This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology

Pushing the Limits of Deep CNNs for Pedestrian Detection

Qichang Hu, Peng Wang, Chunhua Shen, Anton van den Hengel, Fatih Porikli

***Abstract*—Compared to other applications in computer vision, convolutional neural networks have under-performed on pedestrian detection. A breakthrough was made very recently by using sophisticated deep CNN models, with a number of hand-crafted features [2], or explicit occlusion handling mechanism [44]. In this work, we show that by re-using the convolutional feature maps (CFMs) of a deep convolutional neural network (DCNN) model as image features to train an ensemble of boosted decision models, we are able to achieve the best reported accuracy without using specially designed learning algorithms. We empirically identify and disclose important implementation details. We also show that pixel labelling may be simply combined with a detector to boost the detection performance. By adding complementary hand-crafted features such as optical flow, the DCNN based detector can be further improved. We advance state-of-the-art results by lowering the log-average miss rate from 11.7% to 8.9% on the Caltech dataset, 11.2% to 8.6% on the Inria dataset. We also achieve a comparable result to state-of-the-art approaches on the KITTI dataset.**

***Index Terms*—Pedestrian detection, convolutional feature map (CFM), ensemble model.**

I. Introduction

The problem of pedestrian detection has been intensively studied in recent years. Prior to the very recent work in deep convolutional neural networks (DCNNs) based methods [2], [44], the top performing pedestrian detectors are boosted decision forests with carefully hand-crafted features, such as histogram of gradients (HOG) [6], self-similarity (SS) [39], aggregate channel features (ACF) [9], filtered channel features [49] and optical flow [36].

Recently, DCNNs have significantly outperformed comparable methods on a wide variety of vision problems [25], [41], [42], [17], [45], [19], [40], [1]. A region-based convolutional neural network (R-CNN) [17] achieved excellent performance for *generic* object detection, for example, in which a set of potential detections (object proposals) are evaluated by a DCNN model. Later, R-CNN was extended to the Fast R-CNN [16] which significantly increases the detection speed. CifarNet [24] and AlexNet [25] have been extensively evaluated in the R-CNN detection framework in [22] for pedestrian detection. In their work, the best performance is 23.3% log-average miss rate (MR) on Caltech dataset, which was achieved by AlexNet pre-trained on the ImageNet [8] classification dataset. Note that this result is still inferior to conventional pedestrian detectors such as [49] and [36]. The DCNN models in [22] under-perform mainly because the network design is not optimal for pedestrian detection. The performance of R-CNNs for pedestrian detection has further improved to 16.43% MR in [44] through the use of a deeper GoogLeNet model which is fine-tuned using Caltech pedestrian dataset.

To explicitly model the deformation and occlusion, another line of research for object detection is part-based models [11], [12], [28], [18] and explicit occlusion handling [31], [35], [43]. DCNNs have also been incorporated along this stream of work for pedestrian detection [33], [34], [30], but none of these approaches has achieved better results than the best hand-crafted features based method of [49] on the Caltech dataset.

The performance of pedestrian detection is improved over hand-crafted features by a large margin (a $\sim 5\%$ MR gain on Caltech), by two very recent approaches relying on DCNNs: CompACT-Deep [2] combines hand-crafted features and fine-tuned DCNNs into a complexity-aware cascade. Tian *et al.* [44] fine-tuned a pool of part detectors using a pre-trained GoogLeNet, and the resulting ensemble model (refer to as DeepParts) delivers similar results as CompACT-Deep. Both approaches are much more sophisticated than the standard R-CNN framework: CompACT-Deep involves the use of a variety of hand-crafted features, a small CNN model and a large VGG16 model [41]. DeepParts contains 45 fine-tuned DCNN models and needs a set of strategies (including bounding-box shifting handling and part selection) to arrive at the reported result. Note that the high complexity of DCNN models can lead to practical difficulties. For example, it can be too costly to load all 45 DCNN models into a GPU card.

Here we ask a question: Is a complex DCNN based learning approach really a must for achieving the state-of-the-art performance? Our answer to this question is negative. In this work, we propose alternative methods for pedestrian detection, which are simpler in design, with comparable or even better performance. Firstly, we extensively evaluate the CFMs extracted from multiple convolutional layers of a fine-tuned VGG16 model for pedestrian detection. Using only a CFM of a single convolutional layer, we train a boosted-tree-based detector and the resulting model already significantly outperforms all previous methods except the above two sophisticated DCNN frameworks. This model can be seen as a strong baseline for pedestrian detection as it is very simple in

Q. Hu is with The University of Adelaide, Adelaide, SA 5005, Australia, and also with Data61, Canberra, ACT 2601, Australia (E-mail: qichang.hu@adelaide.edu.au).

P. Wang is with The University of Adelaide, Adelaide, SA 5005, Australia (E-mail: p.wang@adelaide.edu.au). Correspondence should be addressed to P. Wang.

C. Shen and A. van den Hengel are with The University of Adelaide, Adelaide, SA 5005, Australia, and also with Australian Centre for Robotic Vision, Brisbane, Qld. 4001, Australia.

F. Porikli is with Data61, Canberra, ACT 2601, Australia.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology

terms of implementation.

We show that the CFMs from multiple convolutional layers can be used for training effective boosted decision forests. These boosted decision forests are combined altogether simply by score averaging. The resulting ensemble model beats all competing methods on the Caltech dataset. We further improve the detection performance by incorporating a semantic pixel labelling model. Next we review some related work.

A. Related Work

1) Convolutional feature maps (CFMs): It has been shown in [38], [20], [48] that CFMs have strong representation abilities for many tasks. Long *et al.* [29] adapt predominant DCNNs into fully convolutional networks and transfer their learned representations by fine-tuning to the semantic segmentation domain. In [20], the CFMs from multiple layers are stacked into one vector and used for segmentation and localization. Ren *et al.* [38] learn a network on the CFMs (pooled to a fixed size) of a pre-trained model.

The work by Yang *et al.* [48] is close to ours, which trains a boosted decision forest for pedestrian detection with the CFM features from the Conv3-3 layer of the VGG16 model [41], and the performance (17.32% MR) on Caltech is comparable to checkerboards [49]. It seems that there is no significant superiority of the CFM used in [48] over hand-crafted features on the task of pedestrian detection. The reason may be two-fold. First, the CFM used in [48] are extract from the pre-trained VGG16 model which is *not fine-tuned on a pedestrian dataset*; Second, CFM features are extracted from only one layer and the multi-layer structure of DCNNs is not fully exploited. We show in this work that both of these two issues are critically important in achieving good performance.

2) Segmentation for object detection: The cues used by segmentation approaches are typically complementary to those exploited by top-down methods. Recently, Yan *et al.* [47] propose to perform generic object detection by labelling super-pixels, which results in an energy minimization problem with data term learned by DCNN models. In [13], [19], segmented image regions (not bounding boxes) are generated as object proposals and then used for object detection.

In contrast to the above region (or super-pixel) based methods, we here exploit at an even finer level of information, that is, pixel labelling. In particular, in this work we demonstrate that we can improve the detection performance by simply re-scoring the proposals generated by a detector, using pixel-level scores.

B. Contributions

We revisit pedestrian detection with DCNNs by studying the impact of a few training details and design parameters. We show that fine-tuning of a DCNN model using pedestrian data is critically important. Proper bootstrapping has a considerable impact too. Besides these findings, other main contributions of this work can be summarized as follows.

1) *The use of multi-layer CFMs for training a state-of-the-art pedestrian detector.* We show that it is possible to train an ensemble of boosted decision forests using multi-layer CFMs that outperform all previous methods. For example, with CFM features extracted from two convolutional layers, we can achieve a log-average miss rate of 10.7% on Caltech, which already perform better than all previous methods, including the two sophisticated DCNNs based methods [2], [44].
2) *Incorporating semantic pixel labelling.* We also propose a combination of sliding-window detectors and semantic pixel-labelling, which outperforms the best of previous methods. To keep the method simple, we use the weighted sum of pixel-labelling scores within a proposal region to represent the score of the proposal.
3) *The best reported pedestrian detector.* A new performance record for Caltech is set by exploiting a DCNN as well as two complementary hand-crafted features: ACF and optical-flow features. This shows that some types of hand-crafted features are complementary to deep convolutional features.

Before we present our methods, we briefly describe the datasets, evaluation metric and boosting models in our experiments. See section A of the supplementary material for the detailed introduction of these datasets.

C. Datasets, Evaluation Metric and Models

Caltech pedestrian dataset The Caltech dataset [10] is one of the most popular datasets for pedestrian detection. It contains 250k frames captured from 10 hours of urban traffic videos. The standard training set and test set consider one out of each 30 frames. In our experiments, the training images are increased to one out of each 4 frames. Note that many competing methods [49], [48], [22] have used the same extended training set or even more data (every third frame). We evaluate the performance of various detectors using the log-average miss rate (MR) which is computed by averaging the miss rate at false positive rates spaced evenly between 0.01 to 1 false-positive-per-image (FPPI) range. The dataset has different test settings with respect to the difficulty of pedestrian height, visibility and aspect ratio. Unless otherwise specified, the detection performance on our experiments shown in the remainder of the paper is the MR on the Caltech Reasonable test setting.

Inria pedestrian dataset The Inria dataset [6] contains 614 positive training images and 288 positive test images. Images of Inria are captured from multiple different scenes. We use the log-average miss rate to evaluate the detection performance as same as the Caltech. All results are reported on the 288 positive test images (negative images are not used).

KITTI pedestrian dataset The KITTI dataset [15] consists of 7481 training images and 7518 test images, comprising more than 80 thousands of annotated objects in traffic scenes. The dataset has three subsets (Easy, Moderate, Hard) with respect to the difficulty of object size, occlusion and truncation. We use the Moderate training subset as the training data in our experiments. Average precision (AP) is used to evaluate the detection performance for KITTI dataset. The average precision summaries the shape of the precision-recall curve, and is defined as the mean precision at a set of evenly spaced

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology

Model	Fine-tuning data	Shrinkage	Avg. miss rate (%)
CFM3a	No fine-tuning	–	18.71
CFM3b	Collected by ACF	–	16.42
CFM3c	Bootstrapping with CFM3b	–	14.54
CFM3	Bootstrapping with CFM3b	0.5	**13.49**

TABLE I: Performance improvements with different fine-tuning strategies and shrinkage (on Reasonable). All boosted decision forests are trained with the CFM extracted from the Conv3-3 layer of VGG16. CFM3a: the original VGG16 model pre-trained on ImageNet is used to extract features. CFM3b: the VGG16 model is fine-tuned with the data collected by an ACF [9] detector. CFM3c and CFM3: the fine-tuning data is obtained by bootstrapping with CFM3b. With the same fine-tuning data, setting the shrinkage parameter of Adaboost to 0.5 brings an additional 1% reduction on the MR

recall levels. All methods are ranked based on the Moderate difficult results.

Boosted decision forest For supervised classification tasks, boosting is a popular method to select features for improving the performance of any given learning algorithm [14], [7], [36], [49]. In this paper, we use the boosted decision forest as a strong classifier which is a convex linear combination of a set of given weak decision trees. The final classification is based on the weighted vote of these decision trees. Unless otherwise specified, we train all our boosted decision forests using the following parameters. The boosted decision forest consists of 4096 depth-5 decision trees, trained via the shrinkage version of real-Adaboost [21]. The size of detection model is set to 128×64 pixels for Caltech and Inria, 64×32 pixels for KITTI. One bootstrapping iteration is implemented to collect hard negatives and re-train the model. The sliding window stride is set to 4 pixels.

II. Boosted Decision Forests with Multi-layer CFMs

In this section, we firstly introduce the general layout of VGG16 model. Then, we show that the performance of boosted decision forests with CFMs can be significantly improved by simply fine-tuning DCNNs with hard negative data extracted through bootstrapping. Next, boosted decision forests are trained with different layers of CFMs, and the resulting ensemble model is able to achieve the best reported result on Caltech dataset.

A. Architecture of the VGG16 model

In this work, VGG16 [41] model is used to extract CFMs. In general, the VGG16 model has 13 convolutional (Conv) layers organized into five convolutional stacks and three fully-connected (FC) layers. We use Conv$\mathbf{Y}$-$\mathbf{x}$ to denote a specific Conv layer, where $\mathbf{Y}$ indicates the $\mathbf{Y}$th Conv stack and $\mathbf{x}$ indicates the $\mathbf{x}$th Conv layer in this stack. FC-6, FC-7, and FC-8 are used to denote three FC layers, respectively. See section B of the supplementary material for the detailed architecture of the VGG16 model.

B. Fine-tuning DCNNs with Bootstrapped Data

As we know, the VGG16 model was originally pre-trained on the ImageNet data with image-level annotations and was not trained specifically for the pedestrian detection task. The CCF framework of [48] extracts CFMs from a single convolutional layer (Conv3-3) of the pre-trained VGG16 model to train the boosted decision forest for diverse detection tasks. To maintain a good generalization ability, the method dose not fine-tune the VGG16 model on any domain-specific datasets. It is expected that the detection performance of boosted decision forests trained with CFMs ought to be improved by fine-tuning the VGG16 model with Caltech pedestrian data. Moreover, We extract CFMs from multiple convolutional layers to train effective boosted decision forests. These boosted decision forests are combined into an ensemble model which further improves the detection performance.

To adapt the pre-trained VGG16 model to the pedestrian detection task, we modify the structure of the model. We replace the 1000-way classification layer with a randomly initialized binary classification layer and change the input size from 224×224 to 128×64 pixels. We also reduce the number of neurons in fully connected layers from 4096 to 2048. We fine-tune all layers of this modified VGG16, except the first 4 convolutional layers since they correspond to low-level features which are largely universal for most visual objects. The initial learning rate is set to 0.001 for convolutional layers and 0.01 for fully connected layers. The learning rate is divided by 10 at every 10000 iterations. For fine-tuning, 30k positive and 90k negative examples are collected by different approaches. The positive samples are those overlapping with a ground-truth bounding box by $[0.5, 1]$, and the negative samples by $[0, 0.25]$. At each stochastic gradient descent (SGD) iteration, we uniformly sample 32 positive samples and 96 negative samples to construct a mini-batch of size 128.

Shallow convolutional layers of the VGG16 contain low-level features which are precise in localization. On the contrary, deep convolutional layers contain discriminative information which are good in classification. According to the evaluation of different CFMs of the VGG16 model in [48], we find that features of Conv3-3 layer provide the best trade-off between the localization information and the discriminative information. It means that these features can achieve the reasonable detection performance and provide effective region proposals simultaneously.

We train boosted decision forests with the CFM extracted from the Conv3-3 layer of differently fine-tuned VGG16 models and the results are shown in Table I. Note that all the VGG16 models in this table are fine-tuned from the original model pre-trained on ImageNet data. It can be observed that the log-average miss rate is reduced from 18.71% to 16.42% by replacing the pre-trained VGG16 model with the one fine-tuned on data collected by applying an ACF [9] detector on Caltech training dataset. The detection performance is further improved to 14.54% MR if it is fine-tuned on the bootstrapped data using the previous trained model CFM3b. Another 1% performance gain is obtained by applying shrinkage to the coefficients of weak learners, with shrinkage parameter being 0.5 (see [37]). The last model (corresponding to row 4 in Table I) is referred to as CFM3 from now on.

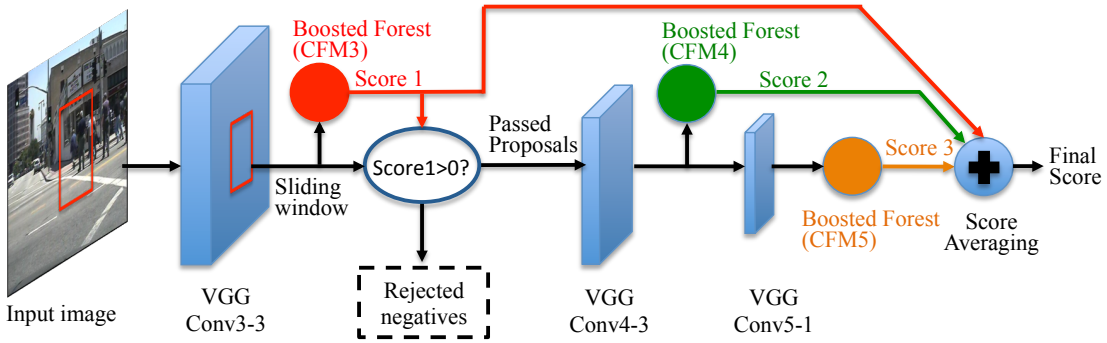


Fig. 1: The framework of an ensemble of boosted decision forests with multi-layer CFMs (CFM3+CFM4+CFM5), which obtain a 10.46% MR on the Caltech Reasonable test setting.

Convolutional layer	# Channels	Down-sampling ratio	Avg. miss rate (%)
Conv3-1	256	4	19.15
Conv3-2	256	4	16.25
Conv3-3 (CFM3)	256	4	**13.49**
Conv4-1	512	8	12.95
Conv4-2	512	8	12.68
Conv4-3 (CFM4)	512	8	**12.21**
Conv5-1 (CFM5)	512	16	**14.17**
Conv5-2	512	16	14.56
Conv5-3	512	16	18.24

TABLE II: Comparison of detection performance (on Reasonable) of boosted decision forests trained on individual CFMs. Note that models with Conv3-x features works as sliding-window detectors, and models with Conv4-x and Conv5-x features are applied to the proposals generated by CFM3. The top performing layers in each convolutional stack are Conv3-3, Conv4-3 and Conv5-1 respectively. The models trained with these three layers are denoted as CFM3, CFM4, and CFM5 respectively

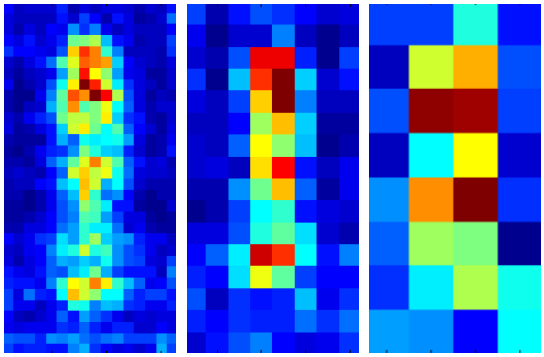


Fig. 2: The spatial distribution of regions of CFMs selected by boosting algorithms. For a 128×64 input image, the size of feature maps are 32×16, 16×8, 8×4 respectively. Red pixels indicate that a large number of features are selected in those regions and blue pixels correspond to low frequency regions. The most discriminative regions correspond to the head, shoulder, waist and feet of a human.

C. Ensemble of Boosted Decision Forests

In the last experiment, we only use a CFM from a single layer of the VGG16 model. In this section, we intensively explore the deep structure of the VGG16 model. We ignore the CFMs of the first two convolutional stacks since they are universal for most visual objects.

We train boosted decision forests with CFMs from individual convolutional layers of the VGG16 model which is the one fine-tuned with bootstrapped data (same as row 4 in Table I). All boosted decision forests are trained with the same data as CFM3. For models with Conv3-x features, the input image are directly applied on the convolutional layers and resulting in a feature map with the down-sampling ratio of 4. The corresponding boosted decision forests work as a sliding-window detector with step-size of 4. In detection, we upsample the image by a factor of 2 as in [49] and the minimum size of the shortest image edge is 72 pixels. The number of scales per each octave is set to 8. For models with Conv4-x and Conv5-x features, they are applied to proposals generated by CFM3 model. This is due to the large downsampling ratio of Conv4-x and Conv5-x. If the step-size of the sliding-window detector is too large, it will hurt the detection performance.

Table II shows the comparison of detection performance of these boosted decision forests on Caltech Reasonable setting. We can observe that the MR is relatively high for the Conv3-1 layer and the Conv5-3 layer. We conjecture that the Conv3-1 layer provides relatively low-level features which result in an under-fitting training. In contrast, the semantic information in the Conv5-3 layer may be too coarse to precisely localize small pedestrians. We also note that Conv5-3 layer performs much worse than Conv5-1 layer. This may be caused by that Conv5-3 has a larger receptive field than Conv5-1, more localization information is lost. The large receptive field of Conv5-3 layer degrades its final detection performance. According to Table II, the best performing layer in each convolutional stack, are from inner layers of Conv3-3 (CFM3), Conv4-3 (CFM4), and Conv5-1 (CFM5) respectively. Fig. 2 shows the spatial distribution of regions of different CFMs selected by boosting algorithms. Features within the warm color area are frequently selected by above three CFM models. We observe that most active regions correspond to the contours of human-body. The head-shoulder area shows to be more discriminative than other body parts.

The boosted decision forests trained with CFMs of these three layers are further combined together simply through score averaging. Fig. 1 shows the framework of the resulting ensemble model. Firstly, CFM3 model works as a sliding-window detector, which rejects the majority of negative exam-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology

Model combination	Avg. miss rate (%)
CFM3+CFM4	10.68
CFM3+CFM5	10.88
CFM3+CFM4+CFM5	10.46
CFM3+CFM4+CFM5+DCNN	**10.07**

TABLE III: The comparison of performance (on Reasonable) of different ensemble models. DCNN: the entire VGG16 model fine-tuned by data collected by CFM3. The combination of multi-layer CFM models improves the detection performance of single-layer CFM models significantly (3%)

ples and pass region proposals to CFM4 and CFM5. Both CFM4 and CFM5 generate the confidence score for each incoming proposal. The CFM3 features are reused in the computation of CFM4 and CFM5 features. A subregion of CFM3 feature map is cropped and fed into the 4/5-th convolutional layers of the VGG16 model to compute CFM4 and CFM5 features. The final score is computed by averaging over the scores output by these three boosted decision forests. *This model delivers the best reported log-average miss rate (*10.46%*) on Caltech* Reasonable *setting without using any sophisticatedly designed algorithms.*

We also evaluate other combinations of the ensemble models. Furthermore, a VGG16 model is fine-tuned with another round of bootstrapping (using CFM3) and its final output is also combined to improve the detection performance. The corresponding results can be found in Table III. We can see that combining two layers already beats all existing approaches on Caltech, and adding the entire large VGG16 model also gives a small improvement.

III. Pixel Labelling Improves Pedestrian Detection

In this section, the sliding-window based detectors are enhanced by semantic pixel labelling. By incorporating DCNNs, the performance of pixel labelling (semantic image segmentation) methods have been recently improved significantly [29], [3], [20], [50], [27]. In general, we argue that pixel labelling models encode information complementary to the sliding-window based detectors. Empirically, we show that consistent improvements are achieved over different types of detectors.

The segmentation method proposed in [3] is used here for pixel labelling, in which a DCNN model (VGG16) is trained on the Cityscapes dataset [5]. The prediction map is refined by a fully-connected conditional random field (CRF) [23] with DCNN responses as unary terms. The Cityscapes dataset that we use for training is similar to the KITTI dataset which contains dense pixel annotations of 19 semantic classes such as road, building, car, pedestrian, sky, etc. Note that our models that exploiting pixel labelling have used extra data for training on top of the Caltech dataset. However, most deep learning based methods [2], [44] have used extra data, at least the ImageNet dataset for pre-training the deep model. Pedestrian detection may benefit from the semantic pixel labelling in the following aspects:

– *Multi-class information:* Learning from multiple classes, in contrast to the object detectors typically trained with two-class data, the pixel labelling model carries richer object-level information.

– *Long-range context:* Using CRFs (especially fully-connected CRFs) as post-processing procedure, many models (for example, [3], [27], [50]) have the ability to capture long-range context information. In contrast, sliding-window detectors only extract features from fixed-sized bounding boxes.

– *Object parts:* The trained pixel labelling model may cater for more fine-grained details, such that they are more insensitive to deformation and occlusion to some extent.

However, it is not straightforward to apply pixel labelling models to pedestrian detection problems. One of the main impediments is that it is difficult to estimate the object bounding boxes from the pixel score map, especially for people in crowds.

To this end, we propose to bring the pedestrian detector and pixel labelling model together. In our framework (see Fig. 3), a sliding-window detector is responsible for providing region proposals and a pixel labelling model is applied to the input image to generate a score map for the "person" class. Next, a spatially weighted mask $\mathbf{M}$ is applied to the proposal region $\mathbf{x}$ of the "person" score map to generate the weighted sum of pixel scores. The weighted sum of the kth region proposal, denoted as $\mathbf{S_k}$, can be calculated by the following equation:

$$\mathbf{S}_k = \sum_{i}^{H \times W} m_i x_i^k \tag{1}$$

where H and W denote the height and width of the mask $\mathbf{M}$, x_i^k denotes the ith local value of the kth region proposal on the "person" score map, and m_i is the corresponding coefficient on the mask. Note that the dimension of each cropped proposal region $\mathbf{x}$ need to be resized to match the dimension of the mask $\mathbf{M}$. Finally, the weighted sum and the detector score for the same proposal are aggregated together as the final score.

To learn the spatially weighted mask, the pixel labelling model is firstly applied to all training images to generate the "person" score maps. Then, ground truth regions are cropped from these score maps and all cropped patches are resized to the dimension of the detection model without padding area (e.g. 100×41 pixels for Caltech). The mask is learned by averaging these cropped patches. See section C of the supplementary material for the visualization of learned masks.

Note that, there are more sophisticated methods for exploiting the labelling scores. For example, one can use the pixel labelling scores as the image features, similar to 'object bank' [26], and train a linear model. In this work, we show that even simply weighted sum of the pixel scores considerably improves the results.

Table IV shows the detection performance of different sliding-window detectors enhanced by pixel labelling. Boosted decision forests are trained here with three types of features, which are ACF [9], checkerboards features [49] and the CFM from the Conv3-3 layer of VGG16 model (CFM3). We can see that the performances of all the three detectors are improved by aggregating pixel labelling models. Fig. 4 presents some region proposals on the original images and the corresponding pixel score maps. Some of false proposals generated by pedestrian

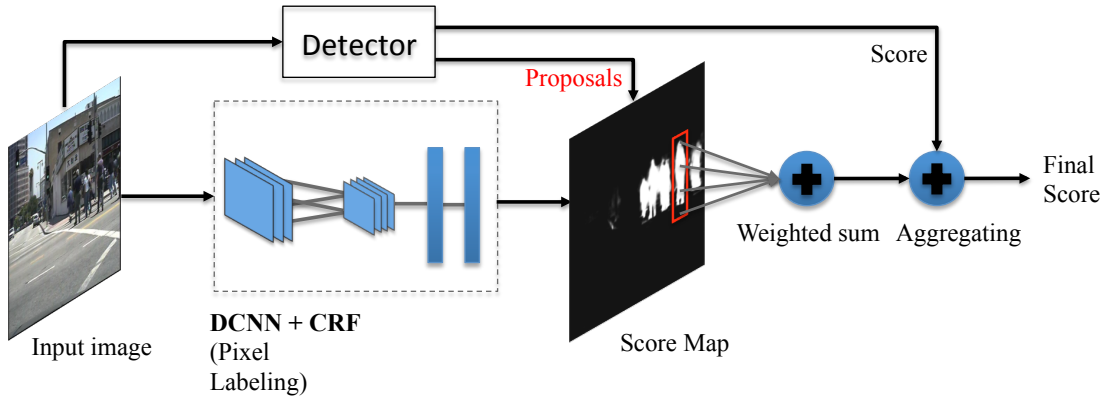
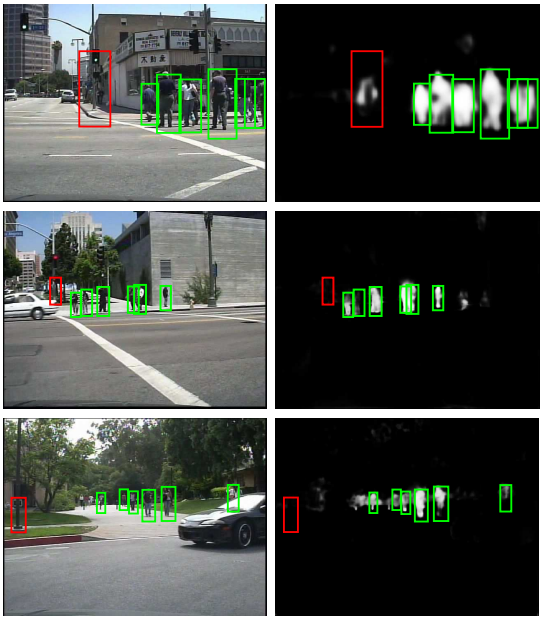


Fig. 3: The framework for pedestrian detection with pixel-labelling. The region proposals and pixel-level score maps are obtained by individually applying the sliding-window detector and the pixel labelling model. Next, the weighted sum of pixel scores within a proposal region is aggregated with the detector score of the same proposal region.

Method	Avg. miss rate (%)	Improve. (%)
ACF [9]	22.23	4.50
ACF+Pixel label.	17.73	
Checkerboards [49]	18.25	3.61
Checkerboards+Pixel label.	14.64	
CFM3 (ours)	13.49	1.91
CFM3+Pixel label.	11.58	

TABLE IV: Performance improvements by aggregating pixel labelling models with sliding-window detectors (on Reasonable). All the three detectors achieve performance gains, which shows that pixel labelling can be used to help detection. Note that the performance of our model 'CFM3 with Pixel labelling' already outperforms the previously best reported result of [2]



(a) proposals by detector (b) pixel score map

Fig. 4: Examples of some region proposals on the original images and the corresponding pixel score maps. A strong complementary relationship can be found in the generated proposals and the pixel score maps.

detectors (CFM3) can be eliminated by considering the context of a larger region (the largest bounding box in the first row in Fig. 4). Some occluded pedestrians have responses on the pixel score map (the rightmost bounding box in the third row in Fig. 4). This clearly illustrates why this combination works.

IV. Fusing Models

A. Overview of the proposed framework

Fig. 5 shows an overview of the proposed pedestrian detection framework. The framework consists of two components: a pedestrian detector and a semantic pixel labelling model. Our pedestrian detector is an ensemble detection model which takes as input an image and outputs a number of proposals with detection scores. The pixel labelling model takes as input an image and proposals within the image. It generates the weighted sum of pixel scores for each proposal. Finally, the confidence score of one proposal is computed by averaging outputs of multiple components. To accelerate the detection speed, the CFM3 detection model can be replaced by a light-weight proposal method, which is described in section IV-E.

B. Using Complementary Hand-crafted Features

The detection performance of the CFM3 model is critical in the proposed ensemble model, since later components often rely on the detection results of this model. In order to enhance the detection performance of the CFM3 model, we make two variants of it by combining two hand-crafted features: the ACF and optical flow. We augment the CFM3 features with the ACF and optical flow features to train an ensemble of boosted decision forests. Optical flow features are extracted the same way as in [36].

Table V shows the detection results of different variants of CFM3 model. With adding the ACF features, the MR of CFM3 detector is reduce by 1.11%. With the extra optical flow features, the MR is further reduced to 11.11%. These experimental results demonstrate that hand-crafted features carry complementary information which can further improve the DCNN convolutional features. Fig. 6 shows the visualization of some intermediate features. We can observe that the ACF

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology

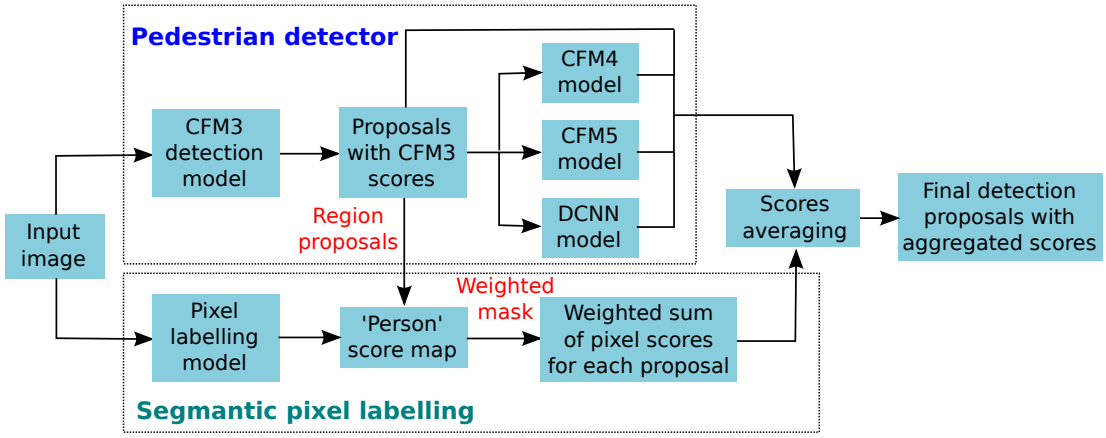


Fig. 5: Overview of our pedestrian detection framework. The framework consists of one pedestrian detector and one pixel labelling model. The final confidence score of one proposal is computed by averaging outputs of multiple components.

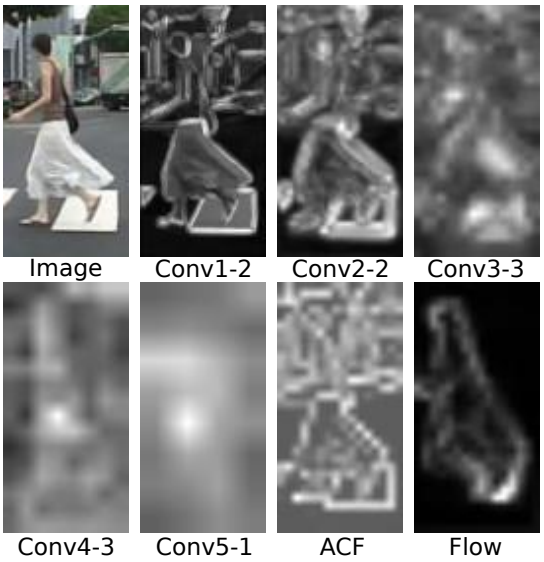


Fig. 6: Visualization of some intermediate features.

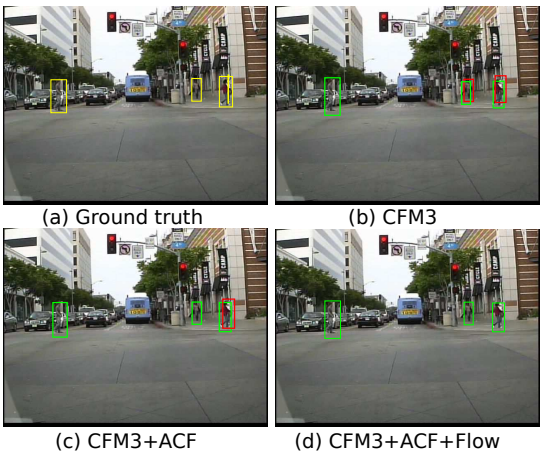


Fig. 7: Visualization of detection results of different variants of the CFM3 detector. Yellow bounding boxes are ground truth, green bounding boxes are true positives, and red bounding boxes are false positives.

Method	Avg. miss rate (%)
CFM3 only	13.49
CFM3+ACF	12.38
CFM3+ACF+Flow	11.11
(CFM3+ACF)+CFM4+CFM5+DCNN	9.37
(CFM3+ACF+Flow)+CFM4+CFM5+DCNN	**9.32**

TABLE V: Comparison of detection results of different variants of the CFM3 detector (on Reasonable). The convolutional features of the Conv3-3 layer are combined with different types of hand-crafted features, and used to train a boosted decision forest. Both the performance of the variants and the ensemble models is improved with these additional features. Flow: optical flow features. DCNN: the entire VGG16 model fine-tuned by data collected by CFM3

features may be viewed as lower-level features, compared with the middle-level features in CFM3. The optical flow clearly encodes motion information which is not in CFM3 features. By adding the other components of the proposed ensemble model, our detector can achieve 9.32% MR. The MR is slightly increased to 9.37% by removing motion information. Fig. 7 shows the visualization of detection results of different variants of the CFM3 detector. By involving these hand-crafted features, more hard false positives can be eliminated by the proposed detector.

C. Pixel Labelling

As shown in Section III, the pixel labelling model is also complementary to convolutional features. Table VI shows the detection performance of different ensemble models enhanced by pixel labelling model. The best result is achieved by combining the most number of different types of models (which is referred to as All-in-one), which reduces the MR on the Caltech Reasonable setting from the previous best 11.7% to 8.9%. Note that the combination rule used by our methods is simple, which implies a potential for further improvement.

D. Ablation Studies

We investigate the overall pipeline of the All-in-one model by adding each component step by step, which is shown

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology

Model	CFM3a	CFM3	CFM3+CFM4	CFM3+CFM4 +CFM5	CFM3+CFM4 +CFM5+DCNN	CFM3+CFM4+CFM5 +DCNN+Label.	All-in-one
Pipeline	CFM3a	fine-tuning	Add CFM4	Add CFM5	Add DCNN	Add Pixel Label.	Use (CFM3+ACF+Flow)
Miss rate (%)	18.71	13.49	10.68	10.46	10.07	9.53	8.93
Improve. (%)	–	+5.22	+2.81	+0.22	+0.39	+0.54	+0.6

TABLE VII: Ablation studies of the All-in-one model on the Caltech Reasonable test setting

Method	Avg. miss rate (%)
CFM3+Pixel label.	11.58
CFM3+CFM4+CFM5+Pixel label.	9.94
CFM3+CFM4+CFM5+DCNN+Pixel label.	9.53
(CFM3+ACF)+CFM4+CFM5+ DCNN+Pixel label.	9.06
(CFM3+ACF+Flow)+CFM4+CFM5+ DCNN+Pixel label. (All-in-one)	**8.93**

TABLE VI: Comparison of detection performance (on Reasonable) of different ensemble models with pixel labelling. DCNN: the entire VGG16 model fine-tuned by hard negative data collected by CFM3; Pixel label.: pixel labelling model; Flow: optical flow. The pixel labelling model consistently improves all the considered models in this table. The All-in-one model set a new record on the Caltech pedestrian benchmark

Method	Avg. miss rate (%)	runtime (s)
CFM3 (proposals)+CFM4+ CFM5+DCNN+Pixel label.	9.53	8.0
ACF (proposals)+CFM3+CFM4+ CFM5+DCNN+Pixel label.	12.20	0.85
Checkerboards (proposals)+CFM3+ CFM4+CFM5+DCNN+Pixel label.	10.65	1.25

TABLE VIII: Comparison of detection performance (on Reasonable) between the original ensemble model and fast ensemble models

in Table VII. As the start point, the CFM3a model with the original VGG16 model pre-trained on ImageNet data achieves a miss rate of 18.71%. A 5.22% performance gain can be obtained by fine-tuning the VGG16 model with bootstrapped data. The detection results can be improved to 10.46% (better than all previous methods) by adding CFM4 and CFM5 models to construct an ensemble model. We obtain 0.39% performance improvement if we use the entire VGG16 model (fine-tuned by bootstrapped data with CFM3) as a component of our ensemble model. Combining the pixel labelling information to detected bounding boxes can further reduce the MR by 0.54%. By replacing the CFM3 model to CFM3+ACF+Flow model, the MR of our ensemble model can eventually achieve 8.93% on the Caltech Reasonable test setting.

E. Fast Ensemble Models

In this section, we investigate the speed issue of the proposed detector. Our All-in-one model takes about 8s for processing one 640×480 image on a workstation with one octa-core Intel Xeon 2.30GHz processor and one Nvidia Tesla K40c GPU. Most of time (about 7s) is spent on the extraction of the CFMs on a multi-scale image pyramid. The remaining components of the ensemble model take less than 1s to process the passed region proposals. The pixel labelling model only uses about 0.25s to process one image since it only need to be applied to one scale. It can be easily observed that the current bottleneck of the proposed detector is the CFM3 which is used to extract region proposals with associated detection scores. The speed of our detector can be accelerated using a light-weight proposal method at the start of the pipeline in Fig. 1.

We use two pedestrian detectors ACF [9] and checkerboards [49] as the proposal methods. Our ACF detector consists of 4096 depth-4 decision trees, trained via real-Adaboost. The model has size 128×64 pixels, and is trained via four rounds of bootstrapping. The sliding window stride is 4 pixels. The checkerboards detector is trained using almost identical parameters as for ACF. The only difference is that the feature channels are the results of convolving the ACF channels with a set of checkerboards filters. In our implementation, we adopt a set of 12 binary 2×2 filters to generate checkerboards feature channels. To limit the number of region proposals, we set a threshold of the above two detectors to generate about 20 proposals per image in average.

Table VIII shows the detection performance of the original ensemble model and fast ensemble models on Caltech Reasonable test setting. We can observe that the quality of proposals are enhanced by a large margin using both ensemble models and the pixel labelling model. The best result of fast ensemble models is achieved by using proposals generated by the checkerboards detector. This method uses the data collected by checkerboards detector as the initial fine-tuning data. With a negotiable performance loss (e.g., 1.12%), it's about 6 times faster than the original method. Note that the fast ensemble model (with checkerboards proposals) also achieves the state-of-the-art results.

F. Comparison to State-of-the-art Approaches

1) Caltech: We compare the detection performance of our models with existing state-of-the-art approaches on the Caltech dataset. Table IX and Fig. 8 compares our models with a wide range of detectors, including boosted decision forests trained on hand-crafted features, RCNN-based methods and the state-of-the-art methods on the Caltech Reasonable test setting. The performance of the first two types are quite close to each other. Using only one single layer of convolutional feature map, our CFM3 model has outperformed all other methods except the two sophisticated methods [44], [2]. Note that the RCNN based methods are based on larger models than CFM3. As feature representation, the CFM from the Conv3-3 layer of our fine-tuned model significantly outperforms all other hand-crafted features. The CFM3+Pixel labelling model already outperforms the state-of-the-art performance achieved by sophisticated methods [44], [2]. Our CFM3+CFM4+CFM5 model performs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology

Type	Method	Miss Rate (%)
Hand-crafted Features	SpatialPooling [37]	29.24
	SpatialPooling+ [36]†	21.89
	LDCF [32]	24.80
	Checkerboards [49]	18.47
	Checkerboards+ [49]†	17.10
RCNN based	AlexNet [22]	23.32
	GoogLeNet [44]	16.43
State-of-the-arts	DeepParts [44]	11.89
	CompACT-Deep [2]	11.75
Ours	CFM3	13.49
	CFM3+Label.	11.58
	CFM3+CFM4+CFM5	10.46
	CFM3+CFM4+CFM5+DCNN+Label.	9.53
	All-in-one†	**8.93**

TABLE IX: Detection performance of different types of detectors on the Caltech Reasonable test setting. Three types of approaches are compared in this table, including boosted decision trees trained on hand-crafted features, RCNN-based methods and the state-of-the-art sophisticated methods. All of our models outperform the first three types of models, and our All-in-one set a new recorded MR on Caltech pedestrian benchmark. † indicates the methods trained with optical flow features

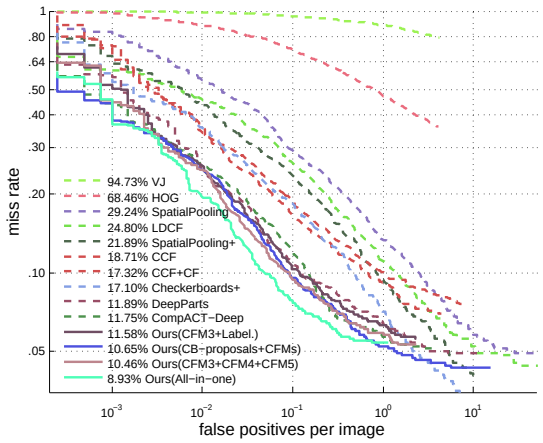


Fig. 8: Comparison to state-of-the-art approaches on the Caltech Reasonable test setting.

even better. Without using hand-crafted features, our model can achieve 9.53% MR. The best result is achieved by the All-in-one model which combines a number of hand-crafted features and CFM models.

2) Inria: Fig. 9 represents the detection results on the Inria dataset. In our experiments, we only apply the fast ensemble model without using the pixel labelling method. Since our pixel labelling model is trained on the Cityscapes dataset which has totally different scenes from the Inria dataset, the improvement of pixel labelling is limited for this dataset. It can be observed that our method achieves the lowest MR of 8.63% outperforming all previously-reported results.

3) KITTI: Table X shows the detection results on the KITTI dataset. Since images of KITTI are larger than in Caltech, the feature extraction of CFM3 model is time-consuming. In our experiments, only the fast ensemble model with Checkerboards proposals is used for testing on KITTI. Our model achieves competitive results, 74.22%, 63.26%, and 56.44% AP on Easy, Moderate, and Hard subsets respectively. Fig. 10 presents the

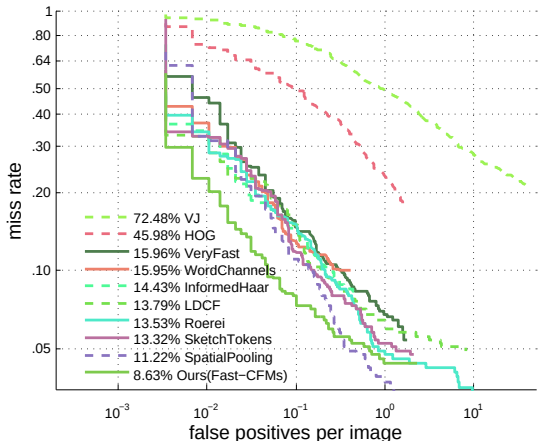


Fig. 9: Comparison to state-of-the-art approaches on the Inria positive test set.

Method	Moderate(%)	Easy(%)	Hard(%)
3DOP* [4]	67.47	81.78	64.70
Fast-CFMs (Ours)	63.26	74.22	56.44
Reionlets [46]	61.15	73.14	55.21
CompACT-Deep [2]	58.74	70.69	52.71
DeepParts [44]	58.67	70.49	52.78
FilteredICF [49]	56.75	67.65	51.12
pAUCEnsT [36]	54.49	65.26	48.60
R-CNN [22]	50.13	61.61	44.79

TABLE X: Detection results (AP) on three KITTI test subsets. Note: * indicates the methods trained with stereo images

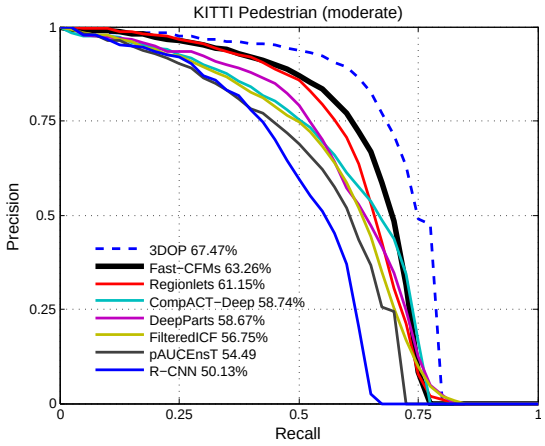


Fig. 10: Comparison to state-of-the-art approaches on the KITTI Moderate test set.

comparison of detection performance on the KITTI Moderate test subset. It can be observed that the proposed detector outperforms all published monocular-based methods. Note that the 3DOP [4] is based on stereo images. The proposed ensemble model is the best-performing detector based on DCNN, and surpasses CompACT-Deep [2] and DeepParts [44] by 4.52% and 4.59% respectively.

V. Conclusion

In this work, we have built a simple-yet-powerful pedestrian detector, which re-uses inner layers of convolutional

features extracted by a properly fine-tuned VGG16 model. This 'vanilla' model has already achieved the best reported results on the Caltech dataset, using the same training data as previous DCNN approaches. With a few simple modifications, its variants have achieved even more significant results.

We have presented extensive and systematic empirical evaluations on the effectiveness of DCNN features for pedestrian detection. We show that it is possible to build the best pedestrian detector, yet avoiding complex custom designs. We also show that a pixel labelling model can be used to improve performance by simply incorporating the labelling scores with the detection scores of a standard pedestrian detector. Note that simple combination rules are used here, which leaves potentials for further improvement. For example the ROI pooling for further speed and performance improvement.

REFERENCES

[1] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. In *Proc. Bri. Conf. Mach. Vis.*, 2014.

[2] Z. Cai, M. Saberian, and N. Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.

[3] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proc. Int. Conf. Learning Representations*, 2015.

[4] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals for accurate object class detection. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 424–432, 2015.

[5] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn. Workshops*, 2015.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2005.

[7] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Mach. Learn.*, 46(1-3):225–254, 2002.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009.

[9] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(8):1532–1545, 2014.

[10] 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.*, 34(4):743–761, 2012.

[11] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 990–997, 2010.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010.

[13] S. Fidler, R. Mottaghi, R. Urtasun, et al. Bottom-up segmentation for top-down detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2013.

[14] Y. Freund and R. E. Schapire. A short introduction to boosting. pages 1401–1406, 1999.

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012.

[16] R. Girshick. Fast R-CNN. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1440–1448, 2015.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014.

[18] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[19] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proc. Eur. Conf. Comp. Vis.*, 2014.

[20] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[21] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[22] J. Hosang, M. Omran, R. Benenson, and B. Schiele. Taking a deeper look at pedestrians. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4073–4082, 2015.

[23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proc. Adv. Neural Inf. Process. Syst.*, 2011.

[24] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 2012.

[26] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei. Object bank: An object-level image representation for high-level visual recognition. *Int. J. Comput. Vision*, 107(1):20–39, 2014.

[27] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv:1504.01013*, 2015.

[28] L. Lin, X. Wang, W. Yang, and J.-H. Lai. Discriminatively trained and-or graph models for object shape detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(5):959–972, 2015.

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015.

[30] P. Luo, Y. Tian, X. Wang, and X. Tang. Switchable deep network for pedestrian detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014.

[31] M. Mathias, R. Benenson, R. Timofte, and L. Van Gool. Handling occlusions with franken-classifiers. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1505–1512, 2013.

[32] W. Nam, P. Dollár, and J. H. Han. Local decorrelation for improved pedestrian detection. In *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[33] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012.

[34] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2013.

[35] W. Ouyang and X. Wang. Single-pedestrian detection aided by multi-pedestrian detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2013.

[36] S. Paisitkriangkrai, C. Shen, and A. v. d. Hengel. Pedestrian detection with spatially pooled features and structured ensemble learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

[37] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In *Proc. Eur. Conf. Comp. Vis.*, 2014.

[38] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. *arXiv:1504.06066*, 2015.

[39] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2007.

[40] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. Int. Conf. Learning Representations*, 2015.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[43] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. *Int. J. Comp. Vis.*, 110(1):58–69, 2014.

[44] Y. Tian, P. Luo, X. Wang, and X. Tang. Deep learning strong parts for pedestrian detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.

[45] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Proc. Adv. Neural Inf. Process. Syst.*, 2014.

[46] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 17–24, 2013.

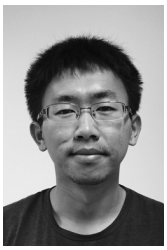
[47] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li. Object detection by labeling superpixels. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[48] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 82–90, 2015.

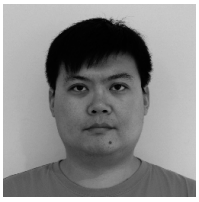
[49] S. Zhang, R. Benenson, and B. Schiele. Filtered channel features for pedestrian detection. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[50] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. *arXiv:1502.03240*, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2017.2648850, IEEE Transactions on Circuits and Systems for Video Technology



Qichang Hu is a PhD Candidature with the Australian Centre for Visual Technologies, University of Adelaide, Adelaide, SA, Australia. He received the bachelor's degree in computer science from the University of Adelaide, Adelaide, SA, Australia in 2012. His research interests include deep learning, object detection, and machine learning.



Peng Wang is a post-doctoral researcher at the University of Adelaide. He received the B.S. degree in electrical engineering and automation, and the PhD degree in control science and engineering from Beihang University, China, in 2004 and 2011, respectively.

Chunhua Shen is a Professor at School of Computer Science, the University of Adelaide. He was with the computer vision program at NICTA (National ICT Australia), Canberra Research Laboratory for about six years. His research interests are in the intersection of computer vision and statistical machine learning.

He studied at Nanjing University, Nanjing, China, and Australian National University, Canberra, ACT, Australia, and received the PhD degree from the University of Adelaide. From 2012 to 2016, he holds an Australian Research Council Future Fellowship.

Anton van den Hengel is a Professor at School of Computer Science, the University of Adelaide. He is also the Founding Director of the Australian Centre for Visual Technologies, Interdisciplinary Research Centre, University of Adelaide, Adelaide, SA, Australia, with a focus on innovation in the production and analysis of visual digital media.

He received the bachelor's degree in mathematical science, the B.L. degree, the master's degree in computer science, and the PhD degree in computer vision from the University of Adelaide in 1991, 1993, 1994, and 2000, respectively.

Fatih Porikli is an IEEE Fellow and a Professor with the Research School of Engineering, Australian National University, Canberra, ACT, Australia. He is also acting as the Leader of the Computer Vision Group at Data61, Canberra, ACT 2601, Australia.

He received the PhD degree from NYU, New York, NY, USA, in 2002. Previously he served as a Distinguished Research Scientist at Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He has contributed broadly to object detection, motion estimation, tracking, image-based representations, and video analytics. He is the coeditor of two books on Video Analytics for Business Intelligence and Handbook on Background Modeling and Foreground Detection for Video Surveillance. He is an Associate Editor of five journals including IEEE Signal Processing Magazine, SIAM Imaging Sciences, EURASIP Journal of Image & Video Processing, Springer Journal on Machine Vision Applications, and Springer Journal on Real-time Image & Video Processing. His publications won three best paper awards and he has received the R&D100 Award in the Scientist of the Year category in 2006. He served as the General and Program Chair of several IEEE conferences in the past.