Deep Pedestrian Detection Using Contextual Information and Multi-level Features

Weijie Kong[1], Nannan Li[1], Thomas H. Li[2], and Ge Li[1(✉)]

[1] School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China
geli@ece.pku.edu.cn
[2] Gpower Semiconductor, Inc., Suzhou, China

Abstract. Recently, Faster R-CNN achieves great performance in deep learning based object detection. However, a major bottleneck of Faster R-CNN lies on the sharp performance deterioration when detecting objects that are small in size or have a similar appearance with their backgrounds. To address this problem, we present a new pedestrian detection approach based on Faster R-CNN, which combines contextual information with multi-level features. The contextual information is embedded by pooling information from a larger area around the original region of interest. It helps pedestrians detection from cluttered backgrounds. The multi-level features can be obtained by pooling proposal-specific features from several shallow but high-resolution layers. These features are more informative for detecting small-size pedestrians. Extensive experiments on the challenging Caltech dataset validate that our approach not only performs better than the baseline of Faster R-CNN but also boosts the detection performance when combined with contextual information and multi-level features. Meanwhile, compared with numerous pedestrian detection approaches, our combined method outperforms all of them and achieves a quite superior performance.

Keywords: Pedestrian detection · Faster R-CNN
Contextual information · Multi-level features

1 Introduction

As a popular topic in computer vision community, pedestrian detection has attracted plenty of attention for decades for its importance in many practical applications, such as video surveillance, tracking, mobile robotics and advanced driver assistance systems (ADAS). During the last decade, numerous methods have been proposed to improve the performance of pedestrian detection.

Recently, taking advantage of convolutional neural networks (CNN), many excellent deep learning models have promoted the object detection performance to a much higher level. As one of the most popular and successful models, Faster R-CNN [1] uses a deep fully convolutional network called Region Proposal Network (RPN) to generate high-quality Region of Interests (RoI), which are fed

K. Schoeffmann et al. (Eds.): MMM 2018, Part I, LNCS 10704, pp. 166–177, 2018.
https://doi.org/10.1007/978-3-319-73603-7_14

Fig. 1. Examples of hard positive and hard negative samples of Caltech dataset. These low-resolution samples are less discriminable from backgrounds. It's difficult to discriminate between them.

into the Fast R-CNN [2] detection network to simultaneously classify the object categories and regress the object bounding. Based on this pioneering work, many state-of-the-art detection models are proposed such as SSD [3], YOLO [4], and R-FCN [5].

Since pedestrian detection is a canonical case of the general object detection problem, many models derived from Faster R-CNN have been brought to this problem and taken the Caltech benchmark [6] top ranks [7–10]. Inspired by the great success of Faster R-CNN, we take the pipeline of [1] as the baseline of our work. However, Faster R-CNN has its shortcomings lie in two main aspects.

Firstly, as shown in Fig. 1, compared to general objects, pedestrians are less discriminable from backgrounds. These pedestrians usually appear in low resolution. Meanwhile, backgrounds such as vertical structures, tree trunk, and traffic lights easily bring about hard negative samples. During detecting, Faster R-CNN only uses information abstracted from a RoI close to the object, which is unable to discriminate between possible pedestrians and backgrounds, resulting in the increase of miss-rate. In addition, Faster R-CNN struggles with detecting small objects. For instance, the feature stride of VGG16 [11] last convolutional layer is 16. In a 600×1000 image, the feature scale of a 32×32 pedestrian will be just 2×2 on the last convolutional layer, which is too coarse to detect small objects.

To address the aforementioned limitations of Faster R-CNN, we explore enhancing Faster R-CNN to include two additional information sources to help pedestrian detection. Firstly, we incorporate contextual information. Context is known to be very useful for improving performance on deep learning based detection methods. It enables the detector to look wider around a pedestrian's RoI and makes a better discrimination between backgrounds and possible pedestrians, which helps reduce more false positive errors. Then, we utilize multi-level features which combine deep, coarse information with shallow, fine information to make features more abundant. These combined features can be more informative for detecting small pedestrians. In summary, the main contributions of this work are:

1. We integrate extra contextual information and multi-level features based on Faster R-CNN, which helps to detect pedestrians from cluttered backgrounds and small pedestrians.

2. Through an extensive experimental evaluation on the challenging Caltech pedestrian benchmark, we demonstrate that our approach not only performs better than the baseline of Faster R-CNN, but also boosts the detection performance when combined with contextual information and multi-level features.
3. Compared with numerous pedestrian detection approaches, our combined model yields a competitive result, which achieves a miss-rate of 14.0%.

This paper is organized as follows. Section 2 covers related works. Section 3 describes how to combine Faster R-CNN with contextual information and multi-level features. Section 4 presents the results of our experiments on Caltech dataset. Finally, Sect. 5 concludes this paper.

2 Related Work

In the past years, various efforts have been proposed to improve the performance of pedestrian detection. Current pedestrian detection methods can be generally grouped into two categories. The first category is known as hand-crafted approaches. Dalal and Triggs [12] firstly use the grids of Histograms of Oriented Gradient (HOG) descriptors, which significantly outperforms previous features. After that, most of the HOG based detection methods are proposed. Wang *et al.* [13] make full use of both HOG feature and LBP feature to handle partial occlusion. Furthermore, Deformable part-based models (DPM) [14] consider the appearance of each part and handle translational movement of parts. Besides, the Integral Channel Features (ICF) [15] and Aggregated Channel Features (ACF) [16] are among the most popular hand-crafted approaches, which efficiently extract features such as local sums, histograms, and Haar features using integral images.

Recently, object detection methods based on CNN have achieved very good performance [1–5]. Some recent works focus on improving the performance of pedestrian detection using CNN and push pedestrian detection results to an unprecedented level. Sermanet *et al.* [17] use an unsupervised method based on convolutional sparse coding to pre-train CNN for pedestrian detection. Based on Fast R-CNN, Li *et al.* [7] introduce multiple built-in sub-networks to detect pedestrians with scales from disjoint ranges. Tian *et al.* [18] improve pedestrian detection by learning high-level features from DNNs of multiple tasks, including pedestrian attribute prediction. Zhang *et al.* [8] use RPN [1] to generate a set of candidate region proposals and re-scored them with decision forest classifier trained over convolutional features. In [9], Tian *et al.* handle partial occlusion by training CNN with automatically selected part pool. Fused DNN [10] use a SSD [3] pedestrian candidate generator, a parallel classification network, and a pixel-wise semantic segmentation network to perform the detection.

Unlike the above CNN-based methods, our methods firstly make full use of contextual information and multi-level features together to perform pedestrian detection. It helps to detect pedestrians from cluttered backgrounds and small pedestrians.

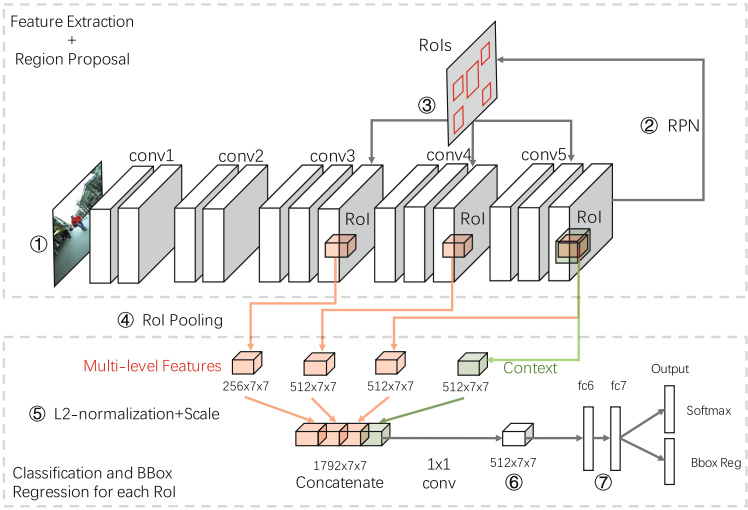


Fig. 2. The architecture of our proposed method. For an image, we extract features based on VGG16 model and generate 2000 RoIs. To evaluate each RoI, we extract contextual information from conv5_3 feature map, and extract multi-level features from several intermediate layers. Contextual information and multi-level features are normalized with L_2-norm, concatenated and compressed to a fixed-length feature descriptor. Finally, these features are fed into two fully-connected layers and produce two outputs: softmax probabilities and bounding-box regression offsets.

3 Proposed Method

In this section, we describe our proposed method. We begin with a brief overview of the entire framework, followed by specific details.

3.1 Framework

The framework of our method is shown in Fig. 2. It consists of two components: a fully convolutional Region Proposal Network (RPN) for proposal generation, and a downstream Fast R-CNN detector taking regions with high foreground likelihood as input. We choose Faster R-CNN not only for its prevalence and state-of-the-art performance, but also generality: our observations should remain mostly effective when similar techniques are applied in other CNN-based pedestrian detectors.

To detect pedestrians, a deep VGG16 model processes an image and generates the hierarchical convolutional feature maps. By performing an RPN on the last feature map conv5_3, we obtain thousands of high-quality region proposals that might contain pedestrians. For each RoI, we first pool contextual information from conv5_3. Then we pool a fixed-length feature descriptor from several layers (conv3_3, conv4_3, and conv5_3) to form the multi-level features. The contextual information and multi-level features are normalized with L_2-norm, concatenated

and compressed (1×1 convolution) to produce a fixed-length feature descriptor of size $512\times7\times7$. Finally, the fixed-length features are fed into two fully-connected layers and produce two sibling output layers.

The first sibling layer outputs softmax probability values, $p = (p_0, p_1)$, over "background" and pedestrian classes. p_0 and p_1 denote the probability values of "background" and pedestrian classes respectively. The second sibling layer outputs the bounding-box regression offsets for pedestrian class, which is denoted as $t = (t_x, t_y, t_w, t_h)$. For each training region proposal, there is a ground-truth class label u and a ground-truth bounding-box regression target v. To jointly train for classification and bounding-box regression, we minimize an objective function using following multi-task loss L on each labeled proposal:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \tag{1}$$

where $L_{cls} = -logp_u$ is log loss for true class u.

The second task loss L_{loc}, which is defined over the true bounding-box regression targets for class u, $v = (t_x, t_y, t_w, t_h)$, and a predicted tuple (formula) again for class u. The Iverson bracket indicator function $[u \geq 1]$ evaluates to 1 when $u \geq 1$ and 0 otherwise. By convention the catch-all background class is labeled $u = 0$. We ignore the L_{loc} for background RoIs since there is no notion of a ground-truth bounding box. For bounding-box regression, we use the loss:

$$L_{loc}(t^u, v) = \sum_{i\in\{x,y,w,h\}} smooth_L(t_i^u - v_i) \tag{2}$$

where the robust loss $smooth_L(\cdot)$ is defined as:

$$smooth_L(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{3}$$

The parameter λ controls the balance between the two task losses. The ground-truth regression targets v is normalized to have zero mean and unit variance. By default we set $\lambda = 1$ in all experiments. We use the stochastic gradient descent (SGD) algorithm to calculate the minimization of loss function.

3.2 Context Embedding

Context is known to be useful for improving the performance of deep learning based detection and segmentation tasks. By visualizing the original Faster R-CNN detection results on Caltech testing set, we observe numerous false positive errors. These false positive errors mainly result from backgrounds of vertical structures, tree leaves, or traffic lights. The behind reason is that Faster R-CNN only uses information involved in a RoI close to the object. However, the region around a RoI always contains additional information that may provide visibility over larger ranges, which enables the detector to look wider around a RoI and makes a better discrimination between possible pedestrians and backgrounds.

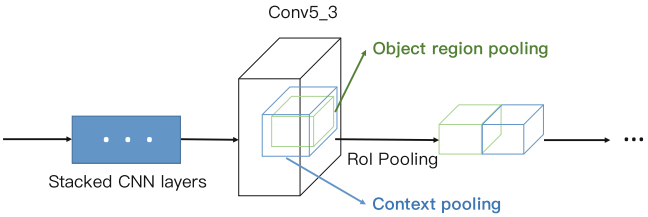


Fig. 3. Contextual information extraction. The green (blue) cubes represent object (context) region pooling. (Color figure online)

This information is known as contextual information. Thus, from the above observation, we hypothesize that adding additional contextual information to Faster R-CNN is helpful to reduce false positive errors. As shown in Fig. 3, in order to gather contextual information on conv5_3, we scale the size of the original proposal box by a factor of 1.5. Specifically, when performing the RoI pooling on conv5_3, we pool from 1.5× larger area than the original proposal region, so that the pooling features contain both original object and contextual information.

3.3 Multi-level Feature Extraction

Faster R-CNN struggles with detecting small objects. It pools from the last convolutional feature map. However, for small pedestrians (50–70 pixels for Caltech), the last feature map always has low resolution (usually with a stride of 16 pixels). When RoI pooling layer is performed on such low-resolution feature map, it will lead to "plain" features caused by collapsing bins. These features are not discriminative on small regions and too coarse for classification of small pedestrians. Thus, in order to detect small pedestrians, the features for region proposal and detection should be more informative and the feature resolution should be more reasonable.

We address this problem by pooling proposal-specific features from several shallow but high-resolution layers to form the multi-level features. For instance, we extract features from RoIs on conv3_3 (of a stride = 4 pixels), conv4_3 (of a stride = 8 pixels) and conv5_3 (of a stride = 16 pixels) and concatenate them together. However, since these features have different scales and norms, they should not be concatenated simply. Naively concatenating features leads to poor performance as the features of larger value will dominate the smaller ones. Thus, inspired by [19], we normalize each individual feature with L_2-norm first, and learn to scale each separately, as described in Sect. 3.4. It makes the training more stable and improves performance. Then we concatenate these normalized features together to form the multi-level features and use a convolutional layer to reduce the final feature to the shape of $512 \times 7 \times 7$, so that it has the correct dimension to feed into the first fully-connected layer (fc6). The convolutional layer is initialized with "Xavier" algorithm [20] and has the filter size of 1×1. In

this way, we combine finer, high-resolution features with coarse, low-resolution features for better classification of small pedestrians.

3.4 L_2-normalization and Scale

For a layer with d-dimensional input $\mathbf{x} = (x_1 \dots x_d)$, we use L_2-norm to normalize it with $\hat{\mathbf{x}} = \mathbf{x}/||\mathbf{x}||_2$, where $||\mathbf{x}||_2 = (\sum_{i=1}^{d} |x_i|^2)^{1/2}$ is the L_2-norm. However, without scaling each input of layer accordingly, simply normalizing the input will change the scale of the layer and will slow down the learning. Thus, it's necessary to learn the scale for each channel by introducing a scaling parameter γ_i, which scales the normalized value by $y_i = \gamma_i \hat{x_i}$. During training, we use back-propagation and chain rule to compute derivatives with respect to scaling factor γ and input data x. We use the implementation of L_2-normalization layer from [19].

4 Experiments

4.1 Dataset for Training and Testing

We choose Caltech dataset [6] to train and test our model. It is one of the most widely used pedestrian datasets. It consists of about 250,000 frames with a total of 350,000 bounding boxes, where 2,300 unique pedestrians are annotated. For all experiments, we train our model on the improved Caltech-10x annotations from [21], which are of higher quality than the original annotations. We train the model with two categories (person and ignore) depending on whether the bounding box is annotated as "ignore". For evaluation, we adopt the original annotations and follow the standard Caltech evaluation [6]: log miss-rate (MR) is averaged over the FPPI (false positives per image) of the range $[10^{-2}, 10^{0}]$. Unless otherwise specified, all experiments are evaluated on the "reasonable" setup (pedestrians over 50-pixel height with no or partial occlusion).

4.2 Experimental Setup

The whole network is trained on NVIDIA Tesla GPU K80 with 12 GB memory. Both region proposal and object detection networks are trained on a single-scale image. We rescale the images so that their shorter side is 600 pixels. For RPN training, it is performed on the last convolutional layer. An anchor is considered as a positive example if it has an Intersection-over-Union (IoU) greater than 0.7 with one ground truth box, and IoU less than 0.3 is considered as negative. To reduce redundancy proposals, we adopt non-maximum suppression (NMS) with IoU threshold at 0.7, which leaves us about 2000 proposal regions per image. We adopt approximate joint training strategy for sharing features between RPN and Fast R-CNN. We train with a learning rate of 0.001 for 60k mini-batches, and 0.0001 for the next 20k mini-batches. We use a momentum of 0.9 and a weight decay of 0.0005. Other details are as in [1] and we adopt the publicly available code of [1].

After training the Baseline-Faster R-CNN model with above parameter settings, it achieves 20.3% miss-rate, which underperforms on the pedestrian detection task. It is because Caltech dataset contains abounding small pedestrians and original Faster R-CNN fails to handle them. To generate more proposals for small sizes, we first slightly modify Faster R-CNN to generate anchors with 10 scales starting from the scale of 2.0 with a stride of 1.3×. Then we adopt anchors of a single aspect ratio of 2.44 (height to width), which is the average pedestrian aspect ratio of Caltech dataset. With this modification, we reduce the miss-rate from 20.3% down to 18.5%, as shown in Table 1.

4.3 Ablation Experiments

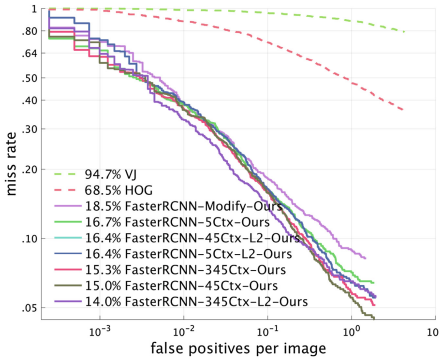
In this subsection, we conduct ablation experiments on the Caltech dataset. We conduct 5 different experiments, each one combined with different extra features. The overall experimental results are presented in Table 1.

Table 1. Combining context with features from different layers. Metric: log-average miss-rate on Caltech benchmark. M: modify the anchor scales and ratio aspects. Ctx: combine with contextual information. C5: combine with features pooled from conv5_3. C4: combine with features pooled from conv4_3. C3: combine with features pooled from conv3_3. $*$: This entry is the unstable result when combine multi-level features without L_2 and scale.

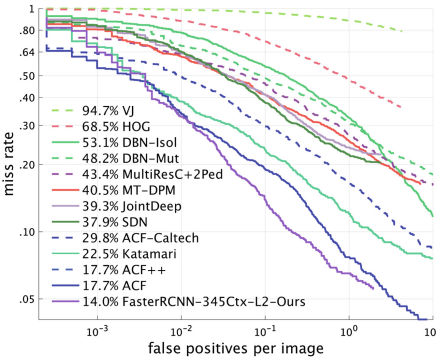
Exp. no.	RoI pooling from	M	Ctx	MR (without $L_2 + \text{Scale}$)	MR (with $L_2 + \text{Scale}$)
1	C5 (baseline)			20.3%	-
2	C5	√		18.5%	18.9%
3	C5	√	√	16.7%	16.4%
4	C5 + C4	√	√	15.0%	14.7%
5	C5 + C4 + C3	√	√	$*$15.3%	**14.0%**

Effects of Multi-level Features. To investigate the impact of multi-level features, we first train several Faster R-CNN models with features that are pooled from different layers. These models are combined with contextual information. When concatenating these features, we adopt L_2-normalization and scale operations to make the features from different layers have similar magnitude. As shown in Table 1, we clearly observe that the miss-rate appears to be lower due to more features pooled from different layers are added (C3 & C4). It indicates that multi-level features can improve the detection performance consistently. This is because lower convolution features (C3 & C4) are finer and more high-resolution, which is more informative for detecting small pedestrians.

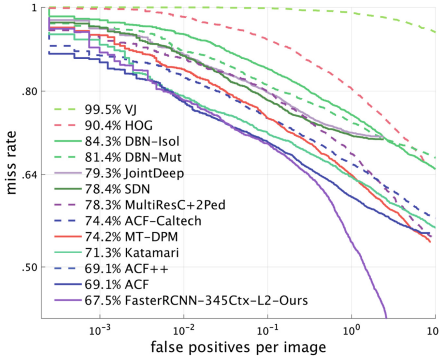
Effects of Contextual Information. Next, we disentangle the influence of contextual information. For this purpose, we train different multi-level models with context embedding as mentioned in Sect. 3.2. L_2-normalization and scale



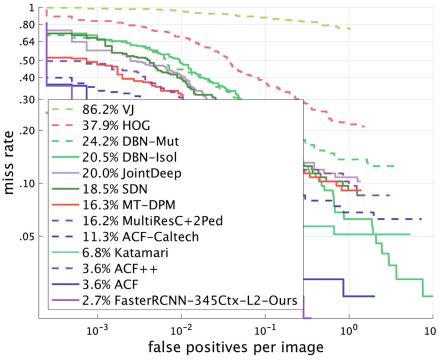
(a) Combining context with features from different layers.



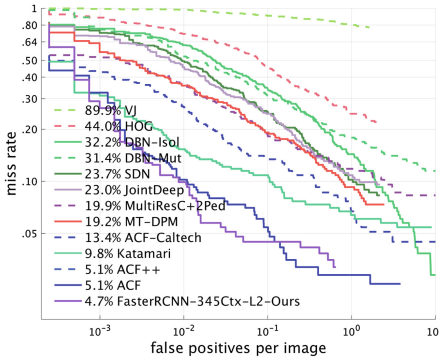
(b) Reasonable ($\geq$ 50 pixels high)



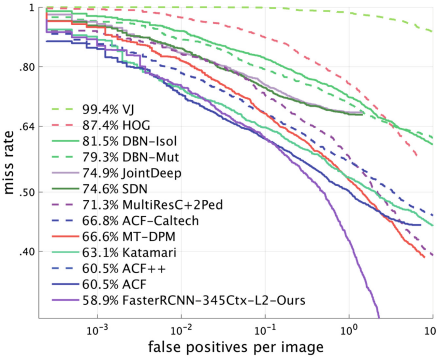
(c) All scales ($\geq$ 30 pixels high)



(d) Large scale($\geq$ 100 pixels high)



(e) Near scale ($\geq$ 80 pixels high)



(f) Medium scale (30 - 80 pixels high)

Fig. 4. The comparison of our approach for pedestrian detection with recent state-of-the-art methods on Caltech benchmark.

operations are also adopted. Table 1 presents that with context embedding (Exp. 3, Exp. 4 & Exp. 5), it leads the miss-rate to be 16.4%, 14.7% and 14.0% receptively, which are all lower than baseline result (20.3%). It demonstrates that the detection performance has been actually boosted by contextual information. When without L_2-normalization and scale, contextual information still leads to the decrease of miss-rate. Figure 4(a) presents the ROC curves when combining context with features from different layers.

Effects of L_2-normalization and Scale. As mentioned above, our detector pools features from multiple layers and combines them for pedestrian detection. From Table 1 (fourth column), we observe that when adding C3, the result of MR (without L2+Scale) becomes worse. This is because features from different layers always have a much different scale and norm, if we simply concatenate the features from each layer and reduce the dimensionality using a 1×1 convolutional operation, the performance gain is unstable. Thus, it's necessary to normalize each individual feature with L_2-norm first, and then learn to scale each separately, so that features pooled from all layers have similar magnitude. As shown in Table 1 (fifth column), after adopting L_2-normalization and scale, the problem is fixed and leads to a competitive result (14.0%).

4.4 Comparison with State-of-the-Art Methods

We compare our method with hand-crafted models such as VJ, HOG, ACF, MT-DPM, MultiResC + 2Ped, Katamari. And we also compare with deep models including DBN-Isol, DBN-Mut, SDN, and JointDeep. The overall experimental results are depicted in Fig. 4(b)–(f). We can clearly observe that:

1. The proposed method performs significantly better than the baseline detector (Faster R-CNN) on reasonable test set (in Fig. 4(b), our miss-rate is **14.0%**, Faster R-CNN miss-rate is 20.3%).
2. As shown in Fig. 4(c)–(f), the proposed method outperforms all the other state-of-the-art methods on all of the four tests, indicating that our method is an effective way for pedestrian detection, especially in multi-scale cases.
3. As shown in Fig. 4(f), the proposed method performs better than all the other methods on medium test sets (pedestrian of 30–80 pixels high, which contains small scale pedestrians), demonstrating that the proposed detector truly benefits from combining features from multi-levels in the training phase.

5 Conclusion

Based on Faster R-CNN, we have presented a method that combines contextual information with multi-level features for pedestrian detection task. On the one hand, the contextual information, which is gathered by enlarging the original object region scale, helps the detector make a better discrimination between possible pedestrians and backgrounds. On the one hand, the multi-level features,

which can be obtained by pooling proposal-specific features from several shallow but high-resolution intermediate layers, are more informative for detecting small-size pedestrians. Extensive experiments validate that the detection performance has been actually boosted by contextual information and multi-level features. Meanwhile, experiments also show that when concatenating features from different level layers, it's necessary to normalize and scale each individual feature. Compared with recent state-of-the-art methods, our combined method achieves a superior result on Caltech dataset.

Acknowledgment. This project was supported by Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (ZDSYS201703031405467), Shenzhen Peacock Plan (20130408-183003656), and National Science Foundation of China (No. U1611461).

References

1. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR abs/1506.01497 (2015)
2. Girshick, R.: Fast R-CNN. In: ICCV, pp. 1440–1448 (2015)
3. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
4. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once - unified, real-time object detection. CoRR cs.CV (2015)
5. Dai, J., Li, Y., He, K., Sun, J.: R-FCN - object detection via region-based fully convolutional networks. In: NIPS (2016)
6. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. **34**(4), 743–761 (2012)
7. Li, J., Liang, X., Shen, S., Xu, T., Yan, S.: Scale-aware fast R-CNN for pedestrian detection. CoRR (2015)
8. Zhang, L., Lin, L., Liang, X., He, K.: Is faster R-CNN doing well for pedestrian detection? In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 443–457. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_28
9. Tian, Y., Luo, P., Wang, X., Tang, X.: Deep learning strong parts for pedestrian detection. In: ICCV, pp. 1904–1912 (2015)
10. Du, X., El-Khamy, M., Lee, J., Davis, L.S.: Fused DNN: a deep neural network fusion approach to fast and robust pedestrian detection, October 2016. arXiv.org
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)
12. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
13. Wang, X., Han, T.X., Yan, S.: An HOG-LBP human detector with partial occlusion handling. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 32–39. IEEE (2009)

14. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
15. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features (2009)
16. Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1532–1545 (2014)
17. Sermanet, P., Kavukcuoglu, K., Chintala, S., LeCun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3626–3633 (2013)
18. Tian, Y., Luo, P., Wang, X., Tang, X.: Pedestrian detection aided by deep learning semantic tasks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 5079–5087 (2015)
19. Liu, W., Rabinovich, A., Berg, A.C.: ParseNet: looking wider to see better, June 2015. arXiv.org
20. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) Proceedings of 13th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, Chia Laguna Resort, PMLR, Sardinia, Italy, vol. 9, pp. 249–256, 13–15 May 2010
21. Zhang, S., Benenson, R., Omran, M., Hosang, J., Schiele, B.: How far are we from solving pedestrian detection? In: CVPR (2016)