Accurate Face Detection for High Performance

Faen Zhang, Xinyu Fan, Guo Ai, Jianfei Song, Yongqiang Qin, Jiahong Wu
AInnovation Technology Ltd, Beijing, China
{zhangfaen,fanxinyu,aiguo}@ainnovation.com
{songjianfei,qinyongqiang,wujiahong}@ainnovation.com

Abstract

Face detection has witnessed significant progress due to the advances of deep convolutional neural networks (CNNs). Its central issue in recent years is how to improve the detection performance of tiny faces. To this end, many recent works propose some specific strategies, redesign the architecture and introduce new loss functions for tiny object detection. In this report, we start from the popular one-stage RetinaNet [20] approach and apply some recent tricks to obtain a high performance face detector namely AInnoFace. Specifically, we apply the Intersection over Union (IoU) loss function [45] for regression, employ the two-step classification and regression [4] for detection, revisit the data augmentation based on data-anchor-sampling [32] for training, utilize the max-out operation [54] for classification and use the multi-scale testing strategy [54] for inference. As a consequence, the proposed face detection method achieves state-of-the-art performance on the most popular and challenging face detection benchmark WIDER FACE [43] dataset.

1 Introduction

Face detection is a tremendously important field in computer vision needed for face recognition, sentiment analysis, video surveillance, and many other fields. Given an arbitrary image, the goal of face detection is to determine whether there are any faces in the image, and if present, return the image location and extent of each face. The recent issue of face detection is how to improve the detection performance in unrestricted scenarios. Because detecting faces in real-world images has many difficulties including occlusion, significant scale variation, different illumination conditions, various facial poses, rich facial expressions, etc. Many works are devoted to solving this issue and great progress has been achieved with the development of deep convolutional neural networks (CNNs). For example, the average precision (AP) performance on the challenging WIDER FACE dataset [43] has been improved from 40% to 90% over recent years.

To improve the performance of face detection in unrestricted scenarios where exists plenty of tiny faces, some works [40, 42, 48, 24, 44] combine traditional methods (*e.g.*, cascade-based mechanism and part-based in DPM) with deep learning methods (*e.g.*, CNN) to perform face detection. A number of works [57, 14, 23] resort to the context information around the face region to find tiny faces based on the Faster R-CNN [27] and SSD [21] detectors. Several works [3, 4, 47, 33] redesign the architecture of modern object detection to better detect tiny faces. A series of works [35, 37, 54, 56, 46] propose some special strategies for tiny faces into the generic object detection methods to improve face detection performance. There are many works [32, 51, 16] present some new data augmentations for tiny faces to improve the performance. Some works [36, 55] introduce the attention mechanism on the feature maps to focus on face regions for better detection performance.

In this work, we first modify the popular one-stage RetinaNet [20] method to perform face detection as our baseline model. Then some recent tricks are applied on this baseline to develop a high performance face detector namely AInnoFace: (1) Employing the two-step classification and regression for

detection; (2) Applying the Intersection over Union (IoU) loss function for regression; (3) Revisiting the data augmentation based on data-anchor-sampling for training; (4) Utilizing the max-out operation for robuster classification; (5) Using the multi-scale testing strategy for inference. Consequently, we achieve some new state-of-the-art AP results on the challenging face detection benchmark WIDER FACE [43] dataset.

2 Related Work

2.1 Traditional Method

Face detection has been extensively studied from its emergence in the 1990s to the present because of its wide practical applications. The pioneering work [34] of Viola and Jones uses the Haar-like feature and the AdaBoost strategy to train several cascaded face detectors, achieving a very good tradeoff between accuracy and efficiency in some simple and fixed scenarios. Afterwards, subsequent works [2, 18] have made great progress by developing more advanced features and more powerful classifiers. Apart from the boosted cascade methods, several studies [39, 58, 40] introduce another famous framework of Deformable Part Model (DPM) [22] to the filed of face detection task, which detect faces by modelling the relationship of deformable facial parts and achieve promising performance in some simple application scenarios. However, these traditional face detectors are unreliable in complex scenarios because they depend on non-robust hand-crafted features and classifiers.

2.2 Deep Learning Method

Deep learning approaches significantly boost the recent progress in the face detection filed and the CNN-based face detectors have achieved the highest performance in the last few years. The cascade CNN-based methods [15, 26] train a series of CNN models separately or jointly to perform face detection, and achieve promising accuracy and efficiency simultaneously. After that, MTCNN [48] and PCN [30] add another extra branch to detect five facial landmarks and predict face angles via the multi-task learning in a coarse-to-fine manner under a cascade-style structure. Faceness [42] obtains different scores according to the spatial structure and arrangement of facial parts to detect faces under severe occlusion and unconstrained pose variations. LDCF+ [24] utilizes the boosted decision tree classifier to detect faces. UnitBox [45] introduces an Intersection-over-Union (IoU) loss to directly minimize the IoUs of the predictions and the ground-truths for more accurate location. ScaleFace [44] detects different scales of faces via applying a specialized set of CNNs with different structures. SAFD [11] develops a scale proposal stage to automatically normalize face sizes prior to detection. Hu *et al.* [14] explore the contextual information with some separate detectors for different scales to find tiny faces. S^2AP [31] finds face via paying attention to specific scales in image pyramid and valid locations in each scales layer. Zhu *et al.* [56] use the Expected Max Overlapping (EMO) score to evaluate the quality of anchor setting. Bai *et al.* [1] generate a clear super-resolution face from a blurry small one via to GAN [9] to detect blurry small faces.

Besides, many state-of-the-are face detectors are evolved from generic object detection methods including the two-stage approach (Faster R-CNN [27], R-FCN [5] and FPN [19]) and the one-stage approach (SSD [21], RefineDet [49] and RetinaNet [20]). Based on Faster R-CNN and R-FCN, some face detection methods (*e.g.*, Face R-CNN [35], Face R-FCN [37], CMS-RCNN [57] and FDNet [46]) design several specific training and testing strategies with the consideration of the characteristics of face detection. Similar to FPN, FANet [47] aggregates higher-level features to augment lower-level features for face detection. Based on SSD, DCFPN [52] and FaceBoxes [53] design a lightweight face detection network to achieve CPU real-time speed with promising result. In contrast, high performance face detectors including S^3FD [54], SFDet [50], SSH [23], PyramidBox [32, 17] and DSFD [16] equip SSD with some specific strategies for better detection of small faces, such as architecture diagram, training strategy, contextual reasoning and multiple layers exploiting. Besides, FAN [36] and DFS [33] use different types of attention mechanism on RetinaNet to handle hard faces. SRN [4] combines the multi-step detection in RefineDet [49] and the focal loss in RetinaNet [20] to perform efficient and accurate face detection. After that, VIM-FD [55] and ISRN [51] combine many previous techniques on SRN and achieve new state-of-the-art performance.

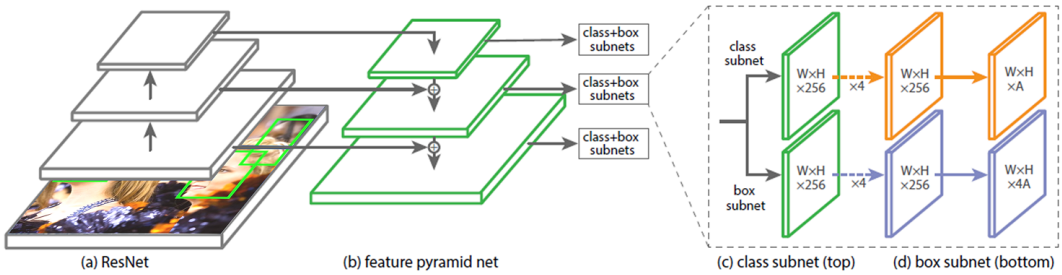


Figure 1: Architecture of our face detector baseline built on RetinaNet [20]. (a) Backbone: a feed-forward ResNet-152 [12] architecture extracts the multi-scale feature maps. (b) Neck: a 6-level Feature Pyramid Network (FPN) [19] structure generates a richer multi-scale convolutional feature pyramid. After that, two shared subnetworks are attached, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). At the end, we use the focal loss [20] for the binary classification and the IoU loss [45] for the regression.

3 Method

Starting from the RetinaNet [20] face detector baseline, we apply some recently proposed strategies to achieve state-of-the-art performance on the challenging WIDER FACE [43] dataset.

3.1 RetinaNet Baseline

RetinaNet is one of the most popular one-stage object detection methods. One-stage detectors that are applied over a regular, dense sampling of possible object locations have the potential to be faster and simpler, but have trailed the accuracy of two-stage detectors because of extreme class imbalance encountered during training. To solve this issue, the focal loss is proposed in RetinaNet as follow:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t). \tag{1}$$

and

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \tag{2}$$

where $y \in \{\pm 1\}$ specifies the ground-truth class, $p \in [0,1]$ is the model's estimated probability for the class with label $y = 1$, α_t is a balanced factor and γ a tunable focusing parameter. The focal loss is the reshaping of cross entropy loss such that it down-weights the loss assigned to well-classified examples. The novel focal loss focuses training on a sparse set of hard examples and prevents the vast number of easy negatives from overwhelming the detector during training.

In this report, we take the modified RetinaNet shown in Figure 1 as our baseline, which is a single and unified network composed of a backbone network, a neck network and two task-specific subnetworks. The backbone and neck networks are responsible for computing a multi-scale convolutional feature maps over an entire input image. The first subnet performs classification on the backbones output and the second subnet performs convolution bounding box regression. Specifically, we adopt the ResNet-152 [12] with 6-level feature pyramid structure as our backbone network. We follow the way in FPN [19] to generate the 6-level feature maps (P2 to P7) for detection.

3.2 IoU Regression Loss

The object detection task consists of the classification subtask and the regression subtask. For regression, the smooth L1 loss [7] is the common loss function used to reduce the difference between anchor boxes and ground-truth bounding boxes, and the Intersection over Union (IoU) is the most popular evaluation metric used in the object detection benchmarks. However, as indicated in [28], there is a gap between optimizing the commonly used smooth L1 distance losses for regressing the parameters of a bounding box and maximizing this IoU metric value. The optimal objective for a metric should be the metric itself, so we follow the UnitBox [45] to minimize the difference between the predictions and the ground-truths via directly using IoU as the regression loss.

The IoU regression loss function is defined as below:

$$L_{IoU} = -\ln \frac{Intersection(B_p, B_{gt})}{Union(B_p, B_{gt})} \tag{3}$$

where $B_p = (x_1, y_1, x_2, y_2)$ and $B_{gt} = (x_1^*, y_1^*, x_2^*, y_2^*)$ are the predicted bounding box and the ground-truth bounding box respectively, $Intersection()$ and $Union()$ indicate the intersection and union area between B_p and B_{gt}.

3.3 Selective Refinement Network

As described in Selective Refinement Network (SRN) [4], using RetinaNet to perform face detection still exists two problems: (a) low recall efficiency: the precision is not high enough at high recall rates, *i.e.*, the Precision-Recall curve extends far enough to the right but not steep enough; (b) low location accuracy: the performance drops dramatically as the IoU threshold increases, *i.e.*, the accuracy of the bounding box location needs to be improved. To solve the aforementioned two issues, Selective Two-step Classification (STC) and Selective Two-step Regression (STR) are proposed in SRN and we follow these designs to further improve the performance of our face detector.

STC conducts two-step classification on three low level detection layers to filter out most simple negatives and reduce the search space for the subsequent classifier. Its loss function is:

$$L_{STC}(\{p_i\}, \{q_i\}) = \frac{1}{N_{s_1}} \sum_{i \in \Omega} L_{FL}(p_i, l_i^*) + \frac{1}{N_{s_2}} \sum_{i \in \Phi} L_{FL}(q_i, l_i^*), \tag{4}$$

STR performs is two-step regression on three high level detection layers to adjust anchors and provide better initialization for the subsequent regressor. Its loss function is:

$$L_{STR}(\{x_i\}, \{t_i\}) = \frac{1}{N_{s_1}} \sum_{i \in \Psi} [l_i^* = 1] L_r(x_i, g_i^*) + \frac{1}{N_{s_2}} \sum_{i \in \Phi} [l_i^* = 1] L_r(t_i, g_i^*), \tag{5}$$

where i is the anchor index, p_i/q_i and x_i/t_i are the prediction of classification and regression in the first/second step, l_i^*/g_i^* are the ground truth of class/location, N_{s_1}/N_{s_2} are the number of positive anchors in the first/second step, Ω/Ψ are the collection of classification/regression samples for the first step and Φ is the collection of samples for the second step, L_{FL} is the sigmoid focal loss and $[l_i^* = 1]\mathcal{L}_{\text{r}}$ indicates that the IoU regression loss is computed only for positive anchors.

3.4 Data Augmentation

Date augmentation is of importance for one-stage detectors to construct a robust model to adapt to variations of objects, especially for face detection where has plenty tiny faces. Following most of the face detectors, we randomly expand and crop the original training images with additional random photometric distortion [13] and flipping to generate the training samples. Besides, with probability of 0.5, we replace the above random cropping operation with the anchor-based sampling like data-anchor-sampling in PyramidBox [32] to diversify the scale distribution of training samples. The anchor-based sampling operation first randomly selects a face of size S_{face} in a batch, then finds its nearest anchor scale S_{anchor}. After that, it chooses a random scale S_{random} around the nearest anchor scale. Finally, it resizes the image by $S^* = S_{random}/S_{face}$ and randomly crops a standard size of the training size containing the selected face to get the anchor-sampled training data.

3.5 Max-out Label

To reduce the tiny false positives from background regions, the max-out operation for the background class is introduced in [54], and then the following work [32] uses this max-out operation on both foreground and background classes. In the proposed face detection method, the max-out operation is applied in the classification subnet to recall more faces and reduce false positives simultaneously. To be more specific, the classification subnet first predicts $c_p + c_n$ scores for each anchor, and then selects $\max\{c_p\}$ and $\max\{c_n\}$ as the final face and no-face confidence score to compute the classification loss. Empirically, we set $c_p = 3$ and $c_n = 3$ to train our final model.

3.6 Multi-scale Testing

Since there are plenty of tiny faces in the challenging WIDER FACE dataset, the multi-scale testing strategy is useful to improve the performance. We use the open source code[1] to conduct the multi-scale testing during inference. It inputs the image to the trained model multiple times with different sizes and then merge these detection results with the bounding box voting operation.

4 Experiment

4.1 Experimental Dataset

We verify the proposed AInnoFace detector on the WIDER FACE [43] dataset, which is a popular face detection benchmark dataset and whose images are selected from the publicly available WIDER [38] dataset. The WIDER FACE dataset contains $32,203$ images and $393,703$ annotated face bounding boxes with a high degree of variability in scale, pose, occlusion, expression, makeup and illumination as depicted in Figure 2. All the images are organized based on 61 event classes and are randomly selected from each event class by $40\%/10\%/50\%$ as training, validation and testing subsets. Based on the detection rate of EdgeBox [59], the validation and testing subsets are divided into three difficulty levels: Easy, Medium, Hard. The Average Precision (AP) is adopted as the evaluation metric. Following MALF [41] and Caltech [6] datasets, this dataset does not release bounding box ground truth for the testing images and researchers are required to submit final prediction files to get the AP performance on the testing subset. The proposed method is trained on the training subset and evaluated on both validation and testing subsets.

Figure 2: Some sample images for each attribute of WIDER FACE [43].

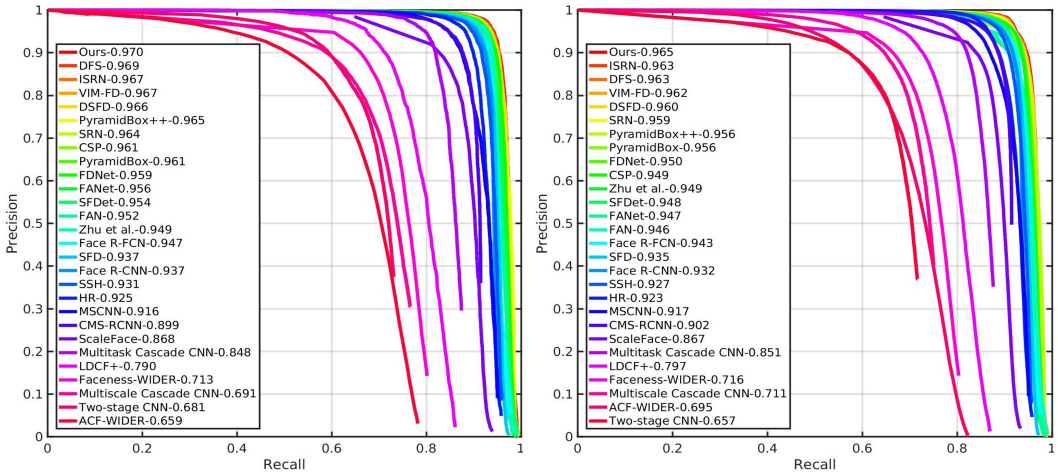
4.2 Anchor Detail

Following [4] we set two $2S$ and $2\sqrt{2}S$ anchor scales (S is the downsampling factor of detection layer) and one 1.25 aspect ratio. Thus, there are $A = 2$ anchors at each location, covering the scale of $8 - 362$ pixels in the 1024×1024 input image. During the training phase, anchors are assigned to ground-truth boxes using the θ_p IoU threshold and to background if their IoU is in $[0, \theta_n)$. The rest anchors in $[\theta_n, \theta_p)$ are ignored. We set $\theta_n = 0.3$ and $\theta_p = 0.7$ for the first step, and $\theta_n = 0.4$ and $\theta_p = 0.5$ for the second step.

4.3 Optimization Detail

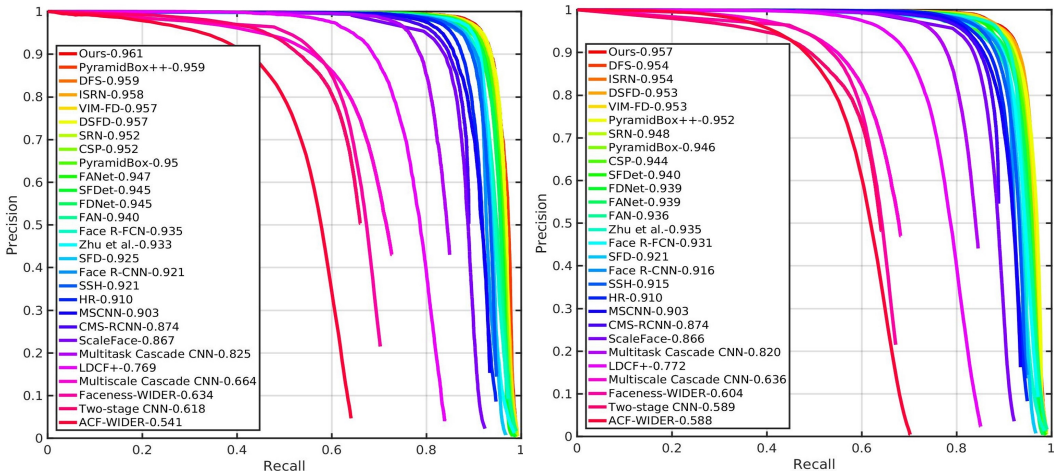
The backbone network in the proposed AInnoFace detector is initialized by the pretrained model on the ImageNet [29] dataset. We use the "xavier" [8] method to randomly initialize the parameters in the newly added convolutional layers. The stochastic gradient descent (SGD) algorithm is used to fine-tune the model with 0.9 momentum, 0.0001 weight decay and batch size 32. The warmup

[1] https://github.com/sfzhang15/SFD/blob/master/sfd_test_code/WIDER_FACE/wider_test.py



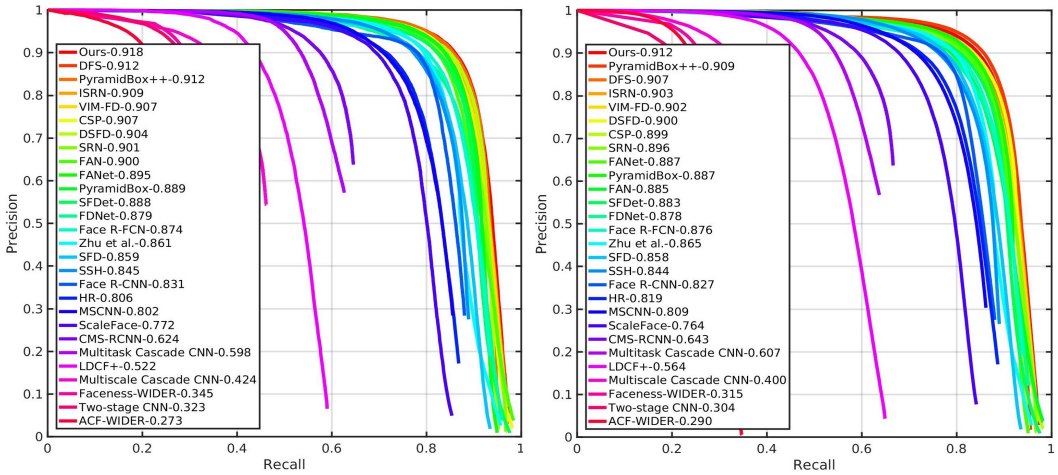
(a) Val: Easy

(b) Test: Easy



(c) Val: Medium

(d) Test: Medium



(e) Val: Hard

(f) Test: Hard

Figure 3: Precision-recall curves on WIDER FACE validation and testing subsets.

strategy [10] is applied to gradually ramp up the learning rate from 0.0003125 to 0.01 at the first 5 epochs. After that, it switches to the regular learning rate schedule, i.e., dividing by 10 at 100 and 120 epochs and ending at 130 epochs. The full training and testing codes are built on the PyTorch library [25].

4.4 Evaluation Result

Figure 3 shows the comparison of the proposed AInnoFace detector with twenty-seven state-of-the-art methods [3, 4, 14, 16, 23, 24, 32, 33, 35, 36, 37, 40, 42, 43, 44, 46, 47, 48, 54, 56, 57, 55, 51, 17, 50] on both the validation and testing subsets based on the precision-recall curve and AP. As shown in Figure 3, our face detector sets some new state-of-the-art results based on the AP score across the three subsets on both validation and testing subsets, *i.e.*, 97.0% (Easy), 96.1% (Medium) and 91.8% (Hard) for validation subset, and 96.5% (Easy), 95.7% (Medium) and 91.2% (Hard) for testing subset. These results outperform all the compared state-of-the-art methods and demonstrate the superiority of our AInnoFace detector.

5 Conclusion

In this report, we present a high performance face detector by equipping the popular one-stage RetinaNet [20] method with some recent tricks: (1) Employing the two-step classification and regression for detection; (2) Applying the Intersection over Union (IoU) loss function for regression; (3) Revisiting the data augmentation based on data-anchor-sampling for training; (4) Utilizing the max-out operation for robuster classification; (5) Using the multi-scale testing strategy for inference. Experiments on the WIDER FACE dataset demonstrate that the proposed AInnoFace detector achieves the state-of-the-art detection performance.

References

[1] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Finding tiny faces in the wild with generative adversarial network. In *CVPR*, 2018.

[2] S. Charles Brubaker, Jianxin Wu, Jie Sun, Matthew D. Mullin, and James M. Rehg. On the design of cascades of boosted ensembles for face detection. *IJCV*, 2008.

[3] Zhaowei Cai, Quanfu Fan, Rogério Schmidt Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, 2016.

[4] Cheng Chi, Shifeng Zhang, Junliang Xing, Zhen Lei, Stan Z. Li, and Xudong Zou. Selective refinement network for high performance face detection. In *AAAI*, 2019.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NIPS*, 2016.

[6] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *TPAMI*, 2012.

[7] Ross B. Girshick. Fast R-CNN. In *ICCV*, 2015.

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[10] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, 2017.

[11] Zekun Hao, Yu Liu, Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu. Scale-aware face detection. In *CVPR*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] Andrew G. Howard. Some improvements on deep convolutional neural network based image classification. *CoRR*, 2013.

[14] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *CVPR*, 2017.

[15] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua. A convolutional neural network cascade for face detection. In *CVPR*, 2015.

[16] Jian Li, Yabiao Wang, Changan Wang, Ying Tai, Jianjun Qian, Jian Yang, Chengjie Wang, Ji-Lin Li, and Feiyue Huang. DSFD: dual shot face detector. In *CVPR*, 2019.

[17] Zhihang Li, Xu Tang, Junyu Han, Jingtuo Liu, and Ran He. Pyramidbox++: High performance detector for finding tiny face. *CoRR*, 2019.

[18] Shengcai Liao, Anil K. Jain, and Stan Z. Li. A fast and accurate unconstrained face detector. *TPAMI*, 2016.

[19] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[20] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *ECCV*, 2016.

[22] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc J. Van Gool. Face detection without bells and whistles. In *ECCV*, 2014.

[23] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, and Larry S. Davis. SSH: single stage headless face detector. In *ICCV*, 2017.

[24] Eshed Ohn-Bar and Mohan M. Trivedi. To boost or not to boost? on the limits of boosted trees for object detection. In *ICPR*, 2016.

[25] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch, 2017.

[26] Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu. Joint training of cascaded CNN for face detection. In *CVPR*, 2016.

[27] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *TPAMI*, 2017.

[28] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[30] Xuepeng Shi, Shiguang Shan, Meina Kan, Shuzhe Wu, and Xilin Chen. Real-time rotation-invariant face detection with progressive calibration networks. In *CVPR*, 2018.

[31] Guanglu Song, Yu Liu, Ming Jiang, Yujie Wang, Junjie Yan, and Biao Leng. Beyond trade-off: Accelerate fcn-based face detector with higher accuracy. In *CVPR*, 2018.

[32] Xu Tang, Daniel K Du, Zeqiang He, and Jingtuo Liu. Pyramidbox: A context-assisted single shot face detector. In *ECCV*, 2018.

[33] Wanxin Tian, Zixuan Wang, Haifeng Shen, Weihong Deng, Binghui Chen, and Xiubao Zhang. Learning better features for face detection with feature fusion and segmentation supervision. *CoRR*, 2018.

[34] Paul A. Viola and Michael J. Jones. Robust real-time face detection. *IJCV*, 2004.

[35] Hao Wang, Zhifeng Li, Xing Ji, and Yitong Wang. Face R-CNN. *CoRR*, 2017.

[36] Jianfeng Wang, Ye Yuan, and Gang Yu. Face attention network: An effective face detector for the occluded faces. *CoRR*, 2017.

[37] Yitong Wang, Xing Ji, Zheng Zhou, Hao Wang, and Zhifeng Li. Detecting faces using region-based fully convolutional networks. *CoRR*, 2017.

[38] Yuanjun Xiong, Kai Zhu, Dahua Lin, and Xiaoou Tang. Recognize complex events from static images by fusing deep channels. In *CVPR*, 2015.

[39] Junjie Yan, Zhen Lei, Longyin Wen, and Stan Z. Li. The fastest deformable part model for object detection. In *CVPR*, 2014.

[40] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z. Li. Aggregate channel features for multi-view face detection. In *IJCB*, 2014.

[41] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z. Li. Fine-grained evaluation on face detection in the wild. In *FG*, 2015.

[42] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, 2015.

[43] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. WIDER FACE: A face detection benchmark. In *CVPR*, 2016.

[44] Shuo Yang, Yuanjun Xiong, Chen Change Loy, and Xiaoou Tang. Face detection through scale-friendly deep convolutional networks. *CoRR*, 2017.

[45] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas S. Huang. Unitbox: An advanced object detection network. In *ACMMM*, 2016.

[46] Changzheng Zhang, Xiang Xu, and Dandan Tu. Face detection using improved faster RCNN. *CoRR*, 2018.

[47] Jialiang Zhang, Xiongwei Wu, Jianke Zhu, and Steven C. H. Hoi. Feature agglomeration networks for single stage face detection. *CoRR*, 2017.

[48] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 2016.

[49] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018.

[50] Shifeng Zhang, Longyin Wen, Hailin Shi, Zhen Lei, Siwei Lyu, and Stan Z Li. Single-shot scale-aware network for real-time face detection. *IJCV*, 2019.

[51] Shifeng Zhang, Rui Zhu, Xiaobo Wang, Hailin Shi, Tianyu Fu, Shuo Wang, Tao Mei, and Stan Z. Li. Improved selective refinement network for face detection. *CoRR*, 2019.

[52] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li. Detecting face with densely connected face proposal network. In *CCBR*, 2017.

[53] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li. Faceboxes: A CPU real-time face detector with high accuracy. In *IJCB*, 2017.

[54] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z. Li. S^3FD: Single shot scale-invariant face detector. In *ICCV*, 2017.

[55] Yundong Zhang, Xiang Xu, and Xiaotao Liu. Robust and high performance face detector. *CoRR*, 2019.

[56] Chenchen Zhu, Ran Tao, Khoa Luu, and Marios Savvides. Seeing small faces from robust anchors perspective. In *CVPR*, 2018.

[57] Chenchen Zhu, Yutong Zheng, Khoa Luu, and Marios Savvides. CMS-RCNN: contextual multi-scale region-based CNN for unconstrained face detection. *CoRR*, 2016.

[58] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012.

[59] C. Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.