Improved HOG Feature Vehicle Recognition Algorithm Based On Sliding Window

Yang Ji¹ and Jun Zhong¹,*

¹College of Electrical Engineering, Sichuan University, Wuhou District, Chengdu, Sichuan, China

*Email address: zhongjun55@163.com

Abstract. With the continuous development in recent years, machine learning has been extensively used in many application scenarios, especially in vehicle recognition. At present, the histogram of oriented gradients (HOG) algorithm has been widely used in vehicle recognition and facial recognition. This paper studies the HOG algorithm, introduces the vehicle recognition using the HOG algorithm, and introduces a detection method based on sliding window. In this research, an improved HOG algorithm based on sliding window is proposed for vehicle recognition and compared with the traditional HOG algorithm. In randomness test, the traditional recognition method often causes its HOG features to be trapped in a large number of invalid features due to the uncertainty of the vehicle distribution. This makes it difficult to be identified by the classifier, resulting in erroneous images, with an accuracy of only 83.33%. The optimized sliding window algorithm eliminates the influence of overlapping windows, and the number of accurately-recognized images is largely increased, with an accuracy of 91.02%.

1. Introduction
The technology of vehicle detection and recognition has become a hot issue in machine learning and even deep learning.[1] It has important application values in safety protection and traffic management, etc. For example, vehicle recognition from surveillance videos acts as an important technical basis to improve the traffic management.[2]

At present, the common vehicle recognition algorithms include YOLO algorithm, SIFT algorithm, HESSIAN algorithm, HAAR algorithm, ORB algorithm, and the forward vehicle recognition methods based on HOG features and SVM, etc. The algorithm based on HOG[3][4] features firstly conducts the feature extraction, and then uses SVM[5][6] classifiers to implement object detection, which reduces the spatial dimension and reduces the amount of calculation. Now, HOG algorithm has been widely applied to the implementation of classic machine learning.

However, the HOG algorithm[7] is time-consuming and labor-intensive in extracting the HOG features of images, and it cannot achieve fast and accurate detection. In view of the above problems, this paper uses a sliding window to optimize the traditional HOG algorithm for vehicle recognition.

2. The improved HOG algorithm based on sliding window for vehicle recognition
This paper proposes a HOG algorithm based on sliding window for vehicle recognition. Specifically, this algorithm first conducts the training using vehicles and non-vehicle objects by a SVM[8][9][10] classifier. Then, the image to be recognized is traversed by the sliding window. Next,
the HOG[11] features of each small window are loaded into the classifier for recognition. If it matches the training set, it is marked with a window. Finally, heatmap threshold filtering is applied to eliminate the influence of overlapping windows, and the vehicles are recognized with a high accuracy. The algorithm flowchart is shown in figure 1.

![Flow chart of the improved HOG algorithm based on sliding window](image1.png)

**Figure 1.** Flow chart of the improved HOG algorithm based on sliding window

3. Training samples

The selection of training set is also very crucial in the proposed algorithm, as the size and typicality of the training set determine the accuracy of the results by the SVM[12][13][14] classifier. The training data selected in this paper is a set of 64 × 64 × 3 RGB images from the front camera of the vehicle, including the two categories of images, with 8792 vehicle images and 8868 non-vehicle images. The camera is the Dianbin general-purpose front-facing camera, with pixels of 1080p, a screen size of 7-10 inches, and a shooting angle of 150°-180°.

These images are obtained from the front camera of the vehicle. All the images have a size of 64 × 64 pixels, and the images shot from different angles, distances, lighting conditions, and occlusion directions are selected, which can ensure the accuracy of the SVM classifier.

Some images of the vehicle and non-vehicle samples in the training set are presented in figure 2.

![Sample images](image2.png)

**Figure 2.** Sample images

After determining the sample images, the dimensions of the extracted HOG features are reduced, which can reduce the detection complexity and improves the accuracy. Then, the features are sent into the SVM classifier for training. Finally, the trained SVM classifier is obtained to provide support for subsequent detection.

4. Detection optimization based on sliding window

The receptive field of a sliding window is actually a three-dimensional block, meaning that the sliding window itself can be seen as a three-dimensional block.[15]

The implementation of the sliding window is to first convert the fully connected layer into a convolution layer, as shown in figure 3. Specifically, an image of 14 × 14 × 3 is input into the object
detection algorithm, the filter size is \(5 \times 5\), the number of the sliding windows is 16, the image map is \(10 \times 10 \times 16\) and then pooled to \(5 \times 5 \times 16\), two fully connected layers that connect 400 units are added, and finally Y is output through softmax.

**Figure 3** Flowchart of performing a sliding window convolution on a fully connected layer

The method to convert the above fully connected layer to convolution layer is to use 400 filters with size of \(5 \times 5 \times 16\)to convert the image to \(1 \times 1 \times 400\). Mathematically, this is equivalent to the 400 units of the fully connected layer. Similarly, the same method is used to implement the conversion from the fully connected layer to the convolution layer, with the flowchart shown in figure 4.

**Figure 4**. Flowchart of converting a sliding window convolution into a convolution layer

In the testing set, it is assumed that the actual size of an image is \(28 \times 28\), and the convolutional network operation is performed on the entire large image. As shown in figure 5, the entire image is divided into 64 blocks. Each part of the \(8 \times 8\) matrix corresponds to the position of the window scan in the large matrix.

**Figure 5** Flowchart of performing a sliding window convolution

The main idea of sliding window method is elaborated as follows. First, sliding the input image with different window sizes from left to right and from top to bottom. The classification by a classifier is performed on the current window every time sliding is conducted (the classifier is trained in advance). If the current window gets a higher classification probability, the object is considered to be successfully detected. After detecting each sliding window of different sizes, object marks detected by different windows are obtained, and there are overlapping parts in these window sizes. Finally, non-maximum suppression (NMS) is employed for screening and the detected objects are obtained after NMS screening.

In this paper, it is time-consuming to identify HOG features directly from the entire image, and it is difficult to make highly accurate predictions due to the randomness of vehicle distribution. So, referring to the optimization of face recognition algorithm, the following improvements are made to the algorithm.

Four types of sliding windows with different sizes are employed to search for vehicles in the image, and then the HOG features belonging to each window are obtained.

The sizes of sliding windows are presented in Table 1.
### Table 1. The sizes of siding windows

| Type   | Size     | Usage                                      |
|--------|----------|--------------------------------------------|
| Type 1 | 64×64    | Detect vehicles in the long distance       |
| Type 2 | 96×96    | Detect vehicles in the medium distance     |
| Type 3 | 128×128  | Detect vehicles in the short distance      |
| Type 4 | 224×224  | Detect vehicles in the very short distance |

In this way, it is possible to extract features from the windows of vehicles from far to near. This operation simplifies the extraction process, saves the time, and improves the accuracy.

The results obtained by implementing the sliding window operation are shown in figure 6.

![Test image 1](image1.png) ![Test image 2](image2.png) ![Test image 3](image3.png)

(a) Test image 1  (c) Test image 2  (e) Test image 3

(b) Results of sliding window detection  (d) Results of sliding window detection  (f) Results of sliding window detection

**Figure 6.** Results of implementing the sliding window operation in the testing images

It can be seen from the figure that there exits the phenomenon of multi-window overlapping and false positive detection. To eliminate these existing errors, a filter is employed, such as detection of heatmap filtering, so as to optimize the basic vehicle recognition algorithm.

### 5. Eliminating errors filtered by heatmap and threshold

Due to the use of multiple sliding windows with different sizes, a single vehicle image may be captured and detected by multiple windows. Therefore, all the positive detections on an image are recorded to form a detection heatmap. Finally, we perform threshold filtering on the heatmap to filter the erroneous detection results. Thus, the ultimate goal of eliminating errors with NMS is achieved.

The results obtained by applying the heatmap and threshold filtering are shown in figure 7 and figure 8.

![Test image 1](image4.png) ![Test image 2](image5.png) ![Test image 3](image6.png)

(a) Test image 1  (c) Test image 2  (e) Test image 3
6. Results and discussion

In this paper, sliding window method is employed to optimize the vehicle recognition algorithm. In the improved algorithm, four different sliding windows are used to divide the image. These images are detected into small windows of different sizes, so as to achieve more accurate detection of randomly-distributed vehicles.

This paper performs random screenshot detection on the 50s video in the vehicle's driving recorder. A total of 78 frames are taken. The traditional method and the optimized algorithm are used for detection, and their accuracies are compared in Table 2.

Table 2. The correct detection rate of two methods

| Method                        | Number of accurately-recognized images | The number of images with errors | Accuracy  |
|-------------------------------|---------------------------------------|---------------------------------|-----------|
| The optimized sliding window algorithm | 71                                     | 7                               | 91.02%    |
| The traditional recognition algorithm | 65                                     | 13                              | 83.33%    |

Table 2 indicates that the traditional recognition method often causes its HOG features to be trapped in a large number of invalid features due to the uncertainty of the vehicle distribution. This makes it difficult to be identified by the classifier, resulting in 13 erroneous images, with an accuracy
of only 83.33%. The optimized sliding window algorithm proposed in this paper eliminates the influence of overlapping windows, and the number of accurately-recognized images reaches 71, with an accuracy of 91.02%. This is greatly improved compared with the traditional recognition algorithm. Therefore, the optimized results are more accurate than the results by the traditional recognition algorithm.

7. Conclusion
This paper proposes a HOG algorithm based on sliding window for vehicle recognition. After the training by the basic classifier, the image to be recognized is traversed by four sliding windows of different sizes. Next, the HOG features of each small window are loaded into the classifier for recognition and the matched vehicles are labelled. Finally, heatmap threshold filtering is applied to eliminate the influence of overlapping windows, so as to improve the recognition accuracy. The final accuracy reaches 91.02%.

Reference
[1] Gao Jinghong, Yang Yimin. A Comprehensive Review of Inspection Technology and Development of Road Traffic Vehicles [J]. Highway Traffic Technology, 2012 (1): 116-119
[2] BLOSSEVILLE J M. Titan: A Traffic Measurement System Using Image Processing Techniques[C]. IEEE the 2nd International Conference on Road Traffic Monitoring, 1989: 84-88.
[3] DALAL N, TRIGGS B. Histograms of Oriented Gradients for Human Detection [J]. IEEE Computer Vision and Pattern Recognition (CVPR), 2005(1): 886-893
[4] Yu C, Joachims T. Learning structural SVMs with latent variables. Proceedings of 26-th International Conference on Machine Learning. 2009.
[5] CORTES C, VAPNIK V. Support vector networks J]. Machine Learning, 1995, 20(3): 273-297.
[6] Cheng-Lung Huang, Chieh-Jen Wang. A GA-based feature selection and parameters optimization for support vector machines[J]. Expert Systems With Applications. 2005 (2).
[7] Y. Pang, Y. Yuan, X. Li, et al. Efficient HOG human detection [J]. Signal Processing, 2011, 91:773-781.
[8] Bradski G, Pisarevsky V. Open computer vision library version 2.1 [CP/OL]. http://opencv.willowgarage.com. 2010.
[9] Felzenszwalb, P. F, Girshick, R. B, McAllester, D, Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010.
[10] Assheton A, Hunter A. A Shape-based Voting Algorithm for Pedestrian Detection and Tracking. Pattern Recognition. 2010.
[11] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection. Proc. of IEEE Conference on Computer Vision and Pattern Recognition. 2005.
[12] N. Shanthi, K. Duraiswamy. A novel SVM-based handwritten Tamil character recognition system[J]. Pattern Analysis and Applications. 2010 (2).
[13] Smits G F, Jordaan E M. Improved SVM regression using mixtures of kernels. Proc of the 2002 Int Joint Conf on Neural Networks. 2002.
[14] Joachims T. Estimating the generalization performance of a SVM efficiently. Proceedings of the 17th International Conference on Machine Learning. 2000.
[15] Enda Wu, Machine Learning Tutorial