Pedestrian Detection Inspired by Appearance Constancy and Shape Symmetry

Jiale Cao, Yanwei Pang, *Senior Member, IEEE*, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Most state-of-the-art methods in pedestrian detection are unable to achieve a good trade-off between accuracy and efficiency. For example, ACF has a fast speed but a relatively low detection rate, while checkerboards have a high detection rate but a slow speed. Inspired by some simple inherent attributes of pedestrians (i.e., appearance constancy and shape symmetry), we propose two new types of non-neighboring features: side-inner difference features (SIDF) and symmetrical similarity features (SSFs). SIDF can characterize the difference between the background and pedestrian and the difference between the pedestrian contour and its inner part. SSF can capture the symmetrical similarity of pedestrian shape. However, it is difficult for neighboring features to have such above characterization abilities. Finally, we propose to combine both non-neighboring features and neighboring features for pedestrian detection. It is found that non-neighboring features can further decrease the log-average miss rate by 4.44%. The relationship between our proposed method and some state-of-the-art methods is also given. Experimental results on INRIA, Caltech, and KITTI data sets demonstrate the effectiveness and efficiency of the proposed method. Compared with the state-of-the-art methods without using CNN, our method achieves the best detection performance on Caltech, outperforming the second best method (i.e., checkerboards) by 2.27%. Using the new annotations of Caltech, it can achieve 11.87% miss rate, which outperforms other methods.**

***Index Terms*— Pedestrian detection, feature extraction, non-neighboring features, neighboring features, adaboost.**

I. Introduction

PEDESTRIAN detection has the various applications in autonomous driving, auxiliary driving, video surveillance, etc. The process of pedestrian detection mainly consists of feature extraction and classification. And feature extraction is a key for pedestrian detection. There are three main manners for feature extraction: (1) completely Hand-Crafted (HC) features [16], [19], [35], [42], (2) Hand-Crafted candidate features followed by Learning Algorithms (e.g., feature selection algorithm) (HCLA) [11], [13], [36], [53], and (3) Deep Leaning (DL) based features [22], [40], [47]. This paper concentrates on HCLA.

Most state-of-the-art methods generate the candidate features by using local features (e.g., local mean features [4], [11]) or neighboring features (e.g., haar features [53], [54]). These features are the general descriptors. Recently, some specific pedestrian attributes are used for feature design. For example, Zhang *et al.* [52] proposed the InformedHaar features based on that the common sense that pedestrians usually appear up-right. Though its success, it's still the local feature and not good enough to describe the inherent attributes of pedestrian. Appearance constancy and shape symmetry are also two simple and basic attributes. However, the previous methods for pedestrian detection do not make full use of two above attributes. In this paper, we propose two new types of non-neighboring features: side-inner difference features (SIDF) and symmetrical similarity features (SSF). They can describe two above attributes very well. The neighboring features are also proposed for pedestrian detection. The proposed method based on non-neighboring and neighboring features is called NNNF or NNF+NF. NNF contains two types of non-neighboring features: SIDF and SSF. NF means the proposed neighboring features The contributions of the paper are as follows:

1) Appearance constancy and shape symmetry are two inherent attributes of pedestrians. Inspired by these attributes, we propose two new types of features: side-inner difference features (SIDF) and symmetrical similarity features (SSF). SIDF and SSF can abstract the above pedestrian attributes very well. Specifically, SIDF can characterize the difference between the background and pedestrian and the difference between the pedestrian contour and its inner part. SSF can capture the symmetrical similarity of pedestrian shape. Compared to some state-of-the-art features (e.g., LDCF [29] and Checkerboards [53]), our features are oriented non-neighboring features. It's difficult for neighboring features to have such above characterization abilities.
2) Neighboring features (NF) are also designed for pedestrian detection. Both neighboring features (NF) and non-neighboring (NNF) features are assigned large freedom in scale (size), aspect ratio, patch distance, and partition location, resulting in the strong discrimination. We propose to employ both non-neighboring features and neighboring features for

Manuscript received April 25, 2016; revised August 7, 2016; accepted September 1, 2016. Date of publication September 15, 2016; date of current version October 3, 2016. This work was supported in part by the National Basic Research Program of China 973 Program under Grant 2014CB340400 and in part by the National Natural Science Foundation of China under Grant 61632081. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaochun Cao.

J. Cao and Y. Pang are with the School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: connor@tju.edu.cn; pyw@tju.edu.cn).

X. Li is with the State Key Laboratory of Transient Optics and Photonics, Center for OPTical IMagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@outlook.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2609807

pedestrian detection. Among all the selected features, it is found that about 70% of them are neighboring features and 30% of them are non-neighboring ones. Thus, the non-neighboring features are complementary to the neighboring ones. The relationship between our proposed features and some state-of-the-art features is also revealed.

3) Compared to the state-of-the-art methods without using CNN, our method achieves the best detection performance (i.e., 16.20% miss rate on Caltech pedestrian dataset [15]). Meanwhile, our method achieves the best trade-off between efficiency and accuracy only by the common CPU. Using the new annotations of Caltech [54], the miss rate of our method is 11.87%, which also outperforms other methods.

A preliminary version of this work appeared in [8]. This paper extends the earlier work [8] as follows. Firstly, we reveal the relationship between our proposed method and some state-of-the-art methods (e.g., LDCF [29]) through the analysis and the experiment. It demonstrates that our methods incorporate more abundant features. Secondly, we comprehensively compare our method with two state-of-the-art methods without using CNN (i.e., MT-LDCF [51] and Checkerboards [53]) on multiple subsets of Caltech pedestrian dataset. Experimental results show that our method is the best and the most stable. Thirdly, we use more large candidate features or more large training negatives to re-train our method. It achieves a better detection performance than [8]. Using the new and more accurate annotations [54] of Caltech pedestrian dataset, we also re-train our method and compare it with some state-of-the-art methods. Fourth, Experiment results on KITTI dataset [20] are also shown. In addition, we also give more detailed explanations of our method in this paper, for example, the channel-specific normalization in Section III-F.

The rest of this paper is organized as follows. We first review related work in Section II. The proposed methods are then given in Section III. Experimental results are shown in Section IV. Finally, we conclude in Section V.

II. Related Work

Generally, pedestrian detection methods can be divided into three families [2]: DPM (Deformable Part Detectors) variants [18], [19], [27], [46], deep networks [9], [22], [40], [50], and decision forests [13], [34], [53], [54]. Our method can be categorized into the family of decision forests. In this section, we review two important stages in the family of decision forests: feature extraction and classification. Specifically, the process of this kind of methods is as follows: (1) a set of channel images are generated from an input image; (2) then, features are extracted from patches of the channels; and (3) finally, the features are fed into a decision forest learned via AdaBoost.

A. Feature Extraction

Histograms of Oriented Gradients (HOG) [16] is a very famous feature descriptor for pedestrian detection. After that, Dollár *et al.* [13] proposed Integral Channel Features (ICF). Firstly, multiple channel images (HOG+LUV) such as gradient histograms, gradient magnitude, and CIE-LUV color channels are computed. Then, first-order and higher-order features are efficiently generated by using integral images. Based on ICF [13], many methods (e.g., ACF [11], SquaresChnFtrs [4], InformedHaar [52], LDCF [29], and Checkerboards [53]) are proposed. In ACF [8], the original channels (i.e., HOG+LUV) are smoothed and each pixel in the resulting lower resolution channels is used as the feature. SquaresChnFtrs [4] only uses the local sum of squares in each channel as the features. InformedHaar [52] is specifically designed for pedestrian detection where a pool of rectangular templates is tailored to the statistical model of the up-right human body across the channels. LDCF [29] and Checkerboards [53] both convolve original channels with a filter bank. LDCF [29] learns the filter bank by using the technique of Linear Discriminant Analysis (LDA) whereas Checkerboards [53] uses the handcrafted filter bank. In Checkerboards [53], six types of filters are considered: InformedFilters, CheckerboardsFilters, RandomFilters, SquaresChntrs filters, LDCF8 filters, and PcaForeground filters.

SpatialPooling+ [31], [32] does not take channel images as input. Instead, it applies the operator of spatial pooling (e.g., max-pooling) on covariance descriptor and Local Binary Pattern (LBP). Blob-like operator [26] and Torque operator [30] can be also used for image abstraction. Enzweiler and Gavrila [17] proposed a multilevel mixture-of-experts framework by combining the multicue pedestrian classifiers. Khan *et al.* [24] proposed to use action-specific person detection to help the action detection. Depth information [21] is also used to help the detection performance.

Though the above methods have achieved great success in pedestrian detection, they are unable to achieve a satisfying trade-off between accuracy and efficiency. According to [2] and our experimental results, the performance of some above methods can be summarized as follows: On the Caltech pedestrian dataset [7], [15], the decreasing order of the log-average miss rates of the above methods is ICF > ACF > SquaresChnFtrs > InformedHaar > LDCF > SpatialPooling+ > Checkerboards. Loosely speaking, the increasing order of the detection speed of these methods is SpatialPooling+ < ICF < SquaresChnFtrs < Checkerboards < InformedHaar < LDCF < ACF. It can be concluded that it's difficult to simultaneously obtain the lowest log-average miss rate and fastest detection speed.

Recently, the methods based on CNN have achieved great success on pedestrian detection [10], [25], [41], [43], [50]. For example, Tian *et al.* [43] proposed DeepParts to improve the detection performance by handling occlusion with an extensive part pool. Li *et al.* [28] extended Fast R-CNN to scale-aware Fast R-CNN to handle the detection of small pedestrians. CCF [50] extends the original channel features from HOG+LUV based features to CNN based features. In fact, the simple feature design can also be complementary to CNN. For example, by combining the simple local features (e.g, ACF [11], Checkerboards [53], and LDCF [29]) and very deep CNN features (e.g., VGG [38] and AlexNet [23]) , Cai *et al.* [10] could decrease the miss rate from 18.90% to 11.75%. So in this paper, we focus the feature design in the traditional methods.

B. Classification

Once the candidate features are generated, the following step is feature selection and classification. Usually, feature selection and classification are done in a unified framework of AdaBoost. In pedestrian detection, the weak classifiers are either decision forests or simple thresholding classifiers (i.e., the level-1 trees). Weak classifiers are selected and weighted by variants of AdaBoost algorithms.

In order for high efficiency, cascade structure is adopted to select and arrange the weak classifiers in AdaBoost. VJCascade [45] proposed by Viola and Jones is the most classical cascade AdaBoost algorithm. It selects weak classifiers step by step until predefined minimum acceptable detection rate and maximum allowable false positive rate are simultaneously satisfied. Retracting Cascade [5] and BoostChain [49] further improve the detection speed of VJCascade by reusing the weak classifiers of previous stages to construct the new stage. Soft-Cascade [6] sets each weak classifier with a rejection threshold as one stage. Based on the insight that the positive examples rejected by the complete classifier can be safely rejected earlier, MIP [56] was proposed to automatically set the rejection thresholds of a given Soft-Cascade. By adding a complexity term to the objective function, FCBoost [39] jointly accounts for classification accuracy and speed. Crosstalk Cascade [14] makes full use of correlation between detector responses at nearby locations and scales to accelerate detection speed with a little performance loss. To speed up the training of boosted decision trees, Appel *et al.* [1] proposed to prune unpromising features early in the training process.

III. Our Methods

A. Appearance Constancy and Shape Symmetry

Most state-of-the-art features for pedestrian detection are designed to describe the local image variance. Thus, they do not make full use of the inherent attributes of pedestrians. In fact, some attributes of pedestrians can be used to further improve detection performance or accelerate detection speed. For example, prior knowledge that pedestrians above the ground is used for pedestrian detection. Park *et al.* [33] implicitly encoded the information of ground lane to penalize detections which deviate from the ground plane. In [3], Benenson *et al* utilized the information ground lane to accelerate detection speed. Zhang *et al.* [52] incorporated the common sense that pedestrians usually appear up-right into the design of simple and efficient haar-like features. After that, Zhang *et al.* [55] further proposed to use the average contrast maps for pedestrian detection due to the observation that pedestrians indeed exhibit discriminative contrast texture. Though these methods have achieved success, they still does not make full use of the attributes of pedestrians. Appearance constancy and shape symmetry are two simple pedestrian attributes. In this paper, we design two types of no-neighboring features to abstract two above attributes very well. First of all, we give the specific explanations of appearance constancy and shape symmetry. Fig. 1 gives some examples of the cropped pedestrians

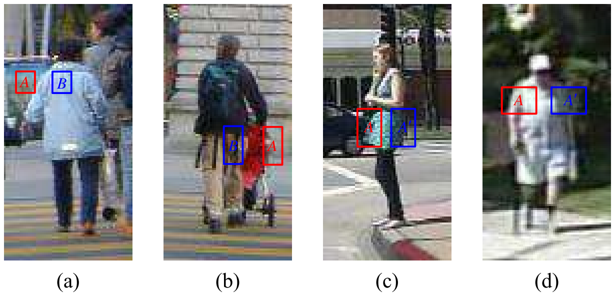


Fig. 1. Some examples of the cropped pedestrians.

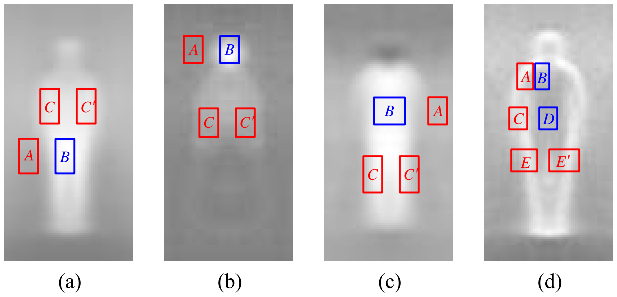


Fig. 2. Average values of channel images . (a) Inversed L (Luminance) channel. (b) U channel. (c) Inversed V channel. (d) G channel.

1) Appearance Constancy: The appearances of pedestrians are usually contrast to the surrounding background. Meanwhile, pedestrians can be seen as three main different parts (i.e., head, upper body, and legs). The appearances of these parts are usually constancy. For example, the woman wears the sky-blue coat and the black pants in Fig. 1(a). We call this inherent attribute of pedestrians *appearance constancy*. Thus, the regions located inside the pedestrians (e.g., patches B in Figs. 1(a) and (b)) are contrast to that located in the background (e.g., patches A in Figs. 1(a) and (b). Note that patches A and B lie in the same horizontal. Patches B are called the inner patches, and patches A are called the side patches.

2) Shape Symmetry: As stated in [52], pedestrians usually appear up-right. Thus, the pedestrian shape is loosely symmetrical in the horizontal direction. For example, the symmetrical regions (patches A and A') in the Figs. 1(c) and (d) have the similar characteristic. This inherent attribute is called *shape symmetry*.

Inspired by the above appearance constancy and shape symmetry, we can design two types of non-neighboring features. The non-neighboring features can describe appearance constancy and shape symmetry. It can be clearly explained by Fig. 2. The average appearances of pedestrians in channel images such as L, U, V, and G are given. Due to the appearance constancy, the pixel values of pedestrians in L, U, and V channel images are similar in the same horizontal, which are different from that of the two-side regions (i.e., background). The difference can be characterized

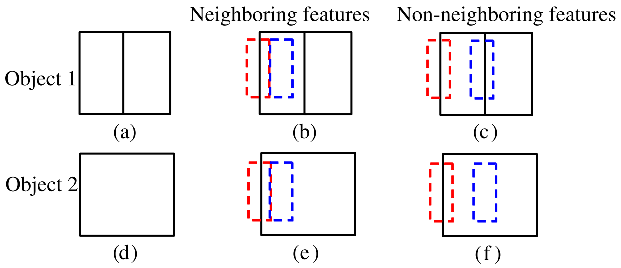


Fig. 3. Demonstration of the discrimination and usefulness of non-neighboring features. (a) and (d) show Object 1 and Object 2, respectively. In (b) and (e), neighboring features are extracted. In (c) and (f), non-neighboring features are extracted.

by the non-neighboring feature formed by patches A and B in Figs. 2(a)-(c). Meanwhile, the pixel values of the inner part of pedestrians in G channel image are constantly small, and the pixel values of pedestrian contour in G channel image are relatively large. Thus, the large difference in G channel image can be characterized by not only the neighboring feature formed by patches A and B but also the non-neighboring feature formed by patches C and D in Fig. 2(d). Due to shape symmetry, the symmetrically non-neighboring regions in the same horizontal have the similar characteristic. For example, the symmetrical patches E and E' in Fig. 2(d) describe the similar edge characteristic, while patches C and C' in Fig. 2 (c) are both bright. Figs. 2(a) and (b) also support it.

The discrimination and usefulness of non-neighboring features are graphically supported by Fig. 3. In Fig. 3, there are two objects (classes) to be classified. We call the object in Fig. 3(a) Object 1 and the object in Fig. 3(d) Object 2. There is a line in the middle of Object 1 whereas the inner part of Object 2 is flat. In both Figs. 3(b) and (e), two neighboring dashed rectangles form a feature. We can see that this neighboring feature is unable to distinguish between Object 1 and Object 2 because the values of neighboring features in Object 1 (i.e., Fig. 3(b)) and Object 2 (i.e., Fig. 3(e)) are equal. Now we use two non-neighboring patches in Figs. 3(c) and (f) to form the features. Because the blue dashed patch in Fig. 3(c) contains a line whereas the blue dashed patch in Fig. 3(f) contains nothing, the non-neighboring features in Object 1 (i.e., Fig. 3(c)) and Object 2 (i.e., Fig. 3(f)) have the different values. Thus, the two objects can be correctly classified according to the different values of the non-neighboring features. It demonstrates the discrimination and usefulness of non-neighboring features.

B. Side-Inner Difference Features Inspired by Appearance Constancy

Inspired by appearance constancy, we design the non-neighboring difference features in the same horizontal. We call the non-neighboring difference feature Side-Inner Difference Feature (SIDF). Fig. 4 gives some possible forms of SIDF. Fig. 4(a) shows that the distance $d(A, B)$ of non-neighboring patches A and B in SIDF can be different. To describe the difference between the background and the pedestrian and the difference between the pedestrian contour and its inner part, the location $l(B)$ of patch B in the interval of the locations $l(A)$ and $l(A')$ where patch A' is the horizontal mirror of patch A. That is, $l(B) \in [l(A), l(A')]$. As demonstrated in Fig. 5, $l(B)$ is randomly sampled from $[l(A), l(A')]$ in our experiments.

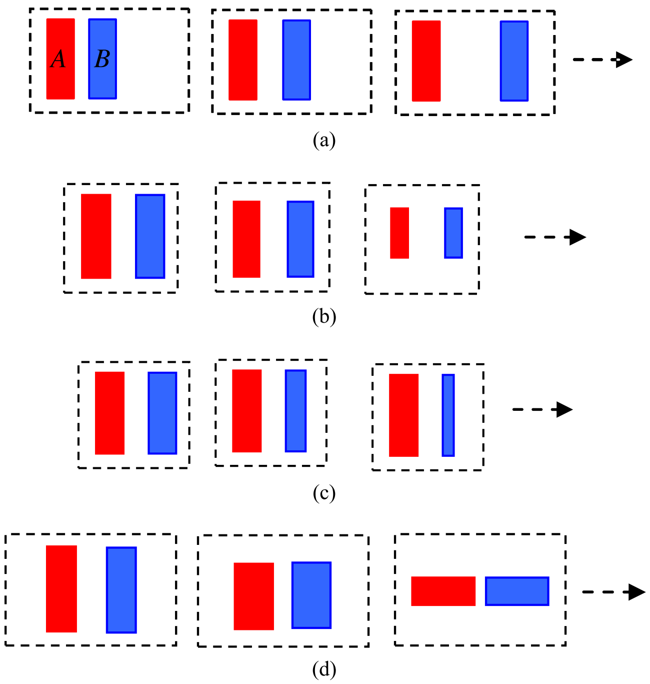


Fig. 4. Some possible forms of side-inner difference features (SIDF). (a) Varying distance between two patches. (b) G Varying size of two patches. (c) Varying size of one patch with the other fixed. (d)Varying aspect ratio.

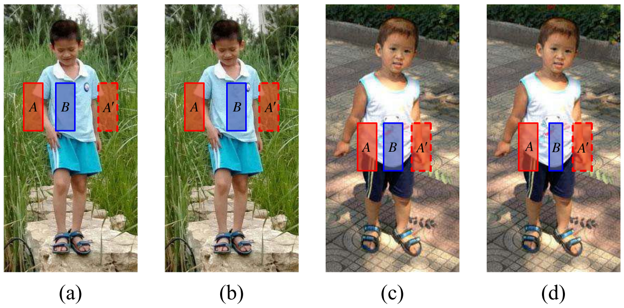


Fig. 5. The patch B is randomly located between the patch A and its horizontal mirror A'. The locations of patch B in (a) and (b) are different. But they are both among patch A and its mirror A'. (c) and (d) show that the width of patch B can be changed.

Both Figs. 4(b) and (c) show varying sizes of patches. But in Fig. 4(b) both two non-neighboring patches equally vary with size (scale) whereas in Fig. 4(c) only one patch varies its size. It's good enough for letting patches A and B have the different width but the same height. Figs. 5(c) and (d) also give an example of the different widths of patches A and B. Fig. 4(d) shows SIDF with varying aspect ratio.

The size of a patch (e.g., patch A in Fig. 5(a)) is allowed to change in a reasonable range. In this paper, the variation of a patch is limited to a maximum square. In other words,

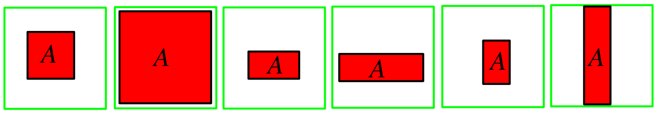


Fig. 6. The size of patch is allowed to change inside the maximum square indicated by green squares.

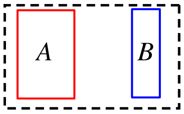


Fig. 7. A side-inner difference feature.

Algorithm 1 Generating the Candidate Features of SIDF

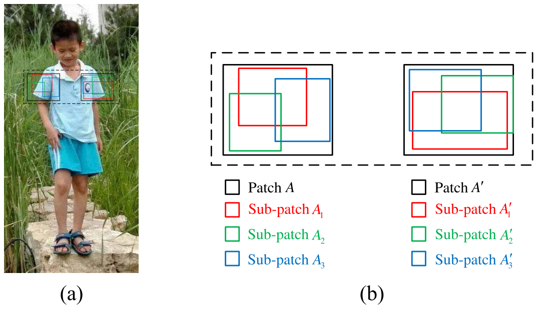
```
Initialize the feature pool $T$ to be empty;
for $y = 1, x = 1$ to $H$, $W$ do
    for $h = 1, w = 1$ to $h_m$, $w_m$ do
        if $x + w - 1 < H$ & $y + h - 1 < W$ then
            $x_A = x, y_A = y, w_A = w, h_A = h$;
            if $x + w - 1 < W/2$ then
                $dst = 2 \times (W/2 - (x + w - 1))$;
                $w_B = randi([1, min(w_m, dst - 1)]), h_B = h$;
                $x_B = randi([x + w + 1, W - x - w + 2 - w_B]), y_B = y$;
                Compute SIDF by Eq. (1) and add it to $T$;
            end if
            if $x > W/2 + 1$ then
                $dst = 2 \times (x - (W/2 + 1))$;
                $w_B = randi([1, min(w_m, dst - 1)]), h_B = h$;
                $x_B = randi([W + 2 - x, x - w_B - 1]), y_B = y$;
                Compute SIDF by Eq. (1) and add it to $T$;
            end if
        end if
    end for
end for
```

the sizes of both patches A and B are allowed to be not larger than the size of the maximum square. The green squares in Fig. 6 are maximum squares and patches have to be inside them. A typical maximum square is of size 8×8 cells (1 cell=2×2 pixels).

Suppose that the side-inner difference feature $f(A, B)$ consists of two patches A and B (see Fig. 7). The numbers of pixels of patches A and B are denoted by N_A and N_B, respectively. Let S_A and S_B be the pixel sum of patches A and B in a channel image, respectively. Then the side-inner difference feature $f(A, B)$ can be calculated by

$$f(A, B) = \frac{S_A}{N_A} - \frac{S_B}{N_B}, \tag{1}$$

where the weights $1/N_A$ and $1/N_B$ are used for normalization. Algorithm 1 gives the process for generating the candidate feature pool of SIDF. W and H is the detection model width and height. w_m and h_m is the maximum width and height of the patch.



(a) (b)

Fig. 8. Examples of symmetrical similarity features.

C. Symmetrical Similarity Features Inspired by Shape Symmetry

As stated in Section III-A, the shape of pedestrian is loosely symmetrical. Thus, patches A and A' in Fig. 5 have the similar characteristic. Inspired by this characteristic, we design the Symmetrical Similarity Features (SSF). The symmetrical similarity features $f(A, A')$ of patches A and A' can be calculated by the following equation:

$$f(A, A') = |f_A - f_{A'}|, \tag{2}$$

where f_A and $f_{A'}$ represent the features of patches A and A', for example, histogram features and local mean features. For the efficiency of computation, we just use the local mean features to represent the patches. Namely, $f_A = S_A/N_A$ and $f_{A'} = S_{A'}/N_{A'}$. Then, Eq. (2) can be written as the following equation:

$$f(A, A') = |\frac{S_A}{N_A} - \frac{S_{A'}}{N_{A'}}|. \tag{3}$$

However, due to the changes of the pedestrian posture, the pedestrian symmetry is relatively loose. It results that Eq. (3) is very sensitive to the pedestrian deformation.

To eliminate the above influence caused by pedestrian deformation, we replace the local mean features of patches by the max-pooling features [48]. In Fig. 8, two symmetrical patches A and A' are represented by three different color sub-patches, respectively. For example, patch A consists of three sub-patches A_1, A_2, and A_3. They are randomly generated inside the patch A. The size and aspect ratio of them can be arbitrary, whereas the area of them should be larger than half of patch A. Then, the feature value of patch A is set as the maximum of mean values of three sub-patches. It can be expressed as:

$$f_M(A) = \max_{i=1,2,3} \frac{S_i}{N_i}. \tag{4}$$

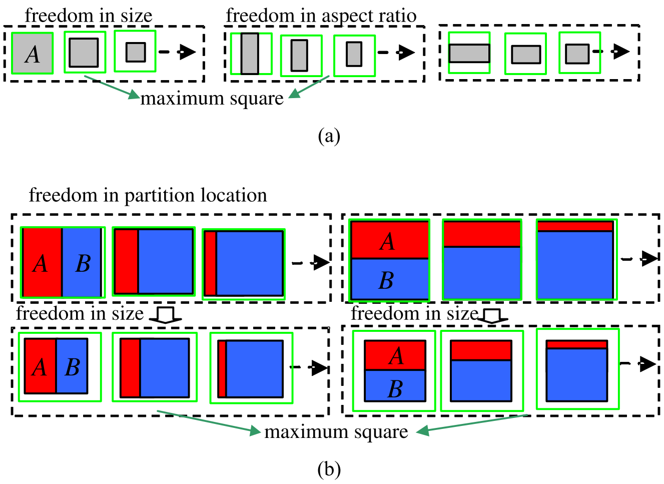
Note that the maximum is replaced by minimum in L and V channel images. Then, the symmetrical similarity feature $f(A, A')$ of patches A and A' is calculated by the following equation:

$$f(A, A') = |f_M(A) - f_M(A')|. \tag{5}$$

Algorithm 2 gives the process for generating the candidate feature pool of SSF. W and H is the detection model width and height. w_s and h_s is the minimum width and height of

Algorithm 2 Generating the Candidate Features of SSF

```
Initialize the feature pool $T$ to be empty;
for $y = 1$, $x = 1$ to $H$, $W$ do
    for $h = h_s$, $w = w_s$ to $h_m$, $w_m$ do
        if $x + w - 1 < W/2$ & $y + h - 1 < W$ then
            $x_A = x, y_A = y, w_A = w, h_A = h$;
            $x_B = W + 1 - x, y_B = y, w_B = w, h_B = h$;
            Randomly sample three sub-patches inside patch
            $A$ and patch $B$, respectively;
            Compute SSF by (5) and add it to $T$;
        end if
    end for
end for
```



(a)

(b)

Fig. 9. Some possible forms of neighboring features. The green squares are called maximum squares. (a) Local mean features. (b) Neighboring difference features.

the patch. w_m and h_m is the maximum width and height of the patch. The size of the symmetrical patches A and A' is allowed to change in a reasonable range, which varies from 6×6 cells to 12×12 cells. As the symmetry in pedestrians mainly exists in L, U, V, and G channel images, we only use the above channel images to generate SSF.

D. Neighboring Features

In Section III-B and III-C, we introduce two types of non-neighboring features (i.e., SIDF and SSF) based on appearance constancy and shape symmetry, respectively. In fact, both non-neighboring features and neighboring features are crucial for pedestrian detection. As stated in Section II-A, most of state-of-the-art features belong to neighboring features. ICF [13], ACF [11], and SquaresChnFtrs [4] are very simple and efficient to be computed by the trick of integral image. InformedFilters [52] and Checkerboards-Filters [53] are rich in representation but in-efficient in computation due to the relatively high complexity of the features.

In this section, we propose to form the pool of neighboring features by using local mean features (see Fig. 9(a)) and neighboring difference features (see Fig. 9(b)) with enough freedom in size, patch direction, aspect ratio, and partition location. The left portion of Fig. 9(a) shows that the size of a feature is allowed to vary in a large extent. Patch direction is either vertical or horizontal. The patch direction in the middle of Fig. 9(a) and the left portion of Fig. 9(b) is vertical whereas the direction in the right portion of Fig. 9(a) and the right portion of Fig. 9(b) is horizontal.

Partition location is illustrated in Fig. 9(b) which is defined as the location where two neighboring patches intersect. Partition direction can be horizontal and vertical. Assigning large freedom in partition location and partition direction can strengthen the representative ability and the discriminative ability of the neighboring difference features.

To avoid the large number of the candidate features, we specify a maximum square. The sizes of local mean features and neighboring difference features are not allowed to be larger than the size of the maximum square. The green squares in Fig. 9 represent the maximum squares. As stated in Section III-B, a typical size of the maximum square is 8×8 cells.

The local mean features and neighboring difference features illustrated in Fig. 9 are suitable to be quickly computed with integral image. Hence the feature extraction process is very efficient. Note that neighboring difference features can be calculated using the same formula (i.e., Eq. (1)) of non-neighboring features.

In our method, both the neighboring features (i.e., local mean features and neighboring difference features) and non-neighboring features (i.e., SIDF and SSF) are used for generating the candidate features. Soft-Cascade AdaBoost [6] is used for learning the strong classifier from the candidate features. The rejected threshold of each stage is the same as the default value (i.e., -1) in [44]. To improve performance [13], [53], level-2 and level-4 decision trees are also used based on AdaBoost.

E. Relation to Other Methods

Figs. 7, 8, and 9 show the proposed non-neighboring features and neighboring features. It can be seen that a non-neighboring feature (e.g., SIDF) is also a difference feature. If two non-neighboring patches are close enough (i.e., adjacent), then the non-neighboring feature becomes neighboring feature. Therefore, neighboring difference features can be seen as the special cases of non-neighboring ones.

ICF [13], ACF [11], and SquaresChnFtrs [4] are subsets of the local mean features shown in Fig. 9(a). LDCF [29] and FCF [53] are neighboring features. The relationship of different features is shown in Table I.

Moreover, binary versions of LDCF features [29] are also special cases of neighboring features shown in Fig. 9. In LDCF, a filter bank is obtained by an LDA technique. Convolving the filter banks with the channel images results in LDCF features. Fig. 10 shows the first three LDCF filters on 10 channel images (i.e., HOG+LUV) for a 4×4 patch. Three filters in the first column of Fig. 10 are chosen and their values are shown in Fig. 11. All the values in Fig. 11(a) are approximately equal. In Fig. 11(b), the values of the first

TABLE I
RELATIONSHIP OF DIFFERENT FEATURES

Feature	ICF, ACF, SquaresChnFtrs, LDCF, FCF	SIDF, SSF
Category	neighboring	non-neighboring

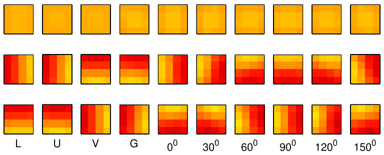


Fig. 10. The first 3 filters on 10 channel images (i.e., HOG+LUV) for a 4 × 4 patch.

$$\begin{bmatrix} 0.234 & 0.252 & 0.252 & 0.233 \\ 0.247 & 0.267 & 0.267 & 0.247 \\ 0.247 & 0.267 & 0.267 & 0.247 \\ 0.233 & 0.252 & 0.252 & 0.234 \end{bmatrix}$$

(a)

$$\begin{bmatrix} -0.308 & -0.130 & 0.143 & 0.317 \\ -0.334 & -0.144 & 0.149 & 0.337 \\ -0.337 & -0.149 & 0.144 & 0.334 \\ -0.317 & -0.143 & 0.130 & 0.308 \end{bmatrix}$$

(b)

$$\begin{bmatrix} -0.316 & -0.343 & -0.338 & -0.306 \\ -0.136 & -0.145 & -0.140 & -0.125 \\ 0.125 & 0.140 & 0.144 & 0.136 \\ 0.306 & 0.338 & 0.343 & 0.316 \end{bmatrix}$$

(c)

Fig. 11. The filter values in the first column of Fig. 10.

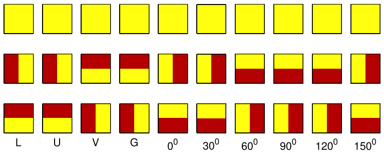


Fig. 12. Binary LDCF filters corresponding to Fig. 10.

two columns are negative and those of the last two columns are positive. In Fig. 11(c), the values of the first two rows are negative and those of the last two rows are positive. Therefore, we propose to binarize the filters by threshold zero. The binarized filters corresponding to Fig. 10 are shown in Fig. 12.

Comparing Fig. 12 with Fig. 9, one can conclude that the binary filters of LDCF are a small subset of the proposed neighboring features/filters. Denote the pedestrian detection method with the binary filters by LDCF-B. The ROC curves of LDCF and LDCF-B on the Caltech pedestrian detection dataset (the Caltech training set is used for training) are shown in Fig. 13. The source codes of LDCF are from Dollar's toolbox [44]. The log-average miss rates of LDCF and LDCF-B are 29.93% and 29.52%, respectively. LDCF-B is comparable and even slightly better than LDCF. Because our method employs much richer and more discriminative features compared to LDCF-B, our method gets much better results than LDCF-B and LDCF as well.

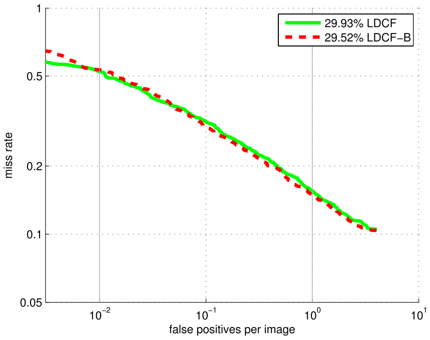


Fig. 13. ROC curves of LDCF and LDCF-B on Caltech dataset [15].

F. Channel-Specific Normalization

In ICF [13], ACF [11], LDCF [29], Checkerboards [53], and so on, the features extracted from channel images are directly used as the input of decision forests and AdaBoost. We propose to properly normalize both the non-neighboring and neighboring features. Many methods adopt the same normalization strategy for all channel images. Instead, we propose a channel-specific normalization scheme.

In our channel-specific normalization scheme, features corresponding to different channels employ different normalization approaches. The reason is that different channels have different physical meaning and property.

In classical sliding-window based detection approach, a detection window slides over an image. Let x be a feature in a detection window. Denote μ and σ as the mean and variance of the corresponding channel images in the detection window. Because L (Luminance) channel is sensitive to variations in lighting conditions, zero-mean and unit variance approach is employed for normalization. Formally, the normalization function $f(x)$ is expressed as:

$$f(x) = \frac{x - \mu}{\sigma}. \tag{6}$$

In fact, the normalization in Eq. (6) has been widely used in many applications [45].

As to the U and V channels, we do not perform normalization because they are relatively stable when there are variations in illumination.

Finally, the normalization function $f(x)$ for the gradient magnitude (G) and six oriented gradients is given by:

$$f(x) = \frac{x}{\mu_G}, \tag{7}$$

where μ_G is the mean of the gradient magnitude channel (G) in the detection window.

Table II clearly shows the proposed scheme of channel-specific normalization.

Our contributions mainly lie in the proposed non-neighboring features (i.e., SIDF and SSF) and neighboring features (NF). But the channel-specific normalization is slightly and stably helpful for improving performance of pedestrian detection.

TABLE II
CHANNEL-SPECIFIC NORMALIZATION

Channel	L	U	V	G	6 Oriented gradients
Method	$\frac{x-\mu}{\sigma}$	x		$\frac{x}{\mu_G}$	

IV. EXPERIMENTS

INRIA dataset [16], Caltech pedestrian dataset [7], [15], and KITTI dataset [20] are employed for evaluation. In the INRIA dataset, there are 1237 pedestrians used for training and 288 pedestrian images used for evaluation. By simple geometric transformation (e.g., flip and translation) of the 1237 pedestrian images [44], there are 22,666 positive samples in our training set.

The Caltech pedestrian dataset is more challenging than the INRIA dataset and hence has become a benchmark of pedestrian detection. It consists of approximately 10 hours of 640 × 480 30Hz video taken from a vehicle driving through regular traffic in an urban environment [7]. The 10 hours data consists of 11 videos with the first 6 videos used for training and the last 5 videos for testing. There are 2300 unique pedestrians in the dataset. The standard positive training data is formed by sampling one image out of each 30 sequential frames. As a result, there are 4250 frames in which there are 1631 positive samples. A positive sample is a subimage which tightly contains a pedestrian. The corresponding training data is called Caltech training set. To enlarge the number of training samples, we sample a frame from every 15 or 3 frames instead of 30 successive frames, respectively. The result training sets are called Caltech 2× and Caltech 10×, respectively [53]. Whenever Caltech 2× training set or Caltech 10× training set is used, the testing set is the same. The testing set consists of 4024 frames among which there are 1014 positive images. The miss rate is log-average over the FPPI of [10^{-2},10^{0}] unless noted otherwise. FPPI means False Positve Per Image.

KITTI dataset [20] is a challenging benchmark used for stereo, optical flow, visual odometry, and object detection. Pedestrian detection is a subtask of object detection on KITTI dataset. For pedestrian detection, the benchmark consists of 7481 training images and 7518 test images. For evaluation of experimental results, precision-recall curve is used.

A. Self-Comparison Using the Caltech 2× Training Set

Before comparing with the state-of-the-art methods, experimental results on Caltech 2× dataset are reported to show how the proposed method works and the importance of each component of the proposed method. Note that the Caltech 2× training set instead of Caltech 10× training set is used.

The experimental setup is as follows. Classical 10 channel images (i.e., HOG+LUV) are used for generating the candidate features. The final classifier consists of 4096 level-2 decision trees. The classifier is learned by five rounds, where the numbers of trees in subsequent rounds are 32, 128, 512, 2048, and 4096, respectively. Each tree is built by randomly sampling 1/32 of features from the large pool of features.

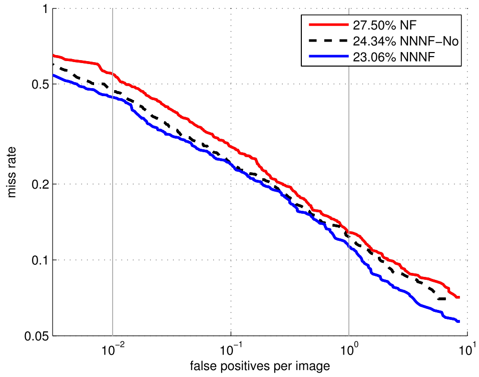


Fig. 14. Self-comparison: ROC curves of NF, and NNNF-No, and NNNF on the Caltech dataset.

TABLE III
COMPARISON OF LOG-AVERAGE MISS RATES

Method	MR	Δ MR
NF	27.50%	N/A
NF+SIDF	25.67%	+1.83%
NF+SSF	25.20%	+2.30%
NNNF-No	24.34%	+3.16%
NNNF	23.06%	+4.44%

5000 hard negatives are added after each round and the cumulative negatives are limited to 15000. The stride of sliding windows is 4 pixels. The model size is 64×128, which consists of 2048 cells (1 cell=2×2 pixels). As the height of the reasonable pedestrian in Caltech dataset is generally taller than 50 pixels, each testing image is upsampled by one octave. ROC curve is generated by pairs of miss rate and FPPI.

In NNNF (a.k.a. NNF+NF), both Non-Neighboring Features (NNF) and Neighboring Features (NF) are employed. In the NNF, there are two types of non-neighboring features: SIDF and SSF. NF+SIDF or NF+SSF means that the neighboring features (i.e., NF) are combined with only one type of non-neighboring features (i.e., SIDF or SSF). In SIDF and NF, the channel-specific normalization stated in Section III-F can be used. We denote NNNF-No the method which is the same as NNNF except that no normalization is conducted in SIDF and NF.

The ROC curves of NF, NNNF-No and NNNF are shown in Fig. 14. It is seen that the performance of NNNF is systematically better than that of NF, meaning that incorporating NNF is useful for improving detection performance. Meanwhile, one can observe that NNNF-No is inferior to NNNF. NNNF employs channel-specific normalization (see Section III-F) in NF and SIDF whereas NNNF-No does not perform normalization. So it is concluded that pedestrian detection benefits from the proposed channel-specific normalization.

The above observation can also be seen from Table III where the log-average miss rates are given. Specifically, the log-average miss rates of NNNF (i.e., NNF+NF), NNNF-No, and NF are 23.06%, 24.34%, and 27.50%, respectively. The log-average miss rate of NNF+NF is 4.44% smaller than that of NF. So it is said that non-neighboring

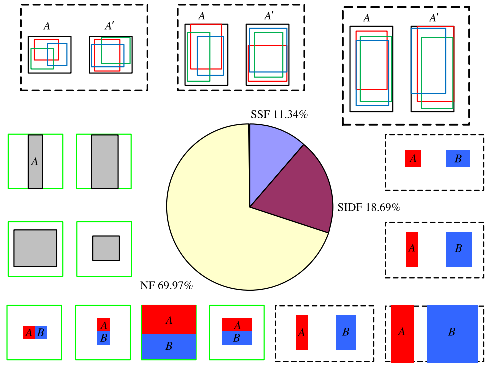


Fig. 15. Among all the selected features, about 30% are non-neighboring features and 70% are neighboring features. Several representative non-neighboring features and neighboring features are also shown.

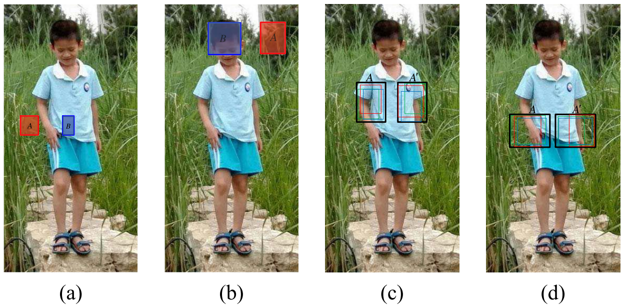


Fig. 16. Several selected non-neighboring features. The first two features are side-inner difference features, and the last two features are symmetrical similarity features.

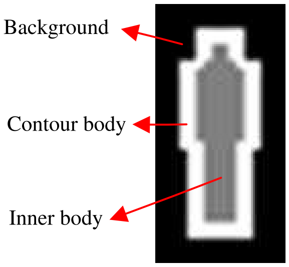


Fig. 17. A ternary model divides a normalized pedestrian image into three regions: background, contour body, and inner body.

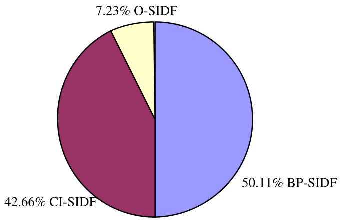


Fig. 18. The proportions of CI-SIDF, BP-SIDF, and O-SIDF features.

features contribute significantly for improving detection performance. Specifically, NF+SIDF and NF+SSF outperform NF by 1.83% and 2.30%, respectively. It means that SIDF and SSF can boost the detection performance of NF by incorporating some complementary information, which is explained in the next paragraph. NNNF outperforms NNNF-No by 1.28%. Though the contribution of channel-specific normalization is not as significant as non-neighboring features, it is steadily helpful for improving detection performance.

Totally, 12288 features are selected, which consist of 3690 non-neighboring features (NNF) and 8598 neighboring features (NF). Among non-neighboring features NNF), there are 2297 side-inner difference features (SIDF) and 1393 symmetrical similarity features (SSF). That is, the proportions of SIDF, SSF, and NF are approximately 18.69%, 11.34% and 69.97% (see Fig. 15). We can conclude that non-neighboring features are complementary to neighboring features. Several representative forms of non-neighboring (SIDF and SSF) and neighboring features (NF) are also shown in Fig. 15.

In Fig. 16, the representative non-neighboring features are also visualized on pedestrian images. The first two images show the side-inner difference features (SIDF), and the last two images show the symmetrical similarity features (SSF).

In fact, SIDF features can be categorized into the following three types. (1) A SIDF feature is called Contour-Inner SIDF (CI-SIDF) feature if one of its patch is located on the contour of a pedestrian and the other patch is inside of the pedestrian; (2) A SIDF feature is called Background-Pedestrian SIDF (BP-SIDF) feature if one of its patch is on the background and the other patch is inside or on the contour of a pedestrian; and (3) A SIDF feature different from CI-SIDF and BP-SIDF features is called Other SIDF (O-SIDF) feature. To know the proportions of the three types of SIDF features, a ternary model (Fig. 17), consisting of background, contour body, and inner body, is created according to average appearance (e.g., Fig. 2(d)) of pedestrians. All the 2297 selected SIDF features are classified to CI-SIDF, BP-SIDF, and O-SIDF by computing the intersection of a SIDF feature and the ternary model. The results given in Fig. 18 indicate that the proportions of CI-SIDF, BP-SIDF, and O-SIDF are 42.66%, 50.11%, and 7.23%, respectively. Fig. 18 tells that SIDF features not only capture the difference the contour of a pedestrian and its inner part but also utilize the difference between the background and a pedestrian. Background can be regarded as the context of a pedestrian image and hence context has been proved to be effective in object detection and recognition. It is difficult for neighboring features to utilize the context information.

B. Comparison With State-of-the-Art Methods on Caltech Dataset

The proposed NNNF method can adopt different levels (depths) of decision trees. In this section, NNNF-L2 stands for the NNNF method where level-2 trees are utilized. The Caltech 2× training set is used for NNNF-L2. All parameters in NNNF-L2 are the same as those in Section IV-A. In NNNF-L4, level-4 trees are employed. The Caltech 10× training set is used for NNNF-L4. The resulting classifier is composed of 4096 level-4 decision trees. Different from [8], each tree is built by sampling all the candidate features from the feature pool. The decision trees are obtained after five rounds with real AdaBoost. In each round, 20000 hard

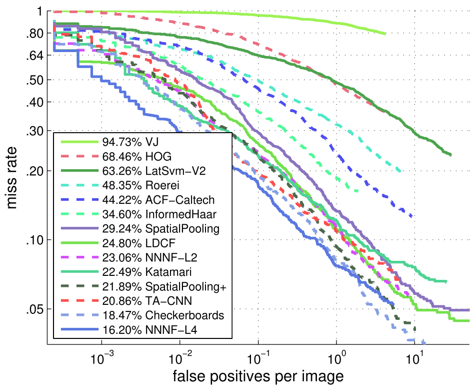


Fig. 19. Comparison with state-of-the-art methods on the Caltech dataset (reasonable).

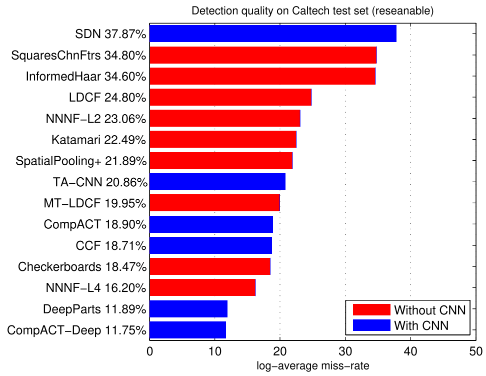


Fig. 20. Miss rate of the state-of-the-art methods. The methods represented by blue bars are based on CNN. The methods with red bars are not using CNN.

TABLE IV

Miss Rates of Some State-of-the-Art Methods Without Using CNN on Various Subsets of the Caltech Dataset Are Shown

Subset	MT-LDCF [51]	Checkerboards [53]	NNNF-L4
Reasonable	19.95%	18.47%	**16.20%**
All	69.98%	68.75%	**67.38%**
None Occlusion	17.75%	16.11%	**14.19%**
Partial Occlusion	36.77%	36.20%	**32.01%**
Heavy Occlusion	76.98%	77.50%	**74.92%**
Mean	44.29%	43.41%	**40.94%**

negatives are added and the cumulative negatives are limited to 50000. Other parameters are the same as those in Section IV-A.

Fig. 19 compares the proposed NNNF-L2 and NNNF-L4 with the state-of-the-art methods: VJ [45], HOG [16], LatSvm-V2 [19], Roerei [4], ACF-Caltech [11], InformedHaar [52], SpatialPooling [31], LDCF [29], Katamari [2], SpatialPooling+ [32], [37], TA-CNN [41], and Checkerboards [53]. In Fig. 19, the curves of ACF-Caltech are obtained when they are trained on the Caltech training set. The models of VJ, HOG, LatSvm-L2, and Roerei are trained on the INRIA dataset. The curves of other methods are obtained when the training set is Caltech 10×. VJ, Roerei, ACF, InformedHaar, SpatialPooling, LDCF, Katamari, SpatialPooling+, and Checkerboards are based on AdaBoost classifier. HOG and LatSvm are based on SVM classifier. TA-CNN is based on CNN. They all utilize the Caltech testing set for evaluation.

The following observations can be seen from Fig. 19. Even the small Caltech 2× training set is used, the proposed NNNF-L2 is better than LDCF [29] which is trained from the large Caltech 10× training set. Specifically, the log-average miss rate of NNNF-2 is 23.06%. It can also be seen from Fig. 19 that the proposed NNNF-L4 is superior to all other methods (VJ [45], HOG [16], LatSvm-V2 [19], Roerei [4], ACF-Caltech [11], InformedHaar [52], SpatialPooling [31], LDCF [29], Katamari [2], SpatialPooling+ [32], [37], and TA-CNN [41]). The log-average miss rate of NNNF-L4 is as small as 16.20% whereas the log-average miss rate of TA-CNN [41] and Checkerboards [53] are 20.86% and 18.47%, respectively. Though the proposed non-neighboring and neighboring features are much simpler than the features in convolutional neural networks [41] and Checkerboards [53], the log-average miss rate of NNNF-L4 is lower than that of TA-CNN [41] and Checkerboards [53] by 4.66% and 2.27%, respectively.

According to whether using CNN or not, Fig. 20 divides the state-of-the-art methods into two classes. In the first class, the methods with red bars do not use CNN. NNNF-L4 achieves the best detection performance, outperforming Checkerboards [53] by 2.27%. In the second class, the methods with blue bars are based on CNN. CompACT-Deep [10] achieves the lowest miss rate (i.e., 11.75%) by combination of some local channel features (e.g., ACF [11], Checkerboards [53], and LDCF [29]) and deep features (e.g., VGG [38]). Though CompACT-Deep has a better performance than NNNF-L4, the improvement of CompACT-Deep are based on very deep CNN model (i.e., VGG [38]). When only using the above local features and small CNN, CompACT can only achieve 18.90%, which is inferior to NNNF-L4. It means that NNNF-L4 are much more effective than the local features used in CompACT. Moreover, with very deep CNN (e.g., VGG [38]), NNNF can also boost the detection performance [9] (i.e., 10.4% miss rate on Caltech).

Table IV further compares two state-of-the-art methods without using CNN (i.e., MT-LDCF [51] and Checkerboards [53]) with our NNNF-L4 under "Reasonable", "All", and three different occlusion subsets of Caltech dataset (i.e, no occlusion, partial occlusion, and heavy occlusion). It can be seen that NNNF-L4 stably outperforms than MT-LDCF [51] and Checkerboards [53] on the all subsets. For example, when the subset is "Heavy Occlusion", NNNF-L4 outperforms MT-LDCF [51] and Checkerboards [53] by 2.06% and 2.48%, respectively. The mean miss rates of MT-LDCF [51], Checkerboards [53], and NNNF-L4 over the all subsets are 44.29%, 43.41%, and 40.94%, respectively. Thus, NNNF outperforms MT-LDCF [51] and Checkerboards [53] by 3.35% and 2.47%, respectively. It means that the detection performance of NNNF-L4 is the best and the most stable in the methods without using CNN.

TABLE V

DETECTION SPEED (FPS) AND MISS RATE (MR) ON THE CALTECH DATASET

Method	FPS	MR
LatSvm-V2 [19]	0.16	63.30%
Crosstalk [14]	14.10	53.90%
ACF [11]	9.49	44.20%
InformedHaar [52]	0.63	34.60%
SpatialPooling [31]	0.12	29.24%
LDCF [29]	3.62	24.80%
SpatialPooling+ [32]	0.12	21.89%
Checkerboards [53]	0.50	18.47%
NNNF-L2	1.61	23.06%
NNNF-L4	1.13	16.20%

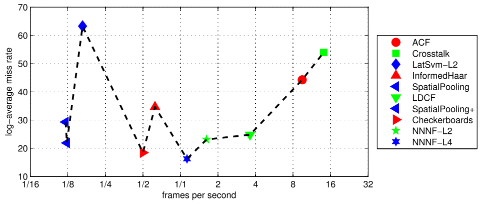


Fig. 21. Miss rates versus frames per second (FPS).

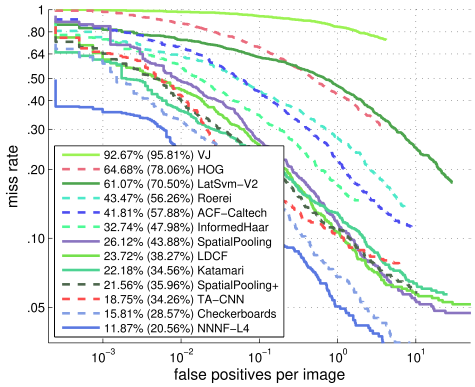


Fig. 22. Comparison with state-of-the-art methods on the Caltech dataset using the new and accurate annotations [54]. MR_{-2} (MR_{-4}) are shown in the legend.

In Table V, the detection speed (FPS) of some state-of-the-art methods without using CNN is shown. The log-average miss rate of NNNF-L4 is lower than that of Checkerboards [53] and the detection speed of NNNF-L4 is also 2.26 times faster than that of Checkerboards. Though Crosstalk [14] has the fastest detection speed, it has the worst detection performance. The log-average miss rates and FPS (Frames per Second) of the methods are also visualized in Fig. 21. It is desirable if miss rate is as small as possible and FPS is as large as possible. So Fig. 21 implies that the proposed NNNF-L4 achieves the best trade-off between miss rate and FPS. Note that the detection speed is measured on a computer with an Intel Core i7 CPU and a 640 × 480 image with the height of pedestrians not less than 50 pixels. GPU is not used in our experiments.

Based on the new and accurate annotations of the Caltech test set [54], Fig. 22 compares NNNF-L4 with some state-of-the-art methods. NNNF-L4 is re-trained based on the new annotations of the Caltech training set. The standard miss rate over the FPPI range of $[10^{-2},10^{0}]$ and more strict miss rate over the FPPI range of $[10^{-4},10^{0}]$ are both shown in the legend [54]. They are represented by MR_{-2} and MR_{-4}, respectively. MR_{-2} and MR_{-4} of NNNF-L4 are 11.87% and 20.56%, respectively. They outperforms that of Checkerboards [53] by 3.94% and 8.02%, respectively. Based on MR_{-2} of Fig. 22, Fig. 23 classifies and ranks the state-of-the-art methods. The methods with red bars do not use CNN. Among these methods, NNNF-L4 also achieves the state-of-the-art.

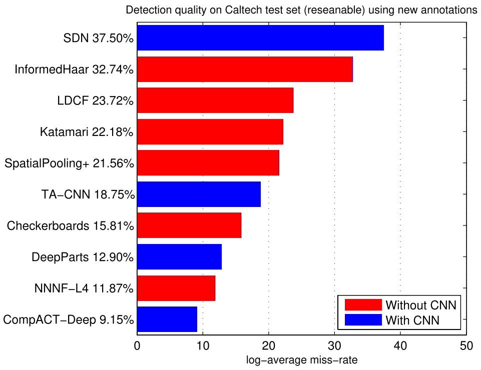


Fig. 23. Miss rate of the state-of-the-art methods using the new and accurate annotations. The methods represented by blue bars are based on CNN. The methods with red bars are not using CNN.

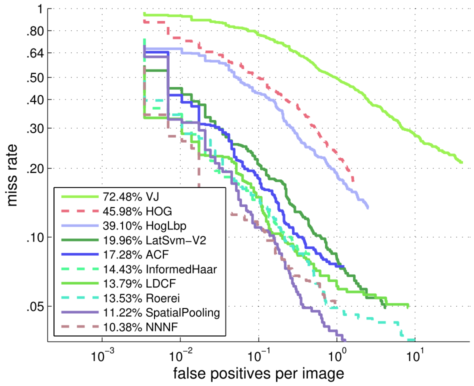


Fig. 24. Comparison with some state-of-the-art methods on the INRIA dataset.

C. Comparison With State-of-the-Art Methods on the INRIA Dataset

Experiments are also conducted on the INRIA dataset. Because pedestrian height in both the training and testing sets are larger than 100 pixels, we train a model with 64 × 128 pixels. Different from [8], the training images are upsampled by one octave in order to enlarge the number of training negatives. In the testing process, the image is not upsampled. The model consists of 2048 level-3 decision trees. The decision trees are obtained after four rounds, where the numbers of trees in each round are 32, 128, 512, and 2048, respectively. 10000 hard negatives are

TABLE VI

DETECTION SPEED (FPS) AND MISS RATE (MR) ON THE INRIA DATASET

Method	FPS	MR
Crosstalk [14]	45.40	20.10%
LatSvm-V2 [19]	0.60	19.96%
ACF [11]	31.90	17.28%
InformedHaar [52]	N/A	14.43%
LDCF [29]	4.70	13.79%
SpatialPooling [31]	0.14	11.22%
NNNF	5.15	10.38%

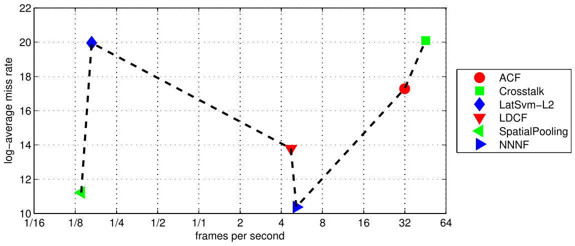


Fig. 25. Miss rates versus frames per second (FPS).

added after each round and the cumulative negatives are limited to 20000.

Experimental results are shown in Fig. 24. It can be observed that the proposed NNNF achieves the best performance (i.e., log-average miss rate is 10.38%). Please note that the miss rate of NNNF is lower than that of [8]. It means that the large number of training negatives is very important. The miss rate of NNNF is 9.58%, 6.90%, 4.05%, and 3.41% lower than that of LatSvm-V2 [19], ACF [11], InformedHaar [52], and LDCF [29], respectively. NNNF outperforms SpatialPooling [31] by 0.84%.

The comparison of detection speed (FPS) and miss rate of different methods is given in Table VI. The image to be detected has 640×480 pixels and the height of pedestrians is not less than 100 pixels. One can see from Table VI that NNNF outperforms all the methods in terms of log-average miss rate. For example, the miss rate of NNNF is 0.84% lower than that of SpatialPooling [31] and the detection speed of NNNF is 36.79 times faster than that of SpatialPooling. Therefore, our method is able to get the best trade-off between miss rate and detection speed. The superiority in trade-off can also be observed from Fig. 25.

D. Comparison With State-of-the-Art Methods on the KITTI Dataset

In this section, NNNF is compared to some state-of-art methods without using CNN (i.e., ACF [11], SquaresChnFtrs [4], SpatialPooling+ [32], and Checkerboards [53]). We also train a model with 64×128 pixels. Because the minimum height of pedestrian for evaluation is 25 pixels, the image is upsampled by two octave. The other training parameters are same as the Section V.B. According to the pedestrian height, occlusion, and truncation, there are three difficult levels for pedestrian detection (i.e., Easy, Moderate, and Hard). All the methods are evaluated on the three difficult levels in terms of average precision (AP). The detection results are given in Table VII. NNNF outperforms the other methods on all the three difficult levels. For example, AP of NNNF is 1.26% higher than that of Checkerboards [53] on the moderately difficult levels. Fig. 26 further gives the average precision curves of these methods on the moderately difficult level. It can be seen that NNNF stably outperform the other methods.

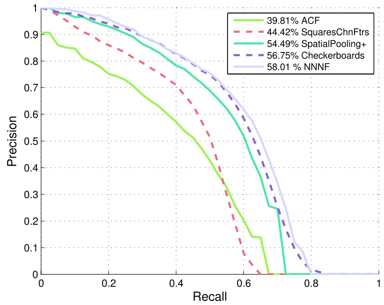


Fig. 26. Precision-recall curves of the moderately difficult level on KITTI.

TABLE VII

AVERAGE PRECISION (AP) OF SOME METHODS WITHOUT USING CNN ON KITTI

Method	Easy	Moderate	Hard
ACF [11]	44.49%	39.81%	37.21%
SquaresChnFtrs [4]	57.33%	44.42%	40.08%
SpatialPooling+ [32]	65.26%	54.49%	48.60%
Checkerboards [53]	67.75%	56.75%	51.12%
NNNF	**69.16%**	**58.01%**	**52.77%**

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented an effective and efficient pedestrian detection method. Two types of non-neighboring features (i.e., side-inner difference features (SIDF) and symmetrical similarity features (SSF)) are proposed. They are found to be complementary to the proposed neighboring features (NF). Among all the selected features, about 1/3 are non-neighboring features (NNF) and 2/3 are NF features. SIDF features characterize not only the difference between contour of a pedestrian and its inner part but also the difference of the background and pedestrian. SSF can capture the symmetrical similarity of pedestrian shape. Though the forms of the proposed NNF and NF features are very simple, combining them in the framework of decision forests results in the best trade-off between log-average miss rate and detection speed. The relationship between our proposed features and some state-of-the-art methods is also revealed. In addition, the proposed channel-specific normalization was also found to be helpful for the improvement of detection performance. In the future work, we will explore that how to use the inherent pedestrian attributes for the structure design of CNN to further improve the performance of pedestrian detection.

REFERENCES

[1] R. Appel, T. Fuchs, and P. Dollar, and P. Perona, “Boosting decision trees-pruning underachieving features early,” in *Proc. Int. Conf. Mach. Learn.*, Jun. 2013, pp. 594–602.

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

[3] R. Benenson, M. Mathias, R. Timofte, and L. Gool, "Pedestrian detection at 100 frames per second," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2903–2910.

[4] R. Benenson, M. Mathias, T. Tuytelaars, and L. V. Gool, "Seeking the strongest rigid detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3666–3673.

[5] S. Brubaker, J. Wu, J. Sun, M. Mullin, and J. Regh, "On the design of cascades of boosted ensembles for face detection," *Int. J. Comput. Vis.*, vol. 77, nos. 1, pp. 65–86, May 2008.

[6] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2005, pp. 236–243.

[7] Accessed on 2009. [Online]. Available: http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

[8] J. Cao, Y. Pang, and X. Li, "Pedestrian detection inspired by appearance constancy and shape symmetry," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1316–1324.

[9] J. Cao, Y. Pang, and X. Li, "Learning multilayer channel features for pedestrian detection," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1603.00124

[10] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3361–3369.

[11] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fastest feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[12] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2010, pp. 68.1–68.11.

[13] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2009, pp. 99.1-99.11.

[14] P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 645–659.

[15] P. Dollar, 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.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2005, pp. 886–893.

[17] M. Enzweiler and D. Gavrila, "A multilevel mixture-of-experts framework for pedestrian classification," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2967–2979, Oct. 2011.

[18] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[19] P. Felzenszwalb, R. 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.

[20] 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.*, Jun. 2013, pp. 3354–3361.

[21] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft Kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Oct. 2013.

[22] J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4073–4082.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[24] F. Khan, J. Xu, J.V. Weijer, A. Bagdanov, R. Anwer, and A. Lopez, "Recognizing actions through action-specific person detection," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4422–4432, Nov. 2015.

[25] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 899–906.

[26] T. Lindeberg, "Detecting salient blob-like image structures and their scales with a scale space primal sketch: A method for focus of attention," *Int. J. Comput. Vis.*, vol. 11, no. 3, pp. 283–318, Dec. 1993.

[27] Y. Li, S. Wang, Q. Tian, and X. Ding, "Learning cascaded shared-boost classifiers for part-based object detection," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1858–1871, Apr. 2014.

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

[29] W. Nam and P. Dollar, and J. H. Han, "Local decorrelation for improved pedestrian detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 424–432.

[30] M. Nishigaki, C. Fermûller, and D. DeMenthon, "The image torque operator: A new tool for mid-level vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 502–509.

[31] S. Paisitkriangkrai, C. Shen, and A.V. Hengel, "Strengthening the effectiveness of pedestrian detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jul. 2014, pp. 546–561.

[32] S. Paisitkriangkrai, C. Shen, and A.V. Hengel, "Pedestrian detection with spatially pooled features and structured ensemble learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1243–1257, Jul. 2016. doi: 10.1109/TPAMI.2015.2474388.2015.

[33] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 241–254.

[34] Y. Pang, J. Cao, and X. Li, "Learning sampling distributions for efficient object detection," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2015.2508603.

[35] Y. Pang, H. Yan, Y. Yuan, and K. Wang, "Robust CoHOG feature extraction in human-centered image/video management system," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 2, pp. 458–468, Apr. 2012.

[36] Y. Pang, H. Zhu, X. Li, and J. Pan, "Motion Blur Detection with an Indicator Function for Surveillance Robots," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5592–5601, Sep. 2016.

[37] Accessed on 2014. [Online]. Available: https://github.com/chhshen/pedestrian-detection

[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[39] M. J. Saberian and N. Vasconcelos, "Boosting algorithms for detector cascade learning," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2569–2605, 2014.

[40] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3626–3633.

[41] 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.*, Jun. 2015, pp. 5079–5087.

[42] S. Tan, F. Zheng, L. Liu, J. Han, and L. Shao, "Dense invariant feature based support vector ranking for cross-Camera person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2016.2555739.

[43] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in *Proc. IEEE Int.Conf. Comput. Vis.*, Dec. 2015, pp. 1904–1912.

[44] Accessed on 2016. [Online]. Available: http://vision.ucsd.edu/∼pdollar/toolbox/doc/

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

[46] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep. 2009, pp. 32–39.

[47] L. Wang, B. Zhang, J. Han, L. Shen, and C. Qian, "Robust object representation by boosting-like deep learning architecture," *Signal Process., Image Commun.*, to be published, doi: 10.1016/j.image.2016.06.002.

[48] 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.

[49] R. Xiao, L. Zhu, and H. Zhang, "Boosting chain learning for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1, pp. 709–715, Oct. 2003.

[50] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Convolutional channel features," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 82–90.

[51] C. Zhu and Y. Peng, "A boosted multi-task model for pedestrian detection with occlusion handling," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5619–5629, Dec. 2015.

[52] S. Zhang, C. Bauckhage, and A. B. Cremers, "Informed haar-like features improve pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 947–954.

[53] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1751–1760.

[54] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1259–1267.

[55] S. Zhang, C. Bauckhage, D. A. Klein, and A. B. Cremers, "Exploring human vision driven features for pedestrian detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 10, pp. 1709–1720, Oct. 2015.

[56] C. Zhang and P. Viola, "Multiple-instance pruning for learning efficient cascade detectors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1681–1688.

Jiale Cao received the B.S. degree in electronic engineering from Tianjin University, Tianjin, China, in 2012, where he is currently pursuing the Ph.D. degree. His research interests include object detection and image analysis, in which he has authored two IEEE Transactions paper and one CVPR paper.

Yanwei Pang (M'07–SM'09) received the Ph.D. degree in electronic engineering from the University of Science and Technology of China in 2004. He is currently a Professor with Tianjin University, China. His current research interests include object detection and image processing, in which he has authored over 100 scientific papers including 24 IEEE Transactions papers.

Xuelong Li (M'02–SM'07–F'12) is currently a Full Professor with the State Key Laboratory of Transient Optics and Photonics, Center for OPTical IMagery Analysis and Learning, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China.