Application of machine learning approach in detection and classification of cars of an image

B. Ashwini*

Department of Information Science and Engineering, NMAMIT (Affiliated to V.T.U. Bellagavi), Nitte, Karnataka 574110, India
Email: ashwinib@nitte.edu.in

B.N. Yuvaraju

Department of Computer Science and Engineering, NIE (Affiliated to V.T.U. Belagavi), Mysuru, Karnataka, India
Email: yuvarajubn@nie.ac.in

Abstract: Support vector machine (SVM) qualified on histogram orientation gradients (HOG) features is a genuine standard across many visual awareness responsibilities. Due to the change in the illumination and scene complexity, moving vehicle detection has become one of the very important components. Therefore, in this paper, a HOG feature descriptor is proposed. HOG features are not perceptive to illumination change and performance is better in characterising object shape and appearance. A feature vector is built by combining all the HOG features, which are required to train a linear SVM classifier for classification of vehicles.

Keywords: classification; HOG features; machine learning; SVM classifier; vehicle detection.

Reference to this paper should be made as follows: Ashwini, B. and Yuvaraju, B.N. (2017) 'Application of machine learning approach in detection and classification of cars of an image', Int. J. Signal and Imaging Systems Engineering, Vol. 10, Nos. 1/2, pp.8–13.

Biographical notes: B. Ashwini working as an Associate Professor in Department of Information Science and Engg. at NMAMIT, Nitte, India. Her research interest is image processing, computer vision, data mining.

B.N. Yuvaraju is working as a Professor in Department of Computer Science and Engineering at NIE, Mysore, India. His research interest is data mining, adhoc networks.

1 Introduction

Traffic administration is an essential part of daily routine because of the enlarged traffic on the roads. A better traffic administration in active highways can be achieved by automatic vehicle detection and the extracted information from these detected vehicles. Various image processing techniques and sensors are used for monitoring and measuring the traffic flow in active highways (Javadzadeh et al., 2015).

For calculating the percentage of the number of vehicle classes that pass through highways, vehicle classification is essential. The use of robotic systems can lead to exact plan of pavements (e.g. the decision about thickness) with valid results in cost and quality. Even in metro areas, there is a need for data about vehicle classes that use a particular street. For a particular traffic scenario, the classification methods provide necessary data (Daigavane et al., 2011).

The importance of vehicle classification in various applications such as security system, traffic surveillance and congestion avoidance, accidents prevention, etc., has evolved as an important part of study. Several algorithms have been implemented for vehicle classification. Detection of vehicles from traffic videos differs from an algorithm to other. Linear support vector machines (SVMs) trained on histogram of oriented gradient (HOG) features are one of the very important procedures for vehicle detection and vehicle classification.

Classification can be generally divided into two methods: discriminative and generative methods. Discriminative classifiers have been broadly used in detection of vehicles, in which a study of decision boundary between two classes is done. Generative classifiers have been less common in the detection of vehicles, in which a study of underlying distribution of a given class is done.

The basic procedure for creating an object classifier is (Hsieh et al., 2006):

- Given an object, a labelled data set with images is collected.
Application of machine learning approach

- The data set is partitioned into training set and test set.
- The features extracted from the training set are used to train the classifier.
- The features extracted from the test set are used to test the classifier.

2 Literature review

Buch et al. (2009) proposed a 3D extended HOG feature for detection and classification of individual vehicles and pedestrians and is implemented using 3D interest points and HOG.

Dhawad and Itkarkar proposed an algorithm known as boosting HOG feature. For boosting HOG features, Adaboost classifier is used. These features detected are used for training a linear SVM classifier. Low altitude airborne platform framework is utilised. The linear SVM classification is used for the final car classification. The classification method is split into two parts which includes feature description and classifier training. One of the main disadvantages of HOG features is high dimensionality. Boosting HOG features overcomes the disadvantage by reducing the dimensionality. Hog features are in fact a histogram. The representatives of the weak classifiers are the bins of the histogram. To translate the classifiers from weak to strong, Adaboost training is necessary. These strong classifiers are collected to build a feature vector. The final stage is to use SVM classifier to get the final output of car detection.

Gupte et al. (2002) proposed an algorithm for vision-based detection and classification of vehicles in monocular image sequences of traffic scenes recorded by a stationary camera. Processing is done at three levels: raw images, region level and vehicle level. Vehicles are modelled as rectangular patches with certain dynamic behaviour. As the vehicles move through the sequence of images, correspondences between the regions and vehicles are established. To demonstrate the effectiveness of the algorithm, experimental results are carried out on several highway scenes. The user selects the image, extract features and for recovering the camera parameters using features an interactive camera calibration tool is developed.

Hsieh et al. (2006) specify that vehicle classification is challenging task due to problems such as motion blurs, varying image resolution, etc. To solve these problems, many algorithms have been proposed. Vehicle classification technique finds major application at toll plaza, traffic signal, etc. SIFT (Scale Invariant Feature Transform) algorithm is proposed to classify the moving objects, which improves the efficiency and reliability of the technique used for classification.

Ambardekar et al. (2014) proposed a system that aims to retrieve the vehicle classes and the vehicles are classified by the concept of images having different geometric variations and photometric conditions. A neoteric descriptor is proposed contingent on trace transform signature which extracts prominent and non-correlated information of the image. The authors also proposed a metric which is used to measure the complexity of vehicle shape based on corner point detection. The result measures the vehicle shape complexity. Three different cameras are used for classifying four different vehicle classes scoring an accuracy of 97.5%.

3 Methodology

The vehicles are detected using HOG feature descriptor and classified using SVM classification method.

3.1 Histogram of oriented gradients

In HOG, first an image is divided into small square cells and gradient histogram is taken, a block-wise pattern is used for normalising the result for each cell and a descriptor is returned. For object detection, an image window descriptor could be used for stacking cells into squared regions in an image, like in SVM classifier.

Syntax and description:

\[
\text{features} = \text{extract HOG Features (I)};
\]

Input image is I. This function returns local shape information. These data are used for detection, tracking and classification.

\[
[\text{features}, \text{validPoints}] = \text{extract HOG Features (I,points)};
\]

Histogram of oriented gradient descriptor can be computed using \(8 \times 8\) cells within the detection window as shown in Figure 1.

**Figure 1** The red box is the detection window (see online version for colours)

Within the image, the gradient vector at each pixel is computed. In all, 64 gradient vectors are chosen (in out \(8 \times 8\) pixel cell) and inserted them into a nine-bin histogram. The range of histogram lies between 0° and 180°, so there are 20°/bin as in Figure 2.

**Figure 2** Histogram of \(8 \times 8\) cell (see online version for colours)
For each gradient vector, its contribution to the histogram is given by the magnitude of the vector. The contribution between the two closest bins is divided. So, for example if a gradient vector has an angle of 85°, addition of one-fourth of its magnitude to the bin centred at 70°, and three-fourth of its magnitude to the bin centred at 90. The gradient histogram is a form of quantisation, where in this case 64 vectors with two components each down to a string of just nine values (the magnitudes of each bin) are reduced.

Performance of the classifier is evaluated by compressing the feature descriptor. The main aim is to generalise the contents of the 8 × 8 cell. If the contents of the 8 × 8 cell are deformed slightly, then, it might still have roughly the same vectors, but they might be in slightly different positions within the cell and with slightly different angles. The histogram bins allow for some play in the angles of the gradients, and certainly in their positions.

Increase in the brightness of the image by increasing the pixel values is not a problem because additional values are eliminated while finding gradient values. If the image contrast is increased, the pixel values are multiplied with a factor. This causes a change in the gradient values. But the object in an image may not be detected.

If we take a vector and multiply it by a factor, its magnitude is increased. But its direction is unchanged. By finding the unit vector, we find vector multiplied by a factor is identical to its original form.

The same concept is applied to the gradient values. First, group the cells into blocks. The blocks used by Dalal and Triggs consisted of two cells by two cells. The blocks have ‘50% overlap’, which is best described through Figure 3a and b.

Figure 3 (a) and (b) Overlap of blocks (see online version for colours)

By concatenating the histograms of the four cells within the block into a vector with 36 (4 histograms × 9 bins per histogram) components, block normalisation is performed. To normalise the components, vector is divided using its magnitude.

The result of the block overlap is that each cell will appear multiple times in the final descriptor, but normalised by a different set of neighbouring cells. (Specifically, the corner cells appear once, the other edge cells appear twice each and the interior cells appear four times each.)

3.2 Support vector machine classifier

Support vector machine is used when the data are separable, non-separable and non-linear transformation with the kernel. The data are classified using SVM by identifying the most suited hyperplane which distinguishes the data points of one class with the other. The data points nearest to the separating hyperplane are the support vectors.

In Figure 4, ‘+’ indicates data points of type 1, and ‘−’ indicates data points of type −1.

Figure 4 Representation of linear SVM

Figure 5 contains two sets of points.

Figure 5 Sample graph (see online version for colours)

For a linearly separable set of 2D points which belong to one of two classes, a separating straight line is found. A criterion is defined to estimate the worth of the lines. A line is bad if it passes too close to the points because it will be noise sensitive and it will not generalise correctly. Hence, in order to minimise noise, the lines that are crossing each other must be as far as possible.
Then, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVM’s theory. Therefore, the optimal separating hyperplane maximises the margin of the training data.

Optimal hyperplane can be computed as in Figure 6.

Figure 6  Optimal hyperplane (see online version for colours)

Let us introduce the notation used to define formally a hyperplane $H_0$:

$$w \cdot x + b = 0$$

where $x$ is a point on the plane and $w$ is the normal vector of plane.

Here, two other hyperplanes $H_1$ and $H_2$ can be selected which also separates the data and has the following equations:

$$w \cdot x + b = \delta$$

and

$$w \cdot x + b = -\delta$$

so that $H_0$ is equidistant from $H1$ and $H2$. Here, the variable $\delta$ is not necessary. To simplify the problem set $\delta = 1$.

$$w \cdot x + b = 1$$

and

$$w \cdot x + b = -1$$

Initially, hyperplane is not selected, later hyperplane that meets the following two constrains is selected:

Either:

$$w \cdot x + b \geq 1 \text{ for } x \text{ having the class 1}$$

or

$$w \cdot x + b \leq -1 \text{ for } x \text{ having class -1}$$

The hyperplane equation can be expressed concisely as

$$| w \cdot x + b = 1 |$$

where $x$ symbolises the training examples closest to the hyperplane. In general, the training examples that are closest to the hyperplane are called support vectors. This representation is known as the canonical hyperplane.

The result of geometry that gives the distance between a point $x$ and a hyperplane $(w, b)$ is used:

$$\text{Distance} = | w \cdot x + b | / || w ||$$

In particular, for the canonical hyperplane, the numerator is equal to one and the distance to the support vectors is:

$$\text{Distance} = | w \cdot x + b | || w || = 1 / || w ||$$

The margin is twice the distance to the closest support vector.

$$M = 2 / || w ||$$

Maximising the margin is same as minimising the norm of $w$.

This gives the following optimisation problem:

Minimise in $(w, b)$

$$|| w ||$$

subject to $y_i (w \cdot x_i + b) \geq 1$, (for any $i = 1, \ldots, n$).

4 Experimental results

Initially, two sets of images are considered, those containing vehicles and those without containing vehicles. These are categorised as training images. Training images have the same size. HOG features are extracted from the training images, and stored in a matrix called the training matrix, with each row containing HOG features of a training image. Create a label array, where its size is equal to the number of rows in training matrix. Consider a labelling scheme where the images with cars are labelled 1 and images without cars are labelled $-1$. Hence, by taking all the images in the same order as we extracted their HOG features, labelling them and saving the labels in the label array. Then by taking the training matrix and the label array which is needed to train the SVM to calculate the optimal hyperplane. Classifier is used on test images.

Test images are several times bigger than the training images. First, a test image with the cars being approximately as big as the ones in the training image is taken. To detect the car in the test image, we use a concept called the sliding window. The sliding window is a window equal to the size of the training images. Test image is seen through this window, the contents that are visible in the window are copied and HOG features are extracted from it. Trained SVM classifier is used to predict whether the features belong to the car group or not. If the result is 1, it means the image belongs to label1, which means a car has been detected in the test image. Rectangular box is drawn to show the detected car. If the result is $-1$, then no car has been detected and the window slides to the right. The process is repeated till the window scans the entire image.

As shown in Figures 7 and 8, Figure 7 is the test image which contains two cars and Figure 8 shows the result of detected cars in the same image. The window detects the same car many times, so there will be many bounding boxes.
Figure 7  Test image

Figure 8  Classifier detects vehicles in the test image

Figure 9 shows the detection of partially occluded image of car, which proves that the algorithm is good enough to detect partially occluded vehicles.

Figure 9  Partial occluded image detection

False positives and false negatives will occur as shown in Figures 10 and 11, but it is possible to rectify them. Window containing the false detection can be cropped and put the image in the appropriate training image set as shown in Figures 12 and 13.

Figure 10  False positive detected

Figure 11  False negative

Figure 12  False positive corrected by putting the cropped falsely detected part in positive training set

Figure 13  Car detected after putting cropped falsely undetected part into the negative training set

Table 1 gives the comparison of manual counting and algorithm count of sample images. Three data sets of 10 images were considered for the algorithm. We could observe that algorithm was nearly 90–95% accurate in detecting the vehicles. The undetected vehicles are at the edges of the images and are occluded.
Table 1  
Comparison of manual count and algorithmic count

<table>
<thead>
<tr>
<th>No. of sample images</th>
<th>Manual counting</th>
<th>Algorithm count</th>
<th>False positive</th>
<th>False negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>31</td>
<td>29</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>26</td>
<td>26</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>21</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

5 Conclusion

Research work was mainly focused on detecting vehicles in images and classifying them. Accordingly, the HOG features of images are determined and given as an input to SVM classifier. Algorithm detects the vehicles in an image. Even partially occluded images were also detected in the image. To avoid false positives and false negatives, the false-positive image from the detected image is cropped and added into the negative training set and false-negative image is cropped and added to positive training images. An algorithm result shows that it is 99% matching with the manual count. Algorithm failed only for the largely occluded vehicles. It can be fine-tuned to eliminate false detections.

References


