Neurocomputing 299 (2018) 42–50

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Face detection using deep learning: An improved faster RCNN approach

Xudong Sun[a], Pengcheng Wu[a], Steven C.H. Hoi[a,b,*]

[a] DeepIR Inc., PR China
[b] School of Information Systems, Singapore Management University, Singapore

ARTICLE INFO

*Article history:*
Received 4 November 2017
Revised 19 February 2018
Accepted 15 March 2018
Available online 21 March 2018

Communicated by Zhu Jianke

*Keywords:*
Face detection
Faster RCNN
Convolutional neural networks (CNN)
Feature concatenation
Hard negative mining
Multi-scale training

ABSTRACT

In this paper, we present a new face detection scheme using deep learning and achieve the state-of-the-art detection performance on the well-known FDDB face detection benchmark evaluation. In particular, we improve the state-of-the-art Faster RCNN framework by combining a number of strategies, including feature concatenation, hard negative mining, multi-scale training, model pre-training, and proper calibration of key parameters. As a consequence, the proposed scheme obtained the state-of-the-art face detection performance and was ranked as one of the best models in terms of ROC curves of the published methods on the FDDB benchmark.[1]



## 1. Introduction

Face detection is a fundamental and important problem in computer vision and pattern recognition, which has been widely studied over the past few decades. Face detection is one of the important key steps towards many subsequent face-related applications, such as face verification [1,2], face recognition [3–5], and face clustering [5], etc. Following the pioneering work of Viola Jones object detection framework [6,7], numerous methods have been proposed for face detection in the past decade. Early research studies in the literature were mainly focused on extracting different types of hand-crafted features with domain experts in computer vision, and training effective classifiers for detection and recognition with traditional machine learning algorithms. Such approaches are limited in that they often require computer vision experts in crafting effective features and each individual component is optimized separately, making the whole detection pipeline often sub-optimal.

In recent years, deep learning, especially the deep convolutional neural networks (CNN), has achieved remarkable successes in various computer vision tasks, ranging from image classification to object detection and semantic segmentation, etc. In contrast to traditional computer vision approaches, deep learning methods avoid the hand-crafted design pipeline and have dominated many well-known benchmark evaluations, such as ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [8]. Along with the popularity of deep learning in computer vision, a surge of research attention has been emerging to explore deep learning for resolving face detection tasks.

In general, face detection can be considered as a special type of object detection task in computer vision. Researchers thus have attempted to tackle face detection by exploring some successful deep learning techniques for generic object detection tasks. One of very important and highly successful framework for generic object detection is the region-based CNN (RCNN) method [9], which is a kind of CNN extension for solving the object detection tasks. A variety of recent advances for face detection often follow this line of research by extending the RCNN and its improved variants.

Following the emerging trend of exploring deep learning for face detection, in this paper, we propose a new face detection method by extending the state-of-the-art Faster R-CNN algorithm [10]. In particular, our scheme improves the existing Faster RCNN scheme by combining several important strategies, including feature concatenation [11], hard negative mining, and multi-scale training, etc. We conducted an extensive set of experiments to evaluate the proposed scheme on the well-known Face Detection

* Corresponding author at: School of Information Systems, Singapore Management University, Singapore.
*E-mail address:* steven@deepir.com (S.C.H. Hoi).

[1] The result of this work ranked #1 on the FDDB leaderboard in Feb 2017. An earlier version of this work was submitted to published in arXiv.org on 28 Jan 2017

https://doi.org/10.1016/j.neucom.2018.03.030
0925-2312/

Dataset and Benchmark (FDDB) [12], and achieved the state-of-the-art performance.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work in face detection literature and recent advances of deep learning approaches. Section 3 presents the proposed deep learning approach for face detection. Section 4 discusses our experiments and empirical results. Section 5 concludes this work.

## 2. Related work

Face detection has been extensively studied in the literature of computer vision. Before 2000, despite many extensive studies, the practical performance of face detection was far from satisfactory until the milestone work proposed by Viola and Jones [6,7]. In particular, the VJ framework [6] was the first one to apply rectangular Haar-like features in a cascaded Adaboost classifier for achieving real-time face detection. However, it has several critical drawbacks. First of all, its feature size was relatively large. Typically, in a $24 \times 24$ detection window, the number of Haar-like features was 160,000 [7]. In addition, it is not able to effectively handle non-frontal faces and faces in the wild.

To address the first problem, much effort has been devoted to coming up with more complicated features like HOG [13], SIFT, SURF [14] and ACF [15]. For example, in [16], a new type of feature called NPD was proposed, which was computed as the ratio of the difference between any two pixel intensity values to the sum of their values. Others aimed to speed up the feature selection in a heuristic way [17,18]. The well known Dlib C++ Library [19] took SVM as the classifier in its face detector. Other approaches, such as random forest, have also been attempted.

Enhancing the robustness of detection was another extensively studied topic. One simple strategy was to combine multiple detectors that had been trained separately for different views or poses [20–22]. Zhu and Ramanan [23] applied multiple deformable part models to capture faces with different views and expressions. Shen et al. [24] proposed a retrieval-based method combined with discriminative learning. Nevertheless, training and testing of such models were usually more time-consuming, and the boost in detection performance was relatively limited. Recently, Chen et al. [25] constructed a model to perform face detection in parallel with face alignment, and achieved high performance in terms of both accuracy and speed.

Recent years have witnessed the advances of face detection using deep learning, which often outperforms traditional computer vision methods significantly. For example, Zhan et al. [26] employed CNN to automatically learn and synthesize feature extractors used for face detection. Li et al. [27] presented a method for detecting faces in the wild, which integrates a ConvNet and a 3D mean face model in an end-to-end multi-task discriminative learning framework. Recently, [28] applied the Faster R-CNN [10], one of state-of-the-art generic object detector, and achieved promising results. In addition, much work has been done to improve the Faster R-CNN architecture. In [29], joint training conducted on CNN cascade, region proposal network (RPN) and Faster R-CNN has realized end-to-end optimization. Wan et al. [30] combined Faster R-CNN face detection algorithm with hard negative mining and ResNet and achieved significant boosts in detection performance on face detection benchmarks like FDDB. In this work, we propose a new scheme for face detection by improving the Faster RCNN framework.

## 3. Our approach

### 3.1. Overview of methodology

Our method follows the similar deep learning framework of Faster RCNN, which has been shown to be a state-of-the-art deep learning scheme for generic object detection [10]. It essentially consists of two parts: (1) a Region Proposal Network (RPN) for generating a list of region proposals which likely contain objects, or called regions of interest (RoIs); and (2) a Fast RCNN network for classifying a region of image into objects (and background) and refining the boundaries of those regions. The two parts share common parameters in the convolution layers used for feature extraction, allowing this architecture to accomplish object detection tasks at a fairly competitive speed. In this work, we propose to extend the Faster RCNN architecture for face detection towards higher recall and accuracy, and train our face detection model by following the proposed procedure as shown in Fig. 1.

First of all, we train the CNN model of Faster RCNN using the WIDER FACE dataset [31]. We further use the same dataset to test the pre-trained model so as to generate hard negatives. These hard negatives are fed into the network as the second step of our training procedure. By training on these hard negative samples, the resulting model is able to generate fewer false positives. This model will be further fine-tuned on the FDDB dataset. During the final fine-tuning process, we apply the multi-scale training process, and adopt a feature concatenation strategy to further boost the performance of our model. For the whole training processes, we follow the similar end-to-end training strategy as Faster RCNN for its simplicity and strong performance. As a final optional step, we convert the resulting detection bounding boxes into ellipses as the regions of human faces are more elliptical than rectangular.

In the following, we discuss five key steps of our solution in detail.

### 3.2. Feature concatenation

For traditional Fast RCNN networks, the RoI pooling is performed on the final feature map layer to generate features of the region, which are then further analyzed by the classification part of the network. This ingenious design makes it possible for the classification network to utilize the features calculated from RPN, saving a lot of unnecessary calculations. However, such an approach is not always optimal and sometimes may omit some important features, as features in deeper convolution layer output have wider receptive fields, resulting in a grosser granularity. In the proposed

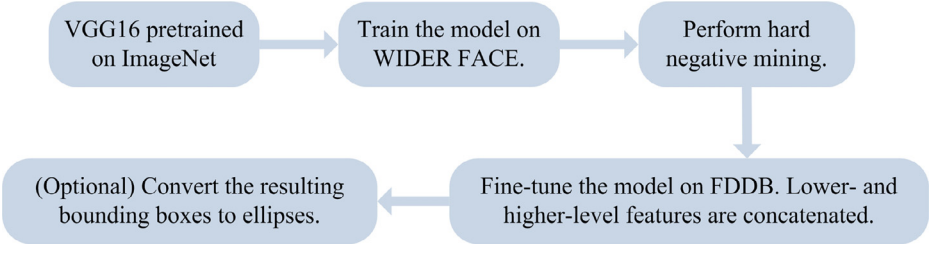


**Fig. 1.** Flowchart of the training procedure of the proposed deep learning scheme.

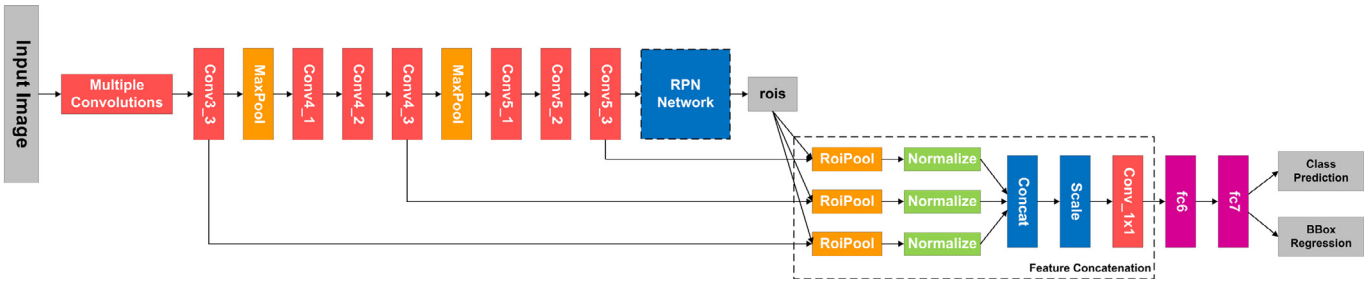


**Fig. 2.** Network architecture of the proposed improved Faster R-CNN scheme.

solution, in order to capture more fine-grained details of the RoIs, we propose to improve the RoI pooling by combining the feature maps of multiple convolution layers, including both lower-level and high-level features. Inspired by the study in [11], we propose to concatenate the pooling result of multiple convolutional feature maps to generate the final pooling features for detection tasks. In other words, we utilize some intermediate results along with the final feature map in RPN, combining them together to yield the final pooling features. Specifically, features from multiple lower-level convolution layers are RoI-pooled and L2-normalized, respectively. Those resulting features are then concatenated and rescaled to match the original scale of the features as if feature-concatenation had not been adopted. A $1 \times 1$ convolution is then applied to match the number of channels of the original network. As to be later illustrated in the experimental section, the invocation of feature concatenation in our network boost the performance by around 1% for the FDDB continuous ROC score, and 0.8% for the discrete one. The detailed architecture of this approach is shown in Fig. 2.

### 3.3. Hard negative mining

Hard negative mining has been shown as an effective strategy for boosting the performance of deep learning, especially for object detection tasks including face detection [30]. The idea behind this method is that, hard negatives are the regions where the network has failed to make correct prediction. Thus, the hard negatives are fed into the network again as a reinforcement for improving our trained model. The resulting training process will then be able to improve our model towards fewer false positives and better classification performance.

In our approach, hard negatives are harvested from the pre-trained model from the first step of our training process. We then consider a region as hard negative if its intersection over union (IoU) over the ground truth region is less than 0.5. During the hard negative training process, we explicitly add those hard negatives into the RoIs for fine-tuning the model, and balance the ratio of foreground and background to be about 1:3, which is the same as the ratio that we use in the first step.

### 3.4. Model pre-training

To adapt Faster RCNN to face detection, we opt to fine-tune a pre-trained model from ImageNet on a face dataset. As a widely acknowledged dataset with unconstrained faces, the FDDB dataset serves this purpose well. However, simply fine-tuning on this dataset may not be a wise option, as this is a relatively small dataset with only 5171 faces in 2845 images. In our approach, before fine-tuning on FDDB, we first pre-train our model on the WIDER FACE dataset, a much larger face dataset with many more difficult examples. To handle those difficult cases, which may disrupt the convergence of the training process, care must be taken to discard some of the training data, where details can be found in the experimental section. In addition, hard negative mining is essential when pre-training on this dataset, so that the number of generated false positives can be minimized. This strategy of pre-training, combined with hard negative mining, gives an increase of around 1.1% on the FDDB continuous ROC score, and 0.6% on the discrete one.

### 3.5. Multi-scale training

The Faster RCNN architecture typically adopt a fixed scale for all the training images. By resizing the images to a random scale, the detector will be able to learn features across a wide range of sizes, thus improving its performance towards scale invariance. In this work, we randomly assign one of three scales for each image before it is fed into the network. The details are given in our experimental section. Our empirical results show that the use of multi-scale training makes our model more robust towards different sizes, and improve the detection performance on benchmark results (around 0.2% boost on FDDB continuous and discrete score).

### 3.6. Number of anchors

In this work, we have tuned several key hyper-parameters in the Faster RCNN architecture, where we have found that, among others, the most crucial one seems to be the number of anchors in the RPN part. Traditional Faster RCNN uses 9 anchors, which sometimes fails to recall small objects. For face detection tasks, however, small faces tend to be fairly common, especially in the case of unconstrained face detection. Therefore, instead of using the default setting, we add a size group of $64 \times 64$ (which is smaller than any of the default size groups), thus increasing the number of anchors to 12. Our experiments show that this modification increases the FDDB continuous ROC by around 1.3%, and discrete ROC by around 2.3%.

## 4. Experiments

### 4.1. Experimental setup

We conduct an empirical study of evaluating the proposed face detection solution on the well-known FDDB benchmark testbed [12], which has a total of 5171 faces in 2845 images, including various detection challenges, such as occlusions, difficult poses, and low resolution and out-of-focus faces.

For implementation, we adopt the Caffe framework [32] to train our deep learning models. VGG16 is selected to be our backbone CNN network, which has been pre-trained on ImageNet. For the first step, WIDER FACE training and validation datasets are selected as our training dataset. We give each ground-truth annotation a difficulty value, according to the standard listed in Table 1. Specifically, all faces are initialized with zero difficulty. If a face region satisfies a certain condition listed in Table 1, we add the corresponding difficulty value. We ignore those annotations with

**Table 1**
Difficulty value assignment strategy.

| Blur | | Expression | Illumination | Occlusion | | Pose |
|---|---|---|---|---|---|---|
| Normal blur | Heavy blur | Extreme expression | Extreme illumination | Partial occlusion | Heavy occlusion | Atypical pose |
| 0.5 | 1 | 1 | 1 | 0.5 | 1 | 1 |

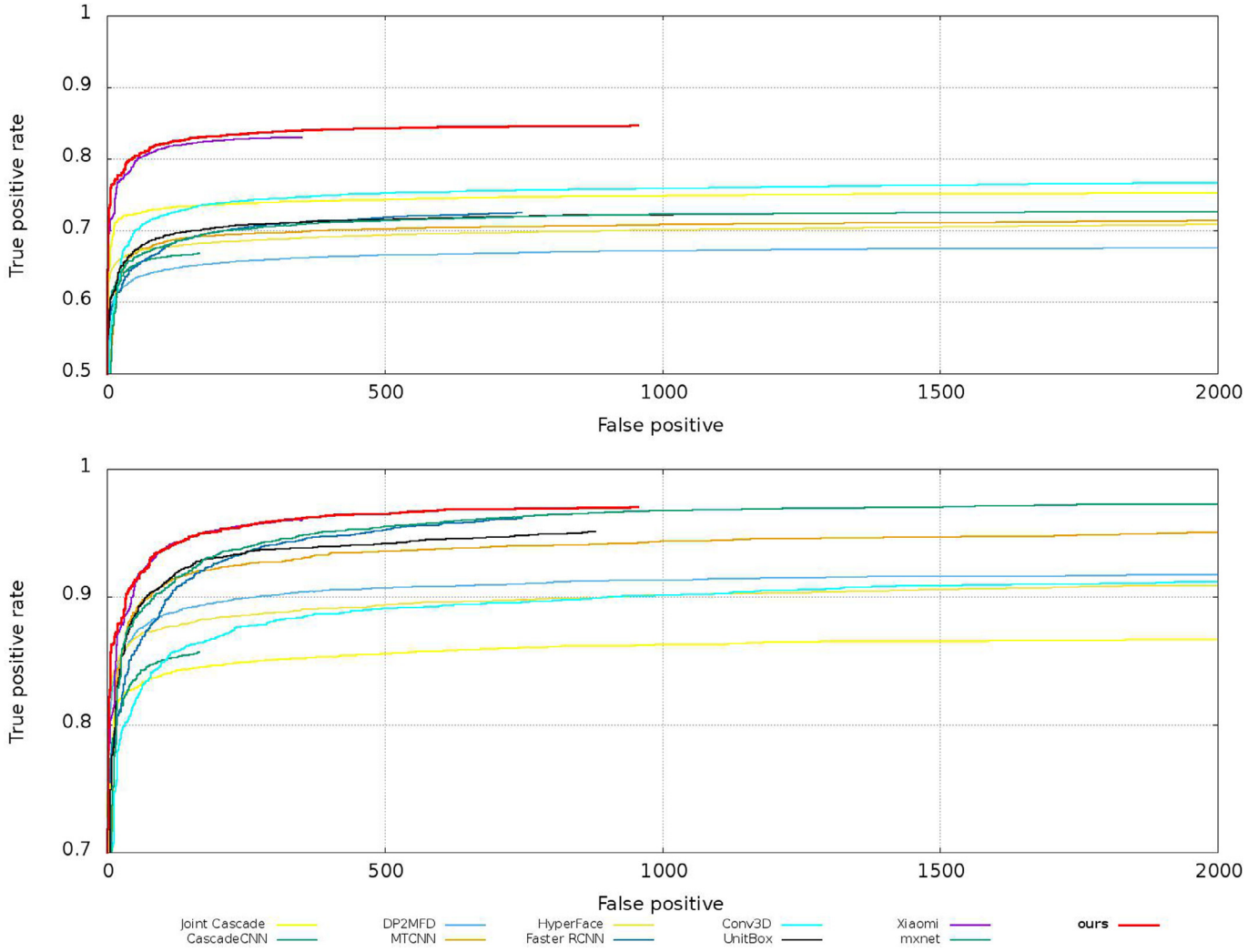


**Fig. 3.** Comparisons of our results with some of recent and most competitive submissions to FDDB (top: continuous ROC result; bottom: discrete ROC result). Our result clearly outperforms those selected in terms of both continuous and discrete ROC. Our result ranked #1 on the leaderboard of FDDB published category in Feb 2017.

difficulty values greater than 2. Further, all the images with more than 1000 annotations are also discarded.

The pre-trained VGG16 model is trained on this aforementioned dataset for 110,000 iterations with the learning rate set to 0.0001. During this training process, images are first re-scaled while always keeping the original aspect ratio. The shorter side is re-scaled to be 600, and the longer side is capped at 1000. Horizontal flipping is adopted as a data augmentation strategy. During the training process, 12 anchors are used for the RPN part, which covers a total size of $64 \times 64$, $128 \times 128$, $256 \times 256$, $512 \times 512$, and three aspect ratios including 1:1, 1:2, and 2:1. After the non-maximum suppression (NMS), 2000 region proposals are kept. For the Fast RCNN classification part, an RoI is treated as foreground if its IoU with any ground truth is greater than 0.5, and background otherwise. To balance the numbers of foregrounds and backgrounds, those RoIs are sampled to maintain a ratio of 1:3 between foreground and background.

For the second step, the aforementioned dataset is fed into the network. Those output regions, whose confidence scores are above 0.8 while IoU values with any ground-truth annotation are less than 0.5, are regarded as the “hard negatives”. The hard negative mining procedure is then taken for 100,000 iterations using a fixed learning rate of 0.0001, where those hard negatives are ensured to be selected along with other sampled RoIs. Finally, the resulting model is further fine-tuned on the FDDB dataset to yield our final detection model.

To examine the detection performance of our face detection model on the FDDB benchmark, we conduct a set of 10-fold cross-validation experiments by following the similar settings in [12]. For each image, in addition to performing the horizontal flipping, we also randomly resize it before feeding it into the network. Specifically, we resize each image such that its shorter side will be one of 480, 600, 750. Similar to the policy taken in the first step, we ensure that the longer side does not exceed 1250.

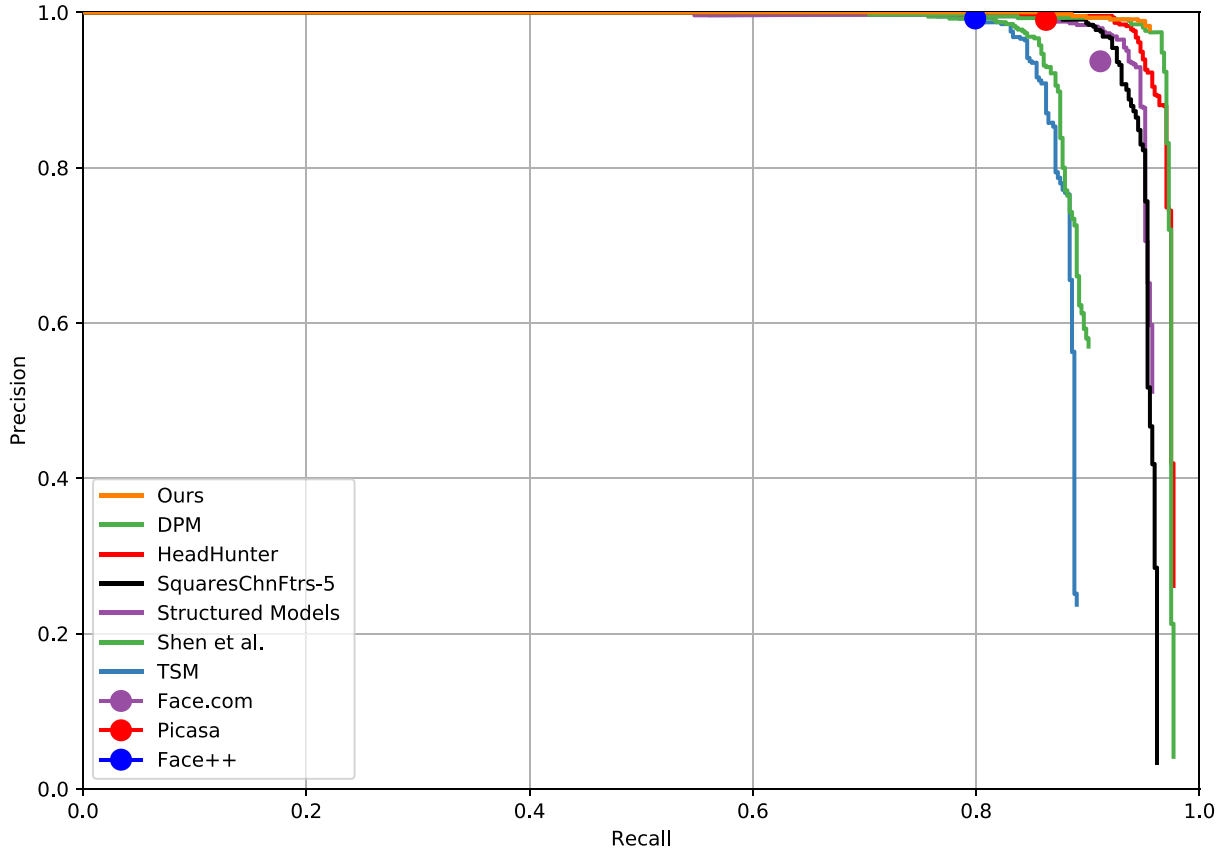


**Fig. 4.** Performance of our model on the AFW test set. Data of other models and evaluation code are acquired from [33].

During the training process, we apply the feature concatenation strategy as introduced in the previous section. Specifically, we concatenate the features pooled from conv3_3, conv4_3, and conv5_3 layers. As illustrated in [11], the scale used after the features being concatenated can be either refined or fixed. Here we use a fixed scale of 4700 for the entire blob, both in the training and test phases. We fine-tune the model for 40,000 iterations using a fixed learning rate of 0.001 to obtain our final models.

For the test phase, a query image is first rescaled by following the same principle as in the first stage. For each image, a total of 100 region proposals are generated by the RPN network during the region proposal generation step. A selected region proposal is regarded as a face if the classification confidence score is greater than 0.8. In our approach, the NMS threshold is set to 0.3. For the analysis purposes, we also output all the region proposals with confidence scores greater than 0.001 in our experiments.

### 4.2. Results

Fig. 3 gives the detailed comparisons of two kinds of ROC scores for some of the most recent / competitive published methods submitted to the FDDB benchmark. The promising results validate the effectiveness of the proposed method for face detection using deep learning techniques.

In addition to the quantitative evaluation results, we also randomly choose some qualitative results of face detection examples for different cases, as shown in Fig. 5 (and more other examples in Fig. 7). Fig. 5a demonstrates that our model is able to detect some difficult cases, such as non-frontal faces, heavily occluded faces, faces with low resolution, and faces with extreme poses and/or illumination. Fig. 5b lists some selected false positives, where it seems that most of the false positives are actually missing annotations. Fig. 5c lists some of the false negatives, which includes some very challenging cases, such as blur faces, heavily occluded faces, and extremely small faces.

The inference speed of this model is around 9 fps on a NVIDIA Geforce GTX 1080 GPU. This is limited by the backbone network of VGG16. Many new networks [34] and optimization techniques [35] exist that could accelerate the model, but the primary focus of this work is to apply training tricks to existing models and improve detection accuracy.

Although our model is fine-tuned on the FDDB dataset, we show in Fig. 4 that it generalizes well on other benchmarks. The FDDB and AFW benchmarks are labeled using completely different guidelines, and yet our model is still able to achieve state-of-the-art performance against the most competitive models.

### 4.3. Ablation experiments

To further gain the deep insights of the improvements obtained by the proposed method, we conduct more additional experiments for ablation studies as listed in Table 2, where we aim to examine the effectiveness and contributions of different strategies used in the proposed method. Fig. 6 shows the detailed experimental results of the ablation studies for examining several different settings.

First of all, by examining the impact of anchor size, we compare the default setting of 9 anchors with our modification of 12. Using this modification would allow our model to detect more small detection boxes (as shown in Experiment ID 1 vs. ID 2).

Second, we examine the impact of pre-training on our model on additional larger-scale face data sets (such as WIDER FACE in our approach), since FDDB is a relatively small dataset. However, the pre-training is not trivial as the WIDER FACE dataset is more

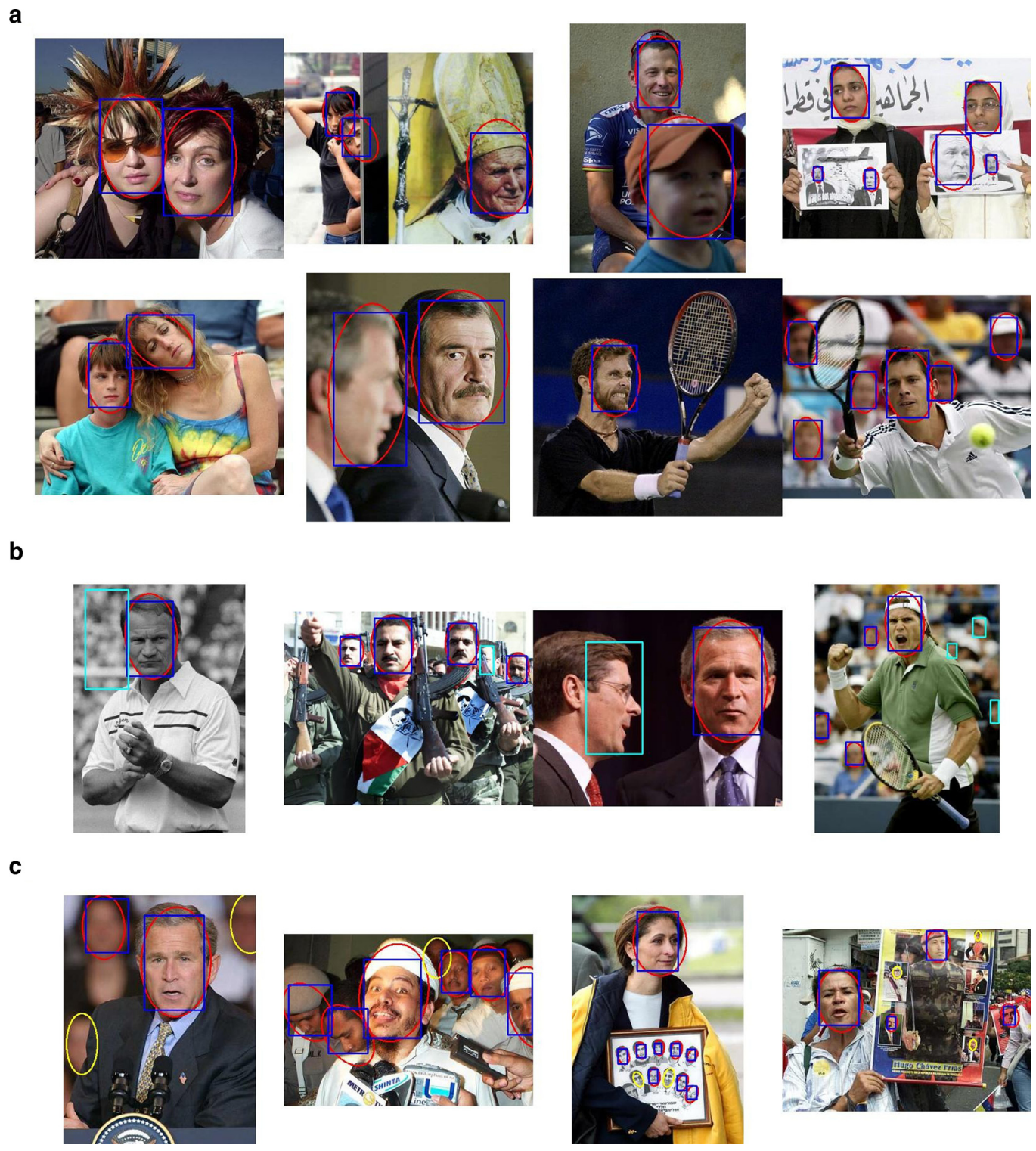


**Fig. 5.** Selected detection results. (a) Correctly detected faces; (b) false positives; (c) false negatives (red: annotation; blue: correctly detected faces; cyan: false positives; yellow: false negatives). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

challenging than FDDB, as it contains many difficult cases. As seen from experiment ID 4, although the detection recall was improved compared with Experiment ID 2, a simple training on WIDER FACE will yield many more false positives. By using the hard negative mining (as shown in Experiment ID 5), the number of false positives was reduced significantly.

Third, we examine the impact of feature concatenation strategy. As shown in our ablation study experiments (ID 2 vs. ID 3, and ID 5 vs. ID 6), feature concatenation turned out to be an effective strategy. By combining features from multiple layers, the model was able to learn features of multiple sizes, and was therefore better at classification.

**Table 2**
Additional experiments for ablation studies of the proposed solution.

| ID | # Anchors | Train with WIDER FACE | Hard negative mining | Feature concatenation | Multi-scale training |
|---|---|---|---|---|---|
| 1 | 9 | No | No | No | No |
| 2 | 12 | No | No | No | No |
| 3 | 12 | No | No | Yes | No |
| 4 | 12 | Yes | No | No | No |
| 5 | 12 | Yes | Yes | No | No |
| 6 | 12 | Yes | Yes | Yes | No |
| 7 | 12 | Yes | Yes | Yes | Yes |

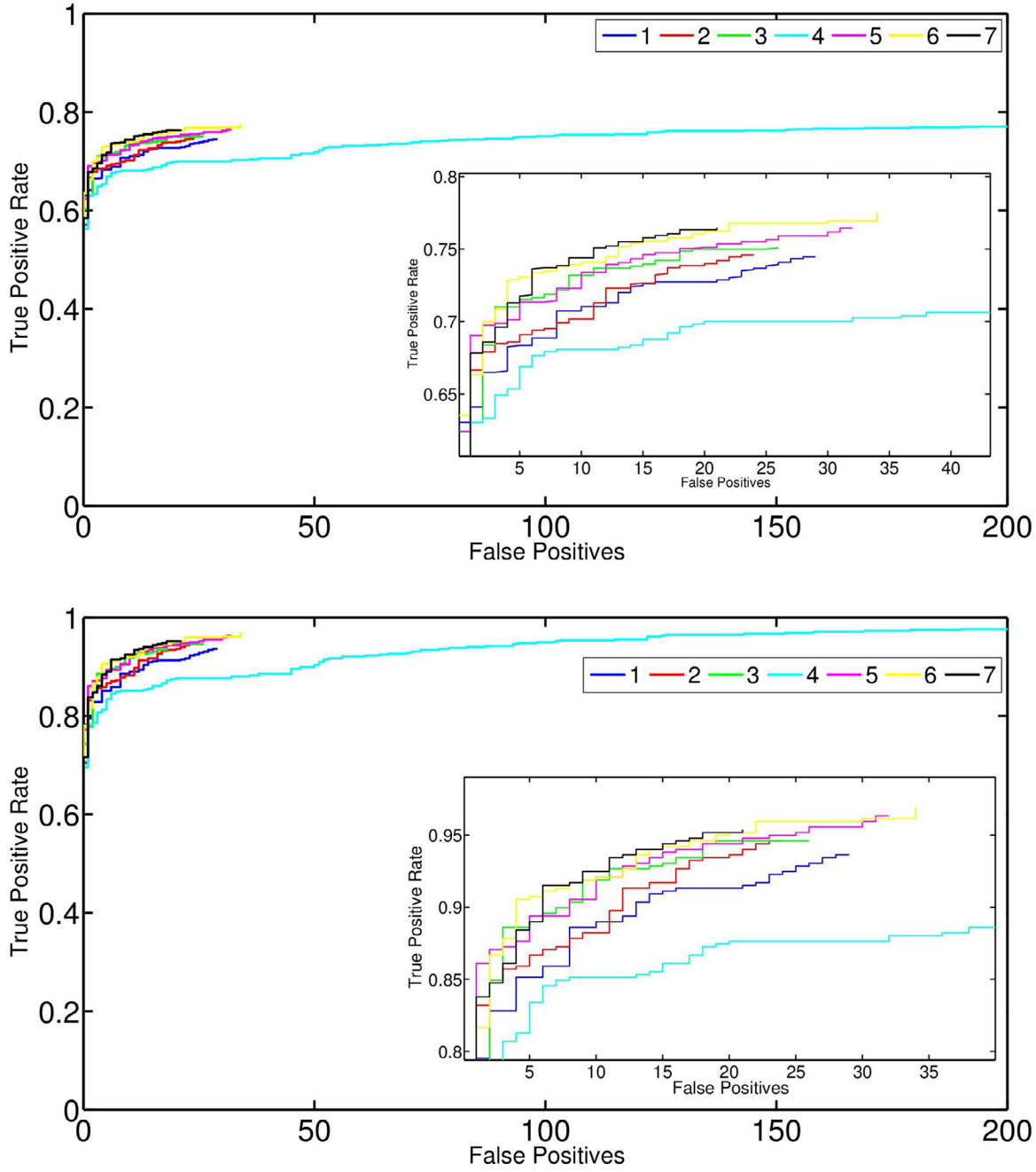


**Fig. 6.** Comparisons of Continuous ROC curves (top) and discrete ROC curves (bottom) for different experimental settings for our ablation studies. These experimental results shown here are only for the fold 7 of our cross-validation experiments; the other experimental folds are similar. The figures on the bottom right are magnified views of the selected regions. The detection bounding boxes are not converted to ellipses. (Best viewed in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**Fig. 7.** More qualitative examples of our face detection results on FDDB (red: annotation; blue: detection results). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fourth, by further examining the impact of multi-scale training, we also observe a positive improvement from our ablation experiments (ID 6 vs. ID 7). Specifically, by adopting the random scaling for data augmentation, the detection performance was further increased.

Finally, combining all the above strategies yielded the best detection performance (as shown in experiment ID 7).

## 5. Conclusions

In this work, we proposed a new method for face detection using deep learning techniques. Specifically, we extended the state-of-the-art Faster RCNN framework for generic object detection, and proposed several effective strategies for improving the Faster RCNN algorithm for resolving face detection tasks, including feature concatenation, multi-scale training, hard negative mining, and proper configuration of anchor sizes for RPN, etc. We conducted an extensive set of experiments on the well-known FDDB testbed for face detection benchmark, and achieved the state-of-the-art results, making it among the best models of all the published methods. Future work will further address the efficiency and scalability of the proposed method for real-time face detection.

## Acknowledgments

This work was fully completed at DeepIR Inc. The authors would like to thank the help and contributions of other colleagues with DeepIR Inc.

## References

[1] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898.
[2] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[3] S. Kumar, S. Prasad, V.B. Semwal, R.C. Tripathi, Real Time face recognition using AdaBoost improved fast PCA algorithm. Int. J. Comput. Sci. Sec., 1(3).
[4] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: Proceedings of the British Machine Vision Conference, 1, 2015, p. 6.
[5] F. Schroff, D. Kalenichenko, J. Philbin, FaceNet: a unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[6] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 2001IEEEI–511.
[7] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[8] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[9] R. Girshick, J. Donahue, T. Darrell, 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.
[10] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[11] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2874–2883.
[12] V. Jain, E.G. Learned-Miller, Fddb: a benchmark for face detection in unconstrained settings, UMass Amherst Technical Report.
[13] Q. Zhu, M.C. Yeh, K.T. Cheng, S. Avidan, Fast human detection using a cascade of histograms of oriented gradients, 2006. in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2, IEEE, 1491–1498.
[14] J. Li, Y. Zhang, Learning surf cascade for fast and accurate object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3468–3475.
[15] B. Yang, J. Yan, Z. Lei, S.Z. Li, Aggregate channel features for multi-view face detection, 2014. in: Proceedings of the International Joint Conference on Biometrics (IJCB), IEEE, 1–8.
[16] S. Liao, A.K. Jain, S.Z. Li, A fast and accurate unconstrained face detector, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 211–223.
[17] M.T. Pham, T.J. Cham, Fast training and selection of Haar features using statistics in boosting-based face detection, IEEE, 2007. in: Proceedings of the Eleventh International Conference on Computer Vision, 1–7.
[18] S.C. Brubaker, J. Wu, J. Sun, M.D. Mullin, J.M. Rehg, On the design of cascades of boosted ensembles for face detection, Int. J. Comput. Vis. 77 (1–3) (2008) 65–86.
[19] D.E. King, Dlib-ml: a machine learning toolkit, J. Mach. Learn. Res. 10 (2009) 1755–1758.
[20] M. Jones, P. Viola, Fast multi-view face detection, Mitsubishi Electr. Res. Lab. 3 (2003) 14. TR-20003-96
[21] S.Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, Proceedings of the European Conference on Computer Vision, Springer, 2002, pp. 67–81.
[22] C. Huang, H. Ai, Y. Li, S. Lao, High-performance rotation invariant multiview face detection, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 671–686.
[23] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 2879–2886.
[24] X. Shen, Z. Lin, J. Brandt, Y. Wu, Detecting and aligning faces by image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3460–3467.
[25] D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun, Joint cascade face detection and alignment, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 109–122.
[26] S. Zhan, Q.Q. Tao, X.H. Li, Face detection using representation learning, Neurocomputing 187 (C) (2016) 19–26. ISSN 0925–2312
[27] Y. Li, B. Sun, T. Wu, Y. Wang, Face detection with end-to-end integration of a convnet and a 3d model, in: European Conference on Computer Vision, Springer, Cham, 2016, pp. 420–436.
[28] H. Jiang, E. Learned-Miller, Face detection with the faster R-CNN, in: Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on IEEE, 2017, pp. 650–657.
[29] H. Qin, J. Yan, X. Li, X. Hu, Joint training of cascaded CNN for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3456–3465.
[30] S. Wan, Z. Chen, T. Zhang, B. Zhang, K.k. Wong, Bootstrapping face detection with hard negative examples, arXiv:1608.02236.
[31] S. Yang, P. Luo, C.C. Loy, X. Tang, Wider face: a face detection benchmark, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5525–5533.
[32] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675–678.
[33] M. Mathias, R. Benenson, M. Pedersoli, L.V. Gool, Face detection without bells and whistles, Springer, 2014. in: Proceedings of the European Conference on Computer Vision, 720–735.
[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[35] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: an extremely efficient convolutional neural network for mobile devices, arXiv:1707.01083.

**Xudong Sun** was a full-time Research staff at DeepIR Inc. from August 2016 to August 2017. He is currently studying the Master Degree of Computer Science at the University of California San Diego. He obtained his Bachelor degree from Peking University.

**Pengcheng Wu** is a Co-founder of DeepIR Inc. He received his Ph.D. degree from the School of Computer Engineering at the Nanyang Technological University, Singapore, and his bachelor degree from Xiamen University, PR China. He was a research fellow in the School of Information Systems, Singapore Management University, and Nanyang Technology University. His research interests include multimedia information retrieval, machine learning and data mining.

**Steven C.H. Hoi** is a Co-founder of DeepIR Inc. He is an Associate Professor of the School of Information Systems, Singapore Management University, Singapore. Prior to joining SMU, he was Associate Professor with Nanyang Technological University, Singapore. He received his Bachelor degree from Tsinghua University, PR China, in 2002, and his Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, in 2006. His research interests are machine learning and data mining and their applications to multimedia information retrieval (image and video retrieval), social media and web mining, and computational finance, etc, and he has published over 150 refereed papers in top conferences and journals in these related areas. He has served as Editor-in-Chief for Neurocomputing Journal, general co-chair for ACM SIGMM Workshops on Social Media (WSM09, WSM10, WSM11), program co-chair for the fourth Asian Conference on Machine Learning (ACML12), book editor for Social Media Modeling and Computing, guest editor for ACM Transactions on Intelligent Systems and Technology (ACM TIST), technical PC member for many international conferences, and external reviewer for many top journals and worldwide funding agencies, including NSF in US and RGC in Hong Kong.