IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2017, accepted January 27, 2018, date of publication February 8, 2018, date of current version March 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2803160

Pedestrian Detection by Feature Selected Self-Similarity Features

XINCHUAN FU[1], RUI YU[2], WEINAN ZHANG[3], LI FENG[4], AND SHIHAI SHAO[1], (Member, IEEE)

[1]National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Department of Computer Science, University College London, London WC1E 6BT, U.K.
[3]Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China
[4]Engineering and Technology College, Sichuan Open University, Chengdu 610073, China

Corresponding author: Xinchuan Fu (xinchuan.fu@foxmail.com)

This work was supported by the National Natural Science Foundation of China under 61771107.

ABSTRACT This paper is concerned with the pedestrian detection problem. In this area, boosted decision tree (BDT) methods are highly successful and very efficient. However, to achieve the best performance, most BDT methods require a large number of input features, which make the algorithm scale poorly to large-scale data. Inspired by the effectiveness of self-similarity (SS) features, we use linear discriminant analysis to select features in the SS features according to their generalized Rayleigh quotient, leading to a small number but most discriminative features. These features are called feature selected self-similarity (FSSS) features. The FSSS features are only used for the late stages of the BDT cascade, making the training and detecting much more efficient. Extensive experiments on four well-known data sets demonstrate that the FSSS features are highly effective and the trained pedestrian detector achieves state-of-the-art performance among all existing non-deep-learning methods on several benchmarks. We also compare our method with deep learning methods and show its superiority in high-quality localization and will be a good complement to deep learning methods.

INDEX TERMS Boosted decision tree, linear discriminant analysis, pedestrian detection, self-similarity features.

I. INTRODUCTION

As a typical problem of object detection, pedestrian detection is an active research topic in recent years. There have been well established benchmark datasets [1]–[5] and a variety of methods published to address this problem [3], [6]–[9]. Great progress has been made in recent years [1], [10]. Most state-of-the-art methods formulate pedestrian detection as a binary classification problem, i.e. to classify all the candidate windows in the target image as pedestrian or not, followed by a non-maximum suppression. Various tools in machine learning have been used to solve pedestrian detection problem, such as sparse representation-based classifier (SRC) [11], support vector machine (SVM) [3], [9], [12], BDT [6], [7], [13] and deep learning [8], [14], [15], etc.

In recent years, BDT and deep learning methods prevail in this research area. Although currently the top performance in pedestrian detection is achieved by deep learning method, BDT methods remain highly competitive in this area. For a typical image in practical application, there are much more background patterns than pedestrian patterns and most of them are very easy to classify, like sky, road, grass, etc. This characteristic of pedestrian detection problem makes boosted cascades especially suitable for this task, as it rejects background patterns in early stages of cascades rather than use the same computation overhead for all the candidate windows. This strategy saves a lot of computation cost and accelerates detection [16]. In fact, many leading deep learning solutions use BDT as the classifier or as a region proposal [15], [17]–[19]. In this paper, we focus on the BDT methods.

Although decision trees are able to select the most discriminative features from the input feature pool, the discriminative ability of the features in the feature pool itself could have a great influence on the performance of the learned classifier. Most state-of-the-art detectors use feature maps created from CIE-LUV color channels, gradient orientation channels, and gradient magnitude channels [20]. These feature maps could be computed efficiently but using these features directly suffers from limited performance. Research [21]–[23] show that the discrimination of these simple features is raised significantly with some linear transformation at a relatively cheap computational cost, depending on the specific

transformation and the number of output feature maps. The type of transformation used are often explored with domain knowledge [21], [22] or by learning from data [23]. Recently, features extracted by deep convolutional neural networks (CNNs) have gained the best performance, but the extraction of these deep features suffers from expensive computational cost and often needs expensive devices like GPUs. On the other hand, handcrafted features are shown to be complementary to CNN features [24], [25].

The main contribution of this paper is to propose a new type of features which achieve a good tradeoff between efficiency and discrimination effectiveness. We call our proposed features as Feature Selected Self-Similarity (FSSS) features as they are similar to self-similarity (SS) features [26] but with feature selection scheme to decrease the dimensionality of feature space. Both types of features compute pixel differences of base features. While SS features consider all possible pairs of the base features, FSSS uses linear discriminant analysis (LDA) to select the most promising pairs from all the possible pairs. The techniques allow us to use more base features than SS. A BDT model is then built based on these FSSS features to perform pedestrian detection efficiently.

The experiments on four well-known benchmark datasets demonstrate that our method leads to the top performance when GPU is not available. We also find that under a stricter criteria, our method even outperforms the top deep learning methods, showing our method has better localization precision than deep learning methods. Hence, our method may serve as a good complement to deep learning methods.

The rest of this paper is organized as follows. We first give a comprehensive review of the feature extraction approaches in Section II. The FSSS features and some auxiliary techniques are described in Section III. Experiment results are given in Section IV. Finally, we conclude in Section V.

II. RELATED WORK

To tackle the task of pedestrian detection, many papers have been published with the focus on different aspects of this problem and the majority of them are about feature design. This is justified by the recent survey [10] which points out that feature design seems more important than the choice of classifier and is a consistent driver for the detection quality improvement.

The first popular feature for pedestrian detection is histogram of oriented gradients (HOG) [3]. Many pedestrian detectors adopt this feature [9], [27], [28]. Based on HOG, [20] proposed to use LUV color channels, gradient magnitude channel and 6 orientation channels as base feature maps and based on these maps, features are extracted by region sums which are efficiently computed using the integral image trick [16].

Later, [6] proposed Aggregated Channel Features (ACF) which aggregate cells of pixels in each feature channel, and then feature extraction is simplified as a single table lookup in the 10 channels. Although very simple, ACF performs well in practice [6]. Due to the advantages of ACF, many variants of these simple features were proposed in the last few years.

Most of the proposed new features can be treated as a linear transformation of ACF features. Some of the methods pre-compute feature maps by performing convolution on the original 10 channels using some filter banks. These are called filter channel feature in [7]. For example, LDCF [23] uses filter banks computed from PCA on local neighborhood features, SquaresChnFtr [29] uses square-shaped uniform filter banks of different sizes, and Checkerboard [7] contains 61 handcrafted filters. As these types of transformations are performed in a convolutional fashion, the same filter is applied in every position in a detection window. Thus there exists possible waste of computation, since some of the features might not be needed at detection time. InformedHaar [21] is a different type of features. Its filter banks are based on the average of many human shapes, thus different positions in a detection window use different filters and a feature is extracted just-in-time when it is needed by a tree node. It seems more economical, but the detection efficiency will be highly reduced if the transformation complexity is high and the feature extraction process is slow.

These linear transformations only exploit feature relationship in local neighborhood. On the other hand, [22] shows that it is effective to use non-neighboring features. This type of features is computed by taking the differences between non-neighboring rectangle regions. Like InformedHaar, it takes advantage of some prior knowledge of a typical pedestrian window, called appearance constancy and shape symmetry. The feature pool is generated exhaustively under some constraints (e.g., in the same horizontal) and then selected by decision trees. Because these features are tailored particularly for human, it cannot be transformed to other detection tasks directly. Moreover, the constraints on the feature pool may result in unnecessary limitations on possible features.

To compute region difference, an integral image [16] needs to be computed, and accessing a feature needs 8 vertex accessing and 7 add/minus operations, which has a high computational cost. An alternative method is pixel difference, which is much easier to compute and is also effective. Inspired by Local Binary Patterns (LBP) [30], the authors in [31] proposed several kinds of pixel differences in a local feature patch. The best one of them is called Total Pixel Differential Features (TPDF), which considers all the possible combinations of pixel differences in a local region. The combination number is $\binom{m}{2}$ where m is the number of pixels in the local region. Apparently, m cannot be set too large, otherwise the dimension of the computed feature vector will be too high which requires a large amount of memory and long training time. In TPDF, the authors only used a 5×5 region. Because many of the created features are correlated, these feature maps are highly redundant. The authors used LDA and principle component analysis (PCA) to perform dimension reduction, which results in extra computational cost.

The SS features [32] used in [25] and [26] can also be treated as a kind of pixel difference features. In fact, it is a

global version of TPDF, which means the corresponding pixel pairs used in the minus operations are not constrained in a local region but in the whole detection window. In various kinds of handcrafted features considered in [25], SS features are proved to be the best kind of features to trade off discrimination effectiveness and efficiency. In order to limit m, SS features need to shrink the original feature map of the detection window to a smaller one in which each pixel corresponds to a local mean of the original feature map. In [25], new feature maps only consider 72 pixels but still lead to 25,560 SS features. This suggests that some sorts of feature selection or dimension reduction may be beneficial, which is the main focus of our work.

Features used in BDT can be divided into two categories. One of them is called pre-computed features, where all the features are computed before they are fed into the BDT. This family includes CNN based features, Checkerboards, LDCF etc. Another family is called just-in-time (JIT) features, where a feature is computed only when it is needed to be evaluated by a tree node. This family includes haar-like features, InformedHaar features etc.

The advantage of pre-computed features is the computed features may be shared among different weak classifier and windows. But for cascade classifier, this method turns out to be inefficient. For example, there are more than 380,000 windows to be classified in a 480×640 image. But usually most spaces in an image are background area, and most of the windows will be rejected at very early stages of the cascade. For example, when using ACF detector with 4096 weak classifiers to detect pedestrians in Fig. 1(a), only a small fraction of the windows will pass the last stage, as shown in Fig. 1(b). In fact, after the first 20 weak classifiers, more than 99% windows are rejected. Thus only a small fraction of features may be used in BDT cascade while most of the feature computation overhead is wasted.

(a) (b)

FIGURE 1. Illustration of the number of weak classifier used in different position of a typical image. (a) A test image from the Caltech dataset [1]. (b) The heatmap of the weak classifier number used in every position of this image. Note in most positions, only a small number of weak classifiers are used.

For JIT features, because the feature extraction happens at every tree node, the complexity of feature extraction should not be too high, otherwise the computational cost is huge. For example, in [23] the authors used oblique splits learned by LDA at each node, which bring considerable computational expense.

Our FSSS features are computed just-in-time, and in detection time the feature extraction only involves two pixel indexing and a minus operation of these two pixels, the complexity is only a slightly higher than ACF features, which only involves a single pixel indexing. To further reduce computational cost, we put the FSSS features to the late stages of the BDT cascade, using the simpler ACF features at early stages as a proposal. After the ACF stages, only a small number of windows are left. These windows are tackled with more complex FSSS features.

III. THE PROPOSED APPROACH

As we stated in the last section, many types of features can be treated as a linear transformation of the ACF features. This is also true for our FSSS features. Throughout this paper, we use the 10 channel ACF features as our base features, but our method is able to extend to other types of feature maps.

Suppose we extract an n dimensional feature vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \Re^n$ in a detection window. The SS features x^{ij} are constructed by all the combinations of feature differences: $x^{ij} = x_i - x_j, 1 \leq i \leq n, i < j \leq n$. These feature differences can be taken as projections from the base feature vector to different projection vectors, that is $x^{ij} = \mathbf{w}^{ij^T}\mathbf{x}$, where $\mathbf{w}^{ij}$ is an n-dimensional vector with only two non-zero elements $\mathbf{w}^{ij}_i = 1$, $\mathbf{w}^{ij}_j = -1$. Thus selecting promising features from all the SS features is equal to finding good projection vectors with only two non-zero elements $+1$ and -1. This inspires us to use LDA for feature selection. We will see below that, under such a constraint (only two non-zero elements $+1$ and -1), the computing become very efficient.

In Section III-A, we first introduce the preliminaries of LDA, then show how LDA is used as a feature selection mechanism for SS features. In Section III-B and Section III-C we describe two auxiliary techniques for our method. Each of them corresponds to one important parameter which will be discussed in detail in Section IV. The overall training process is given in Section III-D. Section III-E gives a visualization of FSSS features to show the reasonability of our method. Section III-F deduces another feature selection mechanism based on Pearson correlation coefficient and show its equivalence with the LDA based method. In Section III-G we introduce the ground plane constraint which is used in our experiments.

A. PIXEL DIFFERENCES SELECTED BY LDA

LDA [33] is a technique to find a projection which maximizes class separability. Specifically, in LDA, class separability criterion is formulated as GRQ

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{S}_B\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}}, \tag{1}$$

where $\mathbf{w}$ is the LDA projection vector. $\mathbf{S}_B$ is the between-class covariance matrix, given by

$$\mathbf{S}_B = (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T \tag{2}$$

$\mathbf{S}_W$ is the total within-class covariance matrix, given by

$$\mathbf{S}_W = \sum_{\mathbf{x}\in C_0}(\mathbf{x}-\boldsymbol{\mu_0})(\mathbf{x}-\boldsymbol{\mu_0})^T + \sum_{\mathbf{x}\in C_1}(\mathbf{x}-\boldsymbol{\mu_1})(\mathbf{x}-\boldsymbol{\mu_1})^T \tag{3}$$

$\boldsymbol{\mu_0}$ and $\boldsymbol{\mu_1}$ are means of negative and positive samples respectively. C_0 and C_1 are sets of negative and positive samples respectively.

The optimal solution is written as

$$\mathbf{w} = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1). \tag{4}$$

LDA could serve as an alternative for linear SVM with lower computational cost and little or no loss in performance [34]. LDA can also be used to train oblique splits when used in boosted tree classifiers [23]. That is, given input feature vector $\mathbf{x}$, at each node of a tree, compute $\mathbf{z} = \mathbf{w}^T\mathbf{x}$ using the projection $\mathbf{w}$ learned by LDA and find the optimal threshold for z . Because in pedestrian detection, a typical BDT contains thousands of trees and $\mathbf{x}$ is usually with a high dimensionality, for fast detection, $\mathbf{w}$ must be sparse. In [23], only a local region of $m \times m$ pixels is used for computing LDA. Moghaddam *et al.* [35] proposed an algorithm to induce sparse LDA (SLDA). In each step, the algorithm use forward selection to choose the feature yielding the maximum generalized eigenvalue. This technique is adopted in [36] to train cascade classifier for pedestrian detection. Note that this technique is to train *one* projection which leads to maximum separability with the constraint of sparsity. So the weak classifiers in the cascade are SLDA, and the features are provided by applying decision dump to the original features (in fact, here the so called original features are formed by applying another traditional LDA to the covariance features [37]). In our paper, we just use LDA to perform feature selection, and leverage the decision tree as the weak classifier. Our FSSS features can be taken as a special sparse LDA with a sparsity of 2.

Because our objective is to select features for pixel deference, we constrain the projection $\mathbf{w}$ to have only 2 non-zero elements, $+1$ and -1. That means we only consider oblique splits of 45° in a 2-dimensional subspace. Unlike the general LDA or sparse LDA, the total number of the projections under such a constraint is relatively small, and we can afford to compute them all. Let $\mathbf{w}^{ij}$ denote a projection vector with the i-th element equal to $+1$ and the j-th element equal to -1, the GRQ of $\mathbf{w}^{ij}$ is

$$J(\mathbf{w}^{ij}) = \frac{\mathbf{w}^{ij^T}\mathbf{S}_B\mathbf{w}^{ij}}{\mathbf{w}^{ij^T}\mathbf{S}_W\mathbf{w}^{ij}}, \tag{5}$$

where the numerator equals

$$\mathbf{w}^{ij^T}\mathbf{S}_B\mathbf{w}^{ij} = \mathbf{S}_B(i,i) + \mathbf{S}_B(j,j) - \mathbf{S}_B(i,j) - \mathbf{S}_B(j,i). \tag{6}$$

Next we show how to compute all the possible $J(\mathbf{w}^{ij})$ (for every i and j). Consider the relative positions in the $\mathbf{S}_B$ matrix

$$\mathbf{S}_B = \begin{bmatrix} \ddots & \vdots & & \vdots & \\ \cdots & \mathbf{S}_B(i,i) & \cdots & \mathbf{S}_B(i,j) & \cdots \\ & \vdots & \ddots & \vdots & \\ \cdots & \mathbf{S}_B(j,i) & \cdots & \mathbf{S}_B(j,j) & \cdots \\ & \vdots & & \vdots & \ddots \end{bmatrix}. \tag{7}$$

We first subtracting the diagonal vector diag($\mathbf{S}_B$) from all the rows (or columns) of $\mathbf{S}_B$ to get $\mathbf{S}'_B$, then compute $\mathbf{J}^{\text{num}} \equiv \mathbf{S}_B'^T + \mathbf{S}'_B$, which contains all the possible numerators of $J(\mathbf{w}^{ij})$.[1] With the same method we compute $\mathbf{J}^{\text{den}}$ which contains all the possible denominators of $J(\mathbf{w}^{ij})$. Then divide $\mathbf{J}^{\text{num}}$ by $\mathbf{J}^{\text{den}}$ using element-wise division, we get the GRQ matrix $\mathbf{J}$ which contains all the possible $J(\mathbf{w}^{ij})$, that is $J(\mathbf{w}^{ij}) = \mathbf{J}(i,j)$.

At this time, a possible choice is to select the k projections corresponding to the k largest GRQs, where k is the feature pool size we predefine. In practice, we find that pixel pairs with similar GRQs tend to come from neighboring areas. Thus if we sort the GRQs from high to low and choose pairs corresponding to the highest k GRQs, the selected features are not diverse. To make the selected features more diverse, we adopt the following strategy: for every pixel in the original feature map, we select the subtrahend with the largest GRQ. While in the top-k strategy, the selected pairs tend to stagnate at some highest discriminative regions but ignore others, for our new strategy, the minuend is forced to traverse the whole feature map and it will also drive the subtrahend to move, which result in a more diverse feature pool.

In this paper, we compute the GRQ matrix $\mathbf{J}$ separately for each channel. Suppose there are 10 channels with 512 features per channel, the number of candidate pixel difference features is 5120, i.e. the same number as of the original feature map. In comparison, if we use SS features, the number of pixel difference features is $\binom{512}{2} \times 10 = 1,308,160$, which makes it infeasible to train without a large amount of memory. With our method, memory constraint is not an issue even if we allow pixel differences to be computed across the 10 channels.

Apart from alleviating memory issue, compared with methods based on the general LDA, like [23], our classifier is much faster to train. The speed improvement comes from two aspects. One is that we do not need to compute the inverse of $\mathbf{S}_W$, which is the bottleneck of computing Eq. (4). Another aspect is that the general LDA only computes one optimal projection. Thus in order to form a feature pool, LDA is applied in many patches of a detection window. For our method, we generate a feature pool by finding many suboptimal projections in single GRQ computation, which brings much higher efficiency.

[1] There is a minus sign, but can be canceled by the denominator.

B. REGION-BASED FSSS FEARTURES

As is stated in [38], it is beneficial for occlusion handling if we constrain all features of a single tree to be selected from a local region. Because when occlusion occurs, only the trees covering the occluded regions are affected. We adopt this strategy in our paper and the FSSS features are computed in a region at each iteration. Using region-based FSSS also accelerate training, since only a part of the base features are involved for GRQ computing at each iteration. In our implementation, we randomly select a region for a weak classifier. The region size is important parameter for our detector. In Section IV we will discuss the region size to choose.

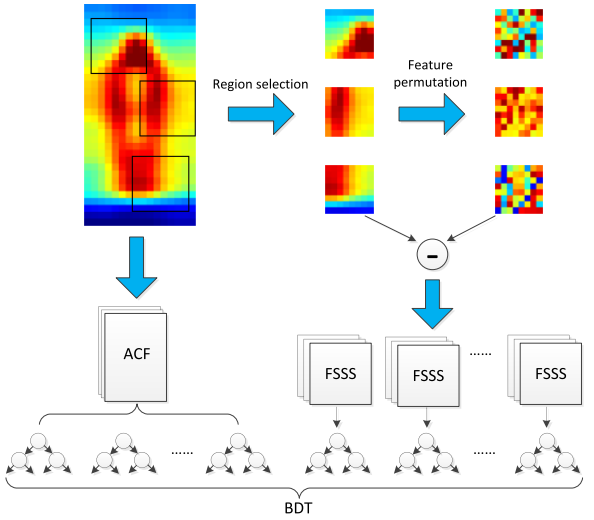


FIGURE 2. The training process of the last training round. FSSS features are generated according to the weights of training samples and are only used at late stages of the cascade.

C. TWO-STAGE TRAINING

Although calculating pixel difference is very efficient, its complexity is still higher than the ACF feature which needs only a single pixel indexing. For higher efficiency, we use ACF as a proposal to FSSS feature just as [39] and many deep learning methods [17], [19] do. The scheme, as illustrated in Fig. 2, is similar to that in [17]: (i) we use ACF features to collect hard negative samples for the last training round; (ii) in the last training round, the cascade is divided into two parts; (iii) the earlier stages of the cascade still use ACF to train, only the later stages of the cascade use FSSS features to train. We will discuss the switching stage from ACF features to FSSS features in Section IV.

As we shown in Fig. 1, most of the negative windows will be filtered out at early stages of the cascade and will not reach the less efficient FSSS stages, thus the increase in detection time is not significant. The speedup is also for the training time. First, the number of weak classifiers using FSSS is reduced by half. Second, when training the later stages of the cascade, many of the sample weights are reduced to zero. These samples are ignored in LDA computation, thus the training will also be faster in later stages compared to that in earlier stages.

D. TRAINING PROCESS

Now we put together all the ingredients stated above and introduce the training process. we use ACF features for hard negative mining before the last training round. At the last training round, the whole cascade include training stages using ACF features and training stages using FSSS features, as illustrated in Fig. 2. In the following, We only illustrate the training stages using FSSS features which are the late stages of the last training round. The algorithm is listed in Algorithm 1.

Algorithm 1 The Training Process Using FSSS Features

Input: A set of labeled training samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where $y_i = \pm 1$; the first part of the BDT cascade trained using ACF features $H_{ACF}(\mathbf{x})$; weak classifier number T for FSSS features; number of channels K.

Output: The final strong classifier $H(\mathbf{x})$.

Initialize sample weights $w_i = \exp[-y_i H_{ACF}(\mathbf{x}_i)]$.
for $t = 1, \ldots, T$ **do**
  Randomly select a region R_t and extract the index vector of features in this region as $\mathbf{s}_t$.
  for $k = 1, \ldots, K$ **do**
    For every $\mathbf{x}_i$, $i = 1, \ldots, n$, choose elements in region R_t and channel k to form a subvector $\mathbf{x}_i^k$.
    Compute between-class covariance $\mathbf{S}_B^k$ and within-class covariance $\mathbf{S}_W^k$ using $\{\mathbf{x}_i^k\}_{i=1}^n$ with weights $\{w_i\}_{i=1}^n$.
    Compute the GRQ matrix $\mathbf{J}^k$ using the method described in Section III-A.
    Compute the permutation vector $\mathbf{p}_t^k$ by selecting the index of the maximum element in each row of $\mathbf{J}^k$

$$\mathbf{p}_t^k(i) = \arg\max_j \mathbf{J}_{ij}^k \tag{8}$$

  Concatenate the permutation vectors $\{\mathbf{p}_t^k\}_{k=1}^K$ for all channels to form the overall permutation vector $\mathbf{p}_t$.
  For every $\mathbf{x}_i$, $i = 1, \ldots, n$, compute a new feature vector

$$\mathbf{x}_i^t = \mathbf{x}_i(\mathbf{s}_t) - \mathbf{x}_i(\mathbf{p}_t) \tag{9}$$

  Train a decision tree $h_t(\mathbf{x}^t)$ using $(\mathbf{x}_1^t, y_1), \ldots, (\mathbf{x}_n^t, y_n)$.
  Update weights $w_i = w_i \exp[-y_i h_t(\mathbf{x}_i^t)]$.
Return the final strong classifier

$$H(\mathbf{x}) = \text{sign}(H_{ACF}(\mathbf{x}) + \sum_{t=1}^{T} h_t(\mathbf{x}(\mathbf{s}_t) - \mathbf{x}(\mathbf{p}_t))) \tag{10}$$

For each iteration using FSSS features, we first randomly select a region which corresponds to an index vector $\mathbf{s}_k$. The elements of this vector denote the indexes of the features in the original feature map. For each channel, we use LDA to find the corresponding feature index which leads to the maximum GRQ for each feature in this region. These indexes form a permutation vector (lines 4-8). Permutation vectors of all channels $\{\mathbf{p}_t^k\}_{k=1}^K$ are concatenated together to form an overall permutation vector $\mathbf{p}_t$. Note the indexes in $\mathbf{p}_t^k$ are region-based. Before concatenation, we need to switch these region-based indexes to their real indexes in the original feature map. Then, we use the index vector $\mathbf{s}_k$ and permutation vector $\mathbf{p}_t$ to compute new feature vectors and use these new feature vectors to train a decision tree. Finally we update sample weights for the next iteration.

The whole classifier is constructed by combining stages using ACF features and stages using FSSS features. Note both $\mathbf{s}_k$ and $\mathbf{p}_t$ are saved in each training iteration and will be used at detection time.

For BDT, sample weights are recomputed after every boosting iteration and training samples are divided into two parts in every tree node split, thus the sample weights distribution are different for each node. In principle, we need to recompute new feature vectors at each node, which takes too much training time. The time cost not only comes from the GRQ computation, but also from quantization of the new feature vectors [40].[2] Thus in our experiments, we renew the feature vectors for every boosting iteration, not for every node.

E. VISUALIZING SELECTED FEATURES

Our FSSS features are intuitive and reasonable. To see this, we choose the #100 feature in each of the 10 channels and show the corresponding GRQ maps of these features in Fig. 3.

The green square indicates the #100 feature, the red square denotes the maximum GRQ in this map, the blue square denotes the maximum GRQ in the yellow region. Some GRQ maps do not show red squares, which means the position of the red square coincides with the blue square. The meaning of this map is that when we use the pixel at the position of the green square as minuend, the corresponding subtrahend is the pixel at the position of the red square for global FSSS, or the pixel at the position of the blue square for region-based FSSS.

Note how interpretable of the GRQ maps and the chosen corresponding pixels are. We take global FSSS as an example. In the first map (L channel), the pixel chosen is at leg. Because the green position in a pedestrian window is often sky which has high luminance and the luminance of the face and upper body are often brighter than the lower body, thus have similar luminance with the sky. Hence choosing pixel at the lower body to perform the differential operation may lead to a better discrimination. In the second map (U channel), the pixel chosen is at face/head, which is a very discriminative position for the U channel [20]. In map 4 (gradient magnitude channel) and map 8 (horizontal gradient channel) the red square is chosen at feet, this position is very informative to imply a pedestrian, which is stated in [31]. In map 9, the choosing position is at shoulder, which is the well-known Ω position for pedestrian detection [41]. The region-based FSSS may give a different choice, and is also very reasonable. In a word, the corresponding pixels automatically chosen by our method are very informative and interpretable. Unlike [21], [22], [38] which use shape prior to generate features, in our method, the pixel pairs are automatically learned from training data. Thus our method is promising to be extended to other object detection tasks.

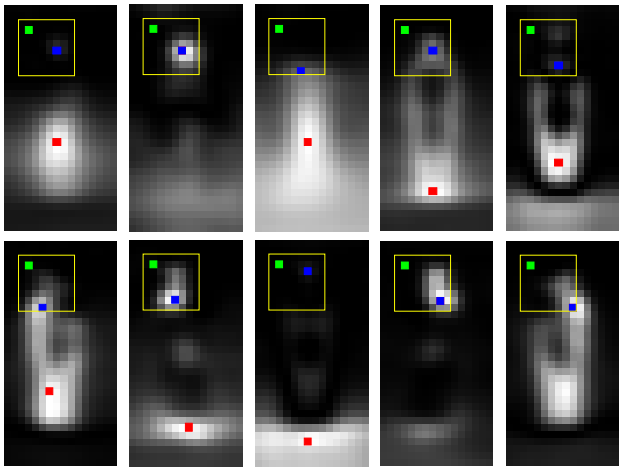


FIGURE 3. The GRQ map of the #100 feature in 10 channels. The green square denotes the #100 feature, the red square denotes the feature corresponding to the maximum GRQ in this map, the blue square denotes the feature corresponding to the maximum GRQ in the yellow region. Some red squares coincide with the blue squares. See the text for details.

F. DISCUSSION

The feature selection procedure can also be performed using Pearson correlation coefficient. Here we will show its correlation with the LDA based method. Given the response variable $c = \{0, 1\}$, our aim is to find out which combination of pixel difference provides the largest absolute correlation coefficient with c. Suppose x and y are arbitrary two features in a feature map, the weighted correlation coefficient between $x - y$ and c is computed by

$$\rho_{x-y,c} = \frac{\sum_i \big((x_i - y_i) - \mu_{x-y}\big)(c_i - p)w_i}{\sigma_{x-y}\sigma_c}, \tag{11}$$

where p is the total weight of all positive examples, μ_{x-y} and σ_{x-y} are the mean and standard deviation of $(x - y)$. Thus the numerator of $\rho_{x-y,c}$ is

$$\sum_i (x_i - \mu_x)(c_i - p)w_i - \sum_i (y_i - \mu_y)(c_i - p)w_i, \tag{12}$$

[2] A feature value is a continuous variable. To find the optimal split efficiently, current implementations of BDT usually quantize the range of the feature value into say, 256 bins. When the feature changes, the range changes, thus we need to re-quantization.

with the first term calculated as

$$\begin{aligned}&\sum_i (x_i - \mu_x)(c_i - p)w_i \\ &= (1-p)\sum_{c_i=1}(x_i - \mu_x)w_i - p\sum_{c_i=0}(x_i - \mu_x)w_i \\ &= p(1-p)(\mu_{x_1} - \mu_{x_0}) \end{aligned} \tag{13}$$

Where μ_{x_1} is the mean of feature x of all positive examples, μ_{x_0} is the mean of feature x of all negative examples. The second term of Eq. (12) has the same form. Thus the denominator of $\rho_{x-y,c}$ can be easily computed and we get

$$\rho_{x-y,c} = \sqrt{p(1-p)}\frac{(\mu_{x_1} - \mu_{x_0}) - (\mu_{y_1} - \mu_{y_0})}{\sqrt{\sigma_x^2 + \sigma_y^2 - 2\sigma_{xy}}}. \tag{14}$$

We prefer to choose the pixel pair with the largest $|\rho_{x-y,c}|$. Suppose the corresponding indexes of x and y are i and j, then $|\rho_{x-y,c}|^2$ is $p(1-p)$ times the $J(\mathbf{w}^{ij})$ (see Eq. (5)). Since both the coefficient $p(1-p)$ and the square operation will not change the order of non-negative values, the index selected by Pearson correlation coefficient coincides with the one selected by GRQ criterion. Besides, $J(\mathbf{w}^{ij})$ could be taken as the Fisher score [42] of $(x-y)$ and $\frac{1}{1+J(\mathbf{w}^{ij})}$ could be taken as the Laplacian score [43] of $(x-y)$.

G. CONTEXTUAL INFORMATION

Ground plane constraint (GPC) is an important context information which is widely used for pedestrian detection [12], [13], [44]–[47]. The key idea of GPC is that under some assumptions [12] which are valid for an onboard camera, the projected height h and the vertical position y of a pedestrian exhibit a linear relationship.

This relationship is used variously in different papers. Park *et al.* [12] and Ohn-Bar and Trivedi [46] re-scored the detection confidence s using SVM with h and y as features. In [13] the authors found that in Caltech dataset the pedestrian window centers are largely (over 99%) located between the rows 140 and 300, thus only scan windows in this region. Kim and Kim [47] modeled the position and the size of a pedestrian in terms of normal distribution and ignored pedestrians out of the 3σ scope.

Currently the top non-deep-learning methods on Caltech benchmark use GPC. To make a fair comparison, we also add this trick in our experiments. We directly use the linear relationship of the heights and vertical positions of pedestrians. Because the assumptions in [12] are not strictly satisfied, the point (h, y) will deviate from the straight line. We argue that the top position and height of the detection window can be bounded by two straight lines as shown in Fig. 4 which shows the distribution of (h, y) of the pedestrians in Caltech training set and the two bounding lines as an example. More than 99.9% pedestrians in the training set belong to this bounding area. When performing detection, we speed up detecting by only searching for pedestrians in this bounding area.

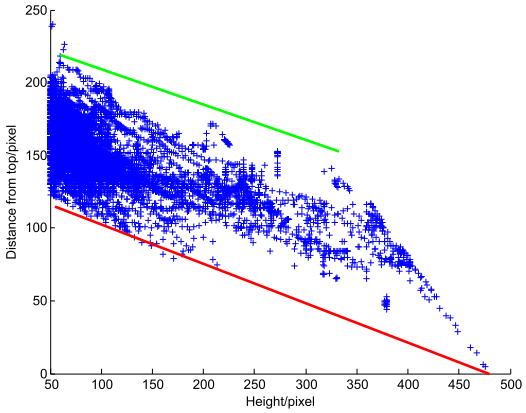


FIGURE 4. (h, y) distribution of the ground truth in Caltech training set. These points can be bounded by two lines.

IV. EXPERIMENTS

We evaluate our proposed method on several standard pedestrian detection datasets which are the most popular and widely used ones in the literature, including INRIA [3], ETH [4], KITTI [2] and Caltech [48] datasets. We use INRIA for parameter tuning, as in [23], [29], and [45]. For INRIA, KITTI and Caltech datasets, we only use their own training data to train the model and test it on their test data. Since there is no training data for ETH, the model trained on INRIA training dataset is used for experiment on ETH. The Intersection-over-Union (IoU) threshold is set to 0.5 to determine true positives for all datasets unless noted otherwise. Results on INRIA, ETH and Caltech are compared using miss rate vs. False-Positive-Per-Image (FPPI) curves, which is the well-recognized evaluation metric for pedestrian detection [1]. Methods are ranked by log average miss rate which is computed by averaging the miss rate at 9 FPPI points that are evenly spaced in the log-space ranging from 10^{-2} to 10^{0} unless noted otherwise. Results on KITTI are compared using precision-recall curves, and methods are ranked by averaging the precision at 11 evenly spaced recall points ranging from 0 to 1.

All the methods involved in comparison in INRIA, ETH and Caltech experiments are listed in the website of Caltech Pedestrian Detection Benchmark[3] and all the methods involved in comparison in KITTI experiment are listed in the website of KITTI Vision Benchmark Suite.[4]

GPC is eligible for transporting scenario when the camera setup is fixed for training and testing. In our experiments, we only use GPC for the Caltech dataset. The result with GPC is denoted as FSSSC. Results of Both FSSS and FSSSC are shown. INRIA is not a transporting dataset, ETH only has testing data, hence both INRAI and ETH can not use GPC. KITTI will benefit from GPC (we got about 1% improvement in our validation set), but the evaluation server only allow one version submission, thus we

[3]http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/
[4]http://www.cvlibs.net/datasets/kitti/

only submit the FSSS result for evaluation to show our main contribution.

A. EVALUATION ON INRIA DATASET

The INRIA dataset includes 614 positive images and 1218 negative images for training. Evaluation results are reported on the 288 positive testing images. The model size is set as 128×64 and the channels are downsampled by 4x, thus the feature map size is 32×16. For multi-scale detection, channels are computed over an image pyramid with 8 scales per octave. The final classifier is built via three rounds of hard negative mining (starting from a forest with 32 trees, and then 256, 1024, 4096 trees). Realboost [49] and level-3 decision trees are used to train our model. In the last round, we switch our RealBoost algorithm to the shrinkage version as is used in [18] and [50]. The shrinkage parameter is set to 0.5. Two hyper-parameters are considered here, one is the region size, another is the switching stage. The default value of the region size is set to 8×8 (half the model width) and the default value of the switching stage is set to 2048 (half the weak classifier number). When we test on one parameter, another is kept as its default value.

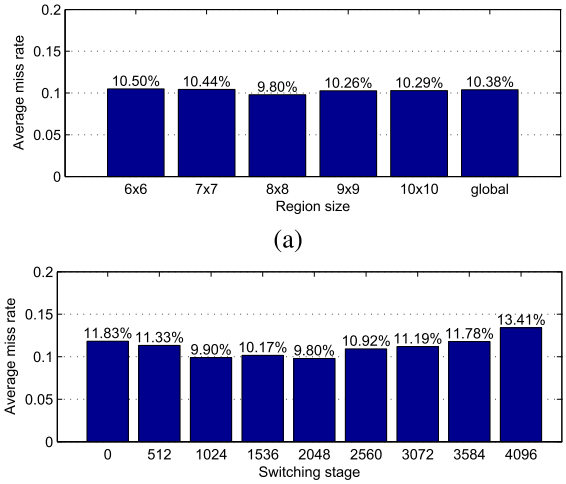


FIGURE 5. Evaluation of different parameters on INRIA test set. (a) Region size. (b) Switching stage.

First, we test the detector trained with different region size, as shown in Fig. 5(a). Note 'global' means we do not use region-based FSSS, e.g., the whole feature map in each channel is involved in GRQ computing. From the figure we see that the best region size is 8×8 which happens to cover half the model width. Intuitively, region size should not be too small or too large. The region size needs to be sufficiently large to ensure it contains enough foreground part. For a 32×16 feature map, if the region size is smaller than 6×6, in some cases it will hardly cover any foreground part (for example in the corner of the feature map), thus feature differences in this region will be not useful. On the other hand, if a region size is too large, the benefit for occlusion handling will be small.

The switching stage from ACF feature to FSSS feature is also important. From Fig. 5(b) we see that though using FSSS will lower the miss rate, ACF is a necessary complement. Switching to FSSS too early will cause performance degradation. Thus using FSSS at the later stages of the cascade not only accelerates detection speed but also improves accuracy. The optimal switching stage is 2048, which is half the total weak classifier number. In the following experiments, we will set the region size to 8×8 and the switching stage to half the weak classifier number.

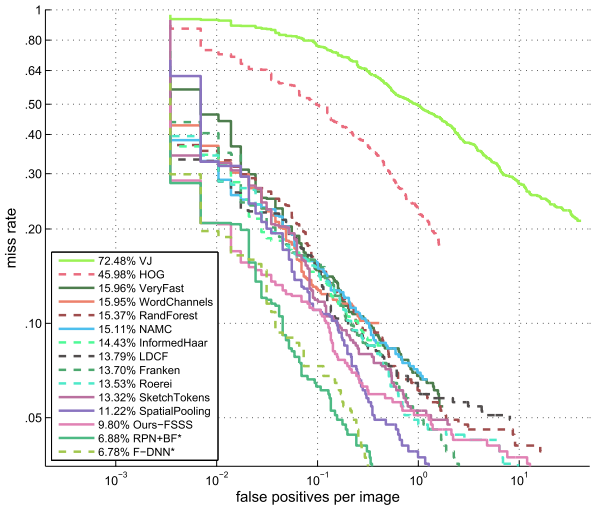


FIGURE 6. Comparison with state-of-the-art methods on the INRIA dataset.

The comparison with state-of-the-art methods is shown in Fig. 6. Our method achieves a 9.80% log average miss rate, which outperforms all non deep learning approaches (the ones without * mark). Compared with the second best method SpatialPooling which achieves a 11.22% log average miss rate, our method not only provides a 12.66% relative improvement, but has a much higher detection speed (See Tabel 1).

TABLE 1. Detection speed (FPS) and miss rate (MR) on the Caltech dataset.

METHOD	FPS	MR
FPDW	2.6	57.4%
ChnFtrs	0.2	56.34%
CrossTalk	14	53.88%
Roerei	1	46.13%
ACF-Caltech	30	44.22%
FastCF	105	37.33%
SquaresChnFtrs	1	34.81%
InformedHaar	0.63	34.6%
SpatialPooling	0.13	29.23%
MRFC	20	19.09%
LDCF84	2.5	17.15%
NNNF-L4	1.14	16.84%
MRFC+Semantic	8	16.83%
Ours-FSSS	3.13	13.96%
Ours-FSSSC	3.46	13.13%

B. EVALUATION ON ETH DATASET

The ETH dataset only contains 1804 images for testing and there is no training data. Thus we use the

detector trained on INRIA for evaluation which is widely adopted in the literature [15], [50], [51]. As there is color offset in this dataset, we apply the automatic color equalization algorithms (ACE) [52], [53] to the images before we extract channel features. This treatment is also adopted in [50]. For the evaluation results are reported on pedestrians taller than 50 pixels, the test images is upsampled by one octave. The result is shown in Fig. 7. Again, our method outperforms all non-deep-learning methods and some the deep learning methods (DBN-IsoI [54], JointDeep [55], DBN-Mut [56], SDN [57]).

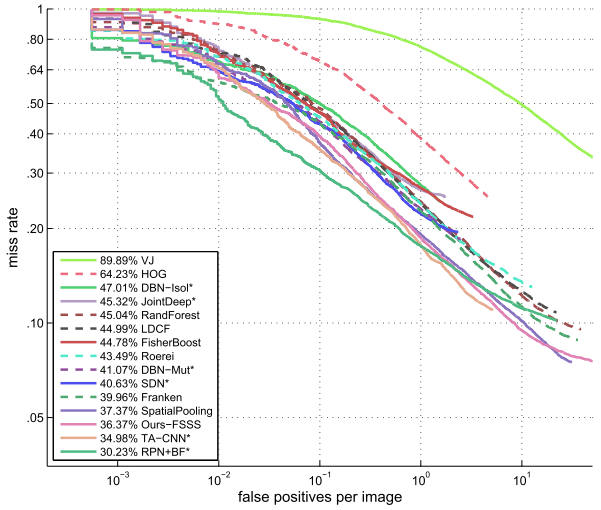


FIGURE 7. Comparison with state-of-the-art methods on the ETH dataset.

C. EVALUATION ON KITTI DATASET

The KITTIs object detection benchmark has 7481 training and 7518 test images. It contains three object classes for evaluation: Car, Pedestrian, and Cyclist. Here we only choose pedestrian class for evaluation. KITTI differentiates the difficulty in identifying pedestrians to three levels: *easy*, *moderate* and *hard*, corresponding to different height, occlusion and truncation. All methods are ranked based on the moderate difficult level (the minimum height of bounding box is 25 pixels, maximum occlusion level is partly occluded and a maximum truncation of 0.30) in the benchmark. In our experiments, most parameters are the same as we used in INRIA experiment, except that we upsample the images by two octaves to detect pedestrians with heights between 25 to 100 pixels and we use level-5 decision trees instead of level-3 trees.

The Precision-recall curves (moderate difficult level) of our method and some published methods on KITTI benchmark are shown in Fig. 8. The average precision of our method is 62.09%, outperforming all the other non deep learning method. The closest one to ours (FilteredICF) only achieves 57.12%. Note the result of Regionlets [58] is achieved by combining the Regionlets and CNN. Some recent deep learning methods which achieve good performance in other dataset are also outperformed by our method (e.g., RPN+BF [15], CompACT-Deep [25]).

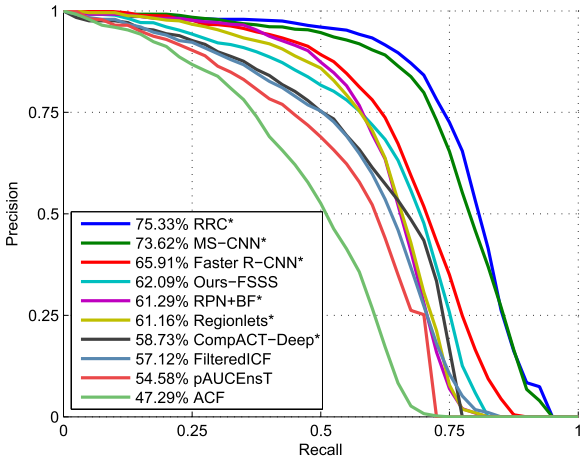


FIGURE 8. Precision-recall curves of the moderately difficult level on KITTI dataset.

Some top deep learning methods (MS-CNN [14], RRC [59]) outperform ours by a large margin. We argue that this has something to do with the KITTI evaluation metrics. In KITTI metrics, there are two disadvantages for our method, which happens to be advantages for some top deep learning methods:

(i) Unlike other datasets which take pedestrians and cyclists as the same class (e.g. Caltech pedestrian dataset), the human data in the KITTI dataset consists of two disjoint subsets (pedestrian class and cyclist class). Our method, like other traditional methods, trained as a binary classifier for pedestrian. At validation time, we find it is difficult for our detector to discriminate pedestrian class and cyclist class separately due to their similar appearance. This will lead to false positives. On the other hand, some CNN methods trained a multiclass detector and will naturally distinguish pedestrian and cyclist.

(ii) The evaluation in INRIA, ETH, and Caltech dataset follows the routine in [1] which standardize the aspect ratio of all the ground truth and detected bounding boxes to 0.41. Traditional sliding window method like ours use a fixed aspect ratio which is suitable for this evaluation. KITTI benchmark, on the other hand, does not standardize the ground truth aspect ratio and the detected bounding boxes may not have a good alignment with the ground truth. This is not a problem for some current deep learning method for they are with a bounding box regression operation which may fit any aspect ratio.

We will see these disadvantages more clearly by comparing two CNN-based methods RPN+BF and MS-CNN. The RPN+BF is a binary detector specialized for pedestrians and it only use bounding box regression for Region Proposal Network (RPN). MS-CNN is a multiclass detector which perform bounding box regression at both RPN and the classification layers. Therefore, although RPN+BF outperform MS-CNN on Caltech dataset (see Fig. 10), it is significantly outperformed by MS-CNN.

Based on the two drawbacks stated above, there are two ways to improve the performance of our method on KITTI dataset. One is to train a separate cyclist detector, another is to train multiple detector corresponding to different aspect ratio. Besides, according to [60], combining detectors with multiresolution feature map will also significantly improve the performance for KITTI dataset. All these methods are about model combination and are orthogonal to our method. They can be implemented on top of our method and further improvement in detection accuracy is expected. To benefit from this type of methods, some strategies need to be carefully designed, such as subcategorization, calibration of confidence scores and fusion of detection results. We will explore this type of methods in our future work.

D. EVALUATION ON CALTECH DATASET

Finally we evaluate our method on Caltech pedestrian datasets which is currently the largest and the most widely used pedestrian detection dataset. It enables a comparison among more than 60 state-of-the-art approaches published during recent years. It consists 250,000 labeled 640 × 480 frames (in 137 approximately minute long segments) which are divided into 11 sessions. The first 6 sessions are used for training and the last 5 sessions are used for testing. The standard evaluation is performed on every 30th frame of the test set, which consists 4,024 images in all. The results are evaluated using the reasonable difficulty which means the pedestrian is at least 50 pixels in height and with a visibility of at least 65%.

Training images is collected by sampling one image out of every 4 consecutive frames, which result in 32,077 images. Most parameters are the same as we used in KITTI experiment, except we use twice the weak classifier number in each bootstrapping round (that is, starting from a model with 64 trees, and then 512, 2048, 8196 trees). Note the corresponding switching stage is also doubled to 4096. We also upsample the test images by one octave to detect pedestrians with heights between 50 to 100 pixels.

1) EVALUATION USING STANDARD ANNOTATIONS

Our FSSS features are constructed by feature differences of ACF features. Here we first show our method will indeed outperform the original ACF features. We train another BDT with all the 8192 trees using ACF features and plot the log average miss rate evolve with the weak classifier number. The result is shown in Fig. 9. The blue line shows that the log average miss rate of original ACF features get stuck about 19%. The red line shows the result of our FSSS framework, in which the first 4096 stages still use ACF features (thus the log average miss rate is the same with that of the blue line) and the subsequent 4096 stages use FSSS features. The figure shows that when we switch to FSSS features, the log average miss rate decreases radically. Adding more weak classifiers using ACF features makes no improvement on performance indicating the discrimination power of the ACF features are exhausted after the first 4096 stages. It is

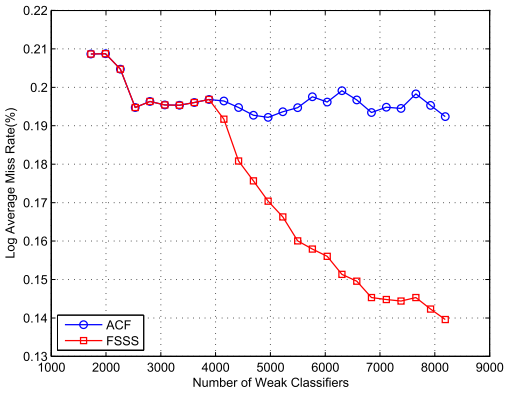


FIGURE 9. Weak classifier number versus the log average miss rate on Caltech test dataset.

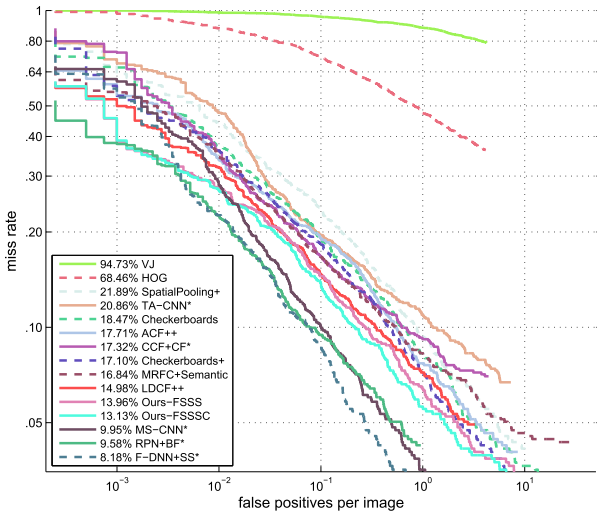
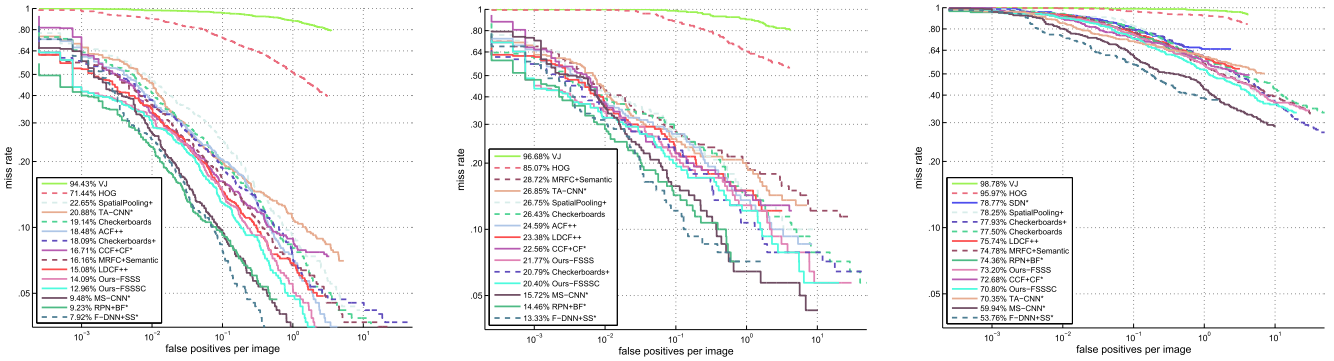


FIGURE 10. Comparison with state-of-the-art methods on the Caltech dataset with standard annotations.

apparently that FSSS features are complementary to the ACF features and will boost the performance.

Comparison with the state of the art methods are shown in Fig. 10. Before using ground plane constrains (GPC), the accuracy of our proposed FSSS (13.96%) already outperform the top non deep learning methods LDCF++ (14.98%) [46] and MRFC+Semantic (16.84%) [13]. Note both these two methods have already used GPC and MRFC+Semantic trained on outside dataset to get semantic channels. By using GPC (denoted as 'FSSSC'), the log average miss rate of our detector is further lowered to 13.13%, providing a 12.35% relative improvement to LDCF++.

We also analyze performance under some difficult conditions on the testing data. Fig. 11 shows the evaluation results under conditions of small scale, atypical aspect ratio and heavy occlusion. In all these circumstances, our FSSS and FSSSC detectors outperform other non deep learning method, except in atypical aspect ratio condition, FSSS is outperformed by checkerbords+ [7]. Checkerbords+ is the



(a) (b) (c)

FIGURE 11. Evaluation results under some difficult conditions on Caltech test set with standard annotations. (a) Small scale (50px $\leq h \leq$ 80px). (b) Atypical aspect ratio ($|w/h - 0.41| \geq 0.1$). (c) Heavy occlusion (35%-80% occluded).

baseline Checkerbords detector enhanced by motion features [61] which need video information while our method is only based on a single frame. In other circumstances, checkerbords+ only achieve a little improvement to checkerbords, but for the atypical aspect ratio condition, it lower the average miss rate by about 6%. This implies that motion information is very helpful to tackle this situation. This may because the atypical aspect ratio often happens when pedestrians are crossing a road, and showing their profiles. In this circumstance, the part-centric motion is significant which is the most useful motion information for pedestrian detection [61]. Our method is hopefully further improved by using motion information and this is left for future work.

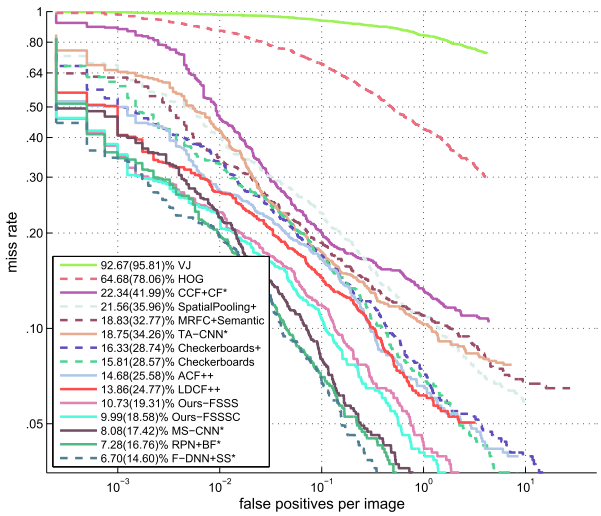


FIGURE 12. Comparison with state-of-the-art methods on the Caltech test set with the new annotations.

2) EVALUATION USING NEW ANNOTATIONS

In [62], Zhang *et al.* conducted a detailed survey and provided a new and more accurate ground truth labeling on Caltech dataset. We also test our trained detector using the new annotations, as shown in Fig. 12. Zhang *et al.* [62] also propose to extend the evaluation FPPI range from traditional $[10^{-2}, 10^{\circ}]$ to $[10^{-4}, 10^{\circ}]$. In our experiments with the new annotations, we use both the standard FPPI range $[10^{-2}, 10^{\circ}]$ and the extended range $[10^{-4}, 10^{\circ}]$. Our method also ranks the first in all non-deep-learning methods. With more accurate annotations, the performance of both LDCF++ and FSSS improves, but improvement of FSSS is more remarkable. With GPC, our FSSSC achieves a log average miss rate of 9.99%, providing a 27.92% relative improvement with respect to LDCF++ (13.86%).

3) RUNTIME ANALYSIS

We measure the detection speed using a single core of Intel i7 6700K CPU (4 GHz) on Caltech dataset. Table 1 provides a comparison of our approach with some state-of-the-art non-deep-learning methods whose execution time are provided. A more intuitive comparison of these methods are also shown in Fig. 13. From the figure we see that our method achieves the top accuracy with a moderate detection speed. Note that the authors of the previous best method LDCF++ [46] only give the computation time of LDCF84, not LDCF++. LDCF++ is an improved version of LDCF84. The differences between them are: (i) LDCF++ does not use the feature pyramid approximation [6] while LDCF84 does; (ii) LDCF++ uses SVM to re-score the detection result. Both changes will lower the detection speed, thus LDCF++ will be slower than LDCF84, and of course, slower than our method.

4) COMPARISON WITH DEEP LEARNING METHODS

Though some deep learning methods outperform our method, their success is based on very deep CNN models and external training data. For example, the currently top method F-DNN+SS [8] combine single shot multibox detector (SSD) [63], GoogleNet [64], ResNet-50 [65] and semantic segmentation (SS) network [66], in which SSD and SS are based on VGG16 network [67]. It runs at 2.48 second per image on TITAN X, while our method runs at 0.289 second per image using CPU. Apart from Caltech training

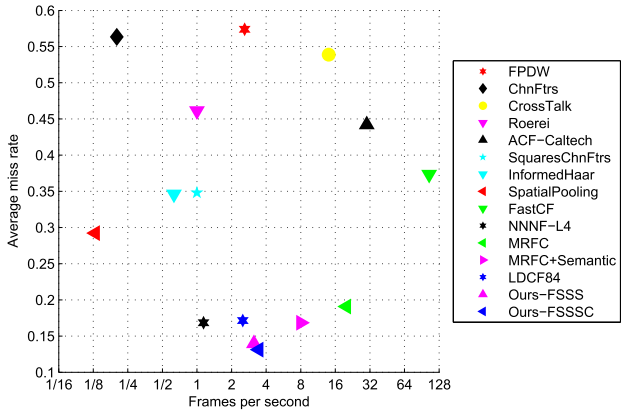


FIGURE 13. Miss rates (MR) versus frames per second (FPS) on the Caltech Dataset.

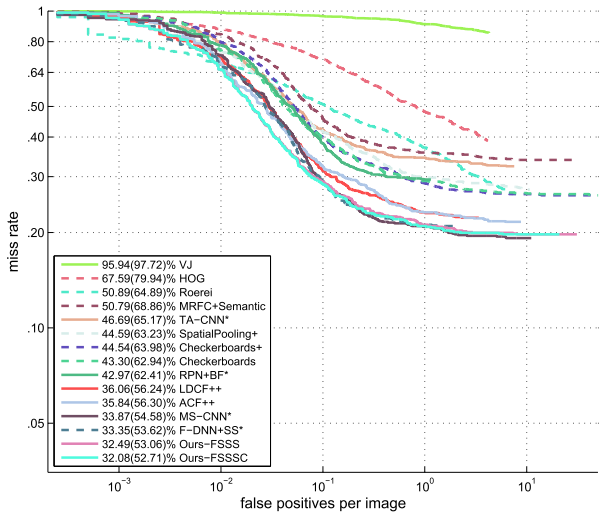


FIGURE 14. Comparison on the Caltech test set with new annotations and *IoU* > 0.7 to determine true positives.

set, F-DNN+SS uses ImageNet [68], Microsoft COCO [69], Cityscape [70], ETH [4] and TudBrussels [5] for training, while we only use Caltech training set.

Moreover, when we raise the IoU threshold from 0.5 to 0.7, all the deep learning methods exhibit dramatic performance degradation and are outperformed by our method, see Fig. 14. When using a larger threshold, in order to be taken as a true positive, a detected bounding box needs to be better aligned with a ground truth bounding box. Hence, increasing the threshold means using a stricter criterion which focuses more on localization quality. At this circumstances, the accuracy of all detectors decreases without doubt, but the accuracy of the deep learning methods deceases more than ours which means the localization quality of our method is better than that of the deep learning methods. As stated in [62], this weakness in localization of current deep learning methods may be due to their feature pooling operation. Tabel 2 summarizes localization performances of top deep learning methods and

TABLE 2. Comparison of mean Iou between top deep learning methods and Fsss at 10^{-1} Fppi.

Method	Miss Rate	Mean IoU
MS-CNN	8.00%	78.05%
RPN+BF	7.02%	75.87%
F-DNN+SS	6.80%	77.97%
FSSS	11.84%	79.50%

FSSS in terms of the mean IoU between detection results and ground truth at 10^{-1} FPPI. From Table 2 we see that although the top deep learning methods have lower miss rate compared to FSSS, their location quality are all inferior than our method. Therefore, when we use a stricter metric, their weakness in precise localization is revealed. Note the RPN+BF has the worst localization quality among them, thus shows the most dramatic performance degradation. Though these deep learning methods use bounding box regression to increase the localization quality, their performances are still inferior than our method. Hence we believe that the proposed approach may provide complementary information for deep learning approaches, and will explore in our future work the integration of these two kinds of approaches to get further improvements.

V. CONCLUSION AND FUTURE WORK

In this paper we have proposed a new type of features based on LDA and SS features. They are incorporated into the BDT model for pedestrian detection. The main contribution of our work is a novel feature selection method which uses LDA to generate a feature pool for classification. The generated FSSS features are only used at late stages of the BDT, hence the increase of computation cost is small. Experiments have shown that our features achieve top accuracy with moderate detection speed. We compare our method with deep learning methods and show its superiority in high-quality localization. We have also discussed some possible directions to further improve the performance of our detector and will explore them in our future work.

REFERENCES

[1] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2011.155

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3354–3361. [Online]. Available: https://doi.org/10.1109/CVPR.2012.6248074

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[4] A. Ess, B. Leibe, K. Schindler, and L. J. V. Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8, [Online]. Available: https://doi.org/10.1109/CVPR.2008.4587581

[5] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Proc. IEEE CVPR*, Jun. 2009, pp. 794–801. [Online]. Available: https://doi.org/10.1109/CVPRW.2009.5206638

[6] P. Dollár, R. Appel, S. J. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014. [Online]. Available: https://doi.org/10.1109/TPAMI.2014.2300479

[7] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1751–1760. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298784

[8] X. Du, M. El-Khamy, J. Lee, and L. S. Davis, "Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Santa Rosa, CA, USA, Mar. 2017, pp. 953–961. [Online]. Available: https://doi.org/10.1109/WACV.2017.111

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010. [Online]. Available: https://doi.org/10.1109/TPAMI.2009.167

[10] R. Benenson, M. Omran, J. H. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Proc. Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 613–627.

[11] X. Zhao, Z. He, S. Zhang, and D. Liang, "Robust pedestrian detection in thermal infrared imagery using a shape distribution histogram feature and modified sparse representation classification," *Pattern Recognit.*, vol. 48, no. 6, pp. 1947–1960, 2015. [Online]. Available: https://doi.org/10.1016/j.patcog.2014.12.013

[12] D. Park, D. Ramanan, and C. C. Fowlkes, "Multiresolution models for object detection," in *Proc. ECCV*, 2010, pp. 241–254.

[13] A. D. Costea and S. Nedevschi, "Semantic channels for fast pedestrian detection," in *Proc. IEEE CVPR*, Jun. 2016, pp. 2360–2368. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.259

[14] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. 14th Eur. Conf. Comput. Vis. ECCV*, Amsterdam, The Netherlands, Oct. 2016, pp. 354–370.

[15] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster R-CNN doing well for pedestrian detection?" in *Proc. ECCV*, 2016, pp. 443–457.

[16] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[17] J. Cao, Y. Pang, and X. Li, "Learning multilayer channel features for pedestrian detection," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3210–3220, 2017. [Online]. Available: https://doi.org/10.1109/TIP.2017.2694224

[18] Q. Hu, P. Wang, C. Shen, A. van den Hengel, and F. M. Porikli, "Pushing the limits of deep cnns for pedestrian detection," *CoRR*, Mar. 2016. [Online]. Available: http://arxiv.org/abs/1603.04525

[19] J. Li, X. Liang, S. Shen, T. Xu, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *CoRR*, Oct. 2015. [Online]. Available: http://arxiv.org/abs/1510.08160

[20] P. Dollár, Z. Tu, P. Perona, and S. J. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, London, U.K., Sep. 2009, pp. 1–11. [Online]. Available: http://dx.doi.org/10.5244/C.23.91

[21] S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 947–954. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.126

[22] J. Cao, Y. Pang, and X. Li, "Pedestrian detection inspired by appearance constancy and shape symmetry," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5538–5551, Dec. 2016. [Online]. Available: https://doi.org/10.1109/TIP.2016.2609807

[23] W. Nam, P. Dollár, and J. H. Han, "Local decorrelation for improved pedestrian detection," in *Proc. Syst. 27th Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 424–432. [Online]. Available: http://papers.nips.cc/paper/5419-local-decorrelation-for-improved-pedestrian-detection

[24] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Convolutional channel features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 82–90. [Online]. Available: https://doi.org/10.1109/ICCV.2015.18

[25] Z. Cai, M. J. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 3361–3369. [Online]. Available: https://doi.org/10.1109/ICCV.2015.384

[26] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3158–3165. [Online]. Available: https://doi.org/10.1109/CVPR.2013.406

[27] Q. Zhu, M. Yeh, K. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, New York, NY, USA, Jun. 2006, pp. 1491–1498. [Online]. Available: https://doi.org/10.1109/CVPR.2006.119

[28] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, USA, Jun. 2008, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CVPR.2008.4587630

[29] R. Benenson, M. Mathias, T. Tuytelaars, and L. J. Van Gool, "Seeking the strongest rigid detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3666–3673. [Online]. Available: https://doi.org/10.1109/CVPR.2013.470

[30] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002. [Online]. Available: https://doi.org/10.1109/TPAMI.2002.1017623

[31] J. Shen, X. Zuo, J. Li, W. Yang, and H. Ling, "A novel pixel neighborhood differential statistic feature for pedestrian and face detection," *Pattern Recognit.*, vol. 63, pp. 127–138, Mar. 2017. [Online]. Available: https://doi.org/10.1016/j.patcog.2016.09.010

[32] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CVPR.2007.383198

[33] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York, NY, USA: Springer-Verlag, 2006.

[34] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *Proc. ECCV*, 2012, pp. 459–472.

[35] B. Moghaddam, Y. Weiss, and S. Avidan, "Fast pixel/part selection with sparse eigenvectors," in *Proc. IEEE ICCV*, Oct. 2007, pp. 1–8.

[36] C. Shen, S. Paisitkriangkrai, and J. Zhang, "Efficiently learning a detection cascade with sparse eigenvectors," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 22–35, Jan. 2011.

[37] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on Riemannian manifolds," in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

[38] Y. Zhao, Z. Yuan, D. Chen, J. Lyu, and T. Liu, "Fast pedestrian detection via random projection features with shape prior," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Santa Rosa, CA, USA, Mar. 2017, pp. 962–970. [Online]. Available: https://doi.org/10.1109/WACV.2017.112

[39] D. Zhang, S. Z. Li, and D. Gatica-Perez, "Real-time face detection using boosting in hierarchical feature spaces," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, Cambridge, U.K., Aug. 2004, pp. 411–414. [Online]. Available: https://doi.org/10.1109/ICPR.2004.1334238

[40] R. Appel, T. J. Fuchs, and P. Dollár, and P. Perona, "Quickly boosting decision trees - pruning underachieving features early," in *Proc. 30th Int. Conf. Mach. Learn. (ICML)*, Atlanta, GA, USA, Jun. 2013, pp. 594–602. [Online]. Available: http://jmlr.org/proceedings/papers/v28/appel13.html

[41] N. Dalal, "Finding people in images and videos," Ph.D. dissertation, Dept. Math. Sci. Technol. Inf., Grenoble Inst. Technol., Grenoble, France, 2006. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00390303

[42] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, 2000.

[43] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Vancouver, BC, Canada, Dec. 2005, pp. 507–514. [Online]. Available: http://papers.nips.cc/paper/2909-laplacian-score-for-feature-selection

[44] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," *Int. J. Comput. Vis.*, vol. 80, no. 1, pp. 3–15, 2008. [Online]. Available: https://doi.org/10.1007/s11263-008-0137-5

[45] J. Marín, D. Vázquez, A. M. López, J. Amores, and B. Leibe, "Random forests of local experts for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 2592–2599. [Online]. Available: https://doi.org/10.1109/ICCV.2013.322

[46] E. Ohn-Bar and M. M. Trivedi, "To boost or not to boost? On the limits of boosted trees for object detection," in *Proc. 23rd Int. Conf. Pattern Recognit., (ICPR)*, Cancún, NM, USA, Dec. 2016, pp. 3350–3355. [Online]. Available: https://doi.org/10.1109/ICPR.2016.7900151

[47] H. K. Kim and D. Kim, "Robust pedestrian detection under deformation using simple boosted features," *Image Vis. Comput.*, vol. 61, pp. 1–11, May 2017. [Online]. Available: https://doi.org/10.1016/j.imavis.2017.02.007

[48] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, Jun. 2009, pp. 304–311. [Online]. Available: https://doi.org/10.1109/CVPRW.2009.5206631
[49] J. H. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.
[50] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Strengthening the effectiveness of pedestrian detection with spatially pooled features," in *Proc. ECCV*, 2014, pp. 546–561.
[51] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5079–5087. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7299143
[52] A. Rizzi, C. Gatta, and D. Marini, "A new algorithm for unsupervised global and local color correction," *Pattern Recognit. Lett.*, vol. 24, no. 11, pp. 1663–1677, Jul. 2003. [Online]. Available: https://doi.org/10.1016/S0167-8655(02)00323-9
[53] P. Getreuer, "Automatic color enhancement (ACE) and its fast implementation," *Imag. Process. Line*, vol. 2, pp. 266–277, Nov. 2012. [Online]. Available: https://doi.org/10.5201/ipol.2012.g-ace
[54] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3258–3265. [Online]. Available: https://doi.org/10.1109/CVPR.2012.6248062
[55] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sydney, NSW, Australia, Dec. 2013, pp. 2056–2063. [Online]. Available: https://doi.org/10.1109/ICCV.2013.257
[56] W. Ouyang, X. Zeng, and X. Wang, "Modeling mutual visibility relationship in pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 3222–3229. [Online]. Available: https://doi.org/10.1109/CVPR.2013.414
[57] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)* Columbus, OH, USA, Jun. 2014, pp. 899–906. [Online]. Available: https://doi.org/10.1109/CVPR.2014.120
[58] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2071–2084, Oct. 2015. [Online]. Available: https://doi.org/10.1109/TPAMI.2015.2389830
[59] J. S. J. Ren *et al.*, "Accurate single stage detector using recurrent rolling convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 752–760. [Online]. Available: https://doi.org/10.1109/CVPR.2017.87 and doi: 10.1109/CVPR.2017.87.
[60] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "Looking at pedestrians at different scales: A multiresolution approach and evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3565–3576, Dec. 2016. [Online]. Available: https://doi.org/10.1109/TITS.2016.2561262
[61] D. Park, C. L. Zitnick, D. Ramanan, and P. Dollár, "Exploring weak stabilization for motion feature extraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2882–2889. [Online]. Available: https://doi.org/10.1109/CVPR.2013.371
[62] S. Zhang, R. Benenson, M. Omran, J. H. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1259–1267. [Online]. Available: https://doi.org/10.1109/CVPR.2016.141
[63] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 21–37.
[64] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9. [Online]. Available: https://doi.org/10.1109/CVPR.2015.7298594
[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90
[66] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.07122
[67] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, Sep. 2014. [Online]. Available: http://arxiv.org/abs/1409.1556
[68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255. [Online]. Available: https://doi.org/10.1109/CVPRW.2009.5206848
[69] T. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zürich, Switzerland, Sep. 2014, pp. 740–755.
[70] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3213–3223. [Online]. Available: https://doi.org/10.1109/CVPR.2016.350

XINCHUAN FU received the B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China in 2009 and 2012, respectively, where he is currently pursuing the Ph.D. degree with the National Key Laboratory of Science and Technology on Communications. His research interests include object detection and image processing.

RUI YU received the B.Sc. and M.Sc. degrees in computer science from the Digital Video Processing Group, School of Computer Science, Northwestern Polytechnical University, in 2009 and 2012, respectively, under supervision of Prof. Y. Zhang and Prof. T. Yang. He is currently pursuing the Ph.D. degree as a member of the Vision and Imaging Science Group, Department of Computer Science, University College London, under the supervision of L. Agapito and C. Russell.

WEINAN ZHANG received the bachelor's degree from the ACM Class of Shanghai Jiao Tong University in 2011, and the Ph.D. degree from the University College London in 2016. He is currently a tenure-track Assistant Professor with the John Hopcroft Center for Computer Science, Department of Computer Science, Shanghai Jiao Tong University. His research interests include machine learning and big data mining, particularly, deep learning, and reinforcement learning techniques for real-world data mining scenarios, such as computational advertising, recommender systems, text mining, web search, and knowledge graphs. He and his teammate received the third place in KDD-CUP 2011 for Yahoo! Music recommendation challenge and the final champion in 2013 Global RTB Advertising Bidding Algorithm Competition.

LI FENG received the B.Eng. and M.Eng. degrees in information engineering from the Southwest University of Science and Technology, in 2004 and 2007, respectively, and the Ph.D. degree from the University of Electronic Science and Technology of China in 2017. He is currently an Associate Professor with the Engineering and Technology College, Sichuan Open University. His research interests include cognitive radio networks, energy-efficient transmission, and radio resource management in relay system.

SHIHAI SHAO (S'05–M'10) received the B.E. and Ph.D. degrees in communication and information systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2003 and 2008, respectively. Since 2015, he has been a Professor with the National Key Laboratory of Science and Technology on Communications, UESTC. His current research interests include the design, modeling, and the analysis of full-duplex transceivers, MIMO detection, and all-digital transceivers.