A method to keep autonomous vehicles steadily drive based on lane detection

Zhenyu Wu¹, Kai Qiu¹, Tingning Yuan¹ and Hongmei Chen²

Abstract
Existing studies on autonomous driving methods focus on the fusion of onboard sensor data. However, the driving behavior might be unsteady because of the uncertainties of environments. In this article, an expectation line is proposed to quantify the driving behavior motivated by the driving continuity of human drivers. Furthermore, the smooth driving could be achieved by predicting the future trajectory of the expectation line. First, a convolutional neural network-based method is applied to detect lanes in images sampled from driving video. Second, the expectation line is defined to model driving behavior of an autonomous vehicle. Finally, the long short-term memory-based method is applied to the expectation line so that the future trajectory of the vehicle could be predicted. By incorporating convolutional neural network- and long short-term memory-based methods, the autonomous vehicles could smoothly drive because of the prior information. The proposed method is evaluated using driving video data, and the experimental results demonstrate that the proposed method outperforms methods without trajectory predictions.

Keywords
Autonomous driving, decision-making, lane detection, LSTM

Date received: 12 January 2021; accepted: 25 February 2021

Introduction
The technology of autonomous driving has been attracting amount of attention of both academia and industry over the last several years. Generally, the studies on autonomous driving could be divided into three different components including perception, decision-making, and control. The perception component aims to efficiently obtain environmental information for autonomous vehicles, and the control component aims to set appropriate mechanical parameters so that the autonomous vehicles could travel forward by following the planned path. The decision-making component tries to determine a perfect path for autonomous vehicles, which could safely navigate the vehicles toward their destinations. Therefore, it is the core component of the three components because it determines all the actions of the autonomous vehicle, such as lane change, brake, and acceleration. Typically, the decision-making component in autonomous driving is realized by the decision-making models incorporated by other components, such as navigation system.

To navigate an autonomous vehicle driving to the target place from the current position, the decision-making component should have a best understanding of the

¹School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China
²College of Electrical Engineering, Henan University of Technology, Zhengzhou, Henan, China

Corresponding author:
Zhenyu Wu, School of Internet of Things, Nanjing University of Posts and Telecommunication, Nanjing 210003, China.
Email: zhenyu.wu@njupt.edu.cn
environments. For example, the Global Positioning System (GPS) is always used as a good assisted tool for the navigation of the autonomous vehicles. However, there are the following drawbacks of the GPS navigation. First, it is not very accurate for GPS navigation. Second, GPS signals are weak in lots of scenarios, such as the tunnel. Therefore, there should be some methods to make up for GPS navigations. In these methods, the onboard sensors should be used, and the local algorithms should be designed to clearly understand the driving environment so that the autonomous vehicles could make correct decisions.

The road or lane is always the key reference during the driving activities. When a human driver drives a car, he or she keeps the vehicle in a specific position of a lane. For example, the vehicle might be in the middle, and it might be on the left side of the lane. Motivated by this, this article tries to transfer this driving behavior of a human to autonomous vehicles. Therefore, the first task is lane detection which is regarded as one of the challenging tasks in the studies of autonomous driving. Lane detection could identify the lanes on the road so that the vehicle location could be obtained. For example, in the case of a tunnel, GPS signals may be lost, the lane detection tasks could help the navigation system locally perceive all kinds of environmental information.

Furthermore, when a human drives a car, he or she is always aware of a short path in front of the vehicle. This behavior is meaningful for driving activities. First, the vehicle could be well controlled when the lanes are missing. Second, the path could be planned in advance which is important for safe driving. Finally, it is useful to keep the vehicle steady. In the previous work, a center line is constructed from a set of predefined waypoints to help autonomous driving. In this article, the line might not be the center line. It should be an expectation line, which is used to quantify the behavior of drivers so that their decisions would not be affected by uncertain environments. The drivers should follow the expectation line which is a fictional line. It is shown in Figure 1. The expectation line is between left lane and right lane. Moreover, it must guarantee the vehicle would not be out of the lane. Furthermore, it is not enough for an autonomous vehicle to recognize its current environment because of the time consumption of the calculation tasks. It must also be able to predict the future states of the environment by remembering the past information.

Therefore, the autonomous driving method could be divided into the following stages in this article. First, lanes should be detected. The deep learning-based methods are applied for this task. Moreover, it is compared with traditional OpenCV-based methods. Second, an expectation line is constructed. Finally, because of recurrent neural networks (RNNs) that have been used in end-to-end scene labeling systems, the future trajectory of the vehicle is predicted by long short-term memory (LSTM) networks which is a kind of RNN. The proposed method is evaluated using autonomous driving data sets.

The remaining of this article is structured as follows: In the second section, the detailed related work about the proposed method is described. In the third section, the study problems are defined. The architecture of system is introduced in the fourth section. The fifth section is about the detailed autonomous driving method using lane detection. Experimental results are analyzed in the sixth section. The final section provides a brief summary of the article.

### Related work

#### Decision-making in autonomous driving

In autonomous driving, the vehicle should select the right route and navigate to the destination. During the process of driving, it should correctly interact with the environment, such as the traffic signals and other vehicles on the road. However, there are a large number of uncertainties over the intentions of other traffic participants in real-world driving. Therefore, decision-making component should be robustly designed so that the autonomous vehicles could handle different uncertainties.

Generally, the decision-making component of autonomous driving could be hierarchically decomposed into the following components. First, a route toward the destination is planned at the highest level. Second, the motion planning module would select a continuous path through the environment to accomplish a locally navigational task. Finally, the control system reactively corrects errors in the execution of the planned motion.

In decision-making of autonomous driving, the intentions of other traffic participants should be well predicted, such as the future trajectories of the other vehicles. The
machine learning-based methods have been proposed to solve the above problems, such as Gaussian mixture models,\textsuperscript{14} Gaussian process regression.\textsuperscript{15} Moreover, some model-based approaches are proposed to directly estimate the intentions from collection of onboard sensors.\textsuperscript{16} In addition, the probabilistic methods are always used to make decision for autonomous vehicles in complex uncertain environments. For example, Markov decision processes (MDPs) and their extended methods are always used to model the decision-making problems of autonomous driving. Brechtel et al. applied the MDP framework to study the behavioral decision-making problem.\textsuperscript{17} Furthermore, they modeled the unobserved driving scenarios and pedestrian intentions explicitly using a partially observable Markov decision process (POMDP) framework.\textsuperscript{18} In their work, specific approximate solution strategies were proposed.

With the development of onboard computation and wireless communication technology, it is possible for autonomous vehicles to exchange information with each other or interact with the road infrastructure. Recently, the studies on connected autonomous vehicles are attracting more and more attention.\textsuperscript{19} By the information shared by connected vehicles, the driving safety could be greatly improved.

**Lane detection**

In fact, lane detection is one of the fundamental problems in autonomous driving. The autonomous vehicle must reliably detect the boundaries of its current lane for accurate localization to guarantee the driving safety. However, the simple appearance of a lane is difficult to provide the distinctive features for lane detection tasks. Moreover, diverse lane patterns, such as solid, broken, splitting, and merging lanes, make independent lane modeling difficult.

Traditionally, algorithms based on highly specialized handcrafted features were used to solve the lane detection problems.\textsuperscript{20} In these algorithms, the handcrafted features include color-based features\textsuperscript{21} and the structure tensor.\textsuperscript{22} Furthermore, the features are combined with Hough transform to evaluate a binary edge image by discretizing all possible lines in the image into an accumulator matrix and counting the edge pixels falling on each line.\textsuperscript{23} There are some assumptions on lanes in existing lane detection methods, such as lanes are parallel\textsuperscript{24} and lanes are straight or close to straight.\textsuperscript{25} However, these assumptions are not always valid especially in urban situations. Recently, methods based on deep neural networks were applied to detect lanes for autonomous vehicles.\textsuperscript{26,27} Especially, the convolutional neural networks (CNNs) had achieved promising results.

There were studies about the lane detection task and the control of the autonomous vehicles. Mcfall et al. proposed an effective lane detection algorithm employing the Hough transform and inverse perspective mapping to estimate distances in real space. Moreover, the estimated distances could be utilized to send steering control commands to an autonomous vehicle.\textsuperscript{28}

**Path planning**

The path planning is also an important task for autonomous driving. Generating the future sequence of driving actions could enable the autonomous vehicles travel successfully and safely.

Montemerlo et al. developed an efficient method for autonomous vehicles to plan path.\textsuperscript{29} In their method, the lateral offset was defined as the perpendicular distance to a fixed baseline which was parallel to the driving path of an autonomous vehicle. Moreover, a cost function was optimized to identify the traveling path. The cost function considering the following two aspects. The one was the distance from the central road. The other was collision obstacle. To avoid obstacles on roads, Chu et al. proposed an approach to solve the path planning problem in autonomous vehicles.\textsuperscript{30} Phung et al. proposed an improved path planning method for autonomous vehicles using discrete particle swarm optimization.\textsuperscript{31} Although these methods could plan a smooth path for autonomous vehicles in a safe way, they could only treat static obstacles. Li et al. had applied discrete optimization to design a path planning method.\textsuperscript{9} In their method, an optimal path could be selected from a collection of possible paths. Moreover, the appropriate vehicle speed and acceleration could be simultaneously determined. Zuo et al. considered the navigation problem for mobile robot in complex environments and proposed a hierarchical path planning method.\textsuperscript{32} To construct a path to the destination avoiding collision, rapidly exploring random trees (RRTs)-based methods were widely applied. Cui et al. proposed an adaptive path planning algorithm to estimate the scalar field over a region of interest.\textsuperscript{33} Deuars et al. developed two efficient sampling-based approaches based on RRT to solve a complex path planning problem.\textsuperscript{34}

To effectively plan paths, the environment could be mapped to the cells set. In the set, each cell was used to represent an obstacle which was appeared at that position in the environment.\textsuperscript{35} Furthermore, the optimal search algorithms including A* and D* could be applied to discover a global optimized path from current position to the destination. Moreover, the obstacles on the path could be avoided. Recently, some improved cell-based methods were proposed to apply in unstructured environments, such as U-turns on a blocked road and parking.\textsuperscript{36} Recently, LSTM or RNN-based methods had also been applied to analyze the vehicle trajectory.\textsuperscript{26,27} Khosroshahi et al. applied LSTM to identify the intention of drivers based on the trajectory data.\textsuperscript{37} Kim et al. had applied LSTM to predict the location of the vehicle after several seconds using the past trajectory data.\textsuperscript{38} Moreover, other deep learning-based methods were also studied for trajectory prediction. Park et al. proposed a deep learning-based vehicle trajectory
prediction technique which could generate the future trajectory sequence of surrounding vehicles in real time.\textsuperscript{39} In addition, Gaussian distribution, dynamic Bayesian network, and fuzzy methods are also used in path planning.\textsuperscript{40–46}

In this article, a driving method for autonomous vehicles would be designed. To solve this problem, the popular lane detection and trajectory prediction method would be used. Moreover, the vehicle model should be investigated to obtain the historical trajectory data of vehicles.

**Problem definition**

In real world, a human could perfectly drive a car in a separate lane. Motivated by this, the goal of this article is to design an autonomous driving approach to keep the vehicle steadily drive, which is similar to human drivers. Specially, we assume that there are no obstacles in the same lane with the autonomous vehicle. Therefore, there are totally three problems that should be defined, including lane model, vehicle model, and trajectory prediction model. The lane model defines the structure used in this article. The vehicle model describes the attributes and dynamic behavior of the autonomous vehicle. The trajectory prediction model introduces the ways to know the future vehicle trajectory.

**Lane model**

In this article, we assume that a lane includes the right side boundary (right lane, RL) and the left side boundary (left lane, LL). Moreover, assuming that LL is the parallel shift of RL. Each side could be modeled by a polynomial model, which could fit the lane boundary edges. The polynomial model could be defined as follows

\[
y = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1
\]

where \(x\) and \(y\) are the coordinates of the polynomial curves. The values of \(A = (a_n, a_{n-1}, \ldots, a_1)^T\) could be obtained by providing a set of \(n\) equations formed by edge pixels

\[
\begin{bmatrix}
x_1^n & \cdots & 1 \\
\vdots & \vdots & \vdots \\
x_n^n & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
a_n \\
\vdots \\
a_1
\end{bmatrix}
= 
\begin{bmatrix}
y_1 \\
\vdots \\
y_n
\end{bmatrix}
\]

The expectation line (EL) is used to measure the appreciate path of the autonomous vehicle. It could be defined as follows

\[
EL = LL_x + |RL_x - LL_x|/N
\]

The integer number \(N\) means that the whole lane would be divided into two parts, the one is \(1/N\) of the whole lane width, and the other is \((N - 1)/N\) of the lane width. When \(N\) equals to 2, the expectation line is the central line of the lane. To prevent the autonomous vehicle out of the boundary of the lane, the following constrains should be defined as

\[
EL > LL_x + w/2 \quad \text{and} \quad EL < RL_x - w/2
\]

where \(w\) is the width of the vehicle.

**Vehicle model**

In this article, the autonomous vehicle model is described in the two-dimensional plane, which is shown in Figure 2. A vehicle \(V\) could be described as follows

\[
VH = (P, V, \Theta)
\]

where \(P = (x, y)\) represents the coordinates of the location of centroid of the vehicle and \(V = (V_x, V_y)\) is the velocity of the vehicle. \(V_x\) and \(V_y\), respectively, refer to the longitudinal velocity and the lateral velocity at the center of mass and \(\Theta\) is the yaw angle, which indicates the orientation of the vehicle. The yaw angle is identified as the angle between the vector of the longitudinal velocity and the tangent to the expectation line.

Given discrete time \(\Delta t\), the autonomous vehicle dynamics could be described as follows

\[
x(t+1) = x(t) + v_y(t) \Delta t \\
y(t+1) = y(t) + v_y(t) \Delta t \\
v(t+1) = v(t) + a(t) \Delta t
\]

where \(a(t)\) is the acceleration of the autonomous vehicle. That is, the autonomous vehicle state at \(t+1\) is determined by the current state and future control input.

**Trajectory prediction model**

In fact, the conditions of the lane are uncertain. Therefore, the prediction of the future trajectories is meaningful for
the safe driving. Generally, the trajectory would not dramatically and frequently change for the comfortable driving. Therefore, the historical trajectory data could be used to predict the future status of the trajectory. Assuming that the historical trajectory data are 

\[ \text{HTD} = \{p_1, p_2, \ldots, p_n\} \]

at time \( T \), where \( p_i (1 < i < n) \) is the passed position of the vehicle. The future positions \( \Delta d \) ahead of the vehicle should be estimated by the trajectory prediction model. The set of future positions \( \Delta \text{HTD} \) could be defined as follows

\[ \Delta \text{HTD} = f(\text{HTD}, \Delta d) \quad (7) \]

That is, a function \( f \) should be designed so that it could learn the patterns from the historical data. Moreover, the position data in future \( \Delta d \) could be predicted. Generally, \( f \) could be implemented using machine learning-based methods.

**System overview**

The proposed method could be used in autonomous driving systems from perception, decision-making to control. The system overview of the proposed autonomous driving method using lane detection is shown in Figure 3.

The whole system includes three layers including perception layer, decision-making layer, and control layer. In perception layer, the lane data are collected by the sensors, such as the camera, which is located on the autonomous vehicle. Generally, it is image or video data. In decision-making layer, the lanes in each captured image would be detected using methods such as CNN. Moreover, the historical trajectory data could be collected. That is, the trajectory could be constructed at a specific time. Furthermore, the LSTM network could be used to predict the future trajectory of the vehicle. In control layer, a steering command would be generated according to LSTM network, and the command could be sent to the control system of the vehicle.

By deploying this system on the autonomous vehicles, the driving behavior of human who drives a car in lane could be modeled.

**Autonomous driving method using lane detection**

In this article, the proposed method includes two steps. The one is lane detection, and the other is trajectory prediction. The structure of the method is shown in Figure 4.

**Lane detection using CNN**

Although traditional OpenCV methods were always used to detect lanes from images, the CNN-based methods had achieved better performance for lane detection task. Assuming that the training data \( D = \{f_1, f_2, \ldots, f_n\} \), where \( f_i (1 < i < n) \) is the continuous frame obtained from the driving video, and each frame is a separate image. The CNN is applied to detect the lane of the current autonomous vehicle.

The detailed steps of the lane detection method are shown in Algorithm 1. The training data \( D \) should be preprocessed, and the lanes in the data set are labeled. The input of the algorithm is \( D \), and output of the algorithm is the detected lane in each image.

**Prediction trajectory using LSTM**

By the vehicle model, the observation of autonomous vehicle at current time \( t \) could be obtained \( O_t = (x_t, y_t, v_{xt}, v_{yt}, \Theta_t) \). Moreover, the lane \( L_t = (LL_t, RL_t) \) has been detected using...
CNN method. The future trajectory of the vehicle would be predicted using LSTM.

First, the expectation line \( EL_t \) should be constructed using \( L_t \). That is, the vehicle would go ahead tracking the expectation line \( EL_t \). In LSTM component, the fully connected (FC) layer is followed by rectified linear unit (ReLU) activation function. The FC layer could transform the five-dimensional input data \( O_t \) into the same dimensional feature needed by LSTM cell. Moreover, it extends the network capacity so that complex structure of the trajectory data could be handled. The output of the FC layer is fed into the LSTM stack which is constructed with 256-dimensional cell memory. The output of LSTM is sent to another FC layers which is followed by a Softmax layer. The output of the FC layers is passed through the Softmax function, and the future trajectory would be predicted.

To train the LSTM trajectory prediction model, the loss function could be defined as follows

\[
L(W) = \sum_{i=1}^{N} \left( \frac{1}{2} \left( |x_{i} + \Delta t - z_{t,x}^\prime|^2 + |y_{i} + \Delta t - z_{t,y}^\prime|^2 + \lambda \mathcal{O}(w) \right) \right)
\]

(8)

where \( w \) is the parameter of the LSTM network, \( N \) is the total number of the training samples, \( z_{t,x}^\prime \) and \( z_{t,y}^\prime \) are the predicted values of LSTM for a specific training sample, and \( \mathcal{O}(w) \) is the regularization term with the parameter \( \lambda \).

Assuming that the trajectory observation set at time \( T \) is \( O(T) = O_1, O_2, \ldots, O_T \), where \( O_i (1 < i < T) \) is a single observation at time \( i \). The detailed expectation line trajectory prediction steps are shown in Algorithm 2.

**Experiments and analysis**

**Experiment setup**

There are two important deep learning-based methods used in this article. The one is CNN and the other is LSTM. Both methods need plenty of computation resources for the training period. Therefore, the experimental environment with GPU configurations is needed.

In this article, the Colab platform provided by Google is used for GPU resources. Moreover, the TensorFlow framework is used as the deep learning platform. Moreover, the OpenCV development environment is configured so that the experimental results obtained by deep learning methods could be compared. Furthermore, the advantages of the deep learning-based methods could be demonstrated.

**Data set**

The data set includes the multi-frame images sampled from the driving video. The resolution of the video is 720 pixels where the vertical axis is 720 pixels and the horizontal axis is 1280 pixels. The frame rate is 30 fps.

The images are analyzed. However, the quality of part of images is poor because of lighting and brightness. This also illustrates the importance of lane prediction. The sample rate of the training images is very important. If the rate is higher, the continuous images might be very similar, which is useless for the model. Therefore, only 1 out of 10 images is selected into the training set.

The binomial curve \( y = ax^2 + bx + c \), where \( a \), \( b \), and \( c \) are coefficients of the curve, is used to fit the vehicle lane. Therefore, the number of training images with different coefficient labels could be counted. The label of the second coefficient \( b \) is studied, and the results are shown in Figure 5, where the horizontal axis stands for the coefficient labels and the vertical axis stands for the image number. It is obvious that most of the images are with the coefficient label “0.” This shows that lanes are lacking of diversity.

To improve the recognition performance of the lane detection model, the training data set should be extended. That is, more images with the curve lane should be added into the training set. First, more extra images with curve lane are collected from video. Second, the images with less number are collected. Moreover, these images are processed to generate some new images. For example, by rotating the images, the images with curve lane would be extended. The extended image numbers with different

---

**Algorithm 1.** Lane detection with CNN.

**Input:** Data \( D = \{f_1, f_2, \ldots, f_n\} \)

**Output:** Lane \( L_t = (L_{Lt}, R_{Lt}) \)

**Begin**

- Initialize parameters of CNN \( C \);
- Divide data \( D \) into training data \( TRD \) and test data \( TED \);
- For \( f \) in \( TRD \) do:
  - Train \( C \) with \( f \);
  - EndFor
- For \( f \) in \( TED \) do:
  - Detect Lane \( (LL, RL) \) using \( C \);
  - EndFor

**End**

**Algorithm 2.** Trajectory prediction with LSTM.

**Input:** Observation \( O(T) \), lane \( L_t = (LL_t, RL_t) \), \( \Delta d \)

**Output:** \( O(T + \Delta d) \)

**Begin**

- Construct expectation line using \( L_t \);
- Initialize the parameters of LSTM network \( LSTM \);
- For \( o \) in \( O(T) \) do:
  - Train \( LSTM \) with \( o \);
  - EndFor
- Predict the trajectory in future \( \Delta d \);
- Return \( O(T + \Delta d) \);

**End**
coefficient labels are shown in Figure 6. The proportion of curve lanes has been significantly improved.

**Experimental results**

Traditionally, the computer vision methods could be used to detect lanes. The steps are as follows: (1) camera calibration and image correction, (2) creating a binarized threshold image to obviously distinguish lanes from the surrounding environment, (3) perspective transformation, and (4) lane detection. In addition to CNN method, the OpenCV method is also implemented to demonstrate the effectiveness of CNN method.

The first evaluation metric is lane features, which would be used to fit the lane model. Obviously, the incorrect lane features would cause the wrong detection of the lane. There are labels for lanes in training data set. The lane features could be determined by computing the absolute difference between the detected lane points and the labeled lane points. Assuming that $P_{GT}(x, y)$ is the coordinate of the lane label value in the image, and $P_{LE}(x, y)$ is the coordinate of the lane feature obtained by the detection algorithms. For a given threshold $\varepsilon$, the true positive (TP) value could be defined as follows

$$\|P_{GT} - P_{LE}\| < \varepsilon$$

(9)

Moreover, the accurate metric to evaluate the lane detection algorithms could be defined as follows

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

(10)

where the $TP$ is true positive value, $TN$ is the true negative value, $FP$ is the false positive value, and $FN$ is the false negative value.

The experimental results are shown in Figure 7, where the horizontal axis stands for the video time (in seconds) and the vertical axis stands for the accuracy value at that time. From 10 s to 20 s, the lane is straight. The others are curved lane. At about 30 s, there are missing part of the lane. The experimental results give the following conclusions. First, the performance of OpenCV-based method and CNN methods is similar in the case of good driving environment. Second, the deep learning-based method outperforms the computer vision-based method in a challenging driving environment, especially when the lane is missing.

The second evaluation metric is the lane position deviation (LPD). It defines the deviation degree of the detected lane from the actual lane. In Figure 8, the virtual line stands for one side of the lane detected by algorithms, and the solid line stands for the actual side of the lane. The average deviation in the $x$-axis for values $y_{\text{max}}$ and $y_{\text{min}}$ between the detected lane and the actual lane could be used to represent the deviation $\delta_{LPD}$, which could be defined as follows

$$\delta_{LPD} = \frac{1}{y_{\text{max}} - y_{\text{min}}} \sum_{y_{\text{min}}}^{y_{\text{max}}} |\delta_i|$$

(11)
where \( y_{\text{min}} < y < y_{\text{max}} \) is the region used to measure the lane deviation.

Similarly, the two lane detection algorithms are evaluated using LPD metric. The experimental results are shown in Figure 9, where the horizontal axis stands for video time (in seconds) and the vertical axis stands for the LPD values. In this experiment, only the first frame in each second is studied. Moreover, only the LPD values of the left side of the lane are calculated. In each image of the selected frame, \( y_{\text{max}} \) is the top terminal point of the left side of the lane, and \( y_{\text{min}} \) is the bottom terminal point of the left side of the lane. Their average values are the pix numbers.

In this figure, it is obvious that the LPD values of CNN methods are lower than those of OpenCV-based methods at most of the time. The results show that the deviation values of CNN method for detecting the left side of the lane are lower than the values of OpenCV-based method. This demonstrates the effectiveness of the deep learning-based methods.

The third metric is to evaluate the computation efficiency and detection accuracy. This metric \( Q_{\text{CEA}} \) could be defined as follows

\[
Q_{\text{CEA}} = \frac{\text{ACC}}{\text{FPS}}
\]  

(12)

where \( \text{ACC} \) is the accuracy value mentioned above and \( \text{FPS} \) is the number of the frames recognized in each second. Evenly, there are about 4.5 frames recognized by OpenCV method, and there are about 30 frames recognized by CNN method. The \( Q_{\text{CEA}} \) values are shown in Figure 10, where the horizontal axis stands for video time (in seconds) and the vertical axis stands for the \( Q_{\text{CEA}} \) values at each time step. The results show that the CNN method outperforms the OpenCV method considering both accuracy and efficiency. Moreover, the CNN method has a higher recognition efficiency in the case of the same accuracy value with OpenCV method.

From the above experiments, the following conclusions could be obtained. The deep learning-based method outperforms the traditionally OpenCV-based method. Especially, when the driving environment is challenging because of the brightness and lightness.

In general, the types of the roads could be divided into the straight road and the curved road. Therefore, the expectation line should be straight or curved. In this article, these two lines are extracted from the detected lanes. Moreover, LSTM is used to predict the future trajectories of the lines.

The expectation line is the middle line of the detected lane, and it is constructed by 100 data points. In both the straight case and the curved case, the first 90 data points are used as training data set, and the last 10 data points are used as test data set. The splits of the data points are shown in Figures 12(a) and 13(a), where the horizontal axis stands
for the successive time steps and the vertical axis stands for the data point position.

The detected lane with expectation line is shown in Figure 11. In this figure, the expectation line is the middle of the two detected lanes. The structure of LSTM is as follows: an LSTM layer with 50 hidden states, a dropout layer, and a dense layer. Moreover, the “ReLU” activation function is applied. The predicted results are respectively shown in Figures 12(b) and 13(b). The r^2 score is used to evaluate the prediction result, and the values are respectively 0.67 and 0.71 in this experiment. The definition of r^2 score is as follows

\[ R^2(y, y') = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - y'_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1} (y_i - \bar{y})^2} \]  

where \( n_{\text{samples}} \) is the number of the samples, \( y_i \) is the \( i \)th sample, \( y'_i \) is the \( i \)th prediction result, and \( \bar{y} \) is the mean of \( y \).

The deviation angle which is the angle deviating from the center line is used to evaluate the effectiveness of LSTM. Total five rounds of experiments are carried out. In each round, the average deviation angle is calculated. The results are shown in Figure 14, where the horizontal axis stands for five rounds and the vertical axis stands for the values of average deviation angle. Obviously, the average deviation angle values in the case of LSTM prediction are lower than those of without LSTM prediction. This shows that there are much more deviation in uncertain environments, such as lane missing and calculation time-consuming. The results demonstrate the effectiveness of LSTM prediction during the process of driving.

**Conclusions**

The decision-making component is the core part of autonomous driving. Several studies have worked on the
decision-making methods, such as the data fusion methods based on the perception data by onboard sensors. Recently, imitation learning is also one of the important research directions for autonomous driving.

In this article, the idea is motivated by the human drives a car in a lane. Both the two sides of the lane and his or her expected driving directions are important for the safe driving. This experience could be transferred to autonomous vehicle for its robust driving. First, the CNN method is applied to detect lanes in video frames captured by camera. Second, the expectation line is defined to direct the driving direction. Finally, LSTM-based method is used to predict the future trajectory of the expectation line to handle the challenges including the lane missing. The experimental results have demonstrated the effectiveness of the proposed method.

In the future, the dynamic model of autonomous vehicle would be further investigated. Moreover, the driving environment is uncertain in real traffic. Therefore, the proposed method would consider much more environment uncertainties in the future work. For example, the expectation line prediction method should be improved in the case of obstacles.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by National Natural Science Foundation of China (No. 61502246), Nanjing University of Posts and Telecommunications Science Foundation (Nos NY215019 and NY220202), and the National Key Research and Development Program of China under grant no. 2017YFB0102603, grants from NVIDIA and the NVIDIA DGX-Station/DRIVE PX 2 AutoChauffeur.

ORCID iDs
Zhenyu Wu https://orcid.org/0000-0003-3628-1142
Hongmei Chen https://orcid.org/0000-0002-4059-4056

References
1. Zhang H and Wang J. Active steering actuator fault detection for an automatically-steered electric ground vehicle. IEEE Trans Veh Technol 2017; 66: 3685–3702.
2. Zhang H, Zhang G, and Wang J. Observer design for LPV systems with uncertain measurements on scheduling variables: application to an electric ground vehicle. IEEE/ASME Trans Mechatron 2016; 21(3): 1659–1670.
3. Chen C, Seff A, Kornhauser A, et al. DeepDriving: learning affordance for direct perception in autonomous driving. In: IEEE international conference on computer vision, Santiago, Chile, 7–13 December 2015, pp. 2722–2730. Piscataway, NJ: IEEE.
4. Hu C, Jing H, Wang R, et al. Robust H∞-H∞; mathContainer loading MathJax, output-feedback control for path following of autonomous ground vehicles. Mech Syst Signal Process 2016; 70: 414–427.
5. Zhang S, Deng W, Zhao Q, et al. Dynamic trajectory planning for vehicle autonomous driving. In: IEEE international conference on systems, man, and cybernetics, Manchester, UK, 13–16 October 2013, pp. 4161–4166. IEEE.
6. Lim W, Lee S, Sunwoo M, et al. Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. IEEE Trans Intell Transp Syst 2018; 19(2): 1–14.
7. Milanes V, Naranjo JE, Gonzalez C, et al. Autonomous vehicle based in cooperative GPS and inertial systems. Robotica 2008; 26(05): 627–633.
8. Wang Z, Ren W, and Qiu Q. LaneNet: real-time lane detection networks for autonomous driving, 2018, arXiv:1807.01726.
9. Li Q, Chen L, Li M, et al. A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. IEEE Trans Veh Technol 2014; 63(2): 540–555.
10. Hu X, Chen L, Tang B, et al. Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. Mech Syst Signal Process 2018; 100: 482–500.
11. Sallab A, Abdou M, Perot E, et al. Deep reinforcement learning framework for autonomous driving. J Electron Imaging 2017; 19: 70–76.
12. Pinheiro PHO and Collobert R. Recurrent convolutional neural networks for scene parsing, 2013, arXiv:1306.2795.
13. Paden B, Cap M, Yong SZ, et al. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Trans Intell Veh 2013; 1(1): 33–55.
14. Havlak F and Campbell M. Probabilistic anticipation for autonomous robots in urban environments. Proc SPIE Int Soc Opt Eng 2013; 30(2): 461–474.
15. Tran Q and Firl J. Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In: IEEE intelligent vehicles symposium (IV), Gold Coast, QLD, Australia, 23–26 June 2013, pp. 334–339. IEEE.
16. Verma R and Vecchio DD. Semiautonomous multivehicle safety. IEEE Robot Autom Mag 2011; 18(3): 44–54.
17. Brechtel S, Gindele T, and Dillmann R. Probabilistic MDP-behavior planning for cars. In: IEEE conference on intelligent transportation systems, Washington, DC, USA, 5–7 October 2011. IEEE.
18. Brechtel S, Gindele T, and Rudiger D. Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs. In: IEEE conference on intelligent transportation systems, Qingdao, China, 8–11 October 2014, pp. 392–399. IEEE.
19. Spalanzani A, Riosmartinez J, Laugier C, et al. Handbook of intelligent vehicles. London: Springer, 2012.
20. Wu PC, Chang CY, and Lin CH. Lane-mark extraction for automobiles under complex conditions. Pattern Recognit 2014; 47(8): 2756–2767.
21. Chiu KY and Lin SF. Lane detection using color-based segmentation. In: *IEEE intelligent vehicles symposium*, Las Vegas, NV, USA, 6–8 June 2005, pp. 706–711. IEEE.
22. Loose H, Franke U, and Stiller C. Kalman particle filter for lane recognition on rural roads. In: *IEEE intelligent vehicles symposium*, Xi’an, China, 3–5 June 2009, pp. 60–65. IEEE.
23. Tan S and Mae J. Real-time lane recognition and position estimation for small vehicle. *Internetworking Indonesia J* 2013; 5(2): 3–8.
24. Yan J, Feng G, and Xu G. Computer vision-based multiple-lane detection on straight road and in a curve. In: *International conference on image analysis signal processing*, Zhejiang, China, 9–11 April 2010, pp. 114–117. IEEE.
25. Jianwei N, Jie L, Mingliang X, et al. Robust lane detection using two-stage feature extraction with curve fitting. *Pattern Recognit* 2016; 59(C): 225–233.
26. Li J, Mei X, and Prokhorov D. Deep neural network for structural prediction and lane detection in traffic scene. *IEEE Trans Neural Networks Learn Syst* 2016; 28(3): 690–703.
27. Huval B, Wang T, Tandon S, et al. An empirical evaluation of deep learning on highway driving, 2015, arXiv:1504.01716.
28. Mcfall K. Using visual lane detection to control steering in a self-driving vehicle. In: *EAI international conference on social innovation and community aspects of smart cities*, Toronto, Canada, 13–16 October 2015.
29. Montemerlo M, Becker J, Bhat S, et al. Junior: the Stanford entry in the Urban Challenge. *J Field Rob* 2008; 25(9): 569–597.
30. Chu K, Lee M, and Sunwoo M. Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE Trans Intell Transp Syst* 2012; 13(4): 1599–1616.
31. Phung MD, Quach CH, Dinh TH, et al. Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection. *Autom Constr* 2017; 81: 25–33.
32. Zuo L, Guo Q, Xu X, et al. A hierarchical path planning approach based on A* and least-squares policy iteration for mobile robot. *Neurocomputing* 2015; 170: 257–266.
33. Cui R, Li Y, and Yan W. Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT*. *IEEE Trans Syst Man Cybern* 2015; 46(7): 1–12.
34. Devaurs D, Thierry S, and Juan C. Optimal path planning in complex cost spaces with sampling-based algorithms. *IEEE Trans Autom Sci Eng* 2016; 13(2): 415–424.
35. Xu W, Pan J, Wei J, et al. Motion panning under uncertainty for on-road autonomous driving. In: *IEEE international conference on robotics and automation*, Hong Kong, China, 31 May–7 June 2014, pp. 2507–2512. IEEE.
36. Mihail P. Differentially constrained mobile robot motion panning in state lattices. *J Field Robot* 2010; 26(3): 308–333.
37. Khosroshahi A, Ohn-Bar E, and Trivedi MM. Surround vehicles trajectory analysis with recurrent neural networks. In: *IEEE international conference on intelligent transportation systems*, Rio de Janeiro, Brazil, 1–4 November 2016, pp. 2267–2272. IEEE.
38. Kim BD, Kang CM, Lee SH, et al. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network, 2017, arXiv:1704.07049.
39. Park SH, Kim BD, Kang CM, et al. Sequence to sequence prediction of vehicle trajectory via LSTM encoder decoder architecture. In: *IEEE intelligent vehicles symposium (IV)*, Changshu, China, 26–30 June 2018. IEEE.
40. Gao H, Zhu J, Zhang T, et al. Situational assessment for intelligent vehicles based on stochastic model and Gaussian distributions in typical traffic scenarios. *IEEE Trans Syst Man Cybern Syst* 2020; 99: 1–11.
41. Li DY and Gao H. A hardware platform framework for an intelligent vehicle based on a driving brain. *Engineering* 2018; 4: 464–470.
42. Xie G, Gao H, Zhang T, et al. Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models. *IEEE Trans Ind Electron* 2018; 65(7): 6599–6608.
43. Gao H, Bi W, Wu X, et al. Adaptive fuzzy region-based control of Euler-Lagrange systems with kinematically singular configurations. *IEEE Trans Fuzzy Syst* 2020. DOI: 10.1109/TFUZZ.2020.2994991.
44. Gao H, Luo J, Pi M, et al. EEG-based volitional control of prosthetic legs for walking in different terrains. *IEEE Trans Autom Sci Eng* 2019. DOI: 10.1109/TASE.2019.2956110.
45. Gao H, Hang S, Cai Y, et al. Trajectory prediction of cyclist based on dynamic Bayesian network and long short-term memory model at unsignalized intersections. *Science China Inform Sci* 2021. DOI: 10.1007/s11432-020-3071-8.
46. Gao H, Cheng B, Wang J, et al. Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment. *IEEE Trans Ind Inform* 2018; 99: 1–10.