Traffic sign detection based on Haar and adaBoost classifier

Wenhao Wang1*, Hui Gao1, Xiaobing Chen1, Zhenyang Yu1, Mingxing Jiang1
1Faculty of Computer and Software Engineering, HuaiYin Institute of Technology, Huai’an, Jiangsu, 223003, China
*Corresponding author’s e-mail: jswwh407@sina.com

Abstract. Intelligent transportation system is a hot issue in the field of computer vision. Traffic sign detection is an important part of intelligent transportation system. A traffic sign detection algorithm based on Haar feature and AdaBoost algorithm is proposed in this paper. Firstly, Haar features are extracted from the positive and negative samples of the training set, and then these features are used to train the AdaBoost cascade classifier. Finally, the Haar features are extracted from the test image, and the possible traffic signs in the image are detected, the experimental results show that the method can achieve good detection effect, and provide a new idea for the detection of traffic signs.

1. Introduction
Traffic sign recognition (TSR) is a kind of traffic sign recognition system, which uses computer vision technology to collect the image of traffic signs, and then detect and recognize the traffic sign image. Finally, the obtained traffic sign information is returned to the driver, so as to enhance the driving assistance function of traffic vehicles. This system can effectively help the driver to know the road conditions in front, and improve the safety rate of traffic travel. At present, the detection of traffic signs is mainly divided into the detection based on color or shape information and the detection based on deep learning [1]. Traffic sign detection based on color space can be divided into RGB space and HSV space. This method can detect candidate regions quickly and efficiently, but they are vulnerable to illumination[2]. Detection method based on shape uses the particularity of the shape of traffic signs to detect their edges, but when the traffic signs are blurred or occluded, the accuracy of the algorithm will decline [3]. With the rapid development of deep learning, image detection and recognition algorithm based on neural network has gradually been widely concerned by scholars [4]. Ciresan et al[5] used the deep network to detect and recognize traffic signs, and achieved better results, but because of the gradient descent method, the training time is too long and the practicability is poor.

A traffic sign detection method based on Haar feature and AdaBoost classifier is proposed in this paper. The Haar features of positive and negative samples are extracted by using integration method, and then the weak classifier is constructed by using these features, and the strong classifier is constructed by AdaBoost. Finally, the trained classifier is used to detect the traffic signs in the real image.

2. Haar features
Haar feature is a kind of rectangular feature proposed by Papageorgiou et al [6]. It is defined as the difference of the sum of gray values of pixels in adjacent regions in the image. That is, the sum of pixels in the white area minus the sum of pixels in the black area, which reflects the gradient change from the white region to the black region. Lienhart et al.[7] further extended the Haar rectangular feature library to 15 feature sets, as shown in Figure 1
Figure 1. Haar rectangle feature

In order to eliminate the impact of different region and ensure that the eigenvalues of all Haar features are absolutely uniform in greyscales distribution, a weight is set. Therefore the feature values for each feature are calculated as formulas:

\[ f_v(x) = \text{weight}_{\text{all}} \times \sum_{\text{pixel} \in \text{all}} \text{pixel} + \text{weight}_{\text{black}} \times \sum_{\text{pixel} \in \text{black}} \text{pixel} \]  \tag{1}

In Figure 1, for the \( x_3 \) and \( y_3 \) rectangle features, \( \text{weight}_{\text{all}} = 1, \text{weight}_{\text{black}} = -3 \); For point rectangle feature, \( \text{weight}_{\text{all}} = 1, \text{weight}_{\text{black}} = -9 \); other rectangle feature, \( \text{weight}_{\text{all}} = 1, \text{weight}_{\text{black}} = -2 \).

In order to compute Haar features, a large number of windows need to be scanned. Such will reduce the speed of training and detection. In order to ensure the real-time requirement, the integral image method can be used to quickly calculate the eigenvalues of the rectangle.

The integral image value at a point \((x, y)\) in image \( I(x, y) \) is defined as follows:

\[ SAT(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \]  \tag{2}

Where \( I(x', y') \) represents the gray pixel value at the \((x', y')\) position of the image.

After getting the integral graph, Haar eigenvalues can be calculated. For feature template \( x_2 \), as shown in Figure 2, the gray values of regions A and B can be expressed as

|   | A \((i_1,j_1)\) | C \((i_3,j_3)\) |
|---|---|---|
|   | B \((i_3,j_2)\) | D \((i_4,j_1)\) |

Figure 2. Example of eigenvalue calculation
\[ R_d = SAT(i_1, j_1), R_a = SAT(i_2, j_2) - SAT(i_1, j_1) \]

Where: \( SAT(i_1, j_1) \) is the pixel of rectangular block A and \( SAT(i_2, j_2) \) is the pixel sum of rectangular blocks A and B. They are the integral image values of points \((i_1, j_1)\) and \((i_2, j_2)\) respectively. Then the eigenvalues is as follows

\[ R_a - R_d = SAT(i_2, j_2) - 2SAT(i_1, j_1) \]

The introduction of integral graph, the eigenvalue calculation of rectangular block is independent of the image coordinate, and the eigenvalue can be obtained by the four endpoint coordinates of the rectangle, which greatly improves the calculation speed.

3. AdaBoost classifier
AdaBoost is a kind of boosting algorithm, which was proposed by Freund [8]. It can train different weak classifiers for the same sample set, and then integrate them by weight to get a strong classifier. The algorithm is as follows:

Suppose the training sample set is:

\[ S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \]

Where, \( N \) is the total number of samples; \( x_i \) represents the characteristics of the image, \( y_i \) represents the category

Step 1: The positive sample weight is \( w_{i,1} = \frac{1}{2m} \), The negative sample weight is \( w_{i,2} = \frac{1}{2n} \), \( m, n \) are the number of positive and negative samples, respectively.

Step 2: Set \( t = 1, 2, \ldots, T \), where \( T \) is the number of weak classifiers, Perform the following operations in a loop.

1. Normalized weight.

\[ q_{t,i} = \frac{w_{i,t}}{\sum_{j=1}^{N} w_{i,j}} \]

2. for each feature \( j \), a weak classifier is trained.

\[ h_j(x) = \begin{cases} 1, & P_j f_j(x) < p_j \theta_j \\ 0, & \text{other} \end{cases} \]

Where, \( P_j = \pm 1 \), which represents the direction of the inequality; \( f_j(x) \) represents the eigenvalue, \( \theta_j \) is the threshold.

3. the weighted error rate of the weak classifier is calculated.

\[ \varepsilon_j = \sum_i q_i \left| h_j(x_i) - y_i \right| \]

4. Selecting the weak classifier with minimum error.

\[ \varepsilon_t = \min_{f, p, \theta} \sum_{i} q_i \left| h_j(x_i) - y_i \right| \]

5. The sample weights are updated according to the optimal weak classifier;

\[ w_{t+1,i} = w_{t,i} \beta_t^{1-\varepsilon_t} \]

Where \( \varepsilon_t = \begin{cases} 0, & h_j(x_i) = y_i \\ 1, & h_j(x_i) \neq y_i \end{cases} \)

\[ \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t} \]

Step 3: Building strong classifiers
\[ R(x) = \begin{cases} 1, & \sum_{t=1}^{T} \delta_i h(x) \geq 1 - \frac{1}{2} \sum_{t=1}^{T} \delta_i, \\ 0, & \text{other} \end{cases} \] (11)

Where \( \delta_i = \log \frac{1}{\beta_i} \).

From the training steps of the algorithm, we can see that each sample is given the same weight initially, and then the weights are updated every iteration, which improves the weight of the wrong classification samples and reduces the weight of the correct classification samples. According to the new sample data, a new weak classifier is obtained. Finally, the weak classifiers are integrated into strong classifiers by weighting.

4. Traffic sign detection based on Haar feature and AdaBoost classifier

Haar features can better describe the structure and edge features of traffic signs, AdaBoost classifier can form a strong classifier by constantly self-adjusting when the sample database is not ideal. Therefore, an traffic sign detection method based on Haar feature and AdaBoost classifier is proposed in this paper.

During training, Haar features are extracted from positive samples and negative samples respectively, and then are sent to AdaBoost classifier for training to obtain the classification model. During detection, Haar features are extracted from each frame of the video, and then is sent to AdaBoost classifier for detection, finally, the detected traffic signal area is labeled. The algorithm block diagram is shown in Figure 3.

![Algorithm Block Diagram](image)

Figure 3. Traffic sign detection based on Haar + AdaBoost

5. Experiment and analysis

In order to verify the effectiveness of the algorithm, 2244 training samples are collected in this paper, including 63 positive samples and 2181 negative samples. The positive samples size is normalized to \(24 \times 24\) before training. The proposed algorithm was implemented by VS2010+OpenCV2.4.3, and the running environment is Intel Core (TM) i3-2350@2.3GHz processor, and 4GB of RAM. Some experimental results are shown in Figure 4 and Figure 5.
As can be seen from Figure 4 and Figure 5, the method proposed in this paper can effectively detect
the traffic signs in the image, and can achieve multi-target detection. This is because Haar features can
better describe the image edge, texture and line features.

Table 1 shows detection time of some traffic signs using our proposed algorithm. It can be seen
from the Table 1 that the detection time is within 0.2 seconds, which can ensure the real-time detection. This
is because the integral image method is used in the calculation of Haar features, which speeds up the
calculation of features.

|                        | No-parking sign | No left-turn sign | No trucks sign | Pedestrian signs | Straight sign |
|------------------------|-----------------|-------------------|----------------|-----------------|--------------|
| first                  | 0.0393s         | 0.0907s           | 0.0485s        | 0.0995s         | 0.0235s      |
| second                 | 0.0397s         | 0.0601s           | 0.0545s        | 0.0961s         | 0.1213s      |
| third                  | 0.0525s         | 0.1372s           | 0.1971s        | 0.0515s         | 0.1676s      |

Table 2 shows the comparison results between our algorithm and reference [9]. It can be seen from
the Table 2 that our algorithm is more efficient. This is because each strong classifier of cascade
classifier can obtain almost all traffic sign samples and a large number of non traffic sign samples.
Although it takes a lot of time to detect traffic signs, it reduces the overall calculation time.

|                        | Our method | reference [9] |
|------------------------|------------|---------------|
| average detection time | 0.1213s    | 4.4654s       |

6. Conclusions
The traffic sign detection algorithm based on Haar feature and AdaBoost proposed in this paper can
effectively detect the traffic signs in the image, and can achieve multi-target detection. No matter what
the traffic environment and traffic signs, the better results can be achieved.
Acknowledgements
This work was supported in part by the Industry university research cooperation project in Jiangsu Province, China (No. BY2020068), Qing Lan Project of JiangSu Province, Opening Project of Henan Engineering Laboratory of Photoelectric Sensor and Intelligent Measurement and Control (Henan Polytechnic University) (No. HELPSIMC-2020-002). The authors would like to thank the anonymous reviewers and the editors for their many helpful suggestions.

References
[1] P. Viola and M. Jones. (2001) Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition. Kauai, HI, USA. pp. 1-9.
[2] T. Çağlayan, H. Ahmadzay and G. Kofraz, (2015) Real-time traffic sign recognition based on shape and color classification. In: Signal Processing and Communications Applications Conference, Malatya. pp.897-1900.
[3] T. Keser, G. Kramar and D. Nožica, (2016) Traffic signs shape recognition based on contour descriptor analysis. In: Smart Systems and Technologies. Osijek, pp.199-204
[4] M. T. Islam, (2019) Traffic sign detection and recognition based on convolutional neural networks. In: Advances in Computing, Communication and Control. Mumbai, India. pp1-6.
[5] D. Ciregan, U. Meier and J. Schmidhuber, (2012) Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition, Providence, RI, pp.3642-3649.
[6] C.Papageorgiou and T.Poggio. (2000) A Trainable System for Object Detection. International Journal of Computer Vision, Vol.38. pp.15–33.
[7] R. Lienhart and J. Maydt, (2002) An extended set of Haar-like features for rapid object detection. In: Image Processing, Rochester, NY, USA. Pp.900-903.
[8] Y.Freund and R.E.Schapire. (1996) Experiments with a new boosting algorithm. In: Machine Learning. Morgan Kaufmann, USA, pp.148-156.
[9] Y.Zhang. et al. (2014) The Traffic Sign Detection based on the visual saliency. Journal of Information & Computing Science. Vol.9,No.3, pp233-239.