Learning Multilayer Channel Features for Pedestrian Detection

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

***Abstract*— Pedestrian detection based on the combination of convolutional neural network (CNN) and traditional handcrafted features (i.e., HOG+LUV) has achieved great success. In general, HOG+LUV are used to generate the candidate proposals and then CNN classifies these proposals. Despite its success, there is still room for improvement. For example, CNN classifies these proposals by the fully connected layer features, while proposal scores and the features in the inner-layers of CNN are ignored. In this paper, we propose a unifying framework called multi-layer channel features (MCF) to overcome the drawback. It first integrates HOG+LUV with each layer of CNN into a multi-layer image channels. Based on the multi-layer image channels, a multi-stage cascade AdaBoost is then learned. The weak classifiers in each stage of the multi-stage cascade are learned from the image channels of corresponding layer. Experiments on Caltech data set, INRIA data set, ETH data set, TUD-Brussels data set, and KITTI data set are conducted. With more abundant features, an MCF achieves the state of the art on Caltech pedestrian data set (i.e., 10.40% miss rate). Using new and accurate annotations, an MCF achieves 7.98% miss rate. As many non-pedestrian detection windows can be quickly rejected by the first few stages, it accelerates detection speed by 1.43 times. By eliminating the highly overlapped detection windows with lower scores after the first stage, it is 4.07 times faster than negligible performance loss.**

***Index Terms*— Pedestrian detection, multi-layer channel features (MCF), HOG+LUV, CNN, NMS.**

I. Introduction

PEDESTRIAN detection based on Convolutional Neural Network (i.e., CNN) has achieved great success recently [3], [20], [27], [41], [51]. The main process of CNN based methods can be divided into two steps: proposal extraction and CNN classification. Firstly, the candidate proposals are extracted by the traditional pedestrian detection algorithm (e.g., ACF [12] and LDCF [33]). Then, these proposals are classified into pedestrian or non-pedestrian by the CNN model [5], [20].

Despite its great success, it still exists some room for improvement. 1) Most methods only use the last layer features in CNN with softmax or SVM to classify the proposals. In fact, the different layers in CNN represents different image characteristic. Based on the descriptions in [23] and [52], the first few layers can better describe the image local variance, whereas the last few layers abstract the image global structure. It means that each layer in CNN contains different discriminative features, which can be used for learning the classifier. 2) Some methods only use the traditional methods based on the handcrafted features (i.e., HOG+LUV [9]) to generate the candidate proposals while ignoring the proposal scores. 3) Due to the large amount of convolutional operations, the methods based very deep CNN (e.g., VGG16 [42]) run very slowly on the common CPU (e.g., about 8s). However, techniques for speeding up CNN on CPU becomes important, because CNN has shown its superior performance on most applications but GPU is not always available.

Recently, researchers have done some work to solve the above problems. Li *et al.* [26] proposed to train the cascaded multiple CNN models of different resolutions. As the low resolution CNN can early reject many background regions, it avoids scanning the full image with high resolution CNN and then reduces the computation cost. Though it's based on cascade structure, the training process of multiple CNN models is relatively complex. Zeng *et al.* [53] proposed a multi-stage contextual deep model to simulate the cascade classifiers. However, it does not use cascade AdaBoost and cannot reject negatives early. CCF [50] learns the AdaBoost classifier based on one convolutional layer. It benefits from the richer capacity in feature representation and the lower cost in computation and storage compared with end-to-end CNN methods. Despite the initial success, CCF employs only one CNN layer for feature extraction. Cai *et al.* [5] proposed the complexity-aware cascade to seamlessly integrate handcrafted features and the last layer features in CNN into a unifying detector. However, it still does not make full use of the multi-layer features in CNN. Yang *et al.* [51] exploited all the layers for object detection. However the proposal scores are not used for object classification and the all convolutional layers still need to be computed. Bell *et al.* [1] concatenated the multiple layers of CNN into the fixed-size ROI pooling. With more abundant feature abstractions, it outperforms fast-RCNN [18]. Though its success, it needs complex operations of L2-normalized, concatenated, scaled, and

Manuscript received September 10, 2016; revised January 23, 2017; accepted April 4, 2017. Date of publication April 26, 2017; date of current version May 9, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61632081, in part by the National Basic Research Program of China 973 Program under Grant 2014CB340400, and in part by the Research Fund of Hainan Tropical Ocean University Grant QYXB201501. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ivana Tosic. *(Corresponding author: Yanwei Pang.)*

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

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

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.2017.2694224

dimension-reduced. Moreover, it ignores the scores of the proposals.

In this paper, we propose a unifying framework, which is called Multi-layer Channel Features (MCF). Firstly, it integrates handcrafted image channels (i.e., HOG+LUV) and each layer of CNN into the multi-layer image channels. HOG+LUV image channels are set as the first layer, which contains 10 image channels. The layers in CNN correspond to the remaining layers, respectively. Secondly, zero-order, one-order, and high-order features are extracted to generate a large number of candidate feature pools in each layer. Finally, a multi-stage cascade AdaBoost is used to select the discriminative features and efficiently classify object and background. The weak classifiers in each stage of multi-stage cascade are learned based on the candidate features from corresponding layer. To further accelerate detection speed, the highly overlapped detection windows with lower scores are eliminated after the first stage. Overall, the contributions of this paper and the merits of the proposed methods (MCF) can be summarized as follows:

1) The unifying framework MCF is proposed. MCF seamlessly integrates HOG+LUV image channels and each layer of CNN into a unifying multi-layer image channels. Due to the diverse characteristic of different layers, these layers can provide more rich feature abstractions.
2) Multi-stage cascade AdaBoost is learned from multi-layer image channels. It can achieve better performance with more abundant feature abstractions and quickly reject many detection windows by the first few stages.
3) The highly overlapped detection windows with lower scores are eliminated after the first stage. Thus, it can further reduce the computation cost of CNN operations. With very little performance loss, it's 4.07 times faster. Finally, it's possible that MCF with very deep CNN (e.g., VGG16 [42]) can run at 0.54 fps on the common CPU, while it achieves 11.05% miss rate on original Caltech pedestrian set.
4) Experiments on Caltech dataset [11], INRIA dataset [8], ETH dataset [13], TUD-Brussels dataset [49], and KITTI dataset [16] are conducted. MCF achieves the state-of-the-art performance on Caltech pedestrian dataset (the log-average miss rate is 10.40%), which outperforms CompACT-Deep [5] by 1.35%. Using new and more accurate annotations [56] of the test set, MCF achieves 7.98% miss rate, which is superior to other methods.

The rest of the paper is organized as follows. Firstly, we give a review about pedestrian detection. Then, our methods are introduced in Section III. Section IV shows the experimental results. Finally, we conclude this paper in Section V.

II. Related Work

Because object detection is a necessary step in many machine vision systems [24], [25], [29]–[31], [48], a lot of pedestrian detection methods and general object detection methods were developed.

According to whether or not CNN is used, pedestrian detection can be divided into two main manners: the handcrafted channels based methods and CNN based methods. Handcrafted channels based methods are relatively simple and efficient, whereas CNN based methods are much more effective but inefficient. We firstly give a review about the handcrafted channels based methods and then introduce some methods based on CNN.

Haar features based cascade AdaBoost detector is one of the most famous object detection methods [46], [37], [38]. It can quickly reject a large number of non-object detection windows by the early stages of the cascade. Dalal and Triggs [8] proposed to use the Histogram of Oriented Gradients (HOG) to describe the image local variance. It can work very well with a linear SVM. To handle pose variations of objects, Felzenszwalb *et al.* [15] proposed the Deformable Part Model (DPM) based on HOG features, which is a mixture of six deformable part models and one root model.

By integrating cascade AdaBoost [46], [4] and HOG features [8], Dollár *et al.* [9] proposed Integral Channel Features (ICF). Firstly, it extracts the local sum features from HOG channels and LUV color channels (i.e., HOG+LUV). Then, cascade AdaBoost [4], [59] is used to learn the classifier. To further speedup the detection, Dollár *et al.* [12] then proposed Aggregated Channel Features (ACF), which downsamples the image channels by a factor of 4.

Following ICF [9], SquaresChnFtrs [2], InformedHaar [54], LDCF [33], Filtered Channel Features (FCF) [55], and NNNF [6] have been also proposed. They all employ the same image channels (i.e., HOG+LUV) as ICF. In SquaresChnFtrs [2], the pixel sums of local square regions in each channel are used for learning the classifier. InformedHaar [54] incorporates the statistical pedestrian model into the design of simple haar-like features. Inspired by [19], Nam *et al.* [33] proposed to calculate the decorrelated channels by convolving the PCA-like [36] filters with HOG+LUV image channels. Recently, Zhang *et al.* [55] proposed to put the above different types of channel features into a unifying framework (i.e., FCF). FCF generates the candidate feature pool by convolving a filter bank (RandomFilters, Checkerboards, etc.) with HOG+LUV image channels. It's found that using the simple Checkerboards filters could achieve very good performance. Based on the appearance constancy and shape symmetry, Cao *et al.* [6] proposed NNNF features.

Recently, deep Convolutional Neural Network (CNN) based methods have also achieved great success in object detection [22], [17], [1], [40], [18], [57], [28], [39], [58]. Generally speaking, it firstly generates the candidate object proposals [7], [45], [21], [14] and then uses the trained CNN model [22], [17] to classify these proposals. Hosang *et al.* [20] generalized CNN model for pedestrian detection after using the handcrafted features based methods to extract the candidate pedestrian proposals. To eliminate the hard negative proposals in the background, Tian *et al.* [43] proposed to jointly optimize pedestrian detection with semantic tasks. Recently, Tian *et al.* [44] proposed to learn deep strong part models to handle the problem of pedestrian occlusion. Li *et al.* [27] proposed the scale-aware fast-RCNN by incorporating a large scale sub-network and a small scale sub-network into a unifying architecture.

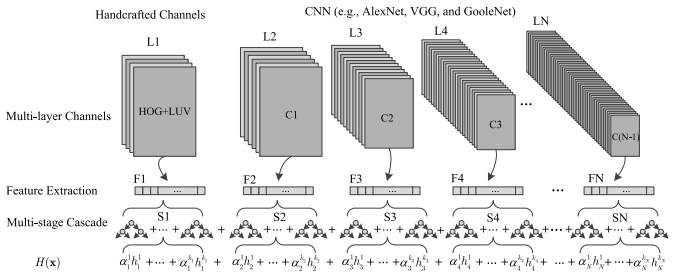

Fig. 1. The basic architecture of MCF. It can be divided into three steps: multi-layer channel generation, feature extraction, and multi-stage cascade AdaBoost classifier.

Despite the success of CNN based pedestrian detection, it still exists some room for improvement. Firstly, the score information of the candidate proposals can be used to boost the detection performance. Secondly, each layer in CNN contains some discriminative features, which can be used for learning the classifier to reject non-pedestrian detection windows early. Cai *et al.* [5] proposed to seamlessly integrate CNN and handcrafted features. Though it uses the proposal score information, it still ignores the features of the inner layers in CNN. Sermanet *et al.* [41] proposed to concatenate the first layer and the second layer together. Bell *et al.* [1] proposed to use skip pooling to integrate multiple layers. It's called skip-layer connections. Despite its success, there is still some problems: 1) It ignores the proposal scores; 2) The proposals need to pass through the whole CNN before classification; 3) The skip-layer operations in [1] is relatively complex.

III. Our Methods

A. Multi-Layer Channel Features (MCF)

The layers in CNN represent the different and diverse image characteristic. Based on the description in [23] and the visualization in [52], it can be concluded that the image channels in the first few layers can better describe the image local variance and the image channels in the last few layers can abstract the image global structure. Meanwhile, the handcrafted image channels (e.g., HOG+LUV) can be also able to describe the image variations very well. HOG channels can describe the image local edge directions and variances. LUV channels capture the image color information. Compared to the layers in CNN, the handcrafted image channels are very simple and the computation is relatively efficient. In this paper, we integrate HOG+LUV and the layers of CNN to construct Multi-layer Channel Features (MCF).

First of all, we give an overview about our proposed Multi-layer Channel Features (i.e., MCF). Fig. 1 shows the basic architecture of MCF. It can be divided into three parts: 1) Firstly, multi-layer image channels from L1 to LN are generated. The traditional handcrafted image channels (i.e., HOG+LUV) are used for the first layer (i.e., L1). The convolutional layers from C1 to C(N-1) in CNN construct the remaining layers from L2 to LN. In each layer, there are multiple image channels. 2) The second step is feature extraction. Zero-order, one-order, and high-order features can be calculated in the image channels of each layer. 3) Finally, the multi-stage cascade AdaBoost is learned from the candidate features of each layer one after another. The weak classifiers in each stage of multi-stage cascade are learned from the candidate features of corresponding layer. For example, the weak classifiers in Stage 2 (i.e., S2) are learned from candidate features F2 of Layer 2 (i.e., L2).

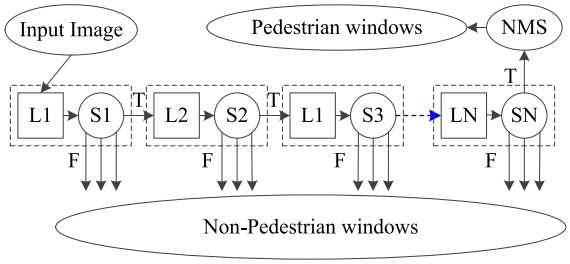

Fig. 2. Test process of basic MCF. As a large number of negative detection windows can be rejected by first few stages, the number of detection windows after several stages is small. Please note that three down arrows at each stage mean that each weak classifier in each stage can reject the windows (i.e., SoftCascade [4]).

TABLE I

Multi-Layer Image Channels. The First Layer is HOG+LUV, and the Remaining Layers are the Convolutional Layers(i.e., C1 to C5) in VGG16

Layer	L1	L2	L3	L4	L5	L6
Name	HOG	VGG16				
	LUV	C1	C2	C3	C4	C5
Size	128×64	64×32	32×16	16×8	8×4	4×2
Num	10	64	128	256	512	512

Fig. 2 shows the test process of MCF. Given the input image, the image channels in L1 (i.e., HOG+LUV) are firstly computed. Detection windows are generated by scanning the input image. These detection windows are classified by S1 using the weak classifiers learned from L1. Some detection windows will be rejected by S1. For the detection windows accepted by S1, the image channels in L2 are computed. Then the accepted detection windows are classified by S2 using the weak classifiers learned from L2. The above process is repeated from L1 to LN. Finally, the detection windows accepted by all the stages (i.e., S1 to SN) will be merged by NMS. The merged detection windows are the final pedestrian windows. As a large number of negative detection windows can be rejected by the first few stages, the number of detection windows after several stages is small. Thus, it can reduce the computation cost and accelerate detection speed. Please note that three down arrows at each stage mean that each weak classifier in each stage can reject the windows (i.e., SoftCascade [4]).

1) Multi-Layer Image Channels: Row 1 in Fig. 1 shows the multi-layer image Channels. It consists of N layers. In each layer, there are multiple image channels. Table I shows the specific parameters of multi-layer image channels based on HOG+LUV and VGG16. It contains six layers from L1 to L6. L1 is the handcrafted image channels (i.e., HOG+LUV). L2-L6 are five convolutional layers (i.e., C1-C5) in VGG16. Please note that the convolutional layers mean the last convolutional layers in each convolutional block of CNN. Row 3 shows the image size in each layer. The image size in L1 is 128×64. The sizes of L2-L6 are 64×32, 32×16, 16×8,

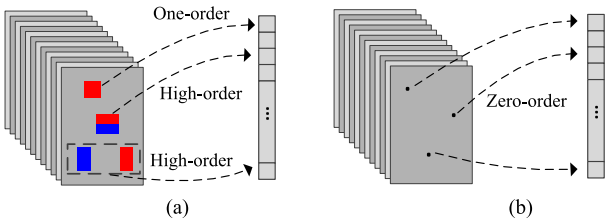

Fig. 3. Feature extraction in multi-layer image channels. (a) feature extraction in L1 (HOG+LUV), where one-order (ACF) and high-order features (NNNF) are used. (b) feature extraction in L2-LN (the layers of CNN), where zero-order features are extracted. Zero-order feature means that a single pixel value in each channel is used as a feature.

8×4, and 4×2, respectively. Row 4 shows the number of the channels in each layer. L1 contains 10 image channels. L2-L6 each have 64, 128, 256, 512, and 512 image channels, respectively. In Table I, all the convolutional layers in CNN (i.e., C1 to C5) are used for constructing the multi-layer image channels. In fact, only part convolutional layers in CNN can also construct the multi-layer image channels. For example, a five-layer image channels can be generated by HOG+LUV and C2-C5 of VGG16, where C1 of VGG16 is not used.

The different image channels of CNN have different characteristic. LeCun *et al.* [23] summarized that CNN exploits the property that many natural signals are compositional hierarchies: in images, local combinations of edges form motifs, motifs assemble into parts, and parts form objects. In [1] and [41], the inner layers of CNN have been explored to improve the detection performance. They uses the skip-layers to extract the information at multiple scales. Thus, multi-layer image channels are complementary and can provide more abundant feature abstractions.

2) Feature Extraction: Features can be divided into three classes: zero-order feature, one-order feature, and high-order feature. In zero-order feature extraction, a single pixel itself is used as a feature and no neighboring pixels are used. One-order feature is defined as the pixel sums or averages in the local or non-local regions in each channel. High-order feature is defined by the difference of the sums or averages of two or more different regions. For L1 (i.e., HOG+LUV), there are many successful methods for feature extraction, including ICF [9], ACF [12], SquaresChnFtrs [2], InformedHaar [54], LDCF [33], FCF [55], and NNNF [6]. ICF, ACF, and SquaresChnFtrs can be seen as one-order features. InformedHaar, LDCF, FCF, and NNNF are high-order features. Among these featues, ACF has the fastest detection speed, and NNNF has the best trade-off between detection speed and detection performance. Due to the simplicity and effectiveness, ACF and NNNF are used for feature exaction in L1. The number of image channels from CNN is relatively large. For example, the fourth convolutional layer (i.e., C4) in VGG16 has 512 image channels (see Table I). To reduce the computation cost and avoid a very large number of candidate features, only zero-order feature is used. It means that each pixel value in image channels of each layer is used as the candidate feature. The specific feature extraction in multi-layer image channels can be seen in Fig. 3.

3) Multi-Stage Cascade AdaBoost: Cascade AdaBoost is a popular method for object detection. Based on multi-layer image channels, we propose the multi-stage cascade AdaBoost for pedestrian detection. Rows 2-4 in Fig. 1 give the specific explanations about multi-stage cascade. The features in Si are learned from the candidate features Fi of Li, where i=1, 2, …, N. Firstly, k_1 weak classifiers in S1 are learned from the candidate features F1 extracted from L1. Based on the hard negative samples and positive samples, k_2 weak classifiers in S2 are then learned from F2. The remaining stages are trained in the same manner. Finally, multi-stage (i.e., N-stage) cascade AdaBoost classifier can be obtained. This strong classifier $H(\mathbf{x})$ can be expressed as the following equation:

$$H(\mathbf{x}) = \sum_{j=1}^{k_1} \alpha_1^j h_1^j(\mathbf{x}) + \ldots + \sum_{j=1}^{k_i} \alpha_i^j h_i^j(\mathbf{x}) + \ldots + \sum_{j=1}^{k_N} \alpha_N^j h_N^j(\mathbf{x}) = \sum_{i=1}^{N} \sum_{j=1}^{k_i} \alpha_i^j h_i^j(\mathbf{x}), \quad (1)$$

where $\mathbf{x}$ represents the samples (windows), $h_i^j(\mathbf{x})$ represents the j-th weak classifier in Stage i, and α_i^j represents the weight of $h_i^j(\mathbf{x})$. k_1, k_2, …, k_N are the number of weak classifiers in each stage, respectively. It is an open and challenging problem to set the optimal values of k_1, k_2, …, k_N because it is infeasible to use maximum accepted false positive rates for choosing the number of weak classifiers. The reason is that the false positive rate in a layer may be zero in the HOG+LUV+CNN framework. In this paper, one simple and empirical structure is used as follows:

$$k_1 = N_{All}/2, \quad k_2 = k_3 = \ldots = k_N = N_{All}/(2 \times (N-1)), \quad (2)$$

where N_{All} represents the number of the total weak classifiers. As k_1 is larger than k_2, k_3, …, k_N, Stage 1 learned from the handcrafted channels can reject a large number of detection windows. Based on SoftCascade [4], the reject thresholds are set after each weak classifier of each stage. It means that each weak classifier in each stage can reject the detection windows.

The advantages about the multi-stage cascade AdaBoost structure can be concluded as the following: 1) Firstly, it avoids learning the classifier from a very large feature pooling (e.g., more than one million); 2) Secondly, it makes full use of the information from multi-layer image channels. Thus, it can enrich the feature abstraction. 3) Finally, many non-pedestrian detection windows can be quickly rejected by the features in the first few layers. Thus, it reduces the computation cost of the remaining layers in CNN and accelerates the detection speed. 4) Our training process is very simple. In [26], it needs to train multiple different CNN models and then integrates them in the cascade structure. MCF only needs to learn one strong classifier by SoftCascade [4].

B. Elimination of Highly Overlapped Windows

Pedestrian detection is a multiple instance problem. Generally, the adjacent area around the pedestrian exists many positive detection windows. Many of these positive detection

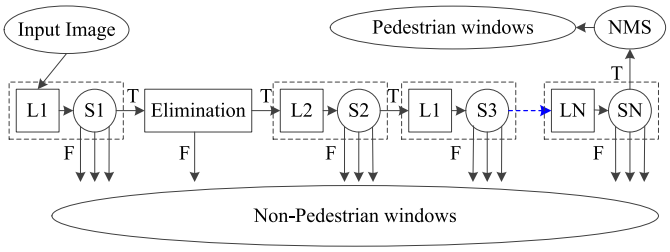

Fig. 4. Test process of MCF-f where the technique of NMS is used to eliminate highly overlapped detection windows with lower scores.

windows around pedestrians highly overlap. Though multi-stage cascade AdaBoost structure can reject many non-pedestrian detection windows, it cannot reject the positive detection windows around pedestrians. When the cascade classifier based on very deep CNN (e.g., VGG16), the computation cost of these positive detection windows are large.

In fact, there is no need to put all the highly overlapped windows accepted by the first stage into the remaining stages. Detection windows accepted by the first stage each have a classification score. The highly overlapped windows with lower scores can be eliminated after the first stage. To eliminate these highly overlapped windows with lower scores, Non-Maximum Suppression (i.e., NMS) is used after the first stage. The overlap ratio $O(w_1, w_2)$ of detection windows can be defined in the following:

$$O(w_1, w_2) = \frac{area(w_1 \cap w_2)}{area(w_1 \cup w_2)}, \tag{3}$$

where w_1 and w_2 are two detection windows. If $O(w_1, w_2) > \theta$, it means that w_1 and w_2 highly overlap. Then the detection window with lower score will be eliminated. Instead of the standard threshold $\theta = 0.5$, a larger threshold is used here. Experimental results show that $\theta = 0.8$ can accelerate the detection speed with little performance loss. Fig. 4 shows the specific test process of MCF by eliminating highly overlapped detection windows. This fast version of MCF by eliminating the highly overlapped detection windows is called MCF-f.

IV. Experiments

The challenging Caltech pedestrian detection dataset [10], [11], the classical INRIA dataset [8], ETH dataset [13], TUD-Brussels dataset [49], and KITTI dataset [16] are employed for the evaluation.

Caltech dataset [10], [11] consists of 11 videos. The first 6 videos are used for training and the remaining videos are used for testing. The raw training images are formed by sampling one image per 30 frames. It results in 4250 images for training, where there are 1631 positive samples. The corresponding training data is called Caltech. To enlarge the training samples, Caltech10x is used. It samples one image per 3 frames in the training videos. As a result, there are 42,782 images in which there are 16,376 positive samples. Please note that the test data is same as [10] and [11] whenever the Caltech or Caltech10x is used. It contains 4024 images in which there are 1014 pedestrians.

INRIA dataset [8] consists of 1237 pedestrians used for training and 288 pedestrian images used for evaluation. By translation and flip, there are 22,666 positive samples for training. There are also 1218 negative images for training. ETH dataset [13] contains three different sequences (i.e., "BAHNHOF", "JELMOLI", and "SUNNY DAY"), where there are totally 1804 test images. TUD-Brussels dataset [49] is captured from a driving car in the urban environment, which contains 508 image pairs with 1326 annotated pedestrians. In KITTI dataset [16], pedestrian detection is a subtask of object detection. For pedestrian detection, it consists of 7481 training images and 7518 test images. For the evaluation on Caltech [10], [11], INRIA [8], ETH [13], and TUD-Brussels [49] datasets, miss rate is used. For the evaluation on KITTI dataset [16], average precision is used.

The first layer in MCF is HOG+LUV image channels [9], which contains one normalized gradient magnitude channel, six histograms of oriented gradient channels, and three color channels. Two popular CNN models (i.e., AlexNet [22] and VGG16 [42]) are used for constructing the remaining layers in MCF. Instead of using original input size 227×227 or 224×224, we use the size 128×64 for pedestrian detection. For AlexNet, stride 4 in the first convolutional layer is replaced by stride 2. The input size 6×6 in the first fully-connected layer is replaced by the size 8×4. For VGG16, the input size 7×7 of the first fully-connected layer is replaced by the size 4×2. The other initial parameters follow the pre-trained models on ImageNet. The final parameters in AlexNet and VGG16 are fine-tuned on the pedestrian dataset.

Feature extraction in L1 (i.e., HOG+LUV) is ACF [12] or NNNF [6]. ACF is used in Section IV-A to demonstrate the effectiveness of the proposed MCF. To achieve the better detecion performance and compare with some state-of-the-art methods, the better NNNF is used in Sections IV-B, IV-C, IV-D, and IV-E. Feature extraction in the remaining layers (i.e., the layers of CNN) is zero-order feature (single pixel). The final classifier consists of 4096 decision trees, unless noted otherwise. The decision tree number of each stage are $k_1 = 2048$, $k_2 = k_3 = \ldots = k_N = 2048/(N-1)$, respectively. N is the number of the layers in MCF.

A. Self-Comparison of MCF

In this section, some intermediate experimental results on original Caltech training set are reported to show how to setup the effective and efficient MCF. Some specific experimental setups are as follows. HOG+LUV are used for the first layer. The convolutional layers in CNN (i.e., AlexNet or VGG16) correspond to the remaining layers. Feature extraction in HOG+LUV is ACF. Feature extraction in the layers of CNN is just zero-order feature (single pixel). To speed up the training, negative samples are generated by five round trainings of original ACF [12], where the number of trees in each round are 32, 128, 512, 1024, and 2048, respectively. Finally, multi-stage cascade which consists of 4096 level-2 decision trees is learned based on these negative samples and positive samples. The first stage contains the first 2048 decision trees. The remaining stages equally split the remaining 2048 decision trees. For

TABLE II

MISS RATES (MR) OF MCF BASED ON HOG+LUV AND THE DIFFERENT LAYERS IN CNN. √ MEANS THAT THE CORRESPONDING LAYER IS USED. HOG+LUV IS ALWAYS USED FOR THE FIRST LAYER. THE LAYERS IN ALEXNET OR VGG16 ARE USED FOR THE REMAINING LAYERS

Name	HOG	AlexNet					MR (%)	Δ MR (%)
	LUV	C1	C2	C3	C4	C5		
MCF-2	√					√	20.08	N/A
MCF-3	√				√	√	18.43	1.65
MCF-4	√			√	√	√	17.40	2.68
MCF-5	√		√	√	√	√	18.01	2.07
MCF-6	√	√	√	√	√	√	17.29	2.79
Name	HOG	VGG16					MR (%)	Δ MR (%)
	LUV	C1	C2	C3	C4	C5		
MCF-2	√					√	18.52	N/A
MCF-3	√				√	√	17.14	1.38
MCF-4	√			√	√	√	15.40	3.12
MCF-5	√		√	√	√	√	14.78	3.74
MCF-6	√	√	√	√	√	√	14.31	4.21

TABLE III

MISS RATES (MR) OF MCF* BASED ON HOG+LUV AND THE DIFFERENT LAYERS IN CNN. √ MEANS THAT THE CORRESPONDING LAYER IS USED. HOG+LUV IS ALWAYS USED FOR THE FIRST LAYER. THE LAYERS IN ALEXNET OR VGG16 ARE USED FOR THE REMAINING LAYERS

Name	HOG	AlexNet					MR (%)	Δ MR (%)
	LUV	C1	C2	C3	C4	C5		
MCF-2*	√	√					30.04	N/A
MCF-3*	√	√	√				24.10	5.94
MCF-4*	√	√	√	√			22.34	7.70
MCF-5*	√	√	√	√	√		19.80	10.24
MCF-6*	√	√	√	√	√	√	17.29	12.75
Name	HOG	VGG16					MR (%)	Δ MR (%)
	LUV	C1	C2	C3	C4	C5		
MCF-2*	√	√					34.08	N/A
MCF-3*	√	√	√				29.22	4.86
MCF-4*	√	√	√	√			23.81	10.27
MCF-5*	√	√	√	√	√		17.42	16.66
MCF-6*	√	√	√	√	√	√	14.31	19.77

example, HOG+LUV and C2 to C5 of CNN construct a five-layer image channels. Then, the corresponding five-stage cascade can be learned. The first stage S1 has 2048 weak classifiers. The remaining stages (i.e., S2-S5) each have 512 weak classifiers. Miss Rates are log-averaged over the range of FPPI = $[10^{-2},10^{0}]$, where FPPI represents False Positive Per Image.

Table II shows Miss Rates (MR) of MCF based on HOG+LUV and the different layers in CNN. The results based on AlexNet and VGG16 are both shown here. √ means that the corresponding layer is used for MCF. HOG+LUV image channels are always used for the first layer. The layers of CNN (i.e., C1, C2, …, or C5) are used for the remaining layers. MCF-N means that there are N layers in MCF. For example, MCF-3 in Row 3 are generated by HOG+LUV, C4 and C5 of AlexNet. The first layer is HOG+LUV image channels. The second layer is the fourth convolutional layer (i.e., C4) of AlexNet. The last layer is the fifth layer (i.e., C5) of AlexNet. Based on multi-layer image channels, the corresponding multi-stage cascade is learned. There are the following observations from Table II: 1) Compared to MCF-2, MCF-N (N>2) usually achieves the better performance. For example, the miss rate of MCF-6 based on VGG16 in the last row is lower than that of MCF-2 by 4.21%; 2) Generally, with the increase of the layer number, the miss rate of MCF becomes lower and the detection performance becomes better. The above observations demonstrate that the middle layers in CNN can enrich the feature abstraction. It means that each layer in CNN contains some discriminative features, which can be used for classification.

In Table III, miss rates of MCF-N versions with incremental layers from C1 to C5 instead of adding from C5 to C1 are also shown. In order to distinguish with Table II, they are represented by MCF-N*. Please note that MCF-6* and MCF-6 are the same. With the incremental layers, the miss rate of MCF becomes lower. For example, the miss rate of MCF-6* based on HOG+LUV and C1-C5 of VGG16 is 19.77% lower than that of MCF-2* based on HOG+LUV and C1 of VGG16. When all the convolutional layers are used, it also achieves the best performance. It also demonstrates that each layer in CNN contains some discriminative features which can contribute to the performance.

TABLE IV

MISS RATES OF MCF-6 AND MCF-ALL. MCF-ALL MEANS ALL THE CONVOLUTIONAL LAYERS IN EACH CONVOLUTIONAL BLOCKS ARE USED

HOG+LUV and AlexNet		HOG+LUV and VGG16	
MCF-6	MCF-All	MCF-6	MCF-All
17.29%	17.45%	14.31%	14.52%

Generally, MCF-6 means that MCF is constructed by the last convolutional layer in each convolutional block. In Table IV, MCF-6 is compared with MCF-All. MCF-All is constructed by all the convolutional layers in each convolutional block. It can be seen that MCF-6 and MCF-All have the similar detection performance. The reason is that the different convolutional layers in the same convolutional block have the similar characteristic. In the following section, MCF-6 is used.

Table V shows the average number and the ratio of detection windows rejected by each stage in MCF-6. MCF-6 is based on HOG+LUV and all the five convolutional layers in CNN. Thus, the multi-stage cascade AdaBoost in MCF has six stages from S1 to S6. '*' means that the average number of detection windows accepted by stage 1, instead of that rejected by stage 1, is shown. As the weak classifiers in S1 are both learned from HOG+LUV, the number of detection windows accepted by S1 are same (i.e., 159). Among the 159 accepted detection windows, about 71.0% and 76.1% detection windows are rejected by the cascade based on AlexNet and that based on VGG16, respectively. Overall, the multi-stage cascade based on VGG16 can reject more detection windows. Specifically,

TABLE V

REJECTED NUMBER AND REJECTED RATIO BY THE STAGES IN MCF-6 ARE SHOWN. '*' MEANS THAT THE AVERAGE NUMBER OF DETECTION WINDOWS ACCEPTED BY STAGE 1 ARE SHOWN

Stage	HOG+LUV and AlexNet		HOG+LUV and VGG16	
	Number	Ratio	Number	Ratio
S1	159*	N/A	159*	N/A
S2	35	22.0%	23	14.5%
S3	35	22.0%	21	13.2%
S4	21	13.2%	33	20.8%
S5	14	8.8%	29	18.2%
S6	8	5.0%	15	9.4%
Total	113	71.0%	121	76.1%

TABLE VI

MISS RATES (MR) AND DETECTION TIME OF MCF-2 AND MCF-6. MCF-2 IS BASED ON HOG+LUV AND C5 OF CNN. MCF-6 IS BASED ON HOG+LUV AND C1-C5 OF CNN

	HOG+LUV and AlexNet		HOG+LUV and VGG16	
	MCF-2	MCF-6	MCF-2	MCF-6
MR (%)	20.08	17.29	18.52	14.31
Time (s)	2.99	2.30	7.69	5.37

the first two stage stages (i.e., S2 and S3) based on AlexNet reject more detection windows than that based on VGG16. The middle two stages (i.e., S4 and S5) based on VGG16 can reject more detection windows than that based on AlexNet.

As multi-stage cascade can reject many detection windows by the first few stages, MCF can accelerate the detection speed. Table VI compares the detection time and detection performance between MCF-2 and MCF-6. MCF-2 uses HOG+LUV and C5 in CNN to construct two-layer image channels. Then two-stage cascade is learned. MCF-6 uses HOG+LUV and all the five convolutional layers from C1 to C5 in CNN to construct six-layer image channels. Then six-stage cascade is learned. The detection time means the average execution time per image on the test dataset, which is tested on the common CPU (i.e., Intel Core i7-3700). No matter the CNN model is AlexNet or VGG16, MCF-6 have the better detection performance and the faster detection speed. For example, based on VGG16, the miss rates of MCF-2 and MCF-6 are 18.52% and 14.31%, respectively. The detection times of MCF-2 and MCF-6 are 7.69s and 5.37s, respectively. Thus, the miss rate of MCF-6 is lower than that of MCF-2 by 4.21%, while the speed of MCF-6 is 1.43 times faster than that of MCF-2. The reasons can be explained as the following: 1) As MCF-6 uses all the layers in CNN to learn the classifier, it can learn more abundant features. Thus, it has a better detection performance. 2) MCF-2 needs to calculate all the layers of CNN (i.e., C1 to C5) before classifying the detection windows accepted by S1. MCF-6 just needs to calculate the first i layers of CNN before classifying the detection windows by Si (i=2,3,…,6). In Table V, MCF-6 rejects 66.7% detection windows before S6. Thus, MCF-6 has faster detection speed than MCF-2.

Though the speed of MCF-6 is faster than that of MCF-2, it's still very slow. To further accelerate the detection speed,

TABLE VII

MISS RATES AND DETECTION TIME VARY WITH θ. MCF USED HERE IS BASED ON HOG+LUV AND ALEXNET

θ	MCF-2		MCF-6	
	MR (%)	Time (s)	MR (%)	Time (s)
INF	20.08	2.99	17.29	2.30
0.50	23.70	0.44	21.65	0.34
0.80	20.97	1.15	18.06	0.86
0.85	20.30	1.76	17.34	1.35
0.90	19.82	2.03	17.32	1.57

TABLE VIII

MISS RATE (MR) AND DETECTION TIME OF MCF-2, MCF-6, AND MCF-6-f. MCF-2 IS BASED ON HOG+LUV AND C5 IN CNN. MCF-6 IS BASED ON HOG+LUV AND C1-C5 IN CNN. MCF-6-f IS THE FAST VERSION OF MCF-6

	HOG+LUV and AlexNet		
	MCF-2	MCF-6	MCF-6-f
MR (%)	20.08	17.29	18.06
Time (s)	2.99	2.30	0.86
	HOG+LUV and VGG16		
	MCF-2	MCF-6	MCF-6-f
MR (%)	18.52	14.31	14.89
Time (s)	7.69	5.37	1.89

the highly overlapped detection windows with lower scores accepted by the first stage (i.e., S1) are eliminated by NMS. As stated in section III-B, the threshold θ is an important factor to balance detection speed and detection performance. Table VII shows that miss rates and detection time vary with θ. MCF-2 and MCF-6 based on HOG+LUV and AlexNet in Table VI are used for the baseline (i.e., θ =INF). When $\theta = 0.5$, the detection speed is very fast, but the detection performance drops rapidly. For example, the detection speed of MCF-6 with $\theta = 0.5$ is 6.76 times faster than original MCF, while the miss rate of MCF-6 with $\theta = 0.5$ is higher than original MCF by 4.36%. Thus, it's not a good choice. When $\theta = 0.9$, the detection performance is almost no loss, while the detection speed is not significantly improved. Thus, the trade-off choice is $\theta = 0.8$. With little performance loss (e.g., 0.77%), MCF-6 is 2.67 times faster than original MCF. In the following section, MCF with $\theta = 0.8$ are called MCF-f.

Table VIII summarizes MCF-2, MCF-6 and MCF-6-f. MCF-6-f is the fast version of MCF-6, where the highly overlapped detection windows are eliminated after the first stage. There are the following observations: 1) MCF-6 and MCF-6-f both have the lower miss rates. Specifically, MCF-6 and MCF-6-f based on AlexNet have lower miss rates than MCF-2 by 2.79% and 2.02%, respectively. MCF-6 and MCF-6-f based on VGG16 have lower miss rates than MCF-2 by 4.21% and 3.63%, respectively. 2) MCF-6 and MCF-6-f is faster than MCF-2. For example, detection time of MCF-2 based VGG16 is 7.69s and that of MCF-6-f based on VGG16 is 1.89s. It means that detection speed of MCF-6-f is 4.07 times faster than that of MCF-2. 3) With little performance loss, MCF-6-f has faster detection speed than MCF-6. The loss of MCF-6-f based on AlexNet is 0.77%, and the loss of MCF-6-f based on VGG16 is 0.58%.

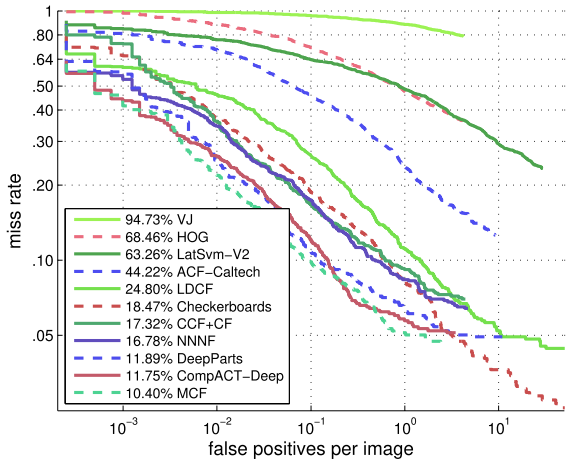

Fig. 5. ROC on Caltech test set (reasonable).

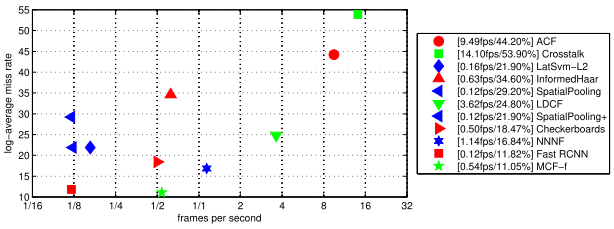

Fig. 6. Miss rates and FPS on Caltech pedestrian dataset are shown. Detection time of the methods are all tested on the common CPU (i.e., Intel Core i7-3700).

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

In this section, MCF based on HOG+LUV and all the five convolutional layers (i.e., C1-C5) in VGG16 is compared to some state-of-the-art methods. The features extracted in the first layer (i.e., HOG+LUV) are NNNF [6] which is one of the state-of-the-art features. The features extracted in the remaining five layers (i.e., C1-C5) are zero-order feature (single pixel). Caltech10x is used for training the final classifier. To speedup the training process, negative samples are accumulated by five rounds of original NNNF, where the number of the trees in each round is 32, 128, 512, 2048, and 4096, respectively. The resulting classifier contains 4096 level-4 decision trees. S1 contains 2048 decision trees. S2-S6 each have 409 decision trees. Zhang *et al.* [56] provided a new, high quality ground truth for the training and test sets. The new annotations of Caltech10x is also used for training MCF. Original Caltech test set and new Caltech test set are both used for the evaluations.

Fig. 5 compares MCF with some state-of-the-art methods on the original annotations of the test set. ACF [12], LDCF [33], Checkerboards [55], CCF+CF [50], NNNF [6], DeepParts [44], and CompACT-Deep [5] are used. ACF [12] are trained on INRIA dataset [8]. The other methods are trained based on Caltech10x dataset. MCF achieves the state-of-the-art performance, which outperforms CompACT-Deep [5], DeepParts [44], NNNF [6], CCF+CF [50], and Checkerboards [55] by 1.35%, 1.49%, 6.38%, 6.92%, and 8.07%, respectively.

Miss rates and Frames Per Second (FPS) of some methods based on CNN are visualized in Fig. 6. Detection time of the methods are all tested on the common CPU (i.e., Intel Core i7-3700). The best choice is that the miss rate is as small as possible while FPS is as large as possible. Though ACF [12] has very fast detection speed (9.49 fps), miss rate of ACF is very large. Fast RCNN reported in [27] has the better detection performance (11.82%), but the speed of Fast RCNN is very slow. MCF-f is the fast version of MCF with little performance loss (0.65%). Compared to Fast RCNN [27], the detection speed of MCF is 4.5 times faster and the miss rate of MCF is 0.77% lower. Therefore, MCF has a better trade-off between detection speed and detection performance. We also implement MCF-f on the GPU (NVIDIA k40c). The speed of MCF-f is 8.13 fps.

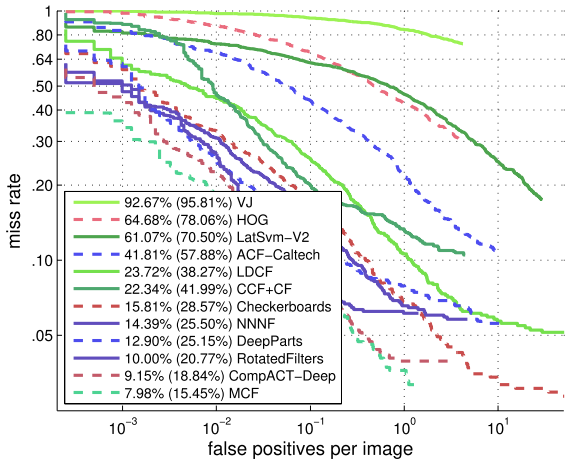

Fig. 7. ROC on Caltech test set using the new and accurate annotations [56]. Miss rates log-averaged over the FPPI range of $[10^{-2},10^{0}]$ and the FPPI range of $[10^{-4},10^{0}]$ are shown. They are represented by MR_{-2} and MR_{-4}, respectively. MR_{-2} (MR_{-4}) are shown in the legend.

Based on the new and accurate annotations of the Caltech test set [56], Fig. 7 further compares MCF with some state-of-the-art methods: CompACT-Deep [5], RotatedFilters [56], DeepParts [44], NNNF [6], Checkerboards [55], and CCF+CF [50]. Miss rates log-averaged over the FPPI range of $[10^{-2},10^{0}]$ and the FPPI range of $[10^{-4},10^{0}]$ are both calculated. They are represented by MR_{-2} and MR_{-4}. MR_{-2} (MR_{-4}) are shown in the legend. MCF and RotatedFilters [56] are trained based on the Caltech10x with the new annotations. MR_{-2} and MR_{-4} of MCF achieve 7.98% and 15.45%, respectively. They are superior to all the other methods. Specifically, MR_{-2} of MCF is 1.17%, 4.92%, 6.41%, and 14.36% lower than that of CompACT-Deep [5], DeepParts [44], NNNF [6], and CCF+CF [50]. Compared to MR_{-2} of MCF, $MR_{-}4$ of MCF has the better performance. Specifically, MR_{-4} of MCF is 3.39%, 9.70%, 10.05%, and 26.54% lower than that of CompACT-Deep [5], DeepParts [44], NNNF [6], and CCF+CF [50]. It means that MCF stably outperforms the other state-of-the-art methods.

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

In this section, we compare MCF with some state-of-the-art methods (e.g., ACF [12], InformedHaar [54], LDCF [33], Roerei [2], SpatialPooling [34], and NNNF [6]) on the INRIA dataset. The size of the negative images is 240×320.

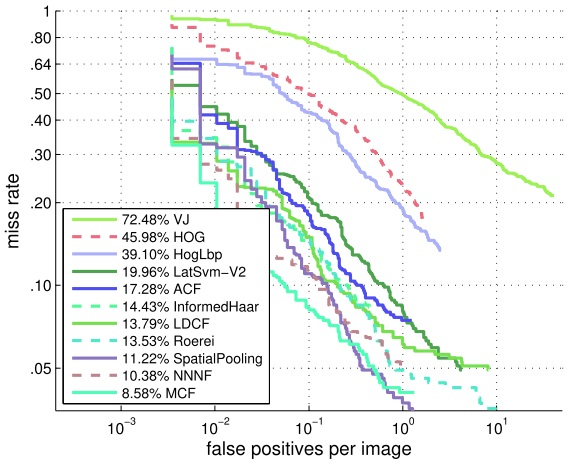

Fig. 8. ROC on INRIA dataset [8].

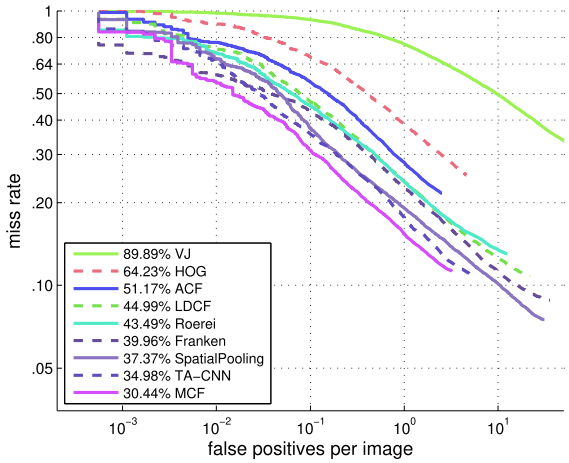

Fig. 9. ROC on ETH dataset [13].

In order to enlarge the number of the negative windows, the training negative images are upsampled by one octave. Negative windows are accumulated by four rounds of original NNNF, where the number of the trees in each round is 32, 128, 512, and 2048, respectively. 10000 hard negatives are added after each round and the cumulative negatives are limited to 15000. As the number of the negatives on the INRIA dataset is relatively limited, the number weak classifiers learned from CNN is only set to 512. The resulting classifier contains 2560 level-3 decision trees. Specifically, S1 contains 2048 decision trees and S2-S6 each have 102 decision trees. Fig. 8 compares MCF with some state-of-the-art methods. The proposed MCF achieves the lowest miss rate (i.e., 8.58%). For example, It outperforms NNNF [6] and SpatialPooling [34] by 1.80% and 2.64%, respectively. It also outperforms than MT-LDCF [60] by 1.92%.

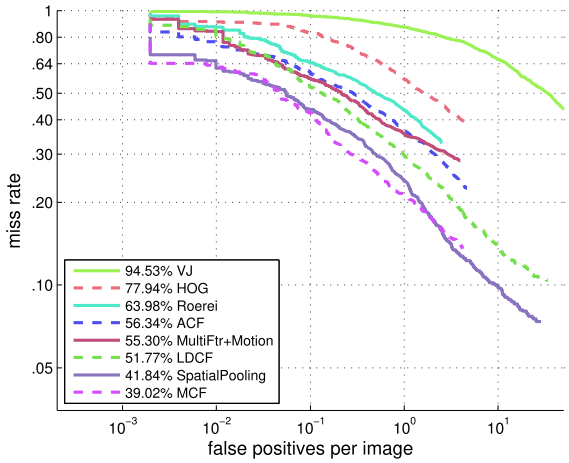

Fig. 10. ROC on TUD-Brussels dataset [49].

D. Comparison With the State-of-the-Art on ETH and TUD-Brussels Datasets

In this section, MCF and some state-of-the-art methods (i.e., ACF [12], LDCF [33], Roerei [2], SpatialPooling [34], Franken [32], TA-CNN [43], and MultiFtr+Motion [47]) are compared on the ETH [13] and TUD-Brussels [49] datasets. Following the experimental setups in [34] and [43], MCF is trained on the INRIA dataset [8]. The parameters are the same as Section IV-C. Fig. 9 and Fig. 10 shows the ROC on ETH dataset and TUD-Brussels dataset, respectively. It can be seen that MCF achieves the state-of-the-art performance on the two datasets. On the ETH dataset, MCF outperforms TA-CNN [43], SpatialPooling [34], Franken [32] by 4.54%, 6.93%, and 9.52%. On the TUD-Brussels dataset, MCF outperforms SpatialPooling [34], LDCF [33], and MultiFtr+Motion [47] by 2.82%, 12.75%, and 16.28%.

E. Comparison With the State-of-the-Art on KITTI Dataset

In this section, MCF and some state-of-the-art methods (i.e., ACF [12], SpatialPooling+ [35], Checkerboards [55], DeepParts [44], and CompACT-Deep [5]) are compared on the KITTI dataset [16]. All the methods are evaluated on the three difficult levels (i.e., Easy, Moderate, and Hard) in terms of average precision (AP). We 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 detection results are given in Table IX. MCF outperforms the other methods on all the three difficult levels. For example, AP of MCF is 0.71% higher than that of CompACT-Deep [5] on the moderately difficult level. Though some state-of-the-art methods (e.g., [51] and [27]) also outperform than MCF, they are scale-aware methods. In the future, we will explore scale-aware MCF.

F. Visualization and Analysis of Detection Results

In this section, the visualization of some detection results about MCF on Caltech dataset [10], [11] is given in Fig. 11. The green rectangle means the true positive, the blue rectangle means the missing positive, and the red rectangle means the false positive. In Figs. 11(a) and (b), MCF can detect all the pedestrians. In Fig. 11(c), one small-scale pedestrian and one heavily occlusion pedestrian are missing. Fig. 11(d), three

TABLE IX
AVERAGE PRECISION (AP) OF SOME METHODS ON KITTI

Method	Easy	Moderate	Hard
ACF [12]	44.49%	39.81%	37.21%
SpatialPooling+ [35]	65.26%	54.49%	48.60%
Checkerboards [55]	67.75%	56.75%	51.12%
DeepParts [44]	70.49%	58.67%	52.78%
CompACT-Deep [5]	70.69%	58.74%	52.71%
MCF	70.87%	59.45%	54.28%

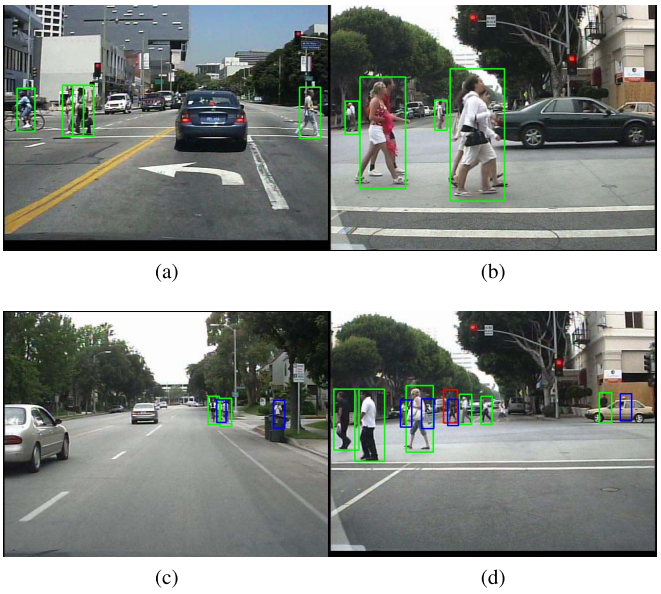

Fig. 11. Some detection results on Caltech test dataset [10], [11]. The green rectangle means the true positive, the blue rectangle means the missing positive, and the red rectangle means the false positive.

heavily occlusion pedestrians are missing, and one small-scale pedestrian cannot be detected accurately. Based on the above observations, it can be seen that the small-scale pedestrian performance and heavily occlusion pedestrian performance of MCF need to be more improved in the future.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a unifying framework, which is called Multi-layer Channels Features (MCF). Firstly, the handcrafted image channels and the layers in CNN construct the multi-layer image channels. Then a multi-stage cascade are learned from the features extracted in the layers, respectively. The weak classifiers in each stage are learned from the corresponding layer. On the one hand, due to the much more abundant candidate features, MCF achieves the state-of-the-art performance on Caltech pedestrian dataset (i.e., 10.40% miss rate). Using the new and accurate annotations of the Caltech pedestrian dataset, miss rate of MCF is 7.98%, which is superior to other methods. On the other hand, due to the cascade structure, MCF rejects many detection windows by the first few stages and then accelerates the detection speed. To further speedup the detection, the highly overlapped detection windows are eliminated after the first stage. Finally, MCF with VGG16 can run on the CPU by 0.54 fps.

In the future work, we will explore the very slim net to further improve detection speed. Based on the slim structure, the computation cost of the first few layers is very small. With the cascade structure, the first few stages can reject many detection widows with little computation cost of the first few convolutional layers.

REFERENCES

[1] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2874–2883.

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

[3] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 1–8.

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

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

[6] 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.

[7] M. Cheng, Z. Zhang, W. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2014, pp. 3286–3293.

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

[9] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proc. Brit. Mach. Vis. Conf.*, 2009, pp. 1–11.

[10] P. Dollár, C. Wojek, B. Schiele, and B. Perona, "Pedestrian detection: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2009, pp. 304–311.

[11] 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.

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

[13] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2008, pp. 1–8.

[14] Z. Fang, Z. Cao, Y. Xiao, L. Zhu, and J. Yuan, "Adobe boxes: Locating object proposals using object adobes," *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4116–4128, Sep. 2016.

[15] 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.

[16] 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.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2014, pp. 580–587.

[18] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Apr. 2015, pp. 1440–1448.

[19] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2012, pp. 459–472.

[20] 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.

[21] Z. Jie, X. Liang, J. Feng, W. Lu, E. H. F. Tay, and S. Yan, "Scale-aware pixel-wise object proposal networks," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4525–4539, Oct. 2016.

[22] 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.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[24] J. Lei, S. Li, C. Zhu, M.-T. Sun, and C. Hou, "Depth coding based on depth-texture motion and structure similarities," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 2, pp. 275–286, Feb. 2015.

[25] J. Lei *et al.*, "A universal framework for salient object detection," *IEEE Trans. Multimedia*, vol. 18, no. 9, pp. 1783–1795, Nov. 2016.

[26] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5325–5334.

[27] J. Li, X. Liang, S. Shen, T. Xu, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *CoRR*, vol. abs/1510.08160, 2015.

[28] J. Li, X. Mei, and D. Prokhorov, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 690–703, Mar. 2017, doi: 10.1109/TNNLS.2016.2522428.

[29] X. Li, Q. Guo, and X. Lu, "Spatiotemporal statistics for video quality assessment," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3329–3342, Jul. 2016.

[30] Y. Li and X. Zou, "Identifying disease modules and components of viral infections based on multi-layer networks," *Sci. China Inf. Sci.*, vol. 59, no. 7, p. 070102:1–070102:15, 2016.

[31] X. Lu, X. Zheng, and X. Li, "Latent semantic minimal hashing for image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 355–368, Jan. 2017.

[32] M. Mathias, R. Benenson, R. Timofte, and L. Van Gool, "Handling occlusions with Franken-classifiers," in *Proc. IEEE Int. Conf. Comput. Vis.*, Aug. 2013, pp. 1505–1512.

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

[34] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Strengthening the effectiveness of pedestrian detection with spatially pooled features," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 546–561.

[35] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Pedestrian detection with spatially pooled features and structured ensemble learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1243–1257, Jun. 2016.

[36] Y. Pang, S. Wang, and Y. Yuan, "Learning regularized LDA by clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2191–2201, Dec. 2014.

[37] Y. Pang, J. Cao, and X. Li, "Learning sampling distributions for efficient object detection," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 117–129, Jan. 2016.

[38] Y. Pang, J. Cao, and X. Li, "Cascade learning by optimally partitioning," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2016.2601438.

[39] Y. Pang, M. Sun, X. Jiang, and X. Li, "Convolution in convolution for network in network," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2017.2676130.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process Syst.*, 2015, pp. 91–99.

[41] 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.*, Apr. 2013, pp. 3626–3633.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[43] 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.*, Sep. 2015, pp. 5079–5087.

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

[45] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

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

[47] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Sep. 2010, pp. 1030–1037.

[48] X. Wu, M. Du, W. Chen, and J. Wang, "Salient object detection via region contrast and graph regularization," *Sci. China Inf. Sci.*, vol. 59, no. 3, pp. 32104:1–321104:14, 2016.

[49] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2009, pp. 794–801.

[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] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2129–2137.

[52] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2013, pp. 818–833.

[53] X. Zeng, W. Ouyang, and X. Wang, "Multi-stage contextual deep learning for pedestrian detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2013, pp. 121–128.

[54] 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.

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

[56] 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.*, Sep. 2016, pp. 1259–1267.

[57] D. Zhang, J. Han, J. Han, and L. Shao, "Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1163–1176, Apr. 2015.

[58] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 215–232, 2016.

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

[60] 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.

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 under the supervision of Prof. Y. Pang. He has authored or co-authored four IEEE Transactions paper and one CVPR paper in his research fields. His research interests include object detection and image analysis.

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. He has authored or co-authored over 100 scientific papers, including 30 IEEE Transactions papers. His current research interests include object detection, image recognition, image processing, and their applications in self-driving cars, visual surveillance, human–machine interaction, and biometrics.

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