Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Vehicle detection in driving simulation using extreme learning machine

Wentao Zhu [a], Jun Miao [a,*], Jiangbi Hu [b], Laiyun Qing [c]

[a] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[b] Transportation Research Center, Beijing University of Technology, Beijing 100124, China
[c] School of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 7 September 2012
Received in revised form
30 April 2013
Accepted 6 May 2013

Keywords:
Extreme learning machine
Driving simulation
Vehicle detection
Road segmentation

ABSTRACT

Automatically driving based on computer vision has attracted more and more attentions from both research and industrial fields. It has two main challenges, high road and vehicle detection accuracy and real-time performance. To study the two problems, we developed a driving simulation platform in a virtual scene. In this paper, as the first step of final solution, the Extreme Learning Machine (ELM) has been used to detect the virtual roads and vehicles. The Support Vector Machine (SVM) and Back Propagation (BP) network have been used as benchmark. Our experimental results show that the ELM has the fastest performance on road segmentation and vehicle detection with the similar accuracy compared with other techniques.

1. Introduction

Long hour driving usually causes drivers loosing focus on recognizing roads and vehicles. Automatically driving has been a goal that researchers are working on in recent years. Google announced its research of auto-driving to prevent accidents in its official blog on October 10th, 2010. Its automatic-driving technique has been tested for 320,000 km, and has no single accident until April 9th, 2012. Google's automatically driving technique uses video camera, radar and laser to detect traffic information and navigate the vehicle through a detailed map. Not only Google, Volvo and German computer expert, Raul Rojas, the Free University of Berlin team (MIG, Germany) have developed their own automatic-driving techniques. Automatic-driving has two main challenges: the accuracy and real-time performance on highway detection and vehicle detection. To study these two problems from computer vision direction, we have developed a driving simulation platform with a virtual camera in a virtual scene. In this paper, as our first step to tackle the challenges, the Extreme Learning Machine (ELM) [1] technique has been used to detect the virtual road and vehicle.

ELM was first proposed in [2], which has overcome some challenging issues, such as slow learning speed, trivial human intervening and poor computational scalability. The essence of the ELM is that hidden layer need not be tuned iteratively. ELM has attracted more and more researchers and engineers, because of its better generalization performance with a much faster learning speed and less human intervening. Taking into account of real-time performance and high accuracy requirements in automatically driving, we use ELM to detect vehicles in our driving simulation.

In a driving simulation system, road segmentation and vehicle detection are required. There are many approaches to detect road. Radar, laser, stereovision [3], Hough transform [4], spline model [5] and steerable filters [6] are used to find road borders or road signs. But these methods can only be applied on structured roads with salient borders and signs. Alon et al. [7] combined region segmentation based on Adaboost with border recognition based on geometric projection to segment drivable road. But it requires many kinds of road images to train classifier for the similar region. Reverse optical flow technique [8] provides an adaptive road region segmentation method. But the estimation of optical flow is not robust to chaotic road when the camera is not fixed. Some methods [9,10] managed to segment drivable road area based on texture information. They detect the vanish point through a voting scheme according to the texture orientation of each pixel. These methods usually fail because of the inaccurate estimation of vanish point. Kong et al. [9] introduced a confidence level method to find vanishing point more accurately. However, this method is not suitable for winding road, and it is fatal to automatically drive when a vehicle is on a turn of the road.

There are also many approaches to detect vehicles. Some vehicle detection methods are based on stereo vision [11], where the disparity map [12] or inverse perspective mapping [3] is utilized.

* Corresponding author. Tel.: +86 10 62600556.
E-mail address: jmiao@ict.ac.cn (J. Miao).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.05.052

These methods are relying on special instruments and they are not suitable for the virtual scene that is obtained by just using a single camera. In methods [13], partial least squares were used to select features, which concatenated histogram of oriented gradients (HOG) [14], color probability maps and pairs of pixels features. Many features were extracted. Feature dimension reducing is a time-consuming process in practice. Some vehicle detection methods are based on constructed vehicle models [15], such as active basis model [16,17]. It is time-consuming on a driving simulation case. The ELM has been used in [18] to detect vehicle based on singular value decompositions. In their work, the camera was fixed, which is not the case in our simulation system.

Inspired by the good performance of ELM, we used ELM method to detect roads and vehicles by using small window sliding technique. In the small window, the corresponding features were extracted.

Color histogram features was obtained to detect roads.

Two sets of features have been used to detect vehicles. One is gray color features, and another is Histogram of Oriented Gradients (HOG) [14] feature. The performance of the vehicle detection was measured on a virtual scene. Our proposed method has a high accuracy and real-time performance on vehicle detection compared with other techniques based on our experimental results.

In this paper, a virtual scene was used for road segmentation and vehicle detection. ELM was used as a two classes' classifier to recognize vehicles. Experiments were carried out to verify ELM's performance on speed and accuracy in the task of road and vehicle detection.

The rest of the paper is organized as follows. ELM is briefly introduced in Section 2. Road segmentation which removed the outlier road patches is introduced in Section 3.1. Vehicle detection in the extended road area is introduced in Section 3.2. Experiments and performance comparisons among ELM, SVM and BP network are introduced in Section 4.

2. ELM

ELM [19] is a single hidden layer forward network (SLFNs). It has many good features, such as fast learning speed, good generalization performance and automatically tuning hidden layer parameters [1].

For N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i) \in (\mathbf{R}^d \times \mathbf{R}^m)$, $\mathbf{x}_i$ is the extracted feature vector and $\mathbf{t}_i$ is the target output label. The mathematical model of ELM with L hidden nodes is

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i g_i(\mathbf{x}_j) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \widehat{\mathbf{t}}_j, \quad j = 1, \ldots, N$$

If $N=L$, ELM can approximate the targets of the distinct N samples with zero error:

$$\sum_{j=1}^{N} \| \widehat{\mathbf{t}}_j - \mathbf{t}_j \| = 0,$$

that is, there exist some set of values $\boldsymbol{\beta}_i, \mathbf{a}_i$ and b_i, such that,

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{t}_j, \quad j = 1, \ldots, N,$$

which is equivalent to, $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$, where,

$$\mathbf{H} = \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L}, \quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix}_{L \times m},$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}_{N \times m}$$

As Huang et al. proved [20], the parameters of hidden layer, $\{\mathbf{a}_i, b_i\}_{i=1}^{L}$, can be randomly generated. There exists $L \le N$, making training error as small as possible with probability one. Training process of ELM is equivalent to solve a least squares problem. That is, $\widehat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}$, where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of hidden layer output matrix $\mathbf{H}$.

The ELM used by the following sections can be summarized in three steps:

Step 1: Assign the parameters of hidden nodes $\mathbf{a}_i, b_i$, $i=1,\ldots,L$ with randomly generated values.

Step 2: Calculate the hidden layer output matrix $\mathbf{H}$.

Step 3: Calculate the output weight $\boldsymbol{\beta}$ by solving the least squares problem: $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$.

3. Vehicle detection

Our approach detected the road first, and then detected the vehicles in an extended road area. The entire detection framework is illustrated in Fig. 1.

In this section, we first introduce how to segment the road, and then discuss how to detect vehicles.

3.1. Road segmentation

Given an $H \times W$ image $\mathbf{I}$ from the video, non-overlapped patches are drawn from it. The size of the patch is $h \times w$, then the image $\mathbf{I}$ was divided into $\lceil H/h \rceil \times \lceil W/w \rceil$ patches, denoted as $p_{11}, \ldots, p_{1,\lceil W/w \rceil}, \ldots, p_{\lceil H/h \rceil, \lceil W/w \rceil}$.

For the road in the virtual scene, our method segmented the road by using a $h \times w$ window to slide over the image and classify whether each patch in the window belongs to the road or not. We extracted color cues in the small window, and use ELM as a classifier. After comparing different features, we found that color histogram was an efficient feature to distinguish the road from other objects in the virtual scene, as illustrated in Fig. 2.

The result of road segmentation is denoted as a matrix of $\lceil H/h \rceil \times \lceil W/w \rceil$. Dimensions are standing for whether patches are within a road area or not. However, there were some outliers mistakenly detected as road patches, as illustrated in Fig. 3.

In the real word, a drivable road is a continuous area and the isolated patch is not likely belonged to the road. Based on this assumption, those road patch outliers were removed from our initial road segmentation result.

3.2. Vehicle detection

In an image, normally the patches of a vehicle which appeared in or above the road area were segmented in the first step. Therefore, vehicle detection was processed in an extended road area. Extended road area is shown in Fig. 4. The width of the extended road area is the same as the width of the image. The

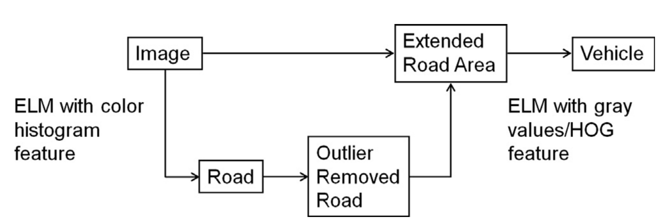


Fig. 1. The framework of vehicle detection.

Please cite this article as: W. Zhu, et al., Vehicle detection in driving simulation using extreme learning machine, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.05.052

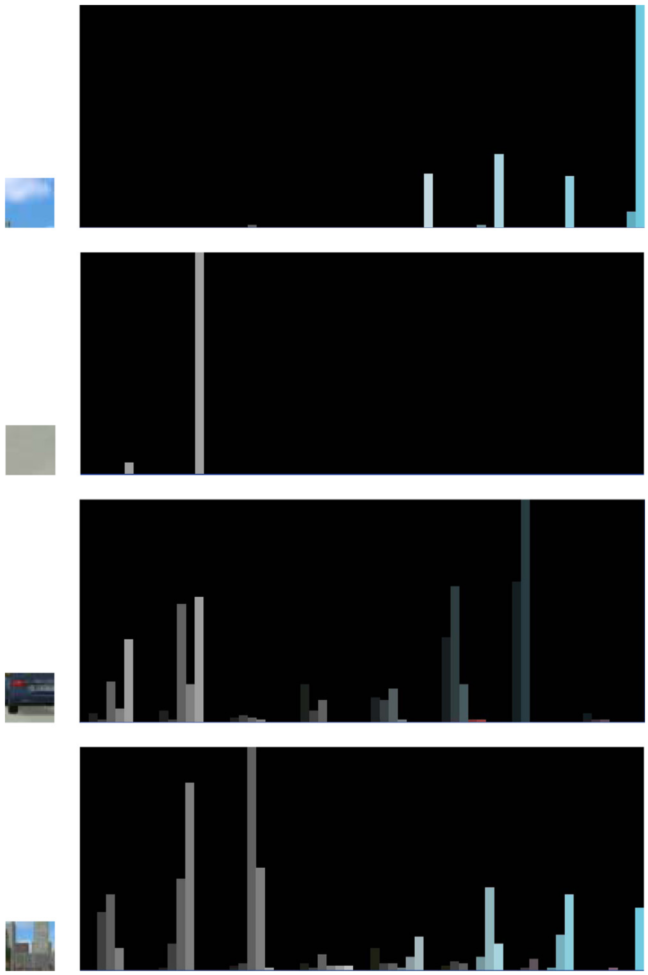


Fig. 2. Color histogram comparison of sky, vehicle, road and buildings. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

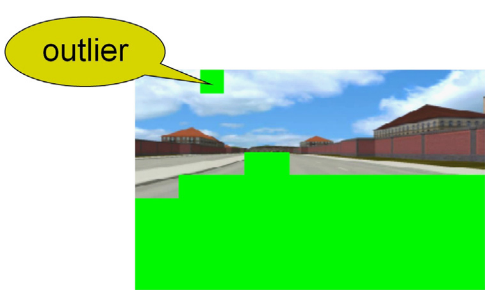


Fig. 3. Outlier illustration in road detection.

Fig. 4. Extended road area.

height of the extended road area is the height of the road plus the average height of the vehicles.

For vehicle detection, the ELM was used with two sets of different color features: grey color value and Histogram of Oriented Gradients [14].

Small sliding window was used on an extended road area to detect vehicles patch by patch.

The gray pixel values of these patches were used to determine whether the patches were the vehicle patches or not.

The rectangular histograms of oriented gradients (R-HOG) for patches in an extended road area were used to decide whether the rectangle was a vehicle or not. The comparison results obtained by using the ELM, kernel SVM and BP network methods based on R-HOG detectors are listed in Section 4.

These implementations have high accuracy and real-time performance. Some results are illustrated in Figs. 5 and 6 in the next section.

4. Experiments

In this section, experimental conditions of the hardware and the software are first introduced, and then the performance comparison of ELM, kernel SVM and BP network classifiers are presented.

4.1. Experimental conditions

In all the experiments, the algorithms were run on the environment: (1) Operating system: Windows XP Professional with service Pack 3. (2) CPU: Pentium(R) Dual-Core T4300. (3) Memory: 2 GB. (4) Software: Driving simulation system is developed in Visual Studio 2008 professional edition with OpenCV 2.3.1 kits [21] and comparison experiments are done in Matlab R2009(a) (7.8). (5) Simulation environment: Virtual driving scenes are shown in the experiment results section.

C-coded ELM algorithm was developed in driving simulation system and its accuracy performance was comparable to the Matlab code from the homepage of ELM's inventor [22]. Matlab coded ELM was used in comparing experiments. BP algorithm was adopted from Matlab *Neural Network Toolbox*^{TM} and the BP network was trained by the resilient back-propagation (RPROP) algorithm [23] to avoid the memory problem. C-coded SVM package from LIBSVM [24] was also used in comparison experiments.

4.2. Experiment results

Classifiers of ELM, SVM and BP networks were constructed to carry out the comparison experiments. Sigmoidal function $g(x) = 1/(1+\exp(-x))$ was chosen as the activation function of ELM. The kernel function of SVM was radical basis function (RBF). The transfer function of BP network was default tangent sigmoidal function $f(x) = (1-\exp(-2x))/(1+\exp(-2x))$. Here BP network was designed to have one hidden layer. The input features were normalized to $[-1,1]$ and the output was 0 or 1, which was a binary classification for targets. The output denotes that the current patch is the road/vehicle or not.

There were three main parameters needed to determine: the number of hidden nodes in ELM, the cost factor in SVM and the number of hidden nodes in BP network. Parameter selection for ELM and BP network was done with the scheme of coarse-to-fine. The initial parameter range and the earlier search step length were different to those later. For example, after the first best parameter was obtained from the first search round, the parameter range was updated to the neighborhood of that obtained parameter and the search step length was decreased. The range of the number of hidden nodes in ELM and BP network was from 20 to 330, which

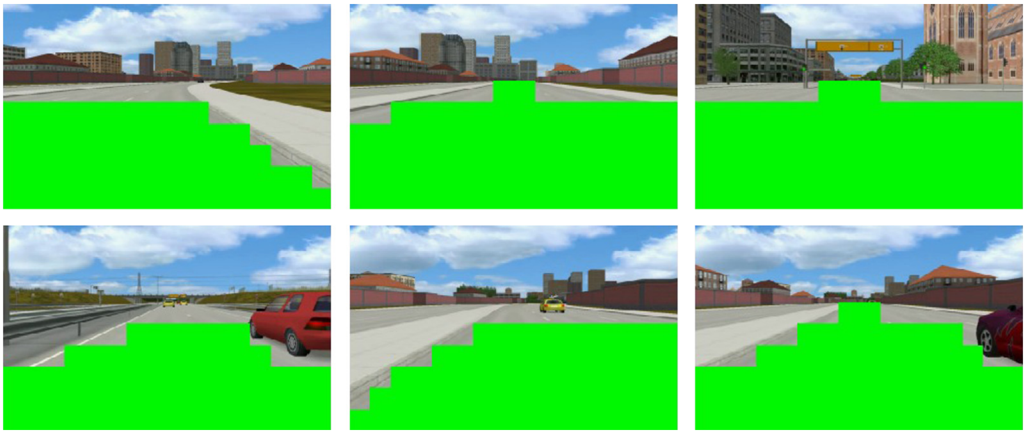


Fig. 5. Road segmentation examples in the driving simulation system. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Fig. 6. Vehicle detection examples in the driving simulation system. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

was decided by our coarse-to-fine search procedure. For parameter selection of SVM, the best parameter was obtained from the candidate sequence $2^{-24}, 2^{-23}, \ldots, 2^{23}, 2^{24}, 2^{25}$ using the same experimental principle as in [19].

For each candidate parameter, three-fold cross validation was conducted to select the parameters of these classifiers. In the road segmentation experiment, test accuracy was used as the performance evaluation metric. In the vehicle detection experiment, true positive rate (TPR) in test procedure was used as the performance evaluation metric. The average value of each corresponding metric in the three cross-validation experiments was used as the performance evaluation metric to search the optimal parameters.

As a result, we obtained the following parameters in details. In the road segmentation experiment, ELM had 195 hidden nodes. The cost factor of SVM was 2^{12}. BP network had 220 hidden nodes. In the first experiment of vehicle detection, ELM was constructed with 153 hidden nodes. The cost factor of SVM was 2^{14}. BP network had one hidden layer with 170 hidden nodes. In the second experiment of vehicle detection, ELM had 163 hidden nodes. The cost factor of SVM was 2^2. BP network had one hidden layer with 23 hidden nodes.

Other parameters were set as follows. The size of patches was 30×30 pixels and the average height of vehicles was set as 30 pixels. Color histogram feature had 78 dimensions in RGB color space and each channel was with 26 dimensions evenly. Other parameters, like kernel parameter γ, were set as default values in the experiments.

The first experiment was carried out for the road segmentation task. Some image frames were randomly snatched from the virtual driving scenes. We collected 7775 patches and 3874 of them were road patches. The number of the training set was 2/3 of the total number of the patches and the rest was the testing set. These samples were randomly selected from the collected patches. Three experiments were carried out independently. The average performance among ELM, SVM and BP network was shown in the table below.

Table 1
Performance comparison among ELM, SVM, BP network on road segmentation.

Algorithms	Training		Testing	
	Accuracy	Time (s)	Accuracy	Time (s)
ELM	0.9873	0.3646	0.9868	0.0885
SVM	0.9907	1.6120	0.9884	0.1300
BP	0.9821	191.6510	0.9799	0.1302

Table 1 shows that ELM can reach the higher accuracy and is the fastest algorithm for training and testing, although the SVM algorithm is implemented in C. Because the color histogram feature extracting and the classification speed are very fast, the method can reach real time performance in our driving simulation system. The effect of our road segmentation in our driving simulation system is illustrated in Fig. 5. The green area is labeled as road.

On vehicle detection, one classifier with two kinds of features were constructed. One is the ELM classifier with pixels' gray value feature. The other one is with HOG feature.

Please cite this article as: W. Zhu, et al., Vehicle detection in driving simulation using extreme learning machine, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.05.052

Table 2
Performance comparison among ELM, SVM, BP network with gray value feature on vehicle detection.

Algorithms	Training				Testing			
	Accuracy	TPR	FPR	Time (s)	Accuracy	TPR	FPR	Time (s)
ELM	0.9385	0.8547	0.0518	0.9531	0.9235	0.8352	0.0615	0.3229
SVM	0.9931	0.9946	0.0070	78.6817	0.9276	0.6671	0.0433	14.1129
BP	0.9291	0.6146	0.0354	195.5729	0.8679	0.3092	0.0691	0.4115

Table 3
Performance comparison among ELM, SVM, BP network with HOG feature on vehicle detection.

Algorithms	Training				Testing			
	Accuracy	TPR	FPR	Time (s)	Accuracy	TPR	FPR	Time (s)
ELM	0.9908	0.9987	0.0119	0.0313	0.9598	0.9843	0.0366	0.0156
SVM	1	1	0	0.0934	0.9879	0.9814	0.0055	0.0313
BP	0.9993	1	0	0.5885	0.9183	0.9168	0.0856	0.0104

Next, we show the performance with the former feature set. Two benchmark classification algorithms, SVM and BP networks, were used to compare the performance of ELM in the vehicle detection. Sample images randomly were obtained from the simulation scenes and 10,000 patches were produced from these samples which had 1015 vehicle patches. The number of the training set was 2/3 of the total number of the patches and the rest was the testing set. These samples were randomly selected from the collected patches. Three experiments were done independently. True positive rate (TPR) and false positive rate (FPR) were used to measure the performance of the vehicle detection methods. TPR was calculated by TP/P, where TP is the number of samples correctly classified in the positive samples and P is the number of positive samples. FPR was calculated by FP/N, where FP is the number of samples classified as positive samples in the negative samples and N is the number of negative samples. The average performance is shown in Table 2.

The results in Table 2 show that although the true positive rate (TPR) of ELM is about 0.17 larger than the result obtained by of the SVM, the accuracy of ELM is similar to that of SVM for there are only 1/10 positive samples. See the following equation:

$$Accuracy = \frac{TPR \times P + (1 - FPR) \times N}{P + N},$$

where P and N are the numbers of positive and negative samples respectively, and $P/(P+N)$ is 1/10. FPR (false positive rate) of SVM is only about 0.02 larger than that of ELM. Based on the above equation, we conclude that the accuracy of ELM is similar to the accuracy provided by the SVM method. The experimental results show that performance of the classifiers cannot just be measured based on their accuracy and speed also should be taken into account. At mean time, the ELM worked well on this experimental case.

The results in Table 2 show that the SVM is the slowest algorithm. Experiment reveals that there were more than 1000 support vectors in the model, which leaded the SVM to be the slowest algorithm in this case.

ELM has a comprehensive advantage over other two algorithms. The vehicle detection method based on ELM can reach real-time in driving simulation system and the performance is illustrated in Fig. 6. The red bounding boxes are the detected vehicles.

In the second experiment, vehicle detection method with HOG feature was implemented. We used OpenCV software kits to extract HOG feature and collected 747 frames from the simulation scenes. There were 371 positive samples and 376 negative samples obtained from these frames. The number of the training set was 2/3 of the total number of the patches and the rest was the testing set. These samples were randomly selected from the collected patches. Three experiments were done independently. Because of the imbalance between the number of samples and the number of HOG feature dimensions, PCA was applied to reduce the dimensions of HOG feature from 3780 to 100. The average performance is shown in Table 3.

The results in Table 3 show that the ELM method has the highest true positive value and a much high speed. High true positive rate means high accuracy in the vehicle sample classification. If the vehicle is wrongly classified into non-vehicle patch, there will be great danger to the safety. And if non-vehicle patch is wrongly classified into vehicle patch, there will be less danger than the opposite case above. So ELM has a comprehensive advantage over other two classifiers when it is used for vehicle detection task. Moreover, ELM is a faster algorithm than SVM although the SVM is C-coded.

5. Conclusions

This paper studied ELM for road segmentation and vehicle detection in a virtual scene served for driving simulation. On road segmentation, one ELM classifier with color histogram features was constructed. Outlier road patches were removed to improve performance. On vehicle detection, one ELM classifier with gray value and HOG feature was conducted. They were compared with the classifiers of SVM and BP Network on time and accuracy. Experimental results indicate that the ELM classifiers have similar accuracy to that of SVM, and have higher accuracy than that of BP. ELM is comprehensively the fastest algorithm in the experiments. They were implemented in the driving simulation system and show a very good performance on vehicle detection. ELM can meet real-time requirement in the simulation system.

Currently, road segmentation results are not very accurate for its rectangular segmentation pattern. Because of the real-time requirement for driving simulation, scale transformation is not conducted for detecting vehicles using a single bounding box. Besides, the virtual scene is ideal and not taken into account for some situations in the real world, like a cloudy weather, gray cars, non-gray road, different lighting conditions and shadows on the road in our simulation system. These will be seriously taken into consideration in our future work.

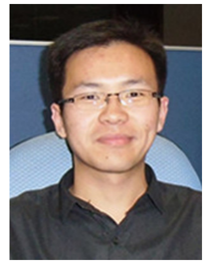
Please cite this article as: W. Zhu, et al., Vehicle detection in driving simulation using extreme learning machine, Neurocomputing (2013), http://dx.doi.org/10.1016/j.neucom.2013.05.052

Acknowledgements

This research is partially sponsored by Natural Science Foundation of China (Nos. 61070116, 61070149, 61001108, 61175115, and 61272320).

References

[1] G.-B. Huang, D. Wang, Y. Lan, Extreme learning machines: a survey, Int. J. Mach. Learn. Cybern. 2 (2011) 107–122.
[2] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of IEEE International Joint Conference on Neural Networks, vol. 2, July 2004, pp. 985–990.
[3] M. Bertozzi, A. Broggi, Gold: a parallel real-time stereo vision system for generic obstacle and lane detection, IEEE Trans. Image Process. 7 (January) (1998) 62–81.
[4] B. Yu, A. Jain, Lane boundary detection using a multiresolution hough transform, in: Proceedings of International Conference on Image Processing, vol. 2, October 1997, pp. 748–751.
[5] Y. Wang, E.K. Teoh, D. Shen, Lane detection and tracking using b-snake, Image Vis. Comput. 22 (4) (2004) 269–280.
[6] W. Freeman, E. Adelson, The design and use of steerable filters, IEEE Trans. Pattern Anal. Mach. Intell. 13 (September) (1991) 891–906.
[7] Y. Alon, A. Ferencz, A. Shashua, Off-road path following using region classification and geometric projection constraints, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, June 2006, pp. 689–696.
[8] A. Lookingbill, J. Rogers, D. Lieb, J. Curry, S. Thrun, Reverse optical flow for self-supervised adaptive autonomous robot navigation, Int. J. Comput. Vis. 74 (2004) 287–302.
[9] H. Kong, J.-Y. Audibert, J. Ponce, Vanishing point detection for road detection, in: IEEE Conference on Computer Vision and Pattern Recognition, June 2009, pp. 96–103.
[10] C. Rasmussen, Grouping dominant orientations for ill-structured road following, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. I–470–I–477.
[11] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection: a review, IEEE Trans. Pattern Anal. Mach. Intell. 28 (May) (2006) 694–711.
[12] M. Bertozzi, A. Broggi, A. Fascioli, S. Nichele, Stereo vision-based vehicle detection, in: Proceedings of the IEEE Intelligent Vehicles Symposium, 2000, pp. 39–44.
[13] A. Kembhavi, D. Harwood, L. Davis, Vehicle detection using partial least squares, IEEE Trans. Pattern Anal. Mach. Intell. 33 (June) (2011) 1250–1265.
[14] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, June 2005, pp. 886–893.
[15] A. Petrovskaya, S. Thrun, Model based vehicle detection and tracking for autonomous urban driving, Auton. Robot. 26 (2009) 123–139.
[16] Y. Wu, Z. Si, H. Gong, S.-C. Zhu, Learning active basis model for object detection and recognition, Int. J. Comput. Vis. 90 (2010) 198–235.
[17] W. Hu, Y.N. Wu, S.-C. Zhu, Image representation by active curves, in: IEEE International Conference on Computer Vision, November 2011, pp. 1808–1815.
[18] Z. Saad, M. Osman, I. Isa, S. Omar, S. Ishak, K. Ahmad, R. Boudville, Vehicle recognition system using singular value decomposition and extreme learning machine, Object Recognition, Dr. T. Cao (Ed.), 2011, ISBN: 978-953-307-222-7, InTech, Available from: 〈http://www.intechopen.com/books/object-recognition/vehicle-recognition-system-usingsingular-value-decomposition-and-extreme-learning-machine〉.
[19] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst. Man Cybern.—Part B: Cybern. 42 (April) (2012) 513–529.
[20] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (July) (2002) 971–987.
[21] 〈http://www.opencv.org.cn/〉.
[22] 〈http://www.ntu.edu.sg/home/egbhuang/〉.
[23] M. RiedmillerOjala, Rprop-Description and Implementation Details, Technical Report, University of Karlsruhe, January 1994.
[24] 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm/〉.



Wentao Zhu received the B.Sc degree in Information and Computing Sciences from Shandong University at Weihai, Weihai, China, in 2012. He is currently working toward the M.Sc degree in computer science at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. His research interests include extreme learning machines, neural networks, and Brain-Mind Developmental Learning.

Jun Miao received the B.Sc. and M.Sc. degrees in computer science from Beijing University of Technology, Beijing, China, in 1993 and 1999, respectively. He received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2005. He is currently an Associated Professor at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. His research interests include artificial intelligence, neural networks, image understanding, and biologically inspired computer vision. He has published more than 40 research articles in refereed journals and proceedings on face detection, visual neural networks, visual neural information coding, neural oscillation, image segmentation, visual perception and cognition. His two main contributions are the technique of Human Face Gravity-Center Template for face detection and the model of Visual Perceiving and Eyeball-Motion Controlling Neural Network for visual search, respectively.

Jiangbi Hu received the B.Sc. and M.Sc. degrees in transportation management and road engineering management from Xi'an Highway Institute, Xi'an, China, in 1987 and 1993, respectively. She received the Ph.D. degree in transportation planning and management from the College of Architecture and Civil Engineering, Beijing University of Technology, Beijing, in 2006. She is currently a Professor at the College of Architecture and Civil Engineering, Beijing University of Technology, Beijing. Her research interests include road safety engineering, design and analysis technique of road life-cycle engineering. She has published more than 60 research articles in refereed journals and proceedings on driving behavior characteristics, driver's driving workload, driving safety and comfort, traffic accidents black-spots, design and analysis technique of road life-cycle engineering. She has published 2 books on road engineering economic and analysis of road life-cycle engineering. Her main contribution is use of driving workload theory in road engineering.

Laiyun Qing is with the School of Information Science and Engineering, Graduate University of the Chinese Academy of Sciences, China. She received her Ph.D. in Computer Science from Chinese Academy of Sciences in 2005. Her research interests include pattern recognition, image processing and statistical learning. Her current research focuses on neural information processing.