ICIC Express Letters
Part B: Applications
Volume **10**, Number **8**, August **2019**

 ISSN 2185-2766
pp. 717–724

DRIVERS' LANE CHANGE RECOGNITION BASED ON HOPFIELD NEURAL NETWORK

Chen Zhao[1,2] and Jiantao Liu[3]

[1]School of Automobile
Chang'an University
Middle-Section of Nan'er Huan Road, Xi'an 710064, P. R. China
273589819@qq.com

[2]China Academy of Safety Science and Technology
Safety Building, No. 32, Beiyuan Road, Chaoyang District, Beijing 100012, P. R. China

[3]Yunnan Traffic Consulting Co., Ltd.
No. 1, Huancheng West Road, Kunming 650499, P. R. China

Received January 2019; accepted April 2019

Abstract. *To address the issue of high false positive rate resulted from ADAS's inability to effectively identify the lane change behavior, lateral velocity, steering wheel angle and yaw rate among other parameters during lane change were acquired through a drive test conducted on an expressway. The collected data were processed by unscented Kalman filtering, normalization and K means clustering for establishing a lane change recognition model based on Hopfield neural network. To further improve the recognition accuracy of the model, the PSO (Particle Swarm Optimization) algorithm is used to optimize the neural network, and the recognition accuracy of the optimized model has reached 91%, which is in compliance with the requirements of ADAS system.*
Keyword: Lane change, Kalman filtering, Neural network, PSO algorithm

1. **Foreword.** Lane change maneuver is one of the most important driving behaviors. Unreasonable lane changes can cause serious traffic conflict and consequent traffic delays [1]. According to previous research, more than 90% of traffic accidents are caused by human factors, and lane changes are the main factor. Research data indicate that the traffic accidents resulted from human factors account for more than 70% of all expressway traffic accidents resulted from lane change [2].

Scholars in China and abroad have conducted a great deal of research on lane change behaviors. Bocklisch et al. used adaptive fuzzy pattern classification to detect the lane change intention in car driving [3]. Woo et al. proposed a new detection method to predict a vehicle's trajectory and used it for detecting lane changes of surrounding vehicles [4]. Kuge et al. built a lane change recognition model based on hidden Markov theory, which can be used to recognize regular lane change, dangerous lane change and lane retaining behaviors according to steering behavioral parameters [5]. Mccall et al. built a sparse Bayesian learning-based lane change intention recognition system with the inputs of transverse vehicular deviation and driver's head posture [6]. Liu and Pentland developed a hidden Markov theory-based lane change intention model with the inputs of transverse distance, steering wheel turn angle and yaw angular velocity [7]. Urun et al. developed a neural network and SVM to predict the driver's behavior according to such parameters as transverse acceleration, time of collision, steering wheel turn angle and distance between lane lines [8]. Yuan et al. built a lane change intention recognition model based on visual characteristics of the driver [9]. Ma et al. developed a BP neural network-based lane change prediction model by using head posture, vehicle motion parameters as well as

DOI: 10.24507/icicelb.10.08.717

spacing of vehicles [10]. Tang et al. built a lane changing predictor based on adaptive fuzzy neural network to predict steering angles [1].

Compared with the above several types of recognition models, neural network has the advantages of good applicability and fault tolerance, even if an error sample is encountered during training, it can be corrected as the training process progresses. Hopfield neural network is an all connection type neural network, which has a leaning capability and learning speed stronger and higher than traditional neural networks.

2. Building of Lane Change Recognition Model.

2.1. Hopfield neural network.

Hopfield neural network was first proposed by Hopfield in 1982 [11]. It runs in kinetic mode. Beginning from its initial state, the neural network evolves gradually in the energy decrementing direction gradually until the neural network reaches a steady state, and then the neural network outputs values in the steady state. In this paper, the discrete Hopfield neural network is used. The output of the discrete Hopfield neural network is only 1 and -1, respectively representing the neural network in active and suppressed states. Structure of discrete Hopfield Neural Network is as shown in Figure 1.

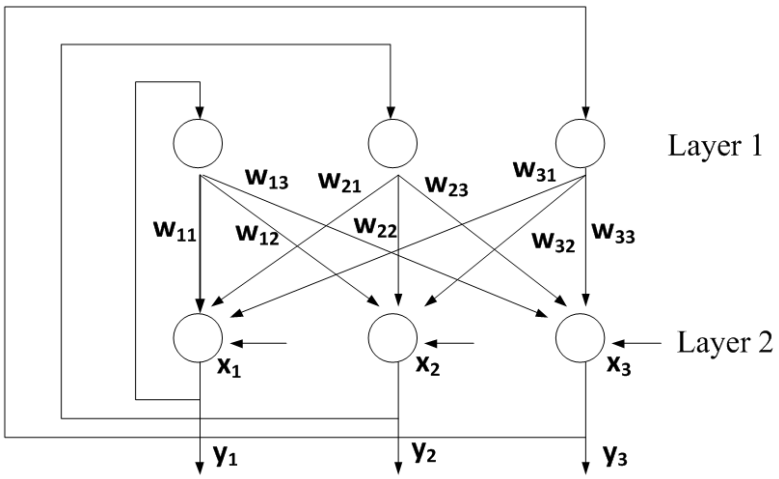

FIGURE 1. Structure of Hopfield neural network

In Figure 1, Layer 1 is used as the input layer of the neural network, only used for value transmission but not for calculation. Layer 2 includes neurone, used for processing inputted information and outputting calculated value after processed by nonlinear function f. Computing formula of Layer 2 is as follows:

$$p_j = \sum w_{ij} y_i + x_j \tag{1}$$

where x_j stands for external input. Computation formula of y_j is given below:

$$\begin{cases} y_j = 1, & u_j \geq a_j \\ y_j = -1, & u_j < a_j \end{cases} \tag{2}$$

Selecting $y_j(t)$ to indicate the status value of the jth nuerone at the moment t, then the status of the moment next to this point can be figured out by using Formulas (3) and (4):

$$y_j(t+1) = f[u_j(t)] = \begin{cases} 1, & u_j(t) \geq 0 \\ -1, & u_j(t) < 0 \end{cases} \tag{3}$$

$$u_j(t) = \sum_{i=1}^{m} w_{ij} y_j(t) + x_j - a_j \tag{4}$$

In this paper, the whole network evolves in the asynchronous working mode of Hopfield neural network. During evolution, the neural network first initializes the entire network, and then selects a neurone $u_j(t)$ on a random basis for computing the output value $v_j(t+1)$ of that neurone, at which point the state of all other neural units in the neural network remains unchanged. And then one has to make a judgment as to whether its state no longer changes beginning from a certain moment. If the conditions are satisfied, the computation should be ended and the state of the neural network at this moment shall be outputted. If any of the conditions is not satisfied, then a neural unit shall be selected and computed again.

2.2. **Acquisition of input parameters.** This paper chooses a section of expressway (G65) near Xi'an City for the test. The expressway is a two-way four-lane road with a central separation zone. The traffic volume of this section is about 1500-2000 vehicles/hour, and the frequency of lane change is higher. In this paper, 15 drivers were selected to complete the real-time test on the highway. The average age of the driver was 41 and the average driving age was 8 years.

The parameters related to the lane change acquired during lane change include: driver's eye moving parameter, heat poster parameter, steering wheel turning parameter, automobile body transverse motion parameter and longitudinal motion parameter, etc. In this paper, transverse velocity, steering wheel turn angle and yaw angular velocity were selected as the input parameters for the neural network model.

2.3. **Filtering of input parameters.** Traditional Kalman filters are linear filters, which can easily introduce linear error in the face of complicated non-linear issues. In this paper, unscented Kalman filter [12] is used for denoising and filtering treatment of the transverse velocity, steering wheel turn angle and yaw angular velocity acquired. Unscented Kalman filtering gives up linear treatment of the non-linear functions, and adopts unscented transformation to solve the issue of covariance non-linear transmission. MATLAB Kalman filtering toolkit is used for data filtering and smoothing. The results are shown in Figures 2, 3, and 4.

It can be known from Figures 2, 3 and 4 that after the unscented Kalman filtering, noises in the data are removed to the maximum extent while the parameter motion tendency is perfectly retained.

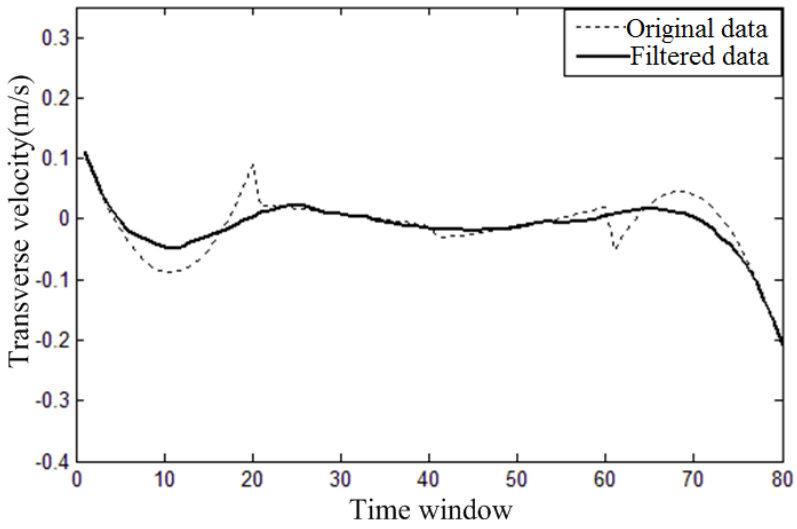

FIGURE 2. Transverse velocity filtering

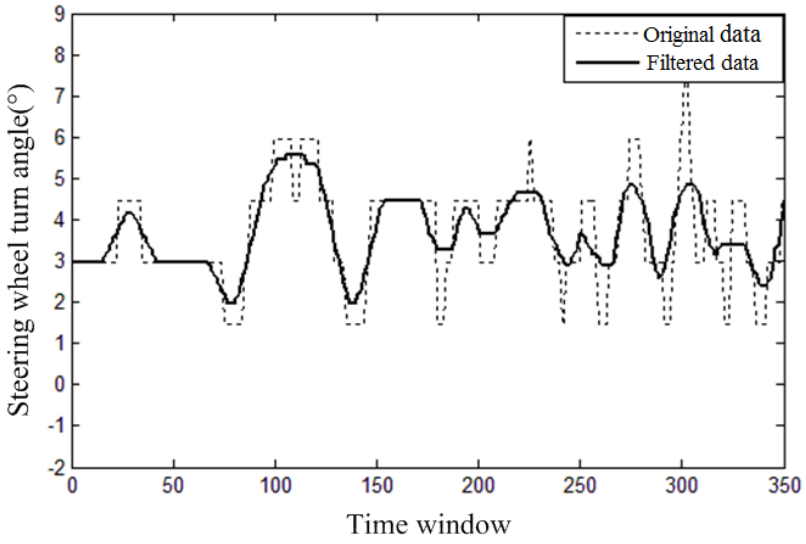

FIGURE 3. Steering wheel turn angle filtering

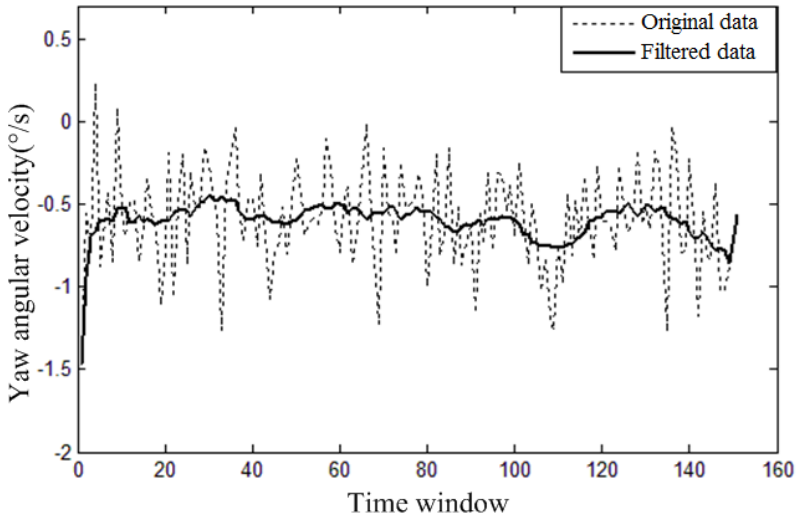

FIGURE 4. Yaw angular velocity filtering

2.4. **Normalization.** The transverse velocities acquired mainly vary within the range of $-5\text{m/s} \sim 5\text{m/s}$, the steering wheel turn angle varies in the range of $-60° \sim 60°$, and the yaw angular velocity mainly varies within the range of $-2°/\text{s} \sim 2°/\text{s}$. For better reflection of data characteristics, normalization is required when data is input. The normalization range selected in this paper is $[0, 1]$. The normalization formula is as follows:

$$x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{5}$$

where $x_{\min}$ refers to the minimum value in the data, and $x_{\max}$ refers to the maximum value in the data.

The normalized processing results are as follows:

Original transverse velocity data: (1.11, 0.73, 0.40, 0.09, −0.16, −0.39, −0.57, −0.71, −0.81, −0.86, −0.87, −0.84, −0.77, −0.65, −0.50)

Data after normalization treatment: (1, 0.88, 0.77, 0.68, 0.59, 0.52, 0.47, 0.42, 0.39, 0.38, 0.37, 0.38, 0.40, 0.44, 0.49)

Original steering wheel turn angle data: (2.97, 4.46, 4.46, 2.97, 2.97, 2.97, 1.48, 1.48, 2.97, 2.97, 2.97, 4.46, 4.46, 5.95, 5.95)

Data after normalization treatment: (0.25, 0.50, 0.50, 0.25, 0.25, 0.25, 0, 0, 0.25, 0.2, 0.25, 0.50, 0.50, 0.74, 0.74)

Original yaw angular velocity data: (−1.47, −0.56, −0.79, 0.23, −0.88, −0.69, −0.43, −0.85, 0.08, −0.55, −0.70, −0.48, −0.48, −0.85, −0.62)

Data after normalization treatment: (0, 0.53, 0.40, 1, 0.34, 0.45, 0.61, 0.36, 0.911, 0.541, 0.45, 0.58, 0.58, 0.36, 0.50)

2.5. **K means clustering.** Sampling frequency of the sensor acquisition system is 10Hz, at which 30 transverse velocity, steering wheel turn angle and yaw angular velocity data are obtained within 1s. If these are directly inputted into the neural network, it will not only increase the neural network computing time, but the large number of redundant data contained in the data will cause over-fitting of the neural network. In this paper, K means clustering method is adopted for clustering of the data, and the center points after clustering are selected as the input data.

Data of steering wheel turn angle, transverse velocity and transverse distance are divided into two categories by using the K means clustering approach. Since the characterization parameter inputs are gradually varied in a certain pattern, Serial Numbers 5 and 9 are respectively selected and used as the initial clustering centers when selecting the initial clustering centers, as shown in Table 1.

Table 1. Initial clustering center

Characterization parameter input	Clustering	
	1	2
Transverse velocity	0.46	0.31
Steering wheel turn angle	8.92	7.15
Yaw angular velocity	−0.50	−0.51

The clustering centers obtained as a result of K means clustering of the data by using K means function in the MATLAB software are as shown in Table 2.

Table 2. Final clustering center

Characterization parameter input	Clustering	
	1	2
Transverse velocity	0.27	0.47
Steering wheel turn angle	6.81	8.79
Yaw angular velocity	−1.21	−0.57

The final clustering centers are used as the input samples, redundant data in the data have been removed, while main information of the data is retained.

2.6. **Model training.** Some 545 sets of samples are selected from the test data, of which 218 sets are lane change samples, accounting for 40% of all samples, and 327 are lane retaining samples, accounting for 60% of all samples. Of the 545 sets of samples, 445 sets of data are selected on a random basis and used for neural network model training, with the remaining 100 sets used for neural network model testing.

Vehicle over lane line is the time point that can cause a traffic conflict. Therefore, the lane change recognition system needs to identify the vehicle's state of motion before it drives over the lane line. It can be known from a statistics of the vehicle line crossing time in the lane change samples that the average line crossing time of the vehicles during

lane change is 3.1s. Within a period of 0.3s $\sim$ 3.0s, the recognition accuracy of the neural network under different time windows is respectively figured out based on a time interval of 0.3s.

It can be known from Table 3 that beginning from 0.3s, recognition accuracy of the neural network persistently increases as the time window increases. At the time window of 2.1s, the recognition accuracy of the neural model exceeds 85%. Thereafter, as the time window increases, the recognition accuracy of the neural network model fluctuates around 85%. This is because as the time window increases, a large amount of redundant information is contained in the information inputted into the model, which affects the recognition effect of the neural network model. At this point, the neural network model has the best recognition effect at the time window of 1.5s. However, the recognition accuracy of the neural network model is still on the low side, and cannot meet the requirements of active safety system.

Table 3. Recognition accuracy under different time windows

Time window (s)	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3.0
Recognition accuracy (%)	69	73	78	82	82	83	86	87	83	88

3. Optimization of Neural Network Model.

3.1. PSO algorithm.

Local search algorithm is used for the neural network parameters during training, which can easily fall into local minimal values when solving sophisticated non-linear issues. Therefore, this paper employs PSO algorithm to optimize the neural network parameters. Optimization process by using PSO algorithm is to first initialize a set of particles in the optimization space. Each particle represents a set of potential optimal parameters of the neural network, and each particle contains three characteristics, namely, velocity, position and fitness value, where the particle is evaluated based on magnitude of the particle fitness value. Individual particle position is updated by using the particle extreme value Pbest and the group extreme value Gbest in the space, of which the particle extreme value Pbest represents the best fitness position obtained during the movement of the particle, and the group extreme value Gbest represents the best fitness positions obtained during the movement of all particles. Through constant update of the group position, the particle extreme value Pbest and the group extreme value Gbest are also persistently recalculated, until the best fitness position in the space is ultimately obtained. The flow of the PSO algorithm to optimize the neural network model is shown in Figure 5.

After optimization of Hopfield neural network by using PSO algorithm, recognition accuracy of the neural network model is figured out under different time windows.

It can be known from Table 4 that recognition accuracies of the neural network optimized by using PSO algorithm are generally improved. At the time window of 1.5s, recognition accuracy of the neural network model has reached 90% or higher. Meanwhile, the time window length is also shortened than 2.1s before the optimization. Recognition accuracy of the optimized neural network model now meets the requirements of the active safety system.

Table 4. Recognition accuracy under different time windows after optimization

Time window (s)	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3.0
Recognition accuracy (%)	71	77	83	88	91	90	91	93	89	90

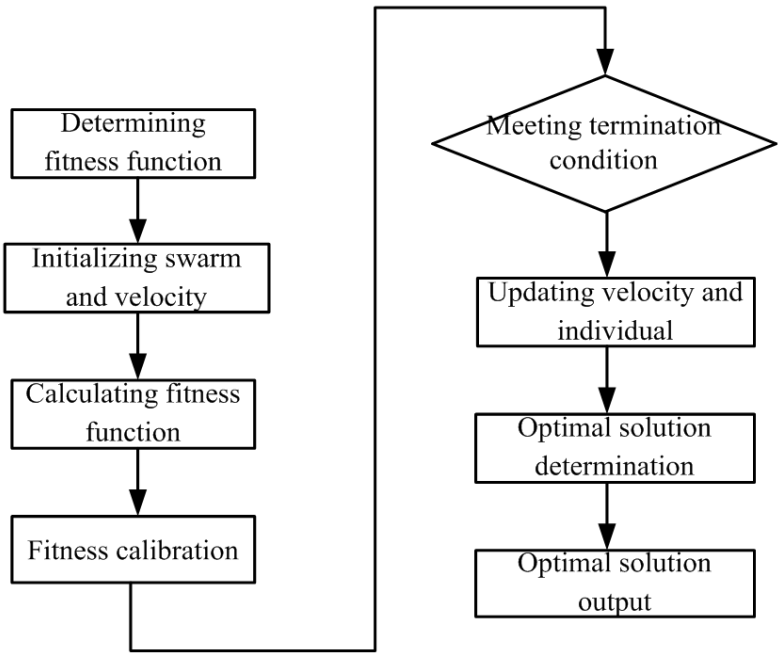

FIGURE 5. Process flow chart for optimizing neural network model by using PSO algorithm

3.2. **Model performance analysis.** Recognition accuracy is a common indicator for evaluating the recognition effect. Recognition accuracy is used for calculating the correct percentage of recognition data samples in the total data samples. In the above part of the text, the recognition effect of the neural network model is evaluated against the recognition accuracy. However, recognition accuracy cannot give an overall indication of the recognition performance of the model, because recognition accuracy cannot reflect the sources of wrong samples as well as the distribution of the wrong samples between in the two states of lane change and lane retaining. With respect to the above circumstances, this paper introduces the two evaluation indicators of a true positive rate and false positive rate to evaluate the classified effects of the neural network. Receiver Operating Characteristic (ROC) curve can be used for an apparent analysis of the equilibrium relationship between true positive rate and false positive rate. In the ROC curve, the false positive rate is shown as the x-axis and the true positive rate is shown as the y-axis in the coordinates. See Figure 6 for the ROC curve.

The recognition effect of the model is then quantitatively evaluated according to the Area Under the ROC Curve (AUC), which shows that the more AUC approaches to 1, the better the recognition effect of the model is. In Figure 6, AUC is 0.9252, indicating that Hopfield neural network model gives an excellent recognition effect no matter it is used for recognizing the lane change state or lane retaining state.

4. **Conclusion.** This paper puts forth a Hopfield neural network-based lane change behavior recognition model to address the lack of effective vehicular lane change behavior recognition in ADAS system. Transverse velocity, steering wheel turn angle and yaw angular speed of vehicles under different states of motion are acquired through real automobile driving test on an expressway. Redundant data in the acquired data are removed through data normalization and K means clustering, whereby main information is extracted. After the Hopfield neural network parameters are optimized by using PSO algorithm, recognition accuracy of the model has reached 91% at the time window of 1.5s, living up to the requirements of the driver assistant system. It can be known through evaluation of the performance of the neural network model by using an ROC curve that when AUC is 0.9252, recognition effect of the neural network model is quite apparent.

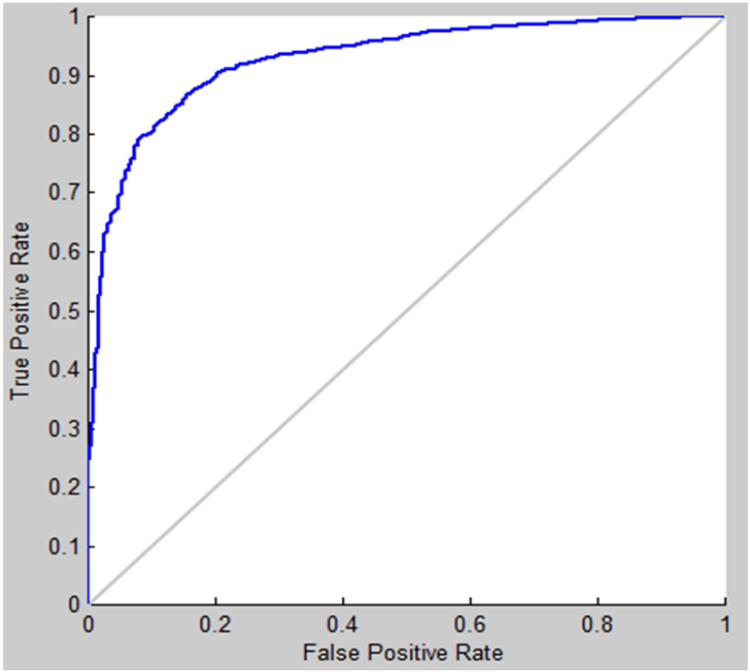

FIGURE 6. ROC curve

Acknowledgment. This work is partially supported by Special funds for basic research of China Academy of Safety Science and Technology (2017JBKY03). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] J. Tang, F. Liu, W. Zhang, R. Ke and Y. Zou, Lane-changes prediction based on adaptive fuzzy neural network, *Expert Systems with Applications*, vol.91, pp.452-463, 2018.

[2] Y. Chu, X. Q. Xiao and J. C. Zhu, Vehicular active safety technology research based on drivers' behaviors and intentions, *Machine Design and Manufacturing*, vol.8, no.1, pp.266-268, 2011.

[3] F. Bocklisch, S. F. Bocklisch, M. Beggiato and J. F. Krems, Adaptive fuzzy pattern classification for the online detection of driver lane change intention, *Neurocomputing*, vol.262, pp.148-158, 2017.

[4] H. Woo, Y. Ji, H. Kono, Y. Tamura, Y. Kuroda, T. Sugano and H. Asama, Lane-change detection based on vehicle-trajectory prediction, *IEEE Robotics and Automation Letters*, vol.2, no.2, pp.1109-1116, 2017.

[5] N. Kuge, T. Yamamura and O. Shimoyama, A driver behavior recognition method based on a driver model framework, *Proc. of the Society of Automotive Engineers World Congress*, 2000.

[6] J. C. Mccall, D. P. Wipf and M. M. Trivedi, Lane change intent analysis using robust operators and sparse Bayesian learning, *IEEE Trans. Intelligent Transportation Systems*, vol.8, no.3, pp.431-440, 2005.

[7] A. Liu and A. Pentland, Towards real-time recognition of driver intentions, *IEEE Conference on Intelligent Transportation System (ITSC)*, pp.236-241, 1997.

[8] D. Urun, J. Edelbrunner and I. Iossifidis, Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior, *Proc. of IEEE International Conference on Robotics & Biomimetics*, pp.1837-1843, 2011.

[9] W. Yuan, R. Fu and Y. S. Guo, Driver's lane change intention recognition based on visual characteristics, *China Journal of Highway and Transport*, vol.26, no.3, pp.132-138, 2013.

[10] Y. Ma, R. Fu and Y. S. Guo, Multi-parameter prediction of dirvers' lane change behaviors based on real automobile test, *Journal of Chang'an University*, vol.34, no.5, pp.101-108, 2014.

[11] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. of the National Academy of Sciences of the United States of America*, vol.79, no.8, pp.2554-2558, 1982.

[12] H. M. T. Menegaz, J. Y. Ishihara, G. A. Borges et al., A systematization of the unscented Kalman filter theory, *IEEE Trans. Automatic Control*, vol.60, no.10, pp.2583-2598, 2015.