camera coordinate system can then be derived as

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + T, \quad (16)$$

where  $R$  is the rotation matrix and  $T$  is the translation vector. Therefore, as shown in Fig. 5, the augmented samples are generated.

2) *Effects of Multiple CNNs in Constructing Cognitive Map*: Within the proposed CMA framework, the CNN regressor takes responsibility for the construction of a basic cognitive map. In a vision based self-driving car, how precisely the visual module can perceive the surrounding environment is of particular importance, and the perception results directly affect the planning and control. In most self-driving scenarios, detections of the free space, the current and adjacent lanes, as well as the obstacle vehicles are primary indicators. Therefore, we mainly evaluate our model on detecting these items.

TABLE I  
PARAMETERS OF THE FIVE CONVOLUTIONAL LAYERS

Layers	Operations	Attributions
1st	Convolution	Size: $[5 \times 5 \times 3 \times 24]$
	Activation	ReLU
	Max pooling	(Not Used)
2nd	Convolution	Size: $[5 \times 5 \times 24 \times 36]$
	Activation	ReLU
	Max pooling	(Not Used)
3rd	Convolution	Size: $[5 \times 5 \times 36 \times 48]$
	Activation	ReLU
	Max pooling	(Not Used)
4th	Convolution	Size: $[5 \times 5 \times 48 \times 64]$
	Activation	ReLU
	Max pooling	(Not Used)
5th	Convolution	Size: $[5 \times 5 \times 64 \times 128]$
	Activation	ReLU
	Max pooling	(Not Used)

Our CNN regressor is built on TensorFlow [35]. As given in Table I, there are 5 convolutional layers in our model. An image with resolution  $320 \times 240$  is processed by those convolutional layers to form a tensor with dimension  $25 \times 33 \times 128$ . And 4 fully connected layers with output size 500, 100, 20, 5 are used to map 128 features to a 5-dimensional vector, which represents the lane boundary and obstacle location in the input image.

**The performance of free-space estimation** is evaluated by a segmentation-based approach. In our approach, we can estimate the free-space in the ego-lane or the adjacent lanes. Free-space in a lane is determined by two lane boundaries and obstacle position in the lane. We evaluate the consistency between the model output and accurately labeled ground truth. Despite that in [36], the authors recommend to estimate the free space in bird's eye view regardless of the type of traffic scenarios, here in our model, we alternatively choose to evaluate by perspective image pixels. Since what we concern indeed is the free-space in a lane, the adopted perspective is more convenient for us. We choose the criteria of *precision*,



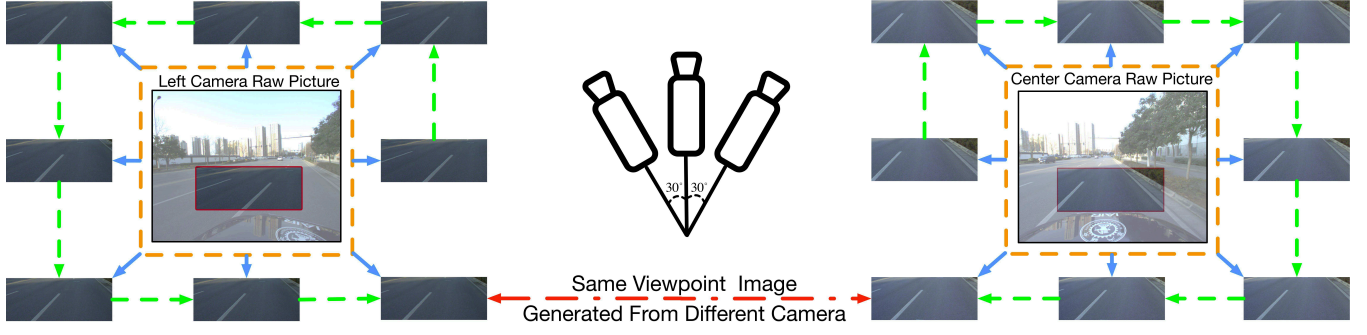


Fig. 5. Illustration of data augmentation process. Images in different viewpoints are generated by the method of viewpoint transformation. The origins are the actual images taken by the three cameras.

*recall* and *F-measure* to evaluate the performance of our model, which are defined by:

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad (17)$$

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad (18)$$

$$F_1 = \frac{2}{Precision^{-1} + Recall^{-1}} = \frac{2N_{TP}}{2N_{TP} + N_{FP} + N_{FN}}, \quad (19)$$

where  $N_{TP}$  is the number of free-space pixels correctly labeled as ground truth,  $N_{FP}$  is the number of free-space pixels in model output but not in the ground truth labeling, and  $N_{FN}$  is the number of free-space pixels in ground truth but not in model output. In our study, three lanes (ego-lane, two adjacent lanes) are captured by three different cameras. We only present the evaluation results on the images of the middle camera and the left camera, since the images of the right camera are similar to those of the left one.

For comparison purpose, the free space detection performance is evaluated on the RVD and Nan's dataset [19], respectively. Table II presents the quantitative analysis of our model in different scenarios. Some results of free-space detection on testing set are shown in Fig. 6.

TABLE II  
THE PERFORMANCE OF FREE-SPACE DETECTION IN OUR RVD DATASET AND NAN'S [19]

Traffic Scene	$Precision$ [%]	$Recall$ [%]	$F_1$ [%]
Urban	98.16	97.51	97.82
Moderate Urban	98.33	97.68	97.98
Complex Illumination	97.88	99.45	98.65
HighWay	98.97	98.94	98.95
HighWay (Left Lane)	99.48	92.02	95.59
Cloudy	98.33	97.68	97.98
Rainy & Snowy Day	96.83	97.99	97.37
Night	97.68	97.70	97.66
Heavy Urban [19]	98.32	96.91	97.58
Highway [19]	99.24	98.43	98.83

**The results of lane boundary detection** is evaluated with the criteria presented in [19]. If the horizontal distance between detected boundary and ground truth labeling is smaller

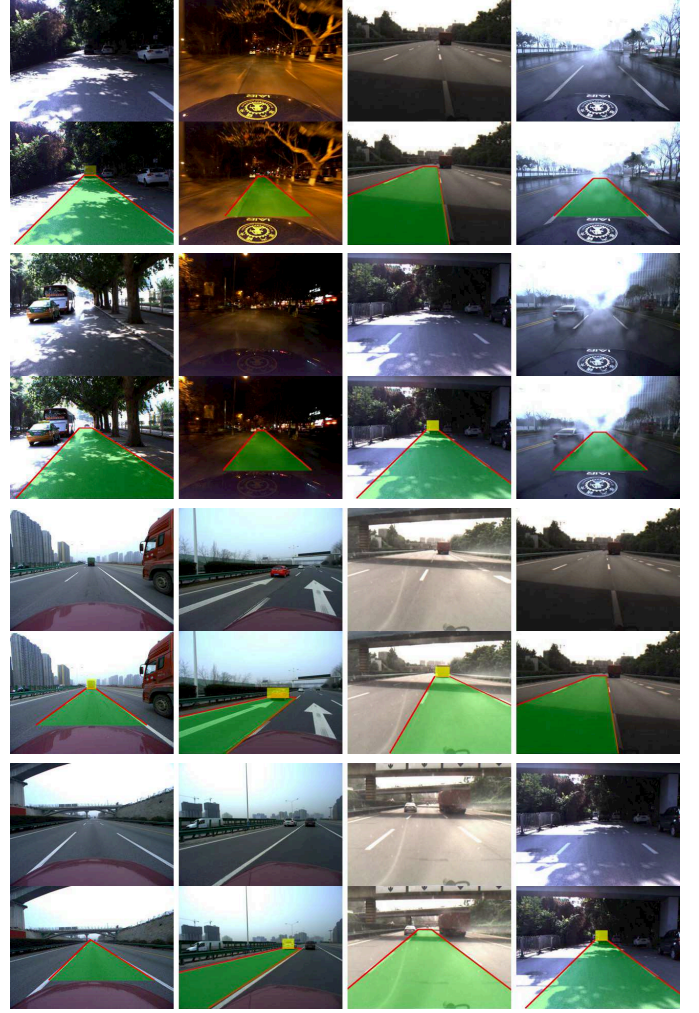


Fig. 6. Free-space detection results. The proposed model is evaluated on our RVD dataset and Nan's [19], where various traffic scenes such as night, rainy day and complex illumination are included.

than a predefined thresholds, the detected boundary will be regard as true positive. We compared our approach with the state-of-art method. The two methods were evaluated respectively in our RVD dataset and Nan's dataset [19]. The experimental results show that our approach exhibits a nearly consistent precision with the state-of-art approach in some typical traffic scenes. However, in challenging scenarios,

TABLE III  
COMPARISON WITH OTHER LANE BOUNDARY DETECTION METHODS IN  
DIFFERENT TRAFFIC SCENES

Traffic Scenes	Methods	Precision[%]	Frame Rate
Highway	Ours	99.9	93
	Nan's	99.9	28
Moderate Urban	Ours	96.2	90
	Nan's	97.7	21
Heavy Urban	Ours	96.1	90
	Nan's	95.4	16
Illumination	Ours	95.8	89
	Nan's	100.0	23
Night	Ours	90.7	94
	Nan's	99.4	35
Rainy & Snowy Day	Ours	87.1	89
	Nan's	47.3	21

such as in rainy, snowy and hazy day, our model shows better performance, indicating that it has a relatively strong robustness. Additionally, with the utilization of convolutional neural network, our model can be processed in parallel on GPU, which leads to a higher frame rate comparing with the state-of-art method.

The quantitative results of our model in lane boundary detection are presented in Table III. The lane boundary detection results of our model in different scenes and datasets are demonstrated in Fig. 7.

**The estimation of obstacle position** with our model is evaluated in perspective image pixels, since the real distance between obstacle vehicle and our car is related to extrinsic parameters of camera which may differ in different self-driving cars. At first we calculate the accuracy of the obstacle detection. The distributions of the distance errors in different scenes are shown in Fig. 8.

### B. Generating Control Command Sequences with Recurrent Neural Network

1) *Experiment Setup*: As shown in Fig. 9, the simulation environment we constructed mainly contains three parts, which are road circumstance, vehicle model and internal controller (driver model included). There are two main proposes for our simulator, one is to evaluate the performance of our method, the other is to augment the data of driver behaviors. In our simulator, the car model interacts with the road circumstance, and their status are sent to the internal controller as inputs to simulate human driver's behaviors, so as to generate control sequences to adjust the attitude of the car. These three modules constitute a complete closed-loop simulation cycle.

2) *Effects of Control Sequence Generation*: Training a recurrent neural network to learn human's driving behavior needs a large-scale driving behavior data. However, lots of unexpected factors may take effect on the data we get from the real world driving. For example, a human's driving behavior is partially based on his subjective decision, which may be absolutely different even in a same circumstance. Existences of such things in our training data may lower the confidence of

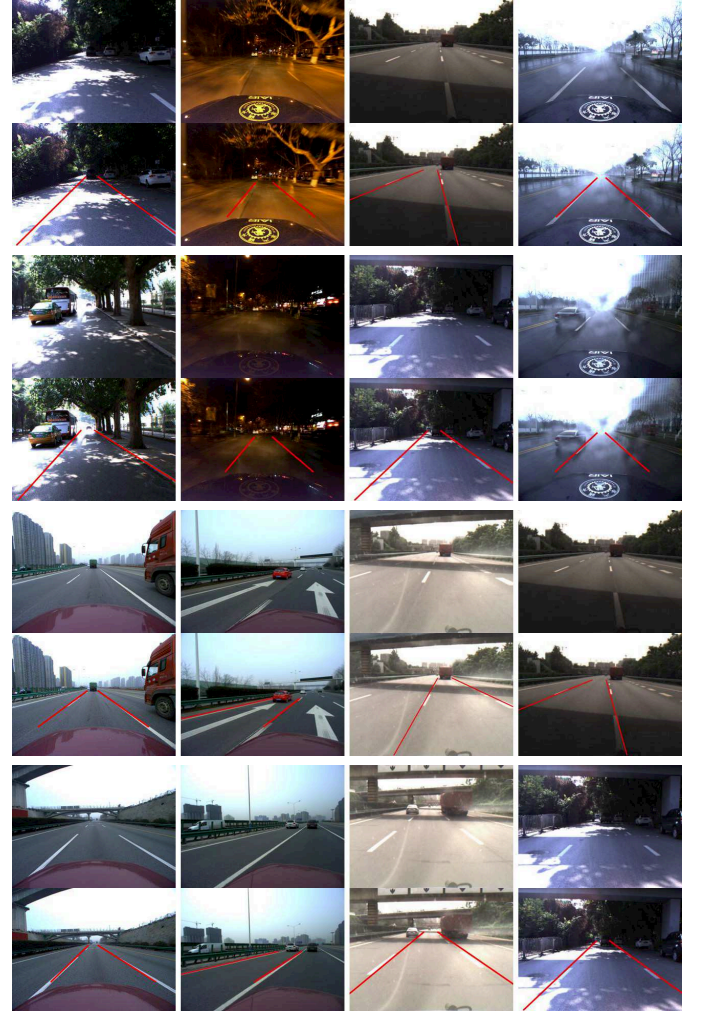


Fig. 7. Lane boundary detection results on different datasets.

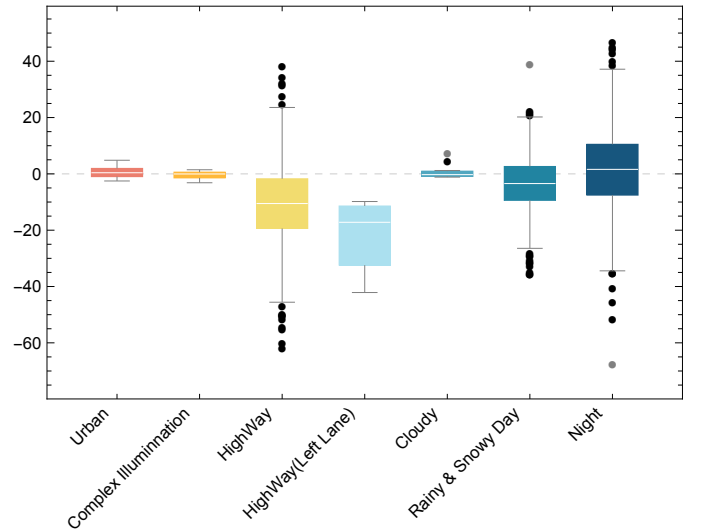


Fig. 8. Distributions of the distance errors.

our network's output. Therefore, to evaluate the planning and control part, we use the data not only from our dataset, which contains both road data we marked and actions the driver actually made, but also from the simulation environment,

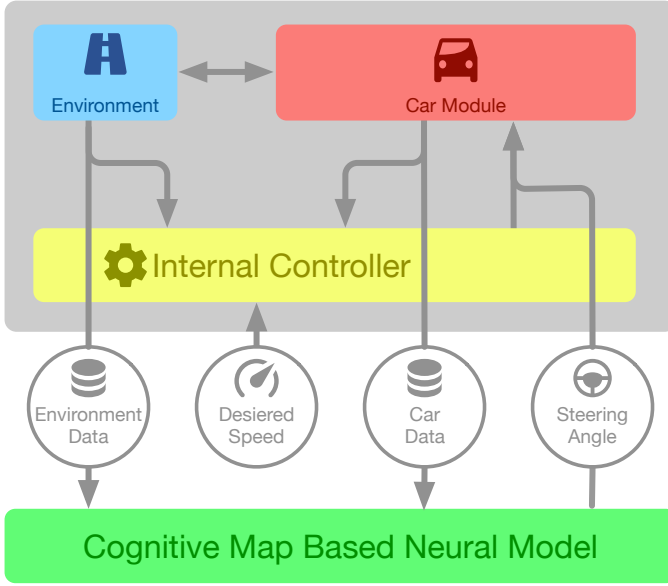


Fig. 9. The block diagram of our simulator, which is used to evaluate the performance of our CMA model in path planning and control. The simulation system mainly consists of three parts, which are internal controller, vehicle model and road circumstance.

shown in Fig. 9. By setting the parameters of both the vehicle and the road module properly, the desired driving data can be reliably generated from the simulation environment, such as the driving trajectory and the commands of steering the wheel of the car. We set multiple parameters for the road module, and run the simulator repeatedly, so as to obtain information of vehicle states (vehicle attitude, speed etc.) and deviation between the vehicle driving trace and the lane to improve and extend our driving behavior dataset.

TABLE IV  
PARAMETERS OF RECURRENT NEURAL NETWORK WITH LSTM BLOCK

Layers	Operations	Attributions
Dense	Input	Size: $[20 \times C^t]$
	Output	Size: $[20 \times 16]$
LSTM	Input	Size: $[20 \times 16]$
	Output	Size: $[20 \times 64]$
LSTM	Input	Size: $[20 \times 64]$
	Output	Size: $[20 \times 64]$
LSTM	Input	Size: $[20 \times 64]$
	Output	Size: $[64]$
Dense	Input	Size: $[64]$
	Output	Steering angle $S_a$

As shown in Table. IV, we build a LSTM network to model the temporal dependencies in driving process. Path planning and vehicle control will be accomplished simultaneously in the LSTM network. In order to quantitative analyze the efficiency of our method in path planning and control, we evaluate the proposed method in the simulator, instead of implementing it on a real vehicle. In the simulator, we set two lanes in the road, and the speed of vehicle is set but not limited to 40 Km/h. In the meanwhile, there will be two obstacles set on the lane to test if our model can achieve the control of

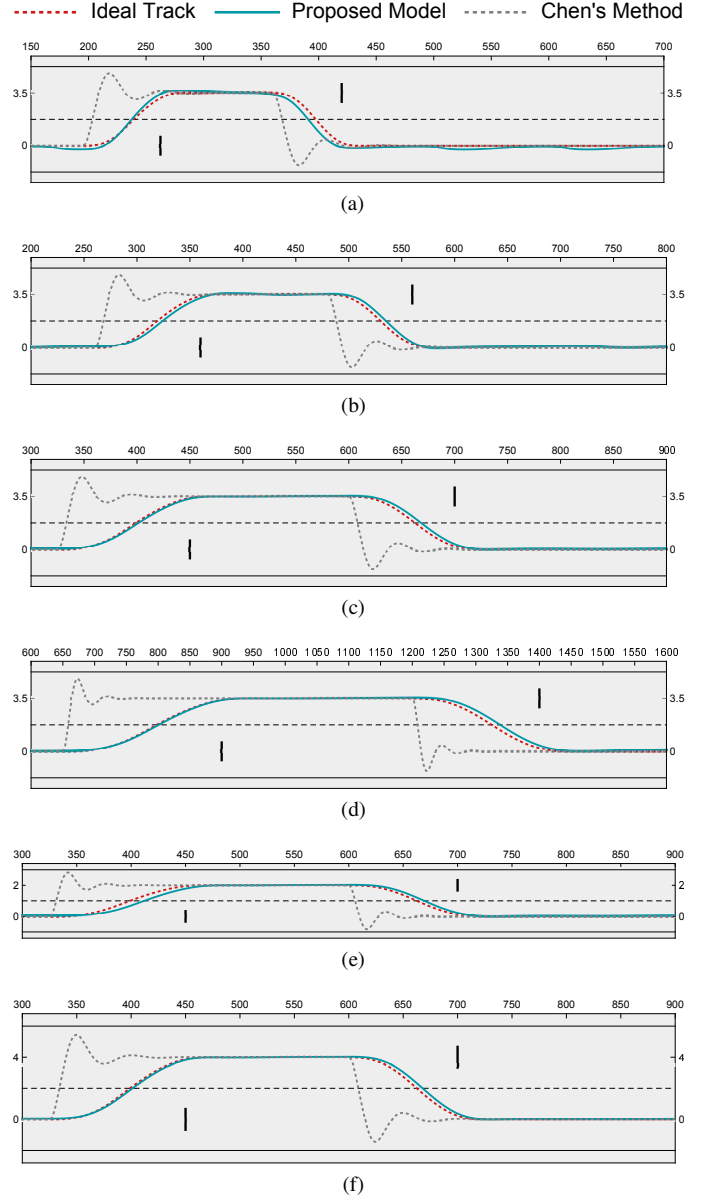


Fig. 10. Simulation of planning and control process with CMA model. Simulative vehicle trajectories on diverse lanes. From 10a to 10f, obstacles are put respectively at 50, 80, 100, 200, 100, 100 meters from the self-driving car, and occurs at random moments to test the planning and control ability of models. In 10a - 10d, the lanes are 3.5 meters in widths, and in 10e - 10f, the widths are of 4 meters.

human-level in the lane changing scenario. As shown in Fig. 9, the proposed method replaces the internal controller (driver model) in the simulator. A steering angle to control the car module in real-time can be generated from the proposed model by using information of car states and environment data.

As shown in Fig. 10, we present the driving trajectories generated by our method and Chen's method [14] in different lane changing scenarios. In testing procedure, we randomly set obstructions in front of the vehicle to test its performance on lane changing operation when faced with obstacles of different distances. As for the lane changing operation, compared with Chen's method, our approach takes account of the temporal dependence, which implies the states of the vehicle over a



period of time are memorized. Therefore, the trajectory of our model is consistent with the ideal curve, and the vehicle can drives in a steady and smooth state. The whole process is similar to the operation of human drivers, where the drivers' behaviors are absolutely learn by model.

## VI. CONCLUSION

In this paper, we proposed a cognitive model with attention for self-driving cars. This model was proposed, inspired by human brain, to simulate human visual and motor cortices for sensing, planning and control. The mechanism of attention was modeled by a recurrent neural network in time. In addition, the concept of cognitive map for traffic scene was introduced and described in detail. Furthermore, a labeled dataset named Road-Vehicle Dataset (RVD) is built for training and evaluating. The performance of the proposed model in planning and control was tested by three visual tasks. Experimental results showed that our model can fulfill some basic self-driving tasks with only cameras.

Besides attention mechanism, the permanent memory plays a crucial role in human cognition. How to incorporate the permanent memory into the proposed cognitive model is left as our future work. In addition, there are many abnormal events in the actual traffic scene. How to develop an efficient cognitive model to deal with these situations is an interesting topic for future study.

## ACKNOWLEDGMENT

This research was partially supported by the National Natural Science Foundation of China (No. 91520000, L1522023), the Programme of Introducing Talents of Discipline to University (No. B13043), 973 Program (No. 2015CB351703).

## REFERENCES

- [1] J.-r. Xue, D. Wang, S.-y. Du, D.-x. Cui, Y. Huang, and N.-n. Zheng, "A vision-centered multi-sensor fusing approach to self-localization and obstacle perception for robotic cars," *Front. Inform. Technol. Electron. Eng.*, vol. 18, no. 1, pp. 122–138, 2017.
- [2] S. Tsugawa, "Vision-based vehicles in japan: Machine vision systems and driving control systems," *IEEE Transactions on Industrial Electronics*, vol. 41, no. 4, pp. 398–405, 1994.
- [3] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "Vits-a vision system for autonomous land vehicle navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 342–361, 1988.
- [4] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman *et al.*, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.
- [5] 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.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [8] N. Zheng, Z. Liu, P. Ren, Y. Ma, S. Chen, S. Yu, J. Xue, B. Chen, and F. Wang, "Hybrid-augmented intelligence: collaboration and cognition," vol. 18, no. 2, pp. 153–179, 2017.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.
- [10] J. Hawkins and S. Blakeslee, *On intelligence*. Macmillan, 2007.
- [11] V. Mountcastle, "An organizing principle for cerebral function: the unit model and the distributed system," in *The Mindful Brain*, G. Edelman and V. Mountcastle, Eds. Cambridge, Mass.: MIT Press, 1978.
- [12] E. C. Tolman *et al.*, "Cognitive maps in rats and men," 1948.
- [13] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M.-B. Moser, "Path integration and the neural basis of the 'cognitive map'," *Nature Reviews Neuroscience*, vol. 7, no. 8, pp. 663–678, 2006.
- [14] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.
- [15] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using b-snake," *Image and Vision computing*, vol. 22, no. 4, pp. 269–280, 2004.
- [16] A. A. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*. IEEE, 2008, pp. 82–88.
- [17] Y. Li, A. Iqbal, and N. R. Gans, "Multiple lane boundary detection using a combination of low-level image features," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 1682–1687.
- [18] J. He, H. Rong, J. Gong, and W. Huang, "A lane detection method for lane departure warning system," in *Optoelectronics and Image Processing (ICOIP), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 28–31.
- [19] Z. Nan, P. Wei, L. Xu, and N. Zheng, "Efficient lane boundary detection with spatial-temporal knowledge filtering," *Sensors*, vol. 16, no. 8, p. 1276, 2016.
- [20] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue *et al.*, "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.
- [21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2016.
- [22] 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.
- [23] D. A. Pomerleau, "Alvinn, an autonomous land vehicle in a neural network," Carnegie Mellon University, Computer Science Department, Tech. Rep., 1989.
- [24] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *NIPS*, 2005, pp. 739–746.
- [25] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," *arXiv preprint arXiv:1612.01079*, 2016.
- [26] J. Koutník, G. Cuccu, J. Schmidhuber, and F. Gomez, "Evolving large-scale neural networks for vision-based torcs," 2013.
- [27] —, "Evolving large-scale neural networks for vision-based reinforcement learning," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, 2013, pp. 1061–1068.
- [28] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.
- [29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [30] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
- [31] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
- [32] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [33] Q. Li, N.-n. Zheng, and X.-t. Zhang, "Calibration of external parameters of vehicle-mounted camera with trilinear method," *Opto-electronic Engineering*, vol. 31, no. 8, p. 23, 2004.
- [34] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

- [35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: A system for large-scale machine learning,” in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. Savannah, Georgia, USA, 2016.
- [36] J. Fritsch, T. Kuhn, and A. Geiger, “A new performance measure and evaluation benchmark for road detection algorithms,” in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 1693–1700.