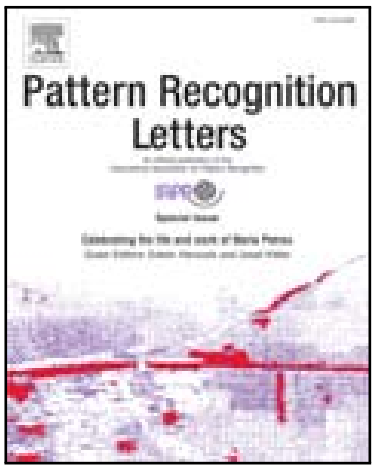


Fast Cascade Face Detection with Pyramid Network

Dan Zeng, Fan Zhao, Shiming Ge, Wei Shen

PII: S0167-8655(18)30212-5
DOI: 10.1016/j.patrec.2018.05.024
Reference: PATREC 7190

To appear in: *Pattern Recognition Letters*

Received date: 14 March 2017
Revised date: 6 March 2018
Accepted date: 30 May 2018

Please cite this article as: Dan Zeng, Fan Zhao, Shiming Ge, Wei Shen, Fast Cascade Face Detection with Pyramid Network, *Pattern Recognition Letters* (2018), doi: 10.1016/j.patrec.2018.05.024

Highlights

- Pyramid network generates multi-scale proposals fast with no more than twice image resizing operations.
- Multi-task learning improves the accuracy of face location.
- Online and offline hard samples mining strengthen the power of networks.
- Increasing convolutional stride saves much computation time in convolutional layers.
- Knowledge distilling and multi-layer merging both improve the performance of small models.

Pattern Recognition Letters
journal homepage: www.elsevier.com

Fast Cascade Face Detection with Pyramid Network

Dan Zeng[a], Fan Zhao[a], Shiming Ge[b,**], Wei Shen[a]

[a]*Key laboratory of Specialty Fiber Optics and Optical Access Networks, Shanghai Institute of Advanced Communication and Data Science, School of Communication and Information Engineering, Shanghai University, 200040, Shanghai*
[b]*Institute of Information Engineering, Chinese Academy of Sciences, 100195, Beijing*

ABSTRACT

Numerous recent face detectors based on convolutional neural networks (CNNs) have significantly improved the detection performance. However, CNNs usually have a huge number of parameters which lead to very low detection speeds. To address this issue, this paper proposes a fast CNNs cascade face detector with multi-task learning and network acceleration techniques. In particular, the first stage of the detector is an elaborately designed fully convolutional network with a novel pyramid architecture, which can generate multi-scale face proposals efficiently with no more than twice image resizing operations. Several network compression and acceleration techniques including multi-layer merging and knowledge distilling are adopted to further improve inference speed. In addition, online and offline hard sample mining are jointly utilised to further strengthen the power of networks. The experimental results on challenging FDDB show that the proposed face detector is comparable in performance with the-state-of-arts while the speed reaches astonishing 165 fps on Titan GPU.

1. Introduction

Face detection is the foundation of many face-related computer vision tasks, such as face tracking [1], facial landmarks detection [2] and face recognition [3]. An excellent face detection method should not only be robust for variations in illumination, facial expression and occlusion, etc., but also be fast. In fact, speed is the biggest bottleneck which hinders the practical deployment of face detectors. Since the tremendous success of the convolutional neural networks (CNNs), many CNNs-based face detectors [4, 5, 6, 7, 8, 9] have significantly improved the detection performance in the wild. However, CNNs generally have a huge number of parameters which cause most of these detectors are very far from realtime.

To address the above issue, this paper proposes a three-stage CNNs cascaded face detector as shown in Fig.1. Cascaded detection is a relatively fast detection framework which can quickly reject most of the background regions in the early stages. However, to detect faces at multi scales, previous cascaded methods [10, 11, 12] must resize the input image to different scales to build an image pyramid, which is time consuming. To accelerate this process, our first stage network is carefully designed with a novel pyramid architecture as shown in Fig.2. It is a fully convolutional network with six branches and can generate multi-scale proposals fast with no more than twice image resizing operations. Concretely, the first network branch can determine whether a 12×12 detection window contains a face. The second branch can process 24×24 windows and so on. This special pyramid network mainly has following three advantages.

(1) Generating multi-scale face proposals only needs to resize image once or twice. The original image and the scaled image with a factor $\sqrt{2}/2$ are fed to the multi-branch network in turn. Then the whole image pyramid space can be completely traversed by sliding detection windows with the size of $12\sqrt{2}^i \times 12\sqrt{2}^i, i \in [0, 1, ..., 11]$.

(2) The network parameters are shared to avoid repeating computation. As shown in Fig.2, the input of high-level branch is the intermediate output of low-level branch, which means high-level branches can be more lightweight with the same classification performance.

(3) Bigger detection windows can achieve higher classification accuracy than smaller windows. It's mainly because a higher-level branch equivalently has a deeper network architecture. In previous methods, an image should be

**Corresponding author:
e-mail: geshiming@iie.ac.cn (Shiming Ge)

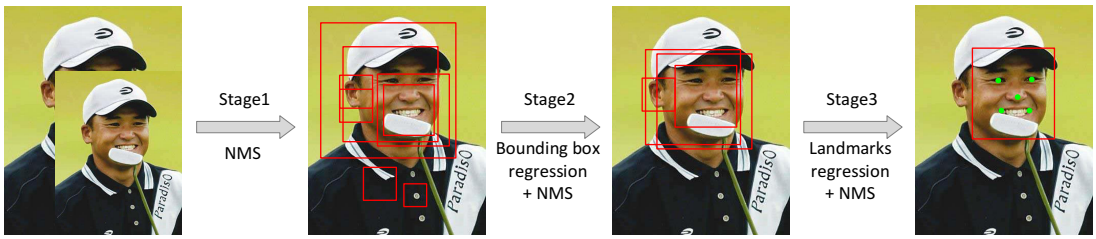


Fig. 1. Pipeline of our three-stage CNNs cascaded face detector. The original image and the scaled image with a factor $\sqrt{2}/2$ are fed into the stage1 pyramid network to generate multi-scale face proposals. After that, stage2 and stage3 are used to refine these proposals in turn.

zoomed out and then fed to the same network to detect bigger faces. It indicates the network will hold the same classification performance for both small and big windows.

After generating face proposals from the pyramid network, Stage2 and Stage3 are used to refine these proposals in turn. These two stages are both dual-task networks, which combine binary classification with bounding box regression and facial landmarks regression respectively to make face location more accurate.

On the other hand, all our three networks are designed and trained with full consideration of computational effectiveness and accuracy performance. The implementation strategies consist of two parts. One is how to accelerate a model and the other is how to promote a small model. For the first aspect, all pooling layers are replaced with increasing convolutional strides to realize feature down-sampling and all batch normalization (BN) [13] layers are merged with their neighbor convolutional layers after training. For the second aspect, there are four techniques adopted: 1)extensive data augmentations including rotating, translating, scaling and Gaussian blurring; 2) online and offline hard samples mining; 3) the knowledge distilling [14] which can produce a better small model from a large one; 4) a novel multi-layer merging method which contributes to training a small model.

The major contributions of this work are the following: (1) We propose a novel pyramid network which can generate multi-scale face proposals efficiently with no more than twice image resizing operations. (2) We combine online and offline hard samples mining to enhance the power of networks. (3) We introduce the knowledge distilling and multi-layer merging to improve the performance of small models. (4) We evaluate our fast face detector on the public benchmark FDDB [15] and achieve comparable results with the state-of-the-arts while the speed reaches astonishing 165 fps on Titan GPU.

The rest of this paper is organized as follows. Section 2 reviews related work, especially the CNNs-based. Section 3 details the proposed fast face detection framework. Section 4 shows the experimental results on public dataset. Finally, Section 5 provides a brief summary of this paper.

2. Related Work

For the past two decades, face detection has always been an important issue in the field of computer vision. All face detection methods can be roughly divided into two categories: traditional machine learning based and CNNs based.

As the most typical traditional method, Viola and Jones [16] built a cascaded detector with Haar features, which could detect front faces at real-time speed. Following this work, many researchers improved the Viola and Jones detector by using advanced features like LBP [17], HOG [18] and SURF [19], or adopting a dividing strategy to assemble several models for multi-view face detection [20, 21, 22]. Another typical traditional method is the deformable part model (DPM) [23], which defined a face as a collection of facial parts. DPM-based methods [24, 25] are robust to partially occluded faces but computationally complex.

Since the enormous success on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, CNNs have been applied in a variety of research fields [26, 27]. Especially, on the challenging Face Detection Data Set and Benchmark (FDDB) [15], public face detection methods with top performance are almost entirely CNNs-based. Benefited from deep features with stronger representation power, DDFD [4] can detect multi-view faces using a single deep model. To deal with severe occlusion and complicated pose variation, some methods combine CNNs with DPM. DP2MFD [28] trained a DPM with pyramid deep features and introduced a normalization layer to handle faces with different size well. Faceness [6] had similarities with the DPM. It used five CNNs models to obtain facial parts responses and then scored a candidate face with the responses spatial structure and arrangement. Besides, to locate faces of varied scales and shapes more accurately, UnitBox [9] introduced an Intersection over Union (IoU) loss to predict four offsets of bounding box jointly.

Recently, multi-task deep learning becomes increasingly popular, which uses a variety of ancillary tasks to improve the performance of the main task. As the most common auxiliary task for face detection, bounding box regression [7, 9, 10, 11, 29] can improve the location accuracy. In addition, MTCNN [10] proposed a cascaded face detector by integrating face classification, bounding box regression and facial landmarks detection. Hyperface [7] further fused face detection, landmarks localization, pose estimation and gender recognition into one model, which generated surprising performance on all four tasks.

However, CNNs models generally have heavy computation

complexity which severely limit their applications especially on mobile platforms. To address this issue, Jiang H et.al [29] trained a Faster R-CNN [30] model for fast face detection. Faster R-CNN is a popular end-to-end object detection framework. It mainly consists of two modules: one is a fully convolutional network for generating object proposals, the other then refines the proposals. DeepIR [31] improved the R-CNN framework by combining a number of tricks including feature concatenation, multi-scale training, model pretraining and so on. R-CNN accelerates the detection speed by avoiding sliding window on an image pyramid. However, it still cannot realize real-time speed because this end-to-end detector needs deeper or wider network to hold performance. Different from R-CNN-based face detectors, cascaded detectors have more advantages in speed. CascadeCNN [11] proposed a CNNs cascade face detector whose speed reached 100 FPS on Titan GPU. Kalinovsky & Spitsyn [12] dramatically improved the speed by implementing SSE/AVX/AVX2 instruction sets on smaller models but the performance decrease significantly. MTCNN [10] also trained a three-stage cascaded model using multi-task learning, which substantially increased the performance while maintaining the same speed as CascadeCNN [11].

On the other hand, mining hard samples is another effective way to strengthen the power of a detector, which mainly includes online and offline hard mining. Wan et.al [32] adopted a boosting strategy to mine hard negative samples and retrain the model several times. Most cascaded methods [10, 11, 12] applied offline hard mining by training present stage with hard samples collected from previous stages. MTCNN [10] proposed a new online hard sample mining strategy which ignores 30% samples with minimum losses during the backward propagation.

3. Our Method

Our detector aims to achieve both real-time speed and excellent performance. Next, we will give the technical details to realize these two targets. Section 3.1 introduces the pyramid network for generating face proposals. Section 3.2 and Section 3.3 detail the techniques used to accelerate and improve the networks respectively.

3.1. Pyramid network for face proposals

The pipeline of our fast face detector is shown in Fig.1. The original image and the scaled image with a factor $\sqrt{2}/2$ are fed into the stage1 pyramid network to generate multi-scale face proposals. After that, Stage2 and Stage3 are used to refine these proposals in turn. In particular, the bounding box regression in Stage2 and the facial landmarks regression in Stage3 make face positions more accurate. Non-maximum suppression (NMS) is used to merge highly overlapped face candidates.

Among multitudinous CNNs-based face detection methods, cascade-based methods [10, 11, 12] are basically the fastest. Since they can quickly reject most of the background regions in the early stages. However, to detect faces with different scales, these methods must resize an image to different scales and slide windows on these scaled images sequentially. Image resizing is time-consuming. Meanwhile, the bigger detection windows cannot achieve higher classification accuracies than smaller windows. In fact, finding bigger faces should be easier. To solve these issues, our first-stage network is carefully designed with a novel pyramid architecture as shown in Fig.2.

Table 1. Comparison of model size, speed and validation accuracy of our six branches and PNet [10]

model	size	1000x Forward Propagation	Validation Accuracy
PNet	27.5Kb	0.044S	0.950
Branch1	17.7Kb	0.019S	0.952
Branch2	9.7Kb	0.017S	0.957
Branch3	8.8Kb	0.015S	0.962
Branch4	8.2Kb	0.014S	0.965
Branch5	8.2Kb	0.014S	0.966
Branch6	8.2Kb	0.014S	0.960

The first stage is a fully convolutional network with 6 branches. The first branch is designed for 12×12 sized detection windows, which has 5 convolutional layers and 4 ReLU activation layers in all. To improve the forward propagation speed, the commonly used pooling layers are abandoned and the stride of the first convolutional layer in each branch is set as 2. The second branch is designed for 24×24 sized windows, whose input comes from the output of the third convolutional layer in branch1. Since this input is advanced feature rather than raw image pixel, a smaller branch model can be easily trained with higher classification accuracy. The remaining branches can process windows with size of 48,96,192 and 384 separately. We compare the model size, inference speed and validation accuracy with the first stage model in MTCNN [10] which is a state-of-the-art CNNs cascaded face detection method. The experimental results are shown in Table 1. And there are a few points to be noted here.

i) The ratio of nonface to face is 3:1 in validation set.

ii) All validation images are larger than 384×384. If the image is too small, the validation performance of high-level branches will decline because of the blurring problem from enlargement.

iii) The forward propagation time is calculated on the Intel Core(TM) i5-6500 CPU.

It is clear that our multi-branch model performs better and faster than PNet in MTCNN [10]. Especially, the validation accuracies generally keep growing with increasing branch levels. It means bigger windows can get higher classification accuracy while previous methods [10, 11, 12] just keep the same performance for both small and big windows. Moreover, the validation accuracy tends to stop growing and even decrease in branch6. It is mainly because training high-level branches lacks big faces as positive samples.

When running detection, the original image is fed to the above pyramid network to detect faces with size $12 \times 2^n, n \in [0, 1 \ldots 5]$. This image is then resized with a scale factor $\sqrt{2}/2$ to further detect faces with size $12\sqrt{2} \times 2^n, n \in [0, 1 \ldots 5]$. Above all, we only need resize image once (if minimum face

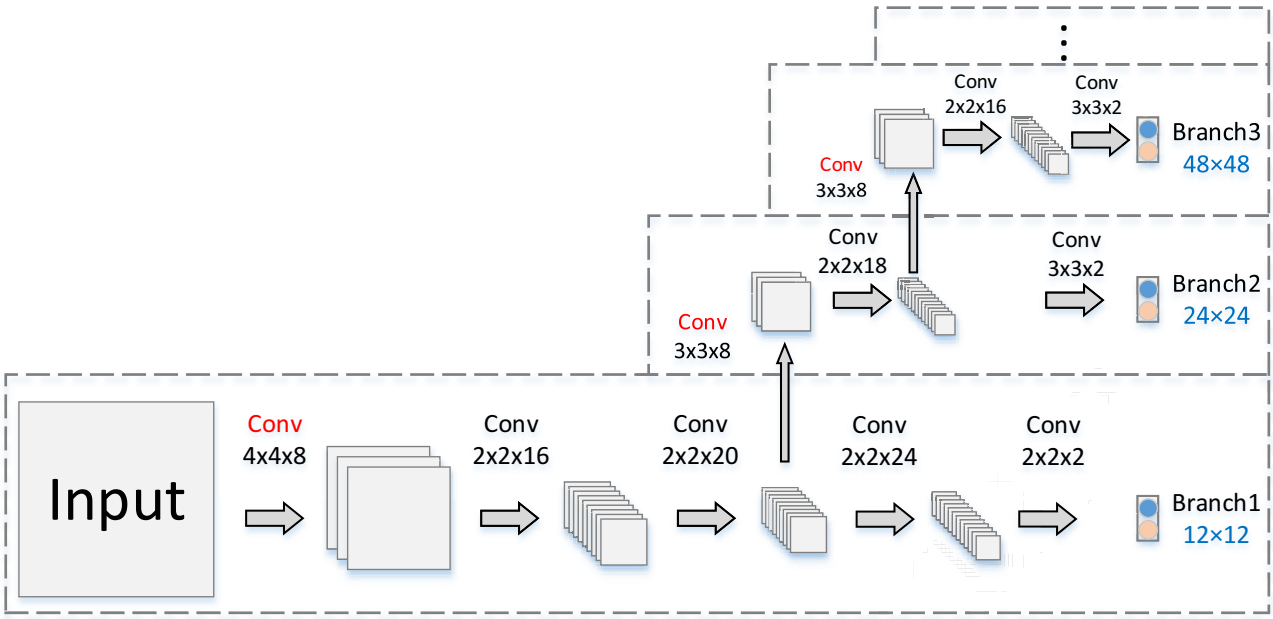


Fig. 2. The architecture of our pyramid network which has totally six branches. The last three branches have the same architecture as the branch3. The "Conv" means a convolutional layer, the expression under it denotes the kernel size. The red "Conv" indicates the convolutional layer has a stride of 2. Each convolutional layer is followed by a ReLU activation function (except the output layer) and each network branch is ended with a binary SoftMax classifier.

size is not 12×12, we need resize it twice) to search the whole image space. Compared with sliding windows on an image pyramid, our method can get both greater performance and faster speed. When the minimum face that can be detected is set to 48×48, then the maximum face we can detect is larger than 2171×2171, which has already met most of practical face detection requirements.

3.2. Network acceleration

Our network acceleration strategies mainly involve two aspects. One is how to accelerate a model with almost equal amount of parameters. The other is how to train a small model with satisfactory performance.

Caffe [33] is used to realize our work, which performs fast convolution by using im2col [34] to reduce the problem to matrix-matrix multiplication. We notice that a convolutional layer costs more forward propagation time than a fully-connected layer with the same amount of parameters. It is mainly because the im2col spends a lot of time stacking data patches into a large dense matrix. To accelerate convolutional computation, all strides of convolutional layers in Stage2 and Stage3 networks are set to 2 as shown in Fig.3. In this case, both convolutional frequency in im2col and computational complexity in matrix multiplication are reduced. Moreover, batch normalization (BN) [13] is introduced into our networks to improve the convergence speed and model performance. After training, we merge a BN layer with its previous neighbor convolutional or fully-connected layer to reduce inference time as following formula.

Table 2. Comparison of model size, speed and validation accuracy of our Stage2 and Stage3 networks to RNet and ONet in MTCNN [10]

model	size	1000x Forward Propagation	Validation Accuracy
RNet	399Kb	0.598S	0.972
Stage2	302Kb	0.127S	0.977
ONet	1522Kb	2.635S	0.984
Stage3	1165Kb	0.605S	0.988

$$\begin{aligned} y_1 &= Conv(x) = wx + b \\ y_2 &= BN(y_1) = \alpha\frac{y_1 - \mu}{\sigma} + \beta = \frac{\alpha w}{\sigma}x + \frac{\alpha b - \alpha\mu + \sigma\beta}{\sigma} \\ &= w'x + b' \end{aligned} \tag{1}$$

Where w and b are the weight and bias of a convolutional layer respectively. μ, σ, α and β are the mean value, variance, scale and shift parameters of BN [13] respectively. After merging, the BN layer is removed and the old weight and bias are replaced with w' and b'.

Besides, we hope to further improve inference speed through reducing the model size. However, a small model usually leads to performance degradation. To address this issue, we employ diverse strategies to strengthen the power of small models in Section 3.3. With these above techniques, our models get significant speed improvements as shown in Table 2.

3.3. Enhance network performance

Speed and performance usually contradict with each other. After network acceleration, we pool a variety of ideas to keep the performance and even improve it.

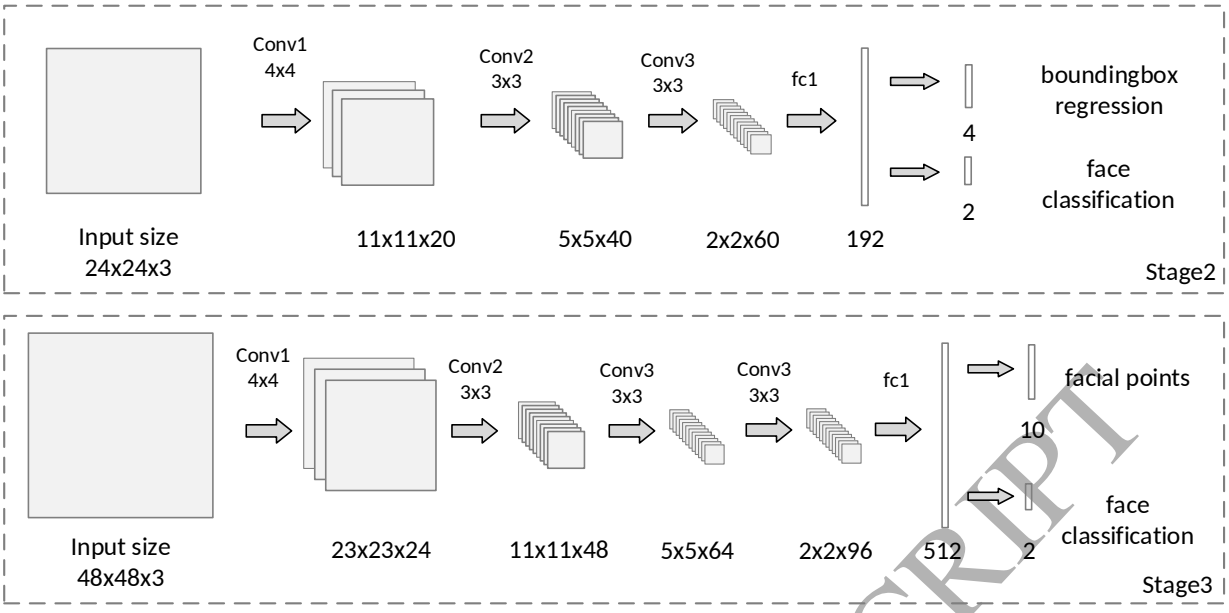


Fig. 3. The architecture of our Stage2 and Stage3 networks, where all convolutional layers have a stride of 2.

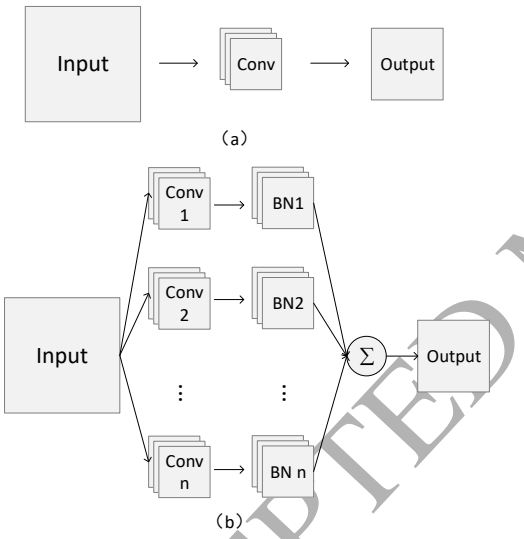


Fig. 4. Merge multi layers. (a) The original single convolutional structure. (b) The decomposition of the structure.

Merge Multi Layers: We propose a novel method of merging multi layers to reduce the difficulty of training a small model and improve its performance. During training, our network is composed of several modules. Fig.4 shows one typical convolutional module, where an original single convolutional structure is decomposed with multi convolutional layers followed by BN layers.

As shown in Fig.4, an original convolution layer is expressed as following formula.

$$y = wx + b = \sum_{i=1}^{n} \alpha_i (w_i^{'} x + b_i^{'}) = (\sum_{i=1}^{n} \alpha_i w_i^{'}) x + (\sum_{i=1}^{n} \alpha_i b_i^{'}) \quad (2)$$

Where $w_i^{'}$ and $b_i^{'}$ are from merging i-th convolutional layer and its followed BN layer using formula (1). n represents the number of decomposition and α_i denotes a corresponding learnable importance coefficient.

Knowledge Distilling: Negative samples with some intersection over ground-truth faces have a big chance of being mistaken for positive samples. Positive samples with more background region also have a big chance of being mistaken for negative samples. However, the common SoftMax classifier ignores the difference of samples in the same class and toughly pushes the output probability to the hard targets 0 or 1, which is an important cause of training difficulty with a small network. To solve this issue, knowledge distilling [14] is adopted to train a small model by using the class probabilities produced by a large model as soft targets. We first train a dense network with classical SoftMax loss. Then the probabilities of all training samples are computed with following formula.

$$p = \frac{e^{\frac{y_1^{'}}{T}}}{e^{\frac{y_0^{'}}{T}} + e^{\frac{y_1^{'}}{T}}} \quad (3)$$

Where $y_0^{'}$ and $y_1^{'}$ are the outputs of the dense network, T is a fixed parameter to control the soft of probability distribution. In our experiments, we set it as 10. Finally, our small network is trained with following loss function.

$$\begin{aligned} L &= \alpha L^{soft} + (1-\alpha) L^{hard} \\ L^{soft} &= -[p \log \frac{e^{\frac{y_1}{T}}}{e^{\frac{y_0}{T}} + e^{\frac{y_1}{T}}} + (1-p) \log \frac{e^{\frac{y_0}{T}}}{e^{\frac{y_0}{T}} + e^{\frac{y_1}{T}}}] \end{aligned} \quad (4)$$

Where L^{hard} is the classical SoftMax loss which is the cross entropy with the correct hard labels. L^{soft} is the cross entropy with the soft targets. y_0 and y_1 are the outputs of present light network. α is a weight factor.

Hard Mining: In our experiments, high training accuracy can be achieved soon after only thousands of training batches, which means most of training samples are easy ones and useless for improving network performance. To solve this issue, we first construct our large training set from hard public dataset WIDER FACE [35], AFLW [36] and Celeba [37]. Then both online and offline hard samples mining are adopted. Specifically, online hard mining [10] ignores 30% samples with minimum losses in the backward propagation and the ratio can even be increased to 60% during the latter of training. Offline hard mining finds error-classifying samples and retrains the model. A boosting-like strategy is also introduced to train present stage network with hard samples collected from previous stages.

4. Experiments

We give some training details in Section 4.1. In Section 4.2, the performance and speed of our face detector are compared with the state-of-the-art methods on FDDB [15]. In Section 4.3, we further evaluate the effectiveness of our multi-layers merging and knowledge distilling strategies.

4.1. Model training

Our face detector has three tasks in total. So there are three kinds of training samples for binary classification, bounding box regression and facial landmarks location respectively. The first two kinds of samples are mainly selected from hard public dataset WIDER FACE [35] and AFLW [36]. WIDER FACE dataset consists of 393703 faces in 32203 images where 50% of them for testing are without released bounding box ground truths. AFLW consists of 25993 labeled faces in 21997 images. Since there are some unlabeled faces in some images, we manually pick out images containing only one face to generate training data to avoid false negative samples. Finally, we crop faces from CelebA dataset [37] to train facial points location task. CelebA contains 202599 images, each of them has only one face with five facial points. Although the number of negative samples is much larger than the other kinds of samples, a sample ratio 3 : 1 (negatives/ positives) is fixed in each training batch.

For the Stage1, the branch1 network is first trained with 12×12 sized images. After that, its last two layers are removed and the parameters of the remaining layers are fixed simultaneously. Then the branch2 is added and trained with 24×24 sized images. By repeating the process above, a pyramid network with six branches is obtained. Since the blur from image magnification has negative effect on training process, we only choose positive samples whose original size are bigger than 32, 32, 32, 64, 128 and 256 for six branches respectively. Unfortunately, most of the labeled faces in WIDER FACE [35] are smaller than 64×64, which means there are not enough big faces as positive training samples to release the potential of high-level branches. It is the main reason why the validation accuracy tends to stop growing and even decrease in branch6 as shown in Table 1. Although the Stage1 network has six branches, it does not cost too much time to be trained. It is because that high-level branches with advanced inputs converge faster as shown in Fig.5.

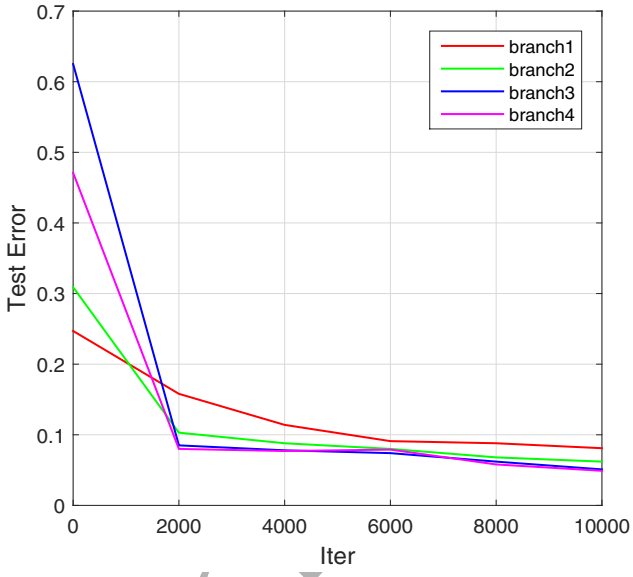


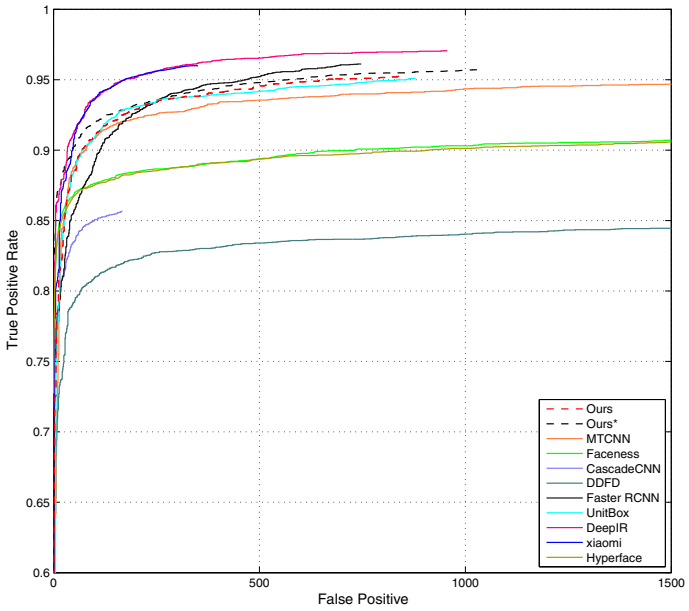
Fig. 5. Comparison of convergence rate of several branches in stage1 network.

For the last two stages, our training draws on diverse experiences from MTCNN [10], including how to prepare training data for different tasks, how to control the ratio of different samples and how to adjust the learning weight of different tasks.

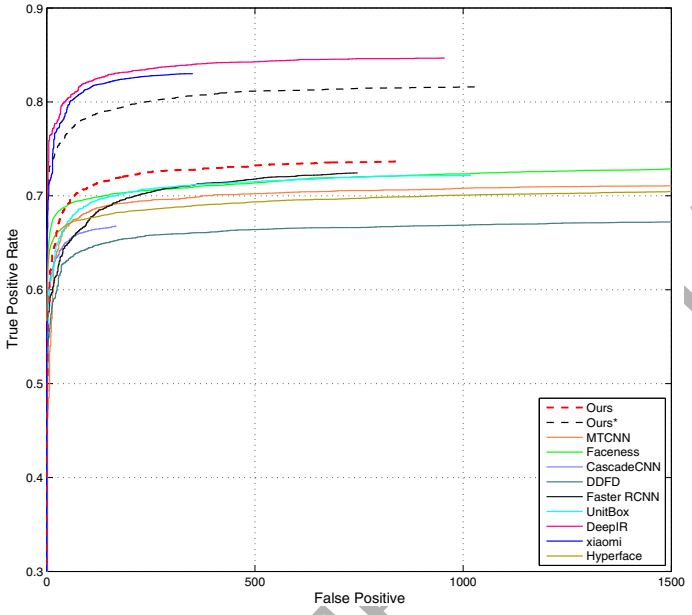
4.2. Performance and speed

In this section, our method is compared with the state-of-the-arts on FDDB [15]. FDDB is a widely-used face detection benchmark which contains 5171 faces with ellipse annotations in 2845 images. We give the Receiver Operating Characteristic (ROC) curves for discrete scores and continuous scores on FDDB as shown in Fig.6. As a result, our proposed face detector achieves comparable accuracy to the-state-of-arts. To show the robustness of our face detector to occlusions, difficult poses, and low resolution and out-of-focus faces, some detection examples on the FDDB are given in Fig. 7.

Although the performance of xiaomi [32], DeepIR [31] and Faster R-CNN [29] are better than ours, they used ResNet-50 [38], VGG-16 and VGG-16 [39] respectively. All of these huge models are too slow to support real-time face detection. Besides, the continuous scores of xiaomi and DeepIR are much larger than ours. This is mainly because they both conducted 10-fold-cross-validation on FDDB which introduces a special elliptical annotation. To give a more fair comparison under the continuous score evaluation, a new third detection stage is trained. Since FDDB has no labeled facial landmarks, we replace the original facial landmarks location task with bounding box regression. Then we perform 10-fold-cross-validation experiments using FDDB. Our new results denoted as "Ours*" are shown in Fig.6. The significantly improved continuous score indicates that the continuous score



(a) Discrete Score



(b) Continuous Score

Fig. 6. Comparisons on FDDB. It's important to note that xiaomi, DeepIR and Faster R-CNN are all huge models and impossible for real-time face detection.

is greatly affected by the different annotation styles between training data and testing data.

Except the discrete scores and continuous scores, the low time cost is our most prominent position. We further compare our detection speed with several state-of-the-art methods and the results are shown in Table 3. The experimental results show that the speed of our face detector obviously exceeds other CNNs-based detectors.

4.3. Multi-layers merging

To evaluate the effectiveness of our proposed multi-layer merging method, a same three-stage cascaded detector is trained without this technology. The recall rate with 500 false detections declines from 0.945 to 0.933. It is clear that our merging multi-layers strategy can significantly improve the performance of small models. A convolutional or fully-connected layer is decomposed into five same branches in our experiments. And more decomposition is not able to bring greater exaltation. It is noted that these five branches are initialized from gaussian distribution with zero means but different variances. And the learning rate of the weight factor α_i in formula (2) is 10x times as basic learning rate.

Fig. 7. Detection examples on the public FDDB dataset.

Table 3. Speed comparison between our method and several state-of-the-art methods

Method	GPU	Speed
CascadeCNN[11]	Titan Black	100FPS
MTCNN[10]	Titan Black	99FPS
Faceness[6]	Titan Black	20FPS
Faster R-CNN[29]	Tesla K40c	2.6 FPS
HyperFace[7]	Titan X	0.3FPS
Ours	**Titan Black**	**165FPS**

4.4. Knowledge distilling

To verify the effectiveness of the knowledge distilling strategy, another face detector with the same architecture is trained without knowledge distilling. Keeping the same recall rate in each stage, the numbers of candidate faces on FDDB [15] are shown in Table 4. Since knowledge distilling can improve networks' classification performance, the number of face proposals can be reduced with the same recall rate, which means the detection speed will be improved greatly.

Table 4. The numbers of candidate faces in each detection stage on FDDB

knowledge distilling	stage1	stage2	stage3
With	246K	24K	5.8K
Without	299K	28K	6.3K

5. Conclusion

In this paper, we presented a fast CNNs cascade face detector whose performance is comparable with the state-of-the-arts.

Our major contribution is the proposed pyramid network for quickly generating multi-scale face proposals, which reduces the major computational complexity in cascaded face detectors. Although our detection speed significantly exceeds other CNNs-based face detectors, generating face proposals still takes up the bulk of our detection time. So, our future work will keep focusing on improving the speed of generating face proposals by designing more efficient proposal network.

Acknowledgments

We would like to thank professor Tian Qi (the University of Texas at San Antonio) and Zhou Xi (Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences) for their suggestions to this project. This work is supported by National Natural Science Foundation of China (61572307).

References

[1] C. Wang, Y. Wang, Z. Zhang, Y. Wang, Face tracking and recognition via incremental local sparse representation, in: Image and Graphics (ICIG), 2013 Seventh International Conference on, IEEE, 2013, pp. 493–498.

[2] Z. He, M. Kan, J. Zhang, X. Chen, S. Shan, A fully end-to-end cascaded cnn for facial landmark detection, in: Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on, IEEE, 2017, pp. 200–207.

[3] C. Wang, Y. Wang, Z. Zhang, Y. Wang, Incremental learning patch-based bag of facial words representation for face recognition in videos, Multimedia Tools and Applications 72 (2014) 2439–2467.

[4] S. S. Farfade, M. J. Saberian, L.-J. Li, Multi-view face detection using deep convolutional neural networks, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ACM, 2015, pp. 643–650.

[5] B. Yang, J. Yan, Z. Lei, S. Z. Li, Convolutional channel features, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 82–90.

[6] S. Yang, P. Luo, C.-C. Loy, X. Tang, From facial parts responses to face detection: A deep learning approach, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3676–3684.

[7] R. Ranjan, V. M. Patel, R. Chellappa, Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition, arXiv preprint arXiv:1603.01249 (2016).

[8] 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, 2016, pp. 420–436.

[9] J. Yu, Y. Jiang, Z. Wang, Z. Cao, T. Huang, Unitbox: An advanced object detection network, in: Proceedings of the 2016 ACM on Multimedia Conference, ACM, 2016, pp. 516–520.

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

[11] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325–5334.

[12] I. A. Kalinovsky, V. G. Spitsyn, Compact convolutional neural network cascade for face detection, arXiv preprint arXiv:1508.01292 (2015).

[13] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).

[14] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, arXiv preprint arXiv:1503.02531 (2015).

[15] V. Jain, E. G. Learned-Miller, Fddb: A benchmark for face detection in unconstrained settings, UMass Amherst Technical Report (2010).

[16] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, IEEE, 2001, pp. I–I.

[17] J. Trefnỳ, J. Matas, Extended set of local binary patterns for rapid object detection, in: Computer Vision Winter Workshop, 2010, pp. 1–7.

[18] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2879–2886.

[19] 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.

[20] C. Huang, H. Ai, Y. Li, S. Lao, High-performance rotation invariant multiview face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007).

[21] A. Torralba, K. P. Murphy, W. T. Freeman, Sharing visual features for multiclass and multiview object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007).

[22] S. Wu, M. Kan, Z. He, S. Shan, X. Chen, Funnel-structured cascade for multi-view face detection with alignment-awareness, Neurocomputing 221 (2017) 138–145.

[23] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[24] G. Ghiasi, C. C. Fowlkes, Occlusion coherence: Detecting and localizing occluded faces, arXiv preprint arXiv:1506.08347 (2015).

[25] M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool, Face detection without bells and whistles, in: European Conference on Computer Vision, Springer, 2014, pp. 720–735.

[26] Z. Zheng, L. Zheng, Y. Yang, Pedestrian alignment network for large-scale person re-identification, arXiv preprint arXiv:1707.00408 (2017).

[27] L. Zheng, Y. Yang, Q. Tian, Sift meets cnn: A decade survey of instance retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence (2017).

[28] R. Ranjan, V. M. Patel, R. Chellappa, A deep pyramid deformable part model for face detection, in: Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on, IEEE, 2015, pp. 1–8.

[29] H. Jiang, E. Learned-Miller, Face detection with the faster r-cnn, arXiv preprint arXiv:1606.03473 (2016).

[30] S. Ren, K. He, R. Girshick, 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.

[31] X. Sun, P. Wu, S. C. Hoi, Face detection using deep learning: An improved faster rcnn approach, arXiv preprint arXiv:1701.08289 (2017).

[32] S. Wan, Z. Chen, T. Zhang, B. Zhang, K.-k. Wong, Bootstrapping face detection with hard negative examples, arXiv preprint arXiv:1608.02236 (2016).

[33] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675–678.

[34] K. Chellapilla, S. Puri, P. Simard, High performance convolutional neural networks for document processing, in: Tenth International Workshop on Frontiers in Handwriting Recognition, Suvisoft, 2006.

[35] 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.

[36] M. Köstinger, P. Wohlhart, P. M. Roth, H. Bischof, Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 2144–2151.

[37] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.

[38] 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.

[39] K. Simonyan, A. Zisserman, Very deep convolutional networks for

large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).