An Evaluation of Boosted Features for Vehicle Detection

Liwei Liu, Genquan Duan, Haizhou Ai
Computer Science and Technology Department,
Tsinghua University, Beijing, China
ahz@mail.tsinghua.edu.cn

Shihong Lao
OMRON Corporation,
Kyoto 619-0283, Japan
lao@ari.ncl.omron.co.jp

Abstract—Vehicle detection in traffic scenes is a fundamental task for intelligent transportation system and has many practical applications as diverse as traffic monitoring, intelligent scheduling and autonomous navigation. In recent years, the number of detection approaches in monocular images has grown rapidly. However, most of them focus on detecting other objects (such as face, pedestrian, cat, dog, etc.) and there lacks vehicle datasets with various conditions for vehicle detection and comprehensive comparisons. To address these problems, we perform an extensive evaluation of many state-of-the-art detection approaches on vehicles. Our main contributions are: (1) we collect a large dataset of real-world vehicles in frontal/rear view with $30^\circ \sim -30^\circ$ yaw changes and $5^\circ \sim 45^\circ$ pitch changes under different weather conditions (snowy, rainy, sunny and cloudy) and illumination variations, and then (2) we evaluate six types of state-of-the-art features in Real AdaBoost framework on the adequate dataset collected by ourselves and a public dataset using the same evaluation protocol. Our study presents a fair comparison and deep analysis of these features in vehicle detection. From these experiments, we explore the characteristics of good features for vehicle detection. (3) Finally, we exploit these characteristics and propose a relatively effective and efficient detector, balancing performance, speed and memory cost which can be put into practical use.

I. INTRODUCTION

Vehicle detection in traffic scenes is of fundamental importance for surveillance system and has apparent commercial value. It exploits robust observation models for vehicle tracking which provides great potentials for many high level computer vision applications such as intelligent scheduling, traffic analysis, abnormal trajectory detection and collision avoiding. Due to its wide spread of potential applications, considerable attentions have been attracted in building automated vision systems for detecting vehicles in recent decade.

Many traditional works [1][2] of vehicle detection depends on background subtraction, which are sensitive to lighting variations. They also assume that vehicles appear in the scene without occlusions for a while to build up proper object models, which may encounter great challenges in crowded scenarios. In contrast to these works, machine learning based vehicle detection techniques are more robust and reliable. But they ([3][4][5]) have a short history compared with face [6][7][8][9][10] and human detection [11][12][13][14][15] and only spring up several years ago. Yet despite its apparent importance, the lack of comprehensive evaluations of existing popular detection techniques on vehicle detection, strongly limits the popularity in practical applications. The main reason is the lack of datasets with various conditions for vehicle detection in traffic surveillance scenes.

In this paper, we propose to evaluate many state-of-the-art detection approaches for vehicle on a relatively adequate dataset collected by ourselves and a public dataset. Our target is to evaluate their effectiveness and efficiency on vehicle detection and attempt to build up a detector for real world applications with well balances of speed, performance and memory cost. The most related work is [16], evaluating state-of-the-art approaches in pedestrian detection. However, not all of their performances are similar on vehicles due to the characteristics of vehicles, for example, vehicles differs much more with viewpoint changes than pedestrians as shown in Fig. 1. A good news is that since vehicles are rigid, their difference in a similar viewpoint is not huge. Therefore, the proposed evaluation on vehicles is of great significance.

In the literature, there are a number of detection approaches, whose key components are the training framework and weak features. In this paper, we do not intend to investigate various training frameworks (Boosting [17], Support Vector Machine (SVM) [11], Random Forest [18] and Latent SVM [19]), but only focus on evaluating the effectiveness of different weak features under the same training framework. Considering the speed, performance and memory issues, the selected training framework is Real AdaBoost, which has significant advantages in speed and is robust and reliable in binary classification problems as reported in [6][7][8][9][4][12]. The selected weak features can be roughly classified into grayscale based [6][10] and gradient based [11][13][14], which have achieved promising results in face and pedestrian detection respectively.

Our main contribution can be summarized in three folds.
(1) We collect a large dataset of vehicles from real-world for training and testing. The viewpoint is frontal/rear with
30° ∼ −30° yaw changes and 5° ∼ 45° pitch changes as shown in Fig.1, which approximates many cameras along the road. Note that, the side view of vehicles is not considered here. (2) We perform an extensive evaluation of many state-of-the-art features in the Real AdaBoost framework using the same evaluation protocol on vehicles detection. And then provide a fair comparison and discussion of these features. (3) Finally, we exploit the most effective and efficient way and learn a vehicle detector for practical applications.

The rest of this paper is organized as follows. Boosting is briefly introduced in section II. In section III, we describe the state-of-the-art features for evaluation. And then implementation details will be provided in section IV. Finally, we will report the results of the performance evaluation, give a deep analysis and propose the most valuable detector for practical applications in section V. Conclusions are made in section VI.

II. BRIEF INTRODUCTION OF BOOSTING

In this section, we briefly summarize some characteristics of AdaBoost, the training framework we adopt to train the vehicle detectors with various features. Let

\[ \mathcal{H} = \{ h : \chi \rightarrow \mathcal{R} \} \] (1)

be the family of “weak classifiers” where \( \chi \) is the sample space, and

\[ F(x) = \sum_{i=1}^{T} \alpha_i h_i(x) \] (2)

be the set of linear combinations of weak-classifiers \( h_i(x) \) which is learnt in the training phase. So the strong classifier can be presented as:

\[ H(x) = \text{sign}(F(x) - b) = \text{sign} \left( \sum_{i=1}^{T} \alpha_i h_i(x) - b \right) \] (3)

with which a mapping from samples \( x \) to class label can be obtained.

We can define the exponential loss

\[ L_{exp}(F(x)) = \sum_{i=1}^{n} w_i \exp(-y_i F(x_i)) \] (4)

which is low when \( F(x_i) \) take values consistent with labeling \( y_i \) (\( n \) is the number of training samples). AdaBoost consists in approximating the mapping

\[ F^* = \arg \min_{F} L_{exp}(F(x)) \] (5)

by successively picking weak classifiers \( h \) and weights \( w \) in the way of reducing \( L_{exp} \) greedily.

III. STATE-OF-THE-ART FEATURES

The previous section has introduced the training framework. Given the weak features, we propose to learn detectors in the Nested Cascade Structure [8] using the framework presented in the previous section. The learning procedure can be found in [17]. In this section, we describe the selected weak features in detail as follows.

A. Haar-like Features (Haar)

Haar features are firstly used in [6] for face detection. Haar features consist of several adjacent rectangles with the same sizes, usually represented in white and black as shown in Fig. 2. The feature value is the absolute difference of the sum of pixels in white regions and black ones. As proposed in [6], Haar features can be efficiently calculated through integral image, where the integral image \( \tilde{I} \) of gray scale image \( I \) can be denoted as:

\[ \tilde{I}(u,v) = \int_{x=0}^{u} \int_{y=0}^{v} I(x,y) dxdy \] (6)

B. Joint Sparse Granular Features (JSGF)

For effective training, Haar features must be devised and enumerated before training [6], which lacks of flexibility and cannot guarantee the most discriminative feature selected in each weak classifier. To address these problems, Huang et al. [10] proposed Joint Sparse Granular Features (JSGF). A JSGF is a Boolean comparison of a pair-wise granules, where granules are square window patches as defined in Granular Space [9]. In formal, a JSGF containing \( n \) pairs of granules can be defined as:

\[ z = f_{JSGF}(x) = [b_1 b_2 ... b_n] \] (7)

where

\[ b_i = g_{i+}(x) \geq g_{i-}(x), i = 1, ..., n \] (8)

g_{i+}(x) and g_{i-}(x) are the \( i \)th pair of granules (in Fig. 3).

Compared with Haar features, JSGF is more discriminative because of the arbitrary combinations of granules and gets rid of integral images with lower computations. To some extent, JSGF is invariant against brightness and contrast changes.
C. Histogram of Oriented Gradient Feature (HOG)

As a well-known gradient based feature, HOG [11] has been proven to be effective for human detection. This feature is based on evaluating well-normalized local histogram of image gradient orientations in a dense grid. In practice, first divide the image window into small spatial regions (cells), and then accumulate a histogram of gradient directions or edge orientations for each cell. Therefore the histogram for each cell:

\[ G_c = (g_c(1), g_c(2), ..., g_c(n))^T \]  

where \( n \) is the dimension of the HOG (in [11], \( n = 9 \)), and \( c \) is a cell in an image. Finally normalize the local responses for better invariance to illumination:

\[ g'_c(n) = \frac{g_c(n)}{\sum_{i \in D} g_c(i)} \]

D. Extended Histogram of Oriented Gradient (EHOG)

In order to extend the description power of HOG, Hou et al. [13] proposed the Extended Histogram of Oriented Gradient (EHOG), adding neighboring bins in the original HOG (1, 2 and 3 bins) as new features to represent the edges as dominant orientations. Therefore, there are totally 27 (=3*9) different dominant orientations, namely 27 weak features, for each cell, which enhanced the description capability of gradient features.

E. Adaptive Contour Feature

Other than directly extending HOG like [13], Gao et al. [14] proposed another way to approximate contours through a set of locations with similar gradients. The so-called feature Adaptive Contour Features (ACF) is based on Oriented Granular Space (OGS) as shown in Fig. 6, which adds orientation information to granules as defined in Sec.B. A ACF is a set of Linked Granules (LG), termed Linked Granules Set (LGS) in [14], where a LG is a chain of a number of oriented granules to describe a continuous contour. There are two types of the value of LG, accumulated edge strengths and relative strengths, in which the relative strengths are normalized ones by the total strengths in the involved granule.

F. Associated Pairing Comparison Feature (APCF)

The previous five kinds of features focus only on either features in grayscale like [6][10] or in gradient like [11][13][14], while these two categories of features may complement each other, which is studied in [15]. Duan et al. [15] proposed a compositional feature, APCF, containing Pairing Comparison of Color (PCC) features and Pairing Comparison of Gradient (PCG) features. A PCC is a Boolean color comparison of two granules and a PCG is a Boolean gradient comparison of two granules (The definition of granule is as in Sec.B). PCC is JSGF in essence and PCG is devised for capturing edge fragments of positive samples (as shown in Fig. 7, large granules for coarse edges and small ones for fine edges). To some extent, APCF describes invariance of color and gradient of an object.

IV. IMPLEMENTATION DETAILS

A. Training Detectors

We train several detectors in the same framework as described in Sec.II. In the training procedure, we set the passed false alarm rate as 0.33 and the passed positive rate as 0.998 and the maximum number of weak classifiers is 100 in each layer, where each layer is a strong classifier and all strong classifiers will form the last detector (Note that the adopted detector structure is Nested Cascade [8]). In addition, all sample size is \( 24 \times 24 \). That is to say, the only differences of these detectors are involved weak features, whose details are described as follows:
Training Data
(a)
Our Testing Dataset
(b)
Public Dataset: i-Lids
(c)
Fig. 8. Some examples of the training data (a) and the test images ((b) our collected dataset and (c) i-Lids dataset).

55,406 positive samples as shown in Fig. 8(a) for training and collect 501 frames for testing which cover the challenges of viewpoint variations, illumination variations and occlusions regularly and frequently in practical applications. The testing dataset is independent from the training dataset.
i-Lids dataset [20]. i-Lids dataset is a public dataset for event detection. Three videos of i-Lids are used to detect vehicle parking event in traffic surveillance scenes. We select 187 images from them and manually label groundtruths for testing vehicle detectors.

B. Comparison

In practical applications of vehicle detection, considerable attentions have been focused on performance effectiveness, time efficiency and memory cost of detectors. Therefore, in this sub-section of experiment, we will compare these state-of-the-art features from performance, speed and memory cost and then give a deep analysis of performance.

We employ the PASCAL protocol for evaluation. When the intersection between a detection response and a groundtruth box is larger than 50% of their union, we consider it to be a successful detection. Only one detection per annotation is counted as correct. Given \( N_d \) merged detected responses and \( N_g \) groundtruths in \( N_t \) testing images, when there are \( N_m \) matched detections to groundtruths, the detection rate (recall) = \( \frac{N_m}{N_g} \) and false alarm per image (FPPI) = \( (N_d - N_m) / N_t \).

Performance. The ROC curves are shown in Fig. 9 (a) and (b). We can see that gradient type features, ACF, EHOG and HOG have similar performances which are much better than Haar features. JSGF outperforms both of Haar features and these gradient type features. With an efficient fusion of JSGF and PCG which capture contour fragments of objects, APCF achieve the most satisfactory performances. In general, JSGF and APCF can improve the detection rate by about 8% on our dataset and 15% on i-Lids at FPPI=1.

Speed. The detection speeds of these features are compared in Tab. 1. Compared to the gradient based features (HOG, EHOG and ACF), the grayscale based features (Haar, JSGF) and the compositional feature (APCF) are much faster (about 3x~8x). In practical applications, taking advantage of Particle Filter, JSGF can process in real-time, more than 22 frames per second on VGA size (640×480) [5].

Memory costs. The memory costs of these features are listed in Tab. 2. Supposing that the image resolution is M×N: since the data type of integral image is float (1 float = 4 bytes), Haar needs 1*4+1=5 M×N bytes (an integral image and a source image). The similar to HOG and EHOG, they
Fig. 9. Performance of detectors trained with state-of-the-arts features: (a) on our dataset; (b) on i-Lids dataset; (c) loose restrictions experiments.

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>SPEED COMPARISONS (ms).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haar</td>
<td>HOG</td>
</tr>
<tr>
<td>640 × 360</td>
<td>161</td>
</tr>
<tr>
<td>720 × 576</td>
<td>307</td>
</tr>
</tbody>
</table>

Taking all these three factors into account, we can easily find that JSGF and APCF are both fast and highly efficient in detection with low computation costs. In order to achieve positive motivations from these features for vehicle detection in practice, we next present a deep analysis of these features on the classification power and effective characteristics.

Classification Power. The more classification power of a feature, the more possibility of the application in real world. A reasonable assumption is that the less number of weak classifiers in a layer of detectors indicates the more classification power because detectors are trained under the same framework with the same setting. Therefore, we compare the number of weak classifiers in each layer of these detectors in Fig. 10. From it, we can see that (1) the three detectors Haar, HOG and EHOG cannot meet the requirement of reducing the false alarm to 0.33 and are terminated when the numbers of weak classifiers reach 100 since the 5th, 6th and 8th layers respectively; and (2) JSGF and APCF can always achieve similar false alarms with the fewest features.

Effective Characteristics. What characteristics can improve the performance of a feature for vehicle detection? We first list some possible characteristics:

- Contour fragment approximation;
- Loose restrictions on features.

and in the next we will show the verifications in details.

For the former one, we can compare ACF with HOG/EHOG and APCF with Haar/JSGF respectively. Although ACF, HOG/EHOG all are gradient based features, ACF achieves higher performance than them as shown in Fig. 9 and is more discriminative than HOG and EHOG as shown in Fig. 10. The main reason may be that ACF can approximate a contour while HOG and EHOG do not. Moreover, as introduced in Sec.III.F, APCF adds gradient information (PCG) into JSGF and PCG can capture edge fragments to some extent. Therefore, these verify the first characteristic.

For the second one, we compare JSGF/APCF with Haar/ACF/HOG/EHOG. The former two features compare the color or gradient in two granules each time, where the distance of a pair of granules is loose restricted, while the basic of the latter four features generally consider the color difference or the gradient of a region each time, where they are calculated through adjacent regions. Thus, we assume the distance restriction is one factor that impacts feature performances. In addition, the granules and rectangles should be the same scale in one LG of a ACF and in a Haar respectively, while they can be different for JSGF and APCF. Therefore, we assume that the scale restriction is another factor to impact feature performances. To know the impactions of these two factors, we take JSGF as an example and train two simplified version JSGF detectors: (1) granules in a comparison pair are adjacent, and (2) granules in a comparison pair are in the same scale. As compared in Fig. 9(c), we can see that the performances of these two simplified versions drop. Then, these verify the second characteristic.

C. Towards vehicle detection

From the experiments in Sec.B, we can come to the conclusion that APCF achieves the most satisfactory performance for vehicle detection. However, its speed is relative slower than JSGF/Haar and its potential is not fully mined. For handling the shortcoming, we train a speedup APCF.
using PCC(JSGF) in first 7 layers (coarse training stages) and both PCC(JSGF) and PCG in the rest layers (fine training stages). From Fig. 9 (a) and (b), we can see that Speedup-APCF achieves similar performance as APCF, and then in some ranges, even improve the performance. The speed is 221ms for 640×360 images (APCF 281ms, JSGF 187ms) and 395ms for 720×576 images (APCF 490ms, JSGF 329ms) which is about 20% faster than APCF. The memory cost is the same as APCF. As a consequence, Speedup-APCF is the most proper feature for vehicle detection, which balancing performance, speed and memory cost very well. Some results of Speedup-APCF are compared with JSGF in Fig. 11. From Fig. 11, we can see that JSGF may lose effectiveness when the appearance of an object is affected by local illumination variation because some Boolean color comparison results will change. But it can be remedied by PCG since the gradient information keeps invariant basically. For further improvements, we can conclude that: 1) in traffic surveillance scenes, the backgrounds are consistent generally so color comparison based features are effective to detect vehicles in these cases; 2) Since there are viewpoint and scale variations of discriminative components of vehicle, loose scale and range restrictions are essential characteristics; 3) Besides the above, discriminative contour fragment approximation can further improve the performance.

VI. CONCLUSION

In this paper, we first collect a large dataset of real-world vehicles in frontal/rear view under different weathers and illumination variations for vehicle detection and we intend to make it public available for academic use on request. And then based on extensive experiments, we analyze the existing state-of-the-arts features in the framework of Real AdaBoost algorithm thoroughly and provide a fair comparison and discussion of these features. Finally, in consideration of important factors in practical applications, we propose Speedup-APCF which is fast in speed and low cost in memory with very high accuracy that outperforms the other existing state-of-the-art features.

REFERENCES