Face Detection with the Faster R-CNN

Huaizu Jiang
University of Massachusetts Amherst
Amherst MA 01003
hzjiang@cs.umass.edu

Erik Learned-Miller
University of Massachusetts Amherst
Amherst MA 01003
elm@cs.umass.edu

***Abstract*— While deep learning based methods for generic object detection have improved rapidly in the last two years, most approaches to face detection are still based on the R-CNN framework [11], leading to limited accuracy and processing speed. In this paper, we investigate applying the Faster R-CNN [26], which has recently demonstrated impressive results on various object detection benchmarks, to face detection. By training a Faster R-CNN model on the large scale WIDER face dataset [34], we report state-of-the-art results on the WIDER test set as well as two other widely used face detection benchmarks, FDDB and the recently released IJB-A.**

I. Introduction

Deep convolutional neural networks (CNNs) have dominated many tasks in computer vision. For object detection, region-based CNN detection methods are now the main paradigm. It is such a rapidly developing area that three generations of region-based CNN detection models, from the R-CNN [11], to the Fast R-CNN [10], and finally the Faster R-CNN [26], have been proposed in the last few years, with increasingly better accuracy and faster processing speed.

Although generic object detection methods have seen large advances in the last two years, the methodology of face detection has lagged behind somewhat. Most of the approaches are still based on the R-CNN framework, leading to limited accuracy and processing speed, reflected in the results on the de facto FDDB [14] benchmark. In this paper, we investigate how to apply state-of-the art object detection methods to face detection and motivate more advanced methods for the future.

Unlike generic object detection, there has been no large-scale face detection dataset that allowed training a very deep CNN until the recent release of the WIDER dataset [34]. The first contribution of this paper is to evaluate the state-of-the-art Faster R-CNN on this large database of faces. In addition, it is natural to ask if we could get an off-the-shelf face detector to perform well when it is trained on one data set and tested on another. Our paper answers this question by training on WIDER and testing on FDDB.

The Faster R-CNN [26], as the latest generation of region-based generic object detection methods, demonstrates impressive results on various object detection benchmarks. It is also the foundational framework for the winning entry of the COCO detection challenge 2015.[1] In this paper, we demonstrate state-of-the-art face detection results using the Faster R-CNN on three popular face detection benchmarks, the widely used Face Detection Dataset and Benchmark (FDDB) [14], the more recent IJB-A benchmark [15], and the WIDER face dataset [34]. We also compare different generations of region-based CNN object detection models, and compare to a variety of other recent high-performing detectors.

Our code and pre-trained face detection models can be found at https://github.com/playerkk/face-py-faster-rcnn.

[1] http://mscoco.org/dataset/#detections-leaderboard

978-1-5090-4023-0/17/$31.00

II. Related Work

In this section, we briefly introduce previous work on face detection. Considering the remarkable performance of deep learning methods for face detection, we simply categorize previous work as non-neural based and neural based methods.

Non-Neural Based Methods. Since the well-known work of Viola and Jones [32], real-time face detection in unconstrained environments has been an active topic of study in computer vision. The success of the Viola-Jones detector [32] stems from two factors: fast evaluation of the Haar features and the cascaded structure which allows early pruning of false positives. In recent work, Chen *et al.* [3] demonstrate better face detection performance with a joint cascade for face detection and alignment, where shape-index features are used.

A lot of other hand-designed features, including SURF [2], LBP [1], and HOG [5], have also been applied to face detection, achieving remarkable progress in the last two decades. One of the significant advances was in using HOG features with the Deformable Parts Model (DPM) [8], in which the face was represented as a single root and a set of deformable (semantic) parts (*e.g.*, eyes, mouth, etc.). A tree-structured deformable model is presented in [37], which jointly addresses face detection and alignment. In [9], it is shown that partial face occlusions can be dealt with using a hierarchical deformable model. Mathias *et al.* [21] demonstrate top face detection performance with a vanilla DPM and rigid templates.

In addition to the feature plus model paradigm, Shen *et al.* [28] utilize image retrieval techniques for face detection and alignment. In [18], an efficient exemplar-based face detection method is proposed, where exemplars are discriminatively trained as weak learners in a boosting framework.

Neural Based Methods. There has been a long history of training neural networks for face detection. In [31], two

CNNs are trained: the first one classifies each pixel according to whether it is part of a face, and the second one determines the exact face position given the output of the first step. Rowley *et al.* [27] present a retinally connected neural network for face detection, which determines whether each sliding window contains a face. In [22], a CNN is trained to perform joint face detection and pose estimation.

Since 2012, deeply trained neural networks, especially CNNs, have revolutionized many computer vision tasks, including face detection, as witnessed by the increasingly higher performance on the Face Detection Database and Benchmark (FDDB) [14]. Ranjan *et al.* [24] introduced a deformable part model based on normalized features extracted from a deep CNN. A CNN cascade is presented in [19] which operates at multiple resolutions and quickly rejects false positives. Following the Region CNN [11], Yang *et al.* [33] proposed a two-stage approach for face detection. The first stage generates a set of face proposals base on facial part responses, which are then fed into a CNN in the second stage for refinement. Farfade *et al.* [7] propose a fully convolutional neural network to detect faces at different resolutions. In a recent paper [20], a convolutional neural network and a 3D face model are integrated in an end-to-end multi-task discriminative learning framework.

Compared with non-neural based methods, which usually rely on hand-crafted features, the Faster R-CNN can automatically learn a feature representation from data. Compared with other neural based methods, the Faster R-CNN allows end-to-end learning of all layers, increasing its robustness.

Using the Faster R-CNN for face detection have been studied in [12], [23]. In this paper, we don't explicity address the occlusion as in [12]. Instead, it turns out the Faster R-CNN model can learn to deal with occlusions purely form data. Compared with [23], we show it is possible to train an off-the-shelf face detector and achieve state-of-the-art performance on several benchmark datasets.

III. Overview of the Faster R-CNN

After the remarkable success of a deep CNN [16] in image classification on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, it was asked whether the same success could be achieved for object detection. The short answer is yes.

A. Evolution of Region-based CNNs for Object Detection

Girshick *et al.* [11] introduced a region-based CNN (R-CNN) for object detection. The pipeline consists of two stages. In the first, a set of category-independent object proposals are generated, using selective search [30]. In the second refinement stage, the image region within each proposal is warped to a fixed size (*e.g.*, 227×227 for the AlexNet [16]) and then mapped to a 4096-dimensional feature vector. This feature vector is then fed into a classifier and also into a regressor that refines the position of the detection.

The significance of the R-CNN is that it brings the high accuracy of CNNs on classification tasks to the problem of object detection. Its success is largely due to transferring the supervised pre-trained image representation for image classification to object detection.

The R-CNN, however, requires a forward pass through the convolutional network for *each* object proposal in order to extract features, leading to a heavy computational burden. To mitigate this problem, two approaches, the SPPnet [13] and the Fast R-CNN [10] have been proposed. Instead of feeding each warped proposal image region to the CNN, the SPPnet and the Fast R-CNN run through the CNN exactly *once* for the entire input image. After projecting the proposals to convolutional feature maps, a fixed length feature vector can be extracted for each proposal in a manner similar to spatial pyramid pooling. The Fast R-CNN is a special case of the SPPnet, which uses a single spatial pyramid pooling layer, *i.e.*, the region of interest (RoI) pooling layer, and thus allows end-to-end fine-tuning of a pre-trained ImageNet model. This is the key to its better performance relative to the original R-CNN.

Both the R-CNN and the Fast R-CNN (and the SPPNet) rely on the input generic object proposals, which usually come from a hand-crafted model such as selective search [30] or EdgeBox [6]. There are two main issues with this approach. The first, as shown in image classification and object detection, is that (deeply) learned representations often generalize better than hand-crafted ones. The second is that the computational burden of proposal generation dominate the processing time of the entire pipeline (*e.g.*, 2.73 seconds for EdgeBox in our experiments). Although there are now deeply trained models for proposal generation, *e.g.*DeepBox [17] (based on the Fast R-CNN framework), its processing time is still not negligible.

To reduce the computational burden of proposal generation, the Faster R-CNN was proposed. It consists of two modules. The first, called the Regional Proposal Network (RPN), is a fully convolutional network for generating object proposals that will be fed into the second module. The second module is the Fast R-CNN detector whose purpose is to refine the proposals. The key idea is to *share* the same convolutional layers for the RPN and Fast R-CNN detector up to their own fully connected layers. Now the image only passes through the CNN once to produce and then refine object proposals. More importantly, thanks to the sharing of convolutional layers, it is possible to use a very deep network (*e.g.*, VGG16 [29]) to generate high-quality object proposals.

The key differences of the R-CNN, the Fast R-CNN, and the Faster R-CNN are summarized in Table I. The running time of different modules are reported on the FDDB dataset [14], where the typical resolution of an image is about 350×450. The code was run on a server equipped with an Intel Xeon CPU E5-2697 of 2.60GHz and an NVIDIA Tesla K40c GPU with 12GB memory. We can clearly see that the entire running time of the Faster R-CNN is significantly lower than for both the R-CNN and the Fast R-CNN.

TABLE I

COMPARISONS OF THE *entire* PIPELINE OF DIFFERENT REGION-BASED OBJECT DETECTION METHODS. (BOTH FACENESS [33] AND DEEPBOX [17] RELY ON THE OUTPUT OF EDGEBOX. THEREFORE THEIR ENTIRE RUNNING TIME SHOULD INCLUDE THE PROCESSING TIME OF EDGEBOX.)

		R-CNN	Fast R-CNN	Faster R-CNN
proposal stage	time	EdgeBox: 2.73s Faceness: 9.91s (+ 2.73s = 12.64s) DeepBox: 0.27s (+ 2.73s = 3.00s)		0.32s
refinement stage	input to CNN	cropped proposal image	input image & proposals	input image
	#forward thru. CNN	#proposals	1	1
	time	14.08s	0.21s	0.06s
total	time	R-CNN + EdgeBox: 14.81s R-CNN + Faceness: 26.72s R-CNN + DeepBox: 17.08s	Fast R-CNN + EdgeBox: 2.94s Fast R-CNN + Faceness: 12.85s Fast R-CNN + DeepBox: 3.21s	0.38s

B. The Faster R-CNN

In this section, we briefly introduce the key aspects of the Faster R-CNN. We refer readers to the original paper [26] for more technical details.

In the RPN, the convolution layers of a pre-trained network are followed by a 3×3 convolutional layer. This corresponds to mapping a large spatial window or *receptive field* (*e.g.*, 228×228 for VGG16) in the input image to a low-dimensional feature vector at a center stride (*e.g.*, 16 for VGG16). Two 1×1 convolutional layers are then added for *classification* and *regression* branches of all spatial windows.

To deal with different scales and aspect ratios of objects, *anchors* are introduced in the RPN. An anchor is at each sliding location of the convolutional maps and thus at the center of each spatial window. Each anchor is associated with a scale and an aspect ratio. Following the default setting of [26], we use 3 scales ($128^2, 256^2$, and 512^2 pixels) and 3 aspect ratios ($1:1, 1:2$, and $2:1$), leading to $k = 9$ anchors at each location. Each proposal is parameterized relative to an anchor. Therefore, for a convolutional feature map of size $W \times H$, we have at most WHk possible proposals. We note that the same features of each sliding location are used to regress $k = 9$ proposals, instead of extracting k sets of features and training a single regressor.t Training of the RPN can be done in an end-to-end manner using stochastic gradient descent (SGD) for both classification and regression branches. For the entire system, we have to take care of both the RPN and Fast R-CNN modules since they share convolutional layers. In this paper, we adopt the approximate joint learning strategy proposed in [25]. The RPN and Fast R-CNN are trained end-to-end as they are independent. Note that the input of the Fast R-CNN is actually dependent on the output of the RPN. For the exact joint training, the SGD solver should also consider the derivatives of the RoI pooling layer in the Fast R-CNN with respect to the coordinates of the proposals predicted by the RPN. However, as pointed out by [25], it is not a trivial optimization problem.

IV. EXPERIMENTS

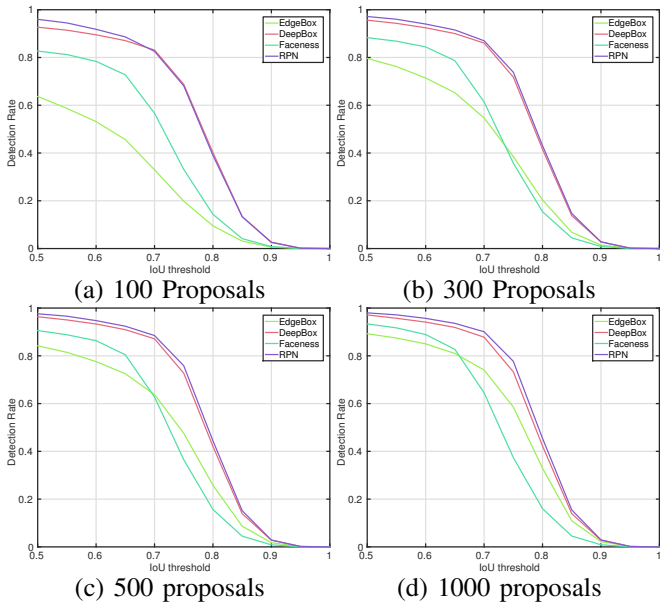
In this section, we report experiments on comparisons of region proposals and also on end-to-end performance of top face detectors.

Fig. 1. Sample images in the WIDER face dataset, where green bounding boxes are ground-truth annotations.

A. Setup

We train a Faster R-CNN face detection model on the recently released WIDER face dataset [34]. There are 12,880 images and 159,424 faces in the training set. In Fig. 1, we demonstrate some randomly sampled images of the WIDER dataset. We can see that there exist great variations in scale, pose, and the number of faces in each image, making this dataset challenging.

We train the face detection model based on a pre-trained ImageNet model, VGG16 [29]. We randomly sample one image per batch for training. In order to fit it in the GPU memory, it is resized based on the ratio $1024/\max(w, h)$, where w and h are the width and height of the image, respectively. We run the stochastic gradient descent (SGD) solver for 50,000 iterations with a base learning rate of 0.001 and run another 30,000 iterations reducing the base learning



(a) 100 Proposals (b) 300 Proposals

(c) 500 proposals (d) 1000 proposals

Fig. 2. Comparisons of face proposals on FDDB using different methods.

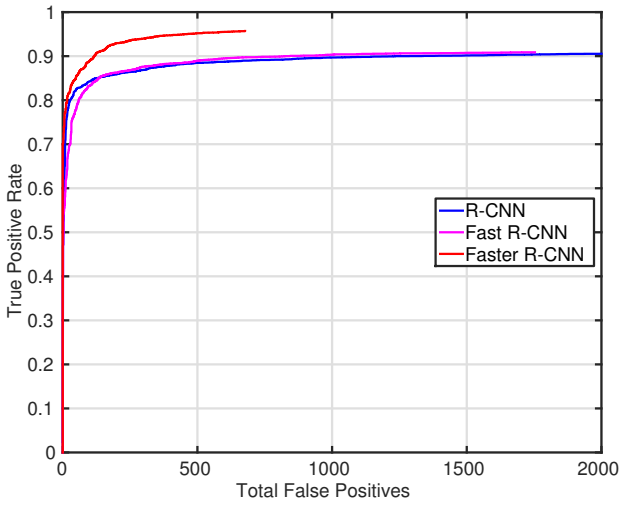


Fig. 3. Comparisons of region-based CNN object detection methods for face detection on FDDB.

rate to 0.0001.[2]

In addition to the extemely challenging WIDER testing set, we also test the trained face detection model on two benchmark datasets, FDDB [14] and IJB-A [15]. There are 10 splits in both FDDB and IJB-A. For testing, we resize the input image based on the ratio $\min(600/\min(w,h), 1024/\max(w,h))$. For the RPN, we use only the top 300 face proposals to balance efficiency and accuracy.

There are two criteria for quantatitive comparison for the FDDB benchmark. For the discrete scores, each detection is considered to be positive if its intersection-over-union (IoU) ratioe with its one-one matched ground-truth annotation is greater than 0.5. By varying the threshold of detection scores, we can generate a set of true positives and false positives and report the ROC curve. For the more restrictive continuous scores, the true positives are weighted by the IoU scores. On IJB-A, we use the discrete score setting and report the true positive rate based on the normalized false positive rate per image instead of the total number of false positives.

[2]We use the author released Python implementation https://github.com/rbgirshick/py-faster-rcnn.

B. Comparison of Face Proposals

We compare the RPN with other approaches including EdgeBox [6], Faceness [33], and DeepBox [17] on FDDB. EdgeBox evaluates the objectness score of each proposal based on the distribution of edge responses within it in a sliding window fashion. Both Faceness and DeepBox re-rank other object proposals, *e.g.*, EdgeBox. In Faceness, five CNNs are trained based on attribute annotations of facial parts including hair, eyes, nose, mouth, and beard. The Faceness score of each proposal is then computed based on the response maps of different networks. DeepBox, which is based on the Fast R-CNN framework, re-ranks each proposal based on the region-pooled features. We re-train a DeepBox model for face proposals on the WIDER training set.

We follow [6] to measure the detection rate of the top N proposals by varying the Intersection-over-Union (IoU) threshold. The larger the threshold is, the fewer the proposals that are considered to be true objects. Quantitative comparisons of proposals are displayed in Fig. 2. As can be seen, the RPN and DeepBox are significantly better than the other two. It is perhaps not surprising that learning-based approaches perform better than the heuristic one, EdgeBox. Although Faceness is also based on deeply trained convolutional networks (fine-tuned from AlexNet), the rule to compute the faceness score of each proposal is hand-crafted in contrast to the end-to-end learning of the RPN and DeepBox. The RPN performs slightly better than DeepBox, perhaps since it uses a deeper CNN. Due to the sharing of convolutional layers between the RPN and the Fast R-CNN detector, the process time of the entire system is lower. Moreover, the RPN does not rely on other object proposal methods, *e.g.*, EdgeBox.

C. Comparison of Region-based CNN Methods

We also compare face detection performance of the R-CNN, the Fast R-CNN, and the Faster R-CNN on FDDB. For both the R-CNN and Fast R-CNN, we use the top 2000 proposals generated by the Faceness method [33]. For the R-CNN, we fine-tune the pre-trained VGG-M model. Different from the original R-CNN implementation [11], we train a CNN with both classification and regression branches end-to-end following [33]. For both the Fast R-CNN and Faster R-CNN, we fine-tune the pre-trained VGG16 model. As can be observed from Fig. 3, the Faster R-CNN significantly outperforms the other two. Since the Faster R-CNN also contains the Fast R-CNN detector module, the performance boost mostly comes from the RPN module, which is based on a deeply trained CNN. Note that the Faster R-CNN also runs much faster than both the R-CNN and Fast R-CNN, as summarized in Table I.

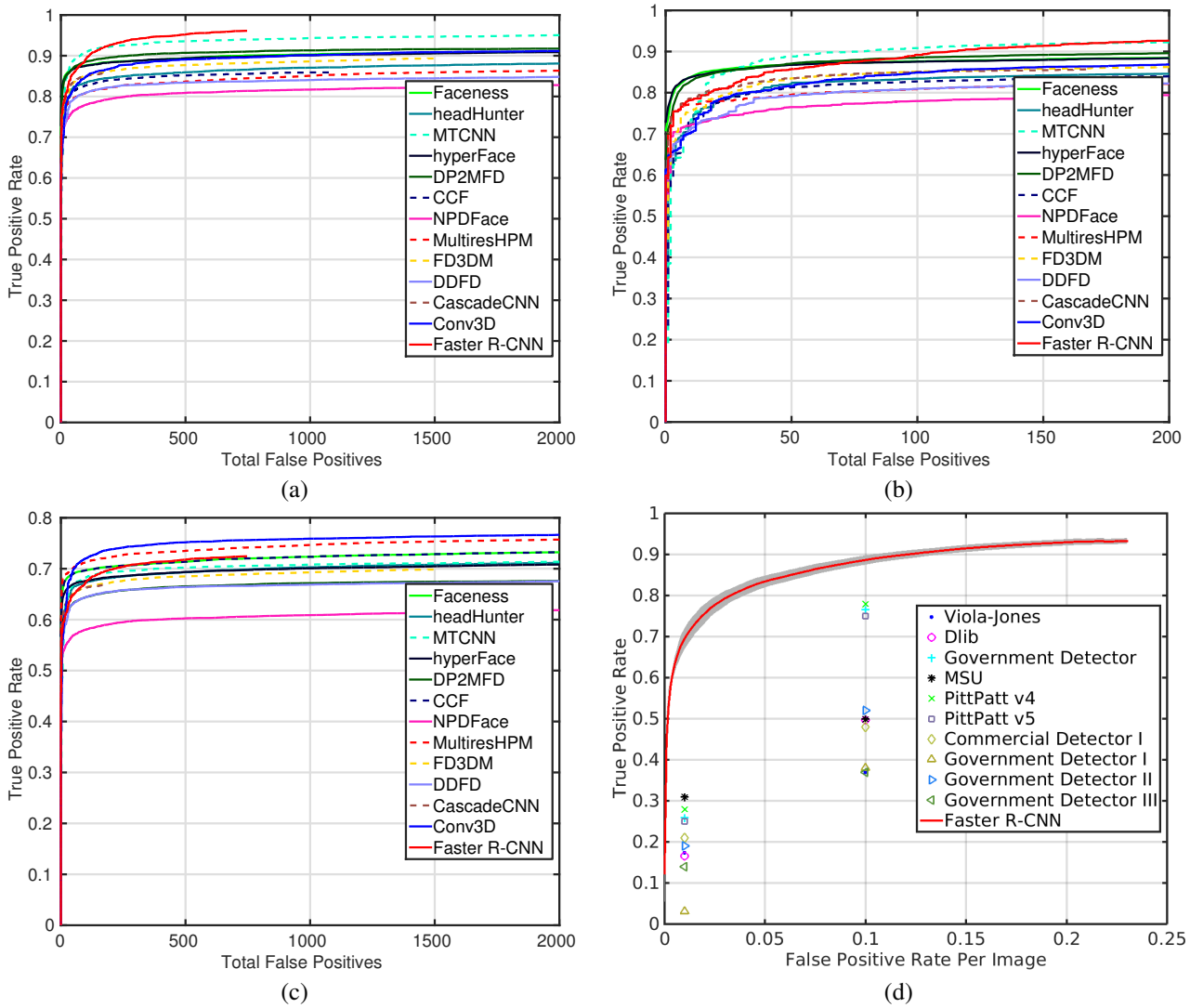


Fig. 4. Comparisons of face detection with state-of-the-art methods. (a) ROC curves on FDDB with discrete scores, (b) ROC curves on FDDB with discrete scores using less false positives, (c) ROC curves on FDDB with continuous scores, and (d) results on IJB-A dataset.

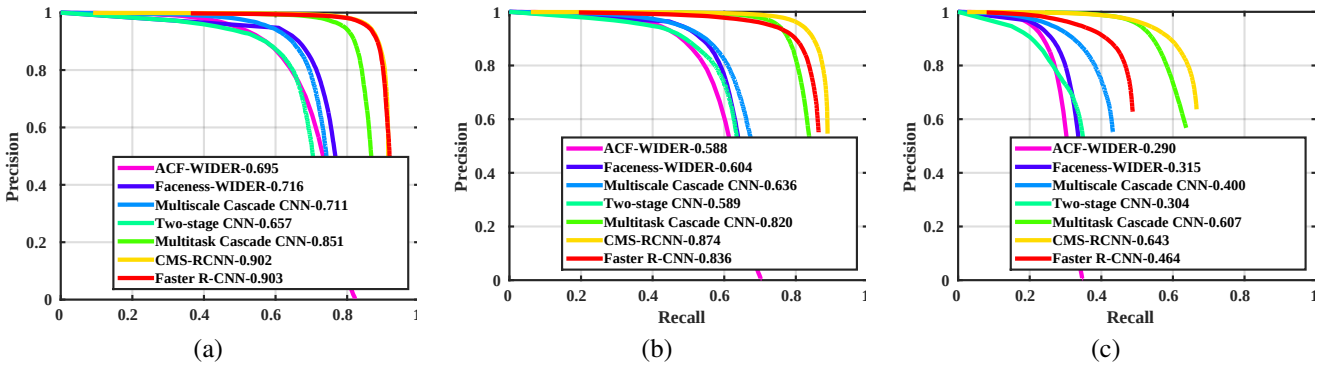
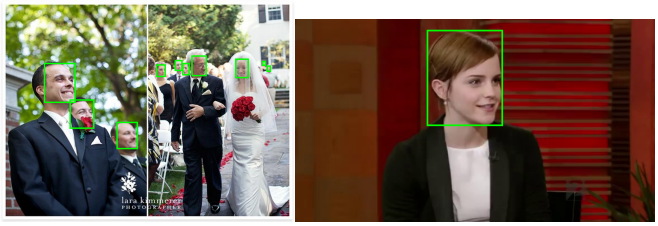


Fig. 5. Comparisons of face detection with state-of-the-art methods on the WIDER testing set. From left to right: (a) on the easy set, (b) on the medium set, and (c) on the hard set.

D. Comparison with State-of-the-art Methods

We compare the Faster R-CNN model trained on WIDER with 11 other top detectors on FDDB, all published since 2015. ROC curves of the different methods, obtained from the FDDB results page, are shown in Fig. 4. For discrete scores on FDDB, the Faster R-CNN performs better than all others when there are more than around 200 false positives for the entire test set, as shown in Fig. 4(a) and (b). With the more restrictive continuous scores, the Faster R-CNN is better than most of other state-of-the-art methods but poorer



(a) (b)

Fig. 6. Illustrations of different annotation styles of (a) WIDER dataset and (b) IJB-A dataset. Notice that the annotation on the left does not include the top of the head and is narrowly cropped around the face. The annotation on the right includes the entire head.

than MultiresHPM [9]. This discrepancy can be attributed to the fact that the detection results of the Faster R-CNN are not always exactly around the annotated face regions, as can be seen in Fig. 7. For 500 false positives, the true positive rates with discrete and continuous scores are 0.952 and 0.718 respectively. One possible reason for the relatively poor performance on the continuous scoring might be the difference of face annotations between WIDER and FDDB.

The testing set of WIDER is divided into easy, medium, and hard according to the detection scores of EdgeBox. From easy to hard, the faces get smaller and more crowded. There are also more extreme poses and illuminations, making it extremely challenging. Comparison of our Faster R-CNN model with other published methods are presented in Fig. 5. Not surprisingly, the performance of our model decreases from the easy set to the hard set. Compared with two concurrent work [35], [36], the performance of our model drops more. One possible reason might be that they consider more cues, such as alignment supervision in [35] and multi-scale convolution features and body cues in [36]. But in the easy set, our model performs the best.

We further demonstrate qualitative face detection results in Fig. 7 and Fig. 9. It can be observed that the Faster R-CNN model can deal with challenging cases with multiple overlapping faces and faces with extreme poses and scales.

In the right bottom of Fig. 9, we also show some failure cases of the Faster R-CNN model on the WIDER testing set. In addition to false positives, there are some false negatives (*i.e.*, faces are not detected) in the extremely crowded images, this suggests trying to integrate more cues, *e.g.*, the context of the scene, to better detect the missing faces.

E. Face Annotation Style Adaptation

In addition to FDDB and WIDER, we also report quantitative comparisons on IJB-A, which is a relatively new face detection benchmark dataset published at CVPR 2015, and thus not too many results have been reported on it yet. As the face annotation styles on IJB-A are quite different from WIDER, when applying the face detection model trained on the WIDER dataset to the IJB-A dataset, the results are very poor, suggesting the necessity of doing annotation style adaptation. As shown in Fig 6, in WIDER, the face annotations are specified tightly around the facial region while annotations in IJB-A include larger areas (*e.g.*, hair).

First, we use *supervised annotation style adaptation*. Specifically, we fine-tune the face detection model on the training images of each split of IJB-A using only 10,000 iterations. In the first 5,000 iterations, the base learning rate is 0.001 and it is reduced to 0.0001 in the last 5,000. Note that there are more then 15,000 training images in each split. We run only 10,000 iterations of fine-tuning to adapt the regression branches of the Faster R-CNN model trained on WIDER to the annotation styles of IJB-A. The comparison is shown in Fig. 4(d). We borrow results of other methods from [4], [15]. As we can see, the Faster R-CNN performs better than all of the others by a large margin. More qualitative face detection results on IJB-A can be found at Fig. 8.

However, in a private superset of IJB-A (denote as sIJB-A), all annotated samples are reserved for testing, which means we can not used any single annotation of the testing set to do supervised annotation style adaptation. Here we propose a simple yet effective *unsupervised annotation style adaptation* method. In specific, we fine-tune the model trained on WIDER on sIJB-A, as what we do in the supervised annotation style adaptation. We then run face detections on training images of WIDER. For each training image, we replace the original WIDER-style annotation with its detection coming from the fine-tuned model if their IoU score is greater than 0.5. We finally re-train the Faster R-CNN model on WIDER training set with annotations borrowed from sIJB-A. With this "cycle training", we can obtain similar performance to the supervised annotation style adaptation.

V. Conclusion

In this report, we have demonstrated state-of-the-art face detection performance on three benchmark datasets using the Faster R-CNN. Experimental results suggest that its effectiveness comes from the region proposal network (RPN) module. Due to the sharing of convolutional layers between the RPN and Fast R-CNN detector module, it is possible to use multiple convolutional layers within an RPN without extra computational burden.

Although the Faster R-CNN is designed for generic object detection, it demonstrates impressive face detection performance when retrained on a suitable face detection training set. It may be possible to further boost its performance by considering the special patterns of human faces.

Acknowledgement

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract number 2014-14071600010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

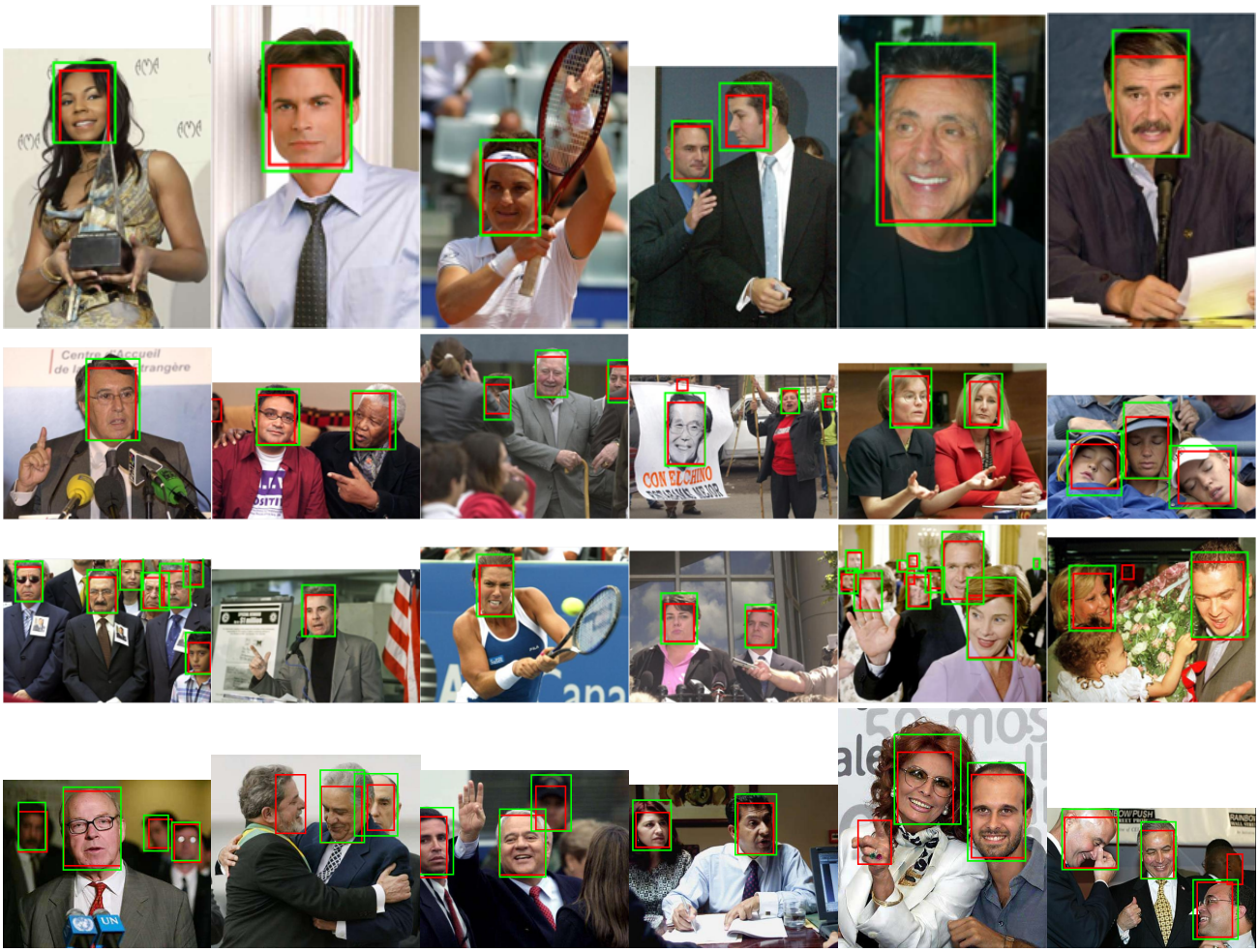


Fig. 7. Sample detection results on the FDDB dataset, where green bounding boxes are ground-truth annotations and red bounding boxes are detection results of the Faster R-CNN.

Fig. 8. Sample detection results on the IJB-A dataset, where green bounding boxes are ground-truth annotations and red bounding boxes are detection results of the Faster R-CNN.

Fig. 9. Sample detection results on the WIDER testing set.

REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):2037–2041, 2006.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[3] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *ECCV*, pages 109–122, 2014.

[4] J. Cheney, B. Klein, A. K. Jain, and B. F. Klare. Unconstrained face detection: State of the art baseline and challenges. In *ICB*, pages 229–236, 2015.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[6] P. Dollár and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(8):1558–1570, 2015.

[7] S. S. Farfade, M. J. Saberian, and L. Li. Multi-view face detection using deep convolutional neural networks. In *ICMR*, pages 643–650, 2015.

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

[9] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *CVPR*, pages 1899–1906, 2014.

[10] R. B. Girshick. Fast R-CNN. In *ICCV*, pages 1440–1448, 2015.

[11] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[12] J. Guo, J. Xu, S. Liu, D. Huang, and Y. Wang. *Occlusion-Robust Face Detection Using Shallow and Deep Proposal Based Faster R-CNN*, pages 3–12. Springer International Publishing, Cham, 2016.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361, 2014.

[14] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[15] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus benchmark A. In *CVPR*, pages 1931–1939, 2015.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[17] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *ICCV*, pages 2479–2487, 2015.

[18] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua. Efficient boosted exemplar-based face detection. In *CVPR*, pages 1843–1850, 2014.

[19] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *CVPR*, pages 5325–5334, 2015.

[20] Y. Li, B. Sun, T. Wu, Y. Wang, and W. Gao. Face detection with end-to-end integration of a convnet and a 3d model. *ECCV*, abs/1606.00850, 2016.

[21] M. Mathias, R. Benenson, M. Pedersoli, and L. J. V. Gool. Face detection without bells and whistles. In *ECCV*, pages 720–735, 2014.

[22] M. Osadchy, Y. LeCun, and M. L. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8:1197–1215, 2007.

[23] H. Qin, J. Yan, X. Li, and X. Hu. Joint training of cascaded cnn for face detection. In *CVPR*, June 2016.

[24] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. In *BTAS*, pages 1–8. IEEE, 2015.

[25] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[26] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[27] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(1):23–38, 1998.

[28] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. In *CVPR*, pages 3460–3467, 2013.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[30] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[31] R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localisation of objects in images. *IEE Proc on Vision, Image, and Signal Processing*, 141(4):245–250, August 1994.

[32] P. A. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[33] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, pages 3676–3684, 2015.

[34] S. Yang, P. Luo, C. C. Loy, and X. Tang. WIDER FACE: A face detection benchmark. In *CVPR*, 2016.

[35] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

[36] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. CMS-RCNN: contextual multi-scale region-based CNN for unconstrained face detection. *CoRR*, abs/1606.05413, 2016.

[37] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012.