IMPROVING TINY VEHICLE DETECTION IN COMPLEX SCENES

Wei Liu[†,1,3], *Shengcai Liao*[*,1,2], *Weidong Hu*[3], *Xuezhi Liang*[1,2], *Yan Zhang*[3]

[1]Center for Biometrics and Security Research & National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing ,China
[2]University of Chinese Academy of Sciences, Beijing ,China
[3]College of Electronic Science, National University of Defense Technology, Changsha, China
{liuwei16, wdhu, atrthreefire}@nudt.edu.cn, scliao@nlpr.ia.ac.cn, liangxuezhi15@mails.ucas.ac.cn

ABSTRACT

Vehicle detection is still a challenge in complex traffic scenes, especially for vehicles of tiny scales. Though RCNN based two-stage detectors have demonstrated considerably good performance, less attention has been paid to the quality of the first stage, where, however, tiny vehicles are very likely to be missed. In this paper, we propose a deep network for accurate vehicle detection, with the main idea of using a relatively large feature map for proposal generation, and keeping ROI feature's spatial layout to represent and detect tiny vehicles. However, large feature maps in lower levels of a deep network generally contain limited discriminant information. To address this, we introduce a backward feature enhancement operation, which absorbs higher level information step by step to enhance the base feature map. By doing so, even with only 100 proposals, the resulting proposal network achieves an encouraging recall over 99%. Furthermore, unlike a common practice which flatten features after ROI pooling, we argue that for a better detection of tiny vehicles, the spatial layout of the ROI features should be preserved and fully integrated. Accordingly, we use a multi-path light-weight processing chain to effectively integrate ROI features, while preserving the spatial layouts. Experiments done on the challenging DETRAC vehicle detection benchmark show that the proposed method largely improves a competitive baseline (ResNet50 based Faster RCNN) by 16.5% mAP, and it outperforms all previously published and unpublished results.

Index Terms— Vehicle detection, Object proposal, Deep neural network

1. INTRODUCTION

Recently, generic object detection has gained great success, especially driven by the deep Convolutional Neural Networks (CNN). However, vehicle detection in complex traffic scenes still encounters a number of challenges, such as various lighting conditions, occlusions, and low resolutions.

† Wei Liu finished his part of work during his visit in CASIA.
* Shengcai Liao is the corresponding author.

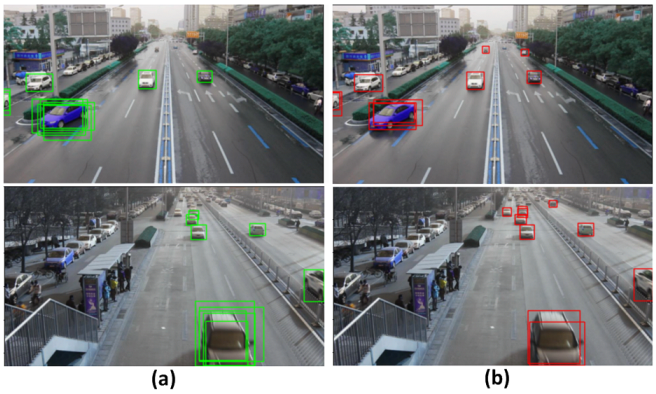

Fig. 1. Top 10 proposals of (a).RPN [1] and (b).BFEN. Visibly, BFEN generates proposals with better accuracy and has a higher recall than RPN (89.68 % vs 63.32 %), especially for tiny vehicles (<20 pixels).

Beyond early studies which focused on hand-craft features [2] and cascade classifier structure [3], RCNN [4] firstly introduced CNN into object detection, followed by Fast-RCNN [5], Faser RCNN [1] and R-FCN [6]. These methods are called two-stage detectors, where proposals are generated in the first stage and refined in the second stage. Although considerably good performance has been achieved [7, 8], less attention has been paid to the quality of the proposals generated in the first stage. A high recall of proposals is critical for the overall vehicle detection performance because missed objects can not be retrieved in the second stage, especially for tiny vehicles (<20 pixels) in traffic scenes. In this paper, we focus on performing vehicle detection with high-recall. Specifically, we firstly answer the question of how to generate high-quality proposals, and secondly how to refine these proposals for better localization accuracy.

Thanks to the strong feature representation of CNN, proposal generation can be embedded in a network, deemed as Region Proposal Network (RPN) in Faster RCNN [1]. Although multi-scale anchors can be assigned on the last layer

978-1-5386-1737-3/18/$31.00 ©2018 IEEE

of a strong backbone network (e.g. ResNet50), RPN has a suboptimal performance on recalling vehicles of various scales, especially the tiny ones, as depicted in Figure 1 (a). This is because the last layer with large respective field are illy matched with small objects. One solution is to utilize multi-layers to generate proposals (e.g. SSD [9] and MSCNN [10]). However, feature representation of lower layers are less discriminative than that of higher layers. To address this problem, we design a backward feature enhancement strategy to transport more semantic information from high layers to low layers, thus features at low layers are fine-grained in spatial resolution and discriminant in representation. This is denoted as Backward Feature Enhancement Network (BFEN). As shown in Figure 1 (b), our BFEN performs considerably well on tiny vehicles and achieves an encouraging recall rate over 99% with only 100 proposals on the DETRAC validation set.

With these high-quality proposals, we aim at further improving localization accuracy in the second stage. To improve the discrimination of the proposals' features, fully-connected (FC) layers are commonly adopted in existing two-stage detectors (i.e.[1] and [11]), but FC layers intrinsically corrupts the spatial layout of features, which is fairly harmful in detecting tiny vehicles. Hence, inspired by [12], we stack two *split-transform-merge* building blocks, which evolve features progressively while preserve the spatial information, resulting in a better detection network.

With the above vehicle detection structure, we achieve a new state of the art on the DETRAC benchmark [8], compared with all published and unpublished results. Notably, the proposed method[1] significantly improves a competitive baseline (ResNet50 based Faster RCNN) by 16.5% mAP.

2. RELATED WORK

Previous works on vehicle detection are mainly based on hand-crafted features [2] and cascade classifier framework [3]. Another line of research introduced deformable part based model (DPM) [13], which explicitly explored spatial structure for vehicle detection, thus yielded good results [14].

Recent works achieved remarkably better performance thanks to the rich feature representation of CNN. Among various CNN detection methods, it can be roughly classified into two categories. The first type is pioneered by RCNN [4] based two-stage methods [5, 1, 6], which first generate plausible region proposals, then refine proposals by another sub-network for bounding box classification and regression. To generate proposals in a unified framework, Faster RCNN [1] devised a Region Proposal Network sharing the base network with detection network, thus can be trained jointly. The second type of methods [9, 15], which are called single-stage methods, aims at speeding up detection by removing the region proposal generation stage. These single-stage detectors directly regress object locations from multiple layers of the base network and thus are more computationally efficient but yield less satisfactory results than two-stage methods.

To achieve strong feature representation, fusing feature maps from different layers has been extensively studied [11, 16]. Especially, [11] proposed a feature pyramid network for object detection, which enhanced feature representation from various levels for generic object detection, while small objects from real applications are not aware of. Motivated by this, we design a backward feature enhancement network, focusing on low-level, large, but boosted feature map for tiny vehicle representation. Perceptual GAN [17] is a recent work for small object detection, which explicitly generates super-resolved feature maps of small objects through GAN, while we remain traditionally in boosting the representation of a basic network.

In two-stage detectors, much progress has been evolved with better CNN architecture in the first stage, while less effort has been made in the second stage. In [18], it is demonstrated that the design of the detection network is as important as the proposal network. As a common practice, FC layers are adopted in two-stage detectors [5, 1, 11] for classification and regression after ROI pooling, which is suboptimal as FC is unable to preserve the spatial layout of input and a large number of parameters in FC also costs a heavy computational burden. Motivated by the simplicity of block design in ResNeXt [12], we simply stack two *split-transform-merge* blocks to evolve the proposals' features progressively, with preserved spatial layouts and boosted discrimination, while with only 1.4% parameters of a two-FC counterpart.

3. PROPOSED METHOD

3.1. Backward Feature Enhancement Network (BFEN)

Feature matters. In Faster RCNN [19], only the final layer is utilized for proposal generation, which is not good for tiny vehicles. A better way is to exploit multiple layers with different resolutions as in SSD [9] and MSCNN [10]. However, large feature maps in lower levels of a deep network generally contain limited discriminant information. In this paper, as depicted in Figure 2 (a), we firstly emanate branches from the last layers of *stage 3, 4 and 5* in ResNet50, respectively, then deliver more semantic information from higher layers to lower layers by the following strategy:

$$\hat{\Phi}_n = c_n(\Phi_n) \oplus d_{n+1}(c_{n+1}(\Phi_{n+1})), n = 3, 4, \quad (1)$$

where Φ_n is the feature map of the last layer of *stage* n, $c_n(.)$ represents a convolution layer to reduce the original feature maps' channel dimensions, $d_n(.)$ represents a deconvolution layer to upsample feature maps by a factor of 2, and $\oplus$ denotes element-wise addition. With this backward transmission, the feature maps from the first two stages are enhanced and thus are more competent for small object detection. For large scale

[1]Code will be available at https://github.com/VideoObjectSearch/

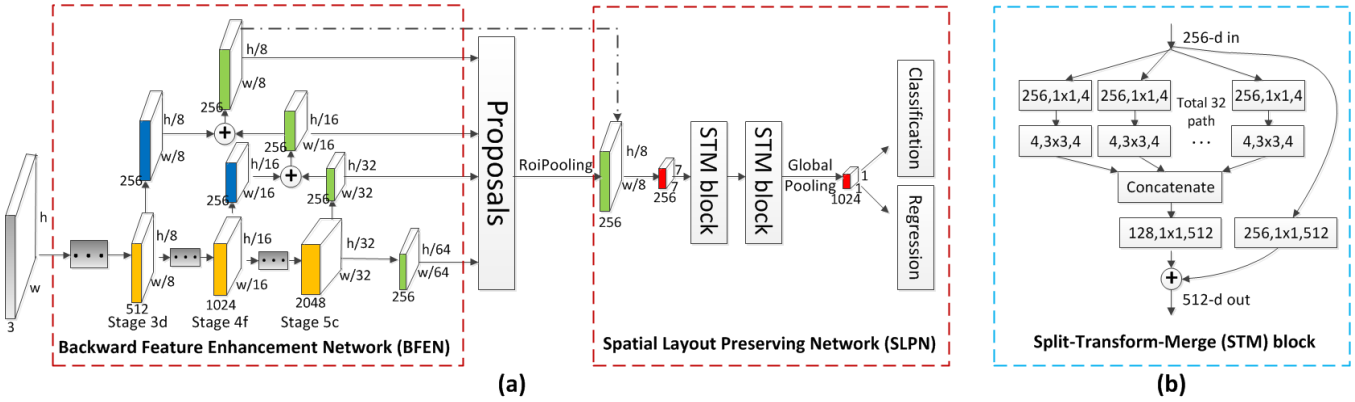

Fig. 2. (a) depicts the detailed architecture of the proposed method. It contains a backward feature enhancement network (BFEN) and a spatial layout preserving network (SLPN), with the formal one responsible for accurate proposal generation and the latter one refining these proposals. (b) demonstrates the *split-merge-transform* block adopted in SLPN.

objects, we attach two convolution layers with a stride of 1 and 2 on Φ_5, generating two branches $\hat{\Phi}_5$ and $\hat{\Phi}_6$, respectively. Then the final set of feature maps are $\{\hat{\Phi}_3, \hat{\Phi}_4, \hat{\Phi}_5, \hat{\Phi}_6\}$, with 256 channels and resolutions of $1/8, 1/16, 1/32$ and $1/64$ of the input image, respectively. For proposal generation, anchors of $\{16^2, (32^2, 64^2), 128^2, 256^2\}$ pixels, with aspect ratios $\{1, 1, (0.5, 1, 2), (0.5, 1, 2)\}$, respectively, are assigned to each level of feature map. Then, we append the same design as in Faster-RCNN [1] for bounding box classification and regression.

3.2. Spatial Layout Preserving Network (SLPN)

To further increase localization accuracy, we design a light-weight yet powerful detection network. We attach a ROI pooling layer [1] after $\hat{\Phi}_3$ to extract feature vectors of fixed dimension (i.e., 7x7x256) to represent proposals. We chose $\hat{\Phi}_3$ primarily because feature maps with higher resolution provide more information for small objects as suggested in [10]. However, a major difference to MSCNN [10] is that, the $\hat{\Phi}_3$ feature maps are enhanced by higher layers with more semantic information. Despite the strong representation power from the enhanced feature map, the spatial layout of ROI pooling features is also critical for vehicle classification and regression, thus should be preserved when evolving progressively in the second stage. However, in the original Faster RCNN, this spatial information are corrupted by FC layers. Motivated by the superior classification performance of ResNext [12] at a considerably lower computational complexity, we choose the *split-transform-merge* building block (illustrated in Figure 2 (b)) as an alternative to FC. In this way, features are evolved progressively while preserving the spatial layout. We simply stack two such blocks, with 512 and 1024 output channels, respectively. The final outputs are then fed to a global average pooling layer and two sibling FC layers for bounding box classification and regression.

3.3. Soft-style Hard Mining

Positive-negative imbalance is a common combat in classification problems. Faster-RCNN randomly selects a fixed set of samples (i.e. 256) with equal weights, while [20] performs a hard-style hard mining by choosing samples with top loss values. In this section, we exploit a soft-style hard mining, which shares a similar idea to [16]. A difference is that, our hard mining is applied on sampled positives S_+ and negatives S_-, with $|S_-| = \alpha|S_+|$. The soft-style hard mining is formulated as:

$$l_{cls} = -\frac{1}{1+\alpha}\sum_{i\in S_+}(1-p_i)^2\log(p_i) - \frac{\alpha}{1+\alpha}\sum_{i\in S_-}p_i^2\log(1-p_i), \tag{2}$$

where p_i is the positive probability score of sample i. With Eq. (2), the contributions of easy samples are down-weighted, thus acting as a kind of hard mining.

3.4. Training

Loss function We choose the ResNet50 [19] pretrained on the ImageNet [21] as the base network. All other layers are randomly initialized with the 'xavier' method. Both BFEN and SLPN have a multi-task loss combining two objectives:

$$L = l_{cls} + \lambda[y = 1]l_{loc}, \tag{3}$$

where the classification loss l_{cls} is defined in Eq. (2), and the regression loss l_{loc} is the same smooth L1 loss adopted in Faster-RCNN [1]. Then our model can be jointly trained by the two-stage loss function:

$$L_{total} = L_{BFEN} + L_{SLPN}, \tag{4}$$

where L_{BFEN} and L_{SLPN} are the objective of BFEN and SLPN respectively.

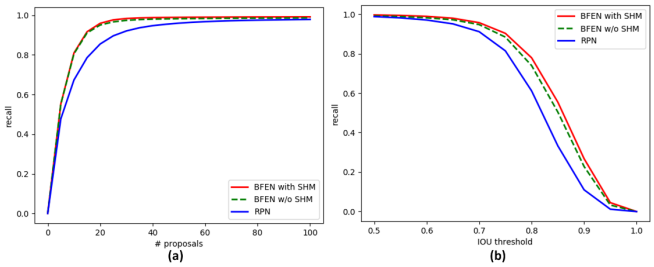

Fig. 3. Proposal recall rate comparison on validation set. (a) Recall v.s. number of proposals at IoU threshold 0.5. (b) Recall v.s. IoU threshold for 300 proposals.

Proposal network warm-up Jointly optimizing Eq. (4) from scratch makes training unstable in early iterations as we found in experiments. Thus we adopt a proposal network warm-up strategy. Specifically, we firstly train BFEN for 20k iterations with a learning rate of 10^{-4}, then the resulting model is used as the start point for jointly optimizing L_{total}. For joint learning, we set initial learning rate as 10^{-5} and 10^{-4} for BFEN and SLPN, respectively, then decrease them by 10 after 60k iterations, with the maximum of iterations 110k.

4. EXPERIMENTS

The proposed method was evaluated on the recent DETRAC benchmark [8] which contains a substantial amount of challenging vehicles like tiny cars, severely overlapped or truncated buses. The DETRAC dataset has 84K images for training and 56K for testing. As the ground truth of test set is not publicly available, we split the training set into 56K images for learning and 28K for validation following EB [7].

We conducted three experiments to evaluate the effectiveness of the proposed method. The first one examined the quality of region proposals, the second one analyzed each component's contribution to the final detection performance, and the last one reported the comparison with state-of-the-art methods on the benchmark test set.

4.1. Region proposal evaluation

For fair comparison, we reimplemented **RPN** in Faster-RCNN [1] with the same backbone network (ResNet50), scales and aspect ratios of default anchors are also the same as our method. To better understand the effectiveness of the Soft-style Hard Mining proposed in Section 3.3, we trained a variant of BFEN without Soft-style Hard Mining, denoted as **BFEN w/o SHM**. Following [22], an object is recalled if any proposal has IoU higher than 0.5. Results are listed in Table 1 and Figure 3(a). With only 10 proposals, our BFEN achieves a recall above 80%, significantly outperforming RPN by 20.3%. Table 2 further demonstrates the effectiveness of

Table 1. Proposal recall rates of different methods on validation set.

# proposals	10	100	300
RPN	0.6733	0.9789	0.9879
BFEN w/o SHM	0.8053	0.9868	0.9915
BFEN with SHM	**0.8097**	**0.9920**	**0.9962**

Table 2. Recall rates of vehicles of different scales on validation set. The scales are defined as the square root of their area in pixels. Results of top 300 proposals at IoU threshold 0.5 are reported.

	Tiny $[0, 20]$	Small $(20, 50]$	Medimum $(50, 150]$	Large $(150,)$
RPN	0.6332	0.9924	0.9965	0.9955
BFEN w/o SHM	0.7730	0.9930	0.9973	**0.9998**
BFEN with SHM	**0.8968**	**0.9986**	**0.9980**	0.9982

BFEN for tiny vehicles(< 20 pixels), with top 300 proposals, BFEN increases the recall of tiny vehicles of RPN by a large margin (63.32% to 89.68%). To better understand the high quality of proposals generated from BFEN, Figure 3 (b) depicts the recall based on different IoU threshold, it highlights the performance gap between RPN and BFEN when IoU threshold increasing progressively, which gives the evidence that BFEN can generate proposals more accurately than RPN. Intriguingly, training with the Soft-style Hard Mining, BFEN can perform slightly better at a higher IoU threshold.

4.2. Detection evaluation

For detection, we feed the top 300 proposals from BFEN to the detection network, and filter out those boxes with detection scores less than 0.1. The final results are obtained by applying NMS with a threshold of 0.7. Table 3 reports the results on validation set, the first two rows are results from the EB paper [7]. We also reimplement a variant of Faster RCNN according to [19], with the same backbone network (ResNet50), denoted as **Faster RCNN*** in Table 3. Firstly, we test BFEN alone as a detector, secondly, we replace the SMT blocks in SLPN with two FC layers with 4096 nodes, which is jointly trained with BFEN (denoted as **BFEN+2FC** in row 5). Then we jointly train BFEN and SLPN from scratch (denoted as **BFEN+SLPN**) and report the results in row 6, finally we jointly trained our model with the Proposal Network Warm-up strategy mentioned in section 3.4 and the results are reported in the last row. It indicates that BFEN can already work well as a detector with a competitive mAP of 82.22 % on the whole validation set, beating the Faster-RCNN* baseline by approximately 10 % in mAP, showing the high quality of our generated proposals. When combining BFEN with a detection network consisting of two fully-connected layers, the

Table 3. Mean average precision (mAP) of different detectors on validation set as well as different subsets. '*' represents the variant of Faster RCNN reimplemented by us as a competitive baseline.

Detectors	Overall	Sunny	Cloudy	Rainy	Night
Faster RCNN[1]	68.58	63.64	70.04	81.56	60.53
EB[7]	84.43	87.48	85.88	85.65	70.86
Faster RCNN*	72.72	88.00	73.96	66.26	68.33
BFEN	82.22	85.38	83.56	80.43	72.69
BFEN+2FC	83.00	83.90	82.73	90.21	74.98
BFEN+SLPN	87.19	89.13	87.35	92.21	76.84
BFEN+SLPN+PNW	**88.71**	**90.01**	**89.46**	**92.64**	**78.48**

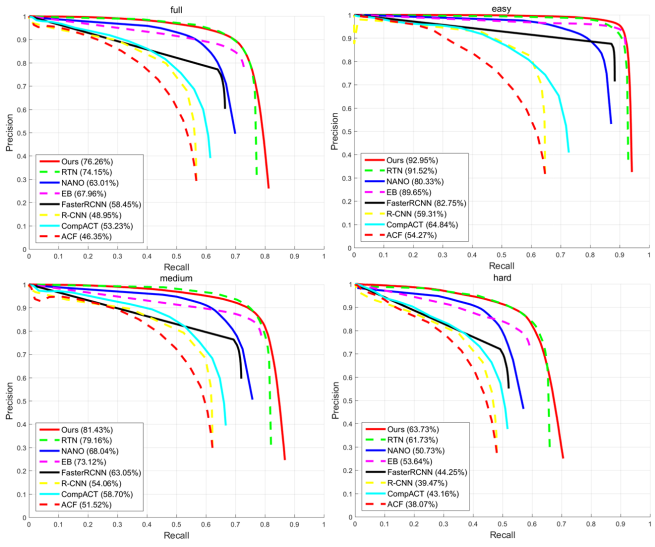

Fig. 4. Precision-recall curves of state-of-the-arts on the full, easy, medium and hard subsets of the DETRAC test set.

improvement is marginal. Jointly training BFEN with SLPN achieves a mAP of 87.19 %. Finally the Proposal Network Warm-up strategy further boosts the performance to an mAP of 88.71 %, significantly outperforming the Faster RCNN* baseline by 16.5 % in mAP.

4.3. Comparisons to the state of the arts

The results of the proposed method trained with both training and validation set were submitted to the DETRAC benchmark. Comparison with previous published and unpublished methods is reported in Table 4. The proposed method achieves a new state-of-the-art on the leaderboard, and ranks first on almost all subsets except for the easy and night subset. Notably, we achieved a significant overall improvement of 8.3 % and 6.4 % mAP over the state-of-the-arts EB [7] and RFCN [6], respectively. Figure 4 further gives the precision-recall curves of state-of-the-arts, it is shown that our method achieves the best performance across all difficulty settings.

5. CONCLUSIONS

We aim at improving the performance of vehicle detection in complex scenes, especially for vehicles of tiny scales. To achieve this target, larger feature maps at lower levels are focused, and a backward feature enhancement network is proposed to generate high-recall proposals. We also show that a spatial layout preserving structure is helpful to the overall detection performance. Combining the above two ideas, the proposed method achieves a new state-of-the-art on the DETRAC benchmark.

Acknowledgment

This work was supported by the National Key Research and Development Plan (Grant No.2016YFC0801002), the Chinese National Natural Science Foundation Projects #61672521, #61473291, #61572501, #61502491, #61572536, CASIA Distinguished Young Cadre Project, and AuthenMetric R&D Funds.

6. REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[2] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[3] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2005, vol. 2, pp. 236–243.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[5] R. Girshick, "Fast r-cnn," in *IEEE International Conference on Computer Cision*, 2015, pp. 1440–1448.

[6] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[7] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, "Evolving boxes for fast vehicle detection," *arXiv preprint arXiv:1702.00254*, 2017.

Table 4. mAP results on the DETRAC leaderboard ('*' represent unpublished works).

Detectors	Overall	Easy	Medium	hard	Sunny	Cloudy	Rainy	Night	Speed	Environment
DPM[13]	25.70	34.42	30.29	17.62	24.78	30.91	25.55	31.77	6s/img	CPU@2.4GHz
ACF[2]	46.35	54.27	51.52	38.07	58.30	35.29	37.09	66.58	1.5s/img	CPU@2.4GHz
RCNN[4]	48.95	59.31	54.06	39.47	59.73	39.32	39.06	67.52	10s/img	GPU@K40
CompACT[23]	53.23	64.84	58.70	43.16	63.23	46.37	44.21	71.16	4.5s/img	GPU@K40
Yolo2[15]	57.72	83.28	62.25	42.44	69.75	57.97	47.84	64.53	-	GPU@1080
Faster RCNN[1]	58.45	82.75	63.05	44.25	62.34	66.29	45.16	69.85	0.09s/img	GPU@TitanX
EB[7]	67.96	89.65	73.12	54.64	72.42	73.93	53.40	83.73	0.11s/img	GPU@TitanX
RFCN[6]	69.87	93.32	75.67	54.31	84.08	74.38	56.21	75.09	0.17s/img	GPU@TitanX
NANO*	63.01	80.33	68.04	50.73	73.89	67.00	55.89	62.20	-	-
HPNDFCN*	71.56	**93.51**	78.00	55.62	84.92	78.84	57.22	**79.37**	0.20s/img	GPU@1080Ti
RTN*	74.15	91.52	79.16	61.73	84.14	77.02	65.27	77.20	0.05s/img	GPU@1080
Ours	**76.26**	92.95	**81.43**	**63.73**	**87.15**	**82.46**	**65.62**	77.21	0.20s/img	GPU@1080Ti

[8] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C Chang, H. Qi, J. Lim, M.-H Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *arXiv preprint arXiv:1511.04136*, 2015.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S.Reed, C.-Y Fu, and A.C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[10] Z. Cai, Q. Fan, R.S.Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.

[11] Y.-T Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *arXiv preprint arXiv:1612.03144*, 2016.

[12] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5987–5995.

[13] P.F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[14] X. Song, T. Wu, Y. Jia, and S.-C Zhu, "Discriminatively trained and-or tree models for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3278–3285.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[16] Y.-T Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, 2017.

[17] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan, "Perceptual generative adversarial networks for small object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[18] S. Ren, K. He, R. Girshickand X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1476–1481, 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[21] J. Deng, W. Dong, R. Socher, L.-J Li, K. Li, and F.-F Li, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[22] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[23] Z. Cai, M.J. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," *IEEE International Conference on Computer Cision*, pp. 3361–3369, 2015.