Neurocomputing 128 (2014) 160–165

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
Available online 25 October 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
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

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

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,

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

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 $\mathbf{a}_i, b_i$, $i = 1,\dots,L$ 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 $p_{11}, \dots, p_{1,\lceil W/w \rceil}, \dots, p_{\lceil H/h \rceil, \lceil W/w \rceil}$.

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

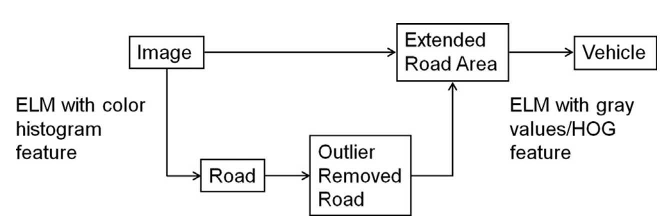

Fig. 1. The framework of vehicle detection.