S^{3}FD: Single Shot Scale-invariant Face Detector

Shifeng Zhang Xiangyu Zhu Zhen Lei* Hailin Shi Xiaobo Wang Stan Z. Li
CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{shifeng.zhang,xiangyu.zhu,zlei,hailin.shi,xiaobo.wang,szli}@nlpr.ia.ac.cn

Abstract

This paper presents a real-time face detector, named Single Shot Scale-invariant Face Detector (S^{3}FD), which performs superiorly on various scales of faces with a single deep neural network, especially for small faces. Specifically, we try to solve the common problem that anchor-based detectors deteriorate dramatically as the objects become smaller. We make contributions in the following three aspects: 1) proposing a scale-equitable face detection framework to handle different scales of faces well. We tile anchors on a wide range of layers to ensure that all scales of faces have enough features for detection. Besides, we design anchor scales based on the effective receptive field and a proposed equal proportion interval principle; 2) improving the recall rate of small faces by a scale compensation anchor matching strategy; 3) reducing the false positive rate of small faces via a max-out background label. As a consequence, our method achieves state-of-the-art detection performance on all the common face detection benchmarks, including the AFW, PASCAL face, FDDB and WIDER FACE datasets, and can run at 36 *FPS on a Nvidia Titan X (Pascal) for VGA-resolution images.*

1. Introduction

Face detection is the key step of many subsequent face-related applications, such as face alignment [50, 61], face recognition [32, 40, 62], face verification [44, 46] and face tracking [17], etc. It has been well developed over the past few decades. Following the pioneering work of Viola-Jones face detector [48], most of early works focus on designing robust features and training effective classifiers. But these approaches depend on non-robust hand-crafted features and optimize each component separately, making the face detection pipeline sub-optimal.

In recent years, convolutional neural network (CNN) has achieved remarkable successes, ranging from image classification [10, 42, 45] to object detection [8, 23, 26, 37, 38], which also inspires face detection. On the one hand, many works [21, 31, 54, 55, 58] have applied CNN as the feature extractor in the traditional face detection framewrok. On the other hand, face detection is regarded as a special case of generic object detection and lots of methods [3, 15, 43, 49, 59] have inherited valid techniques from generic object detection method [38]. Following the latter route, we improve the anchor-based generic object detection frameworks and propose a state-of-the-art face detector.

Anchor-based object detection methods [26, 38] detect objects by classifying and regressing a series of pre-set anchors, which are generated by regularly tiling a collection of boxes with different scales and aspect ratios on the image.

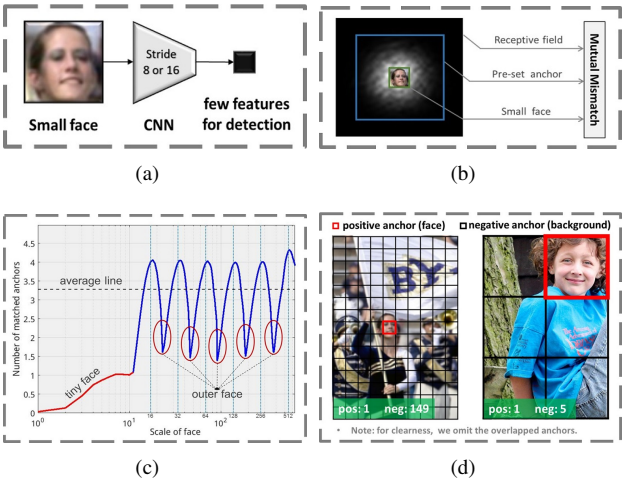


Figure 1. Reasons behind the problem of anchor-based methods. **(a) Few features:** Small faces have few features at detection layer. **(b) Mismatch:** Anchor scale mismatches receptive field and both are too large to fit small face. **(c) Anchor matching strategy:** The figure demonstrates the number of matched anchors at different face scales under current anchor matching method. It reflects that tiny and outer faces match too little anchors. **(d) Background from small anchors:** The two figures have the same resolution. The left one tiles small anchors to detect the small face and the right one tiles big anchors to detect the big face. Small anchors bring about plenty of negative anchors on the background.

*Corresponding author

These anchors are associated with one [38] or several [26] convolutional layers, whose spatial size and stride size determine the position and interval of the anchors, respectively. The anchor-associated layers are convolved to classify and align the corresponding anchors. Comparing with other methods, anchor-based detection methods are more robust in complicated scenes and their speed is invariant to object numbers. However, as indicated in [12], **the performance of anchor-based detectors drop dramatically as the objects becoming smaller**. In order to present a scale-invariant anchor-based face detector, we comprehensively analyze the reasons behind the above problem as follows:

Biased framework. The anchor-based detection frameworks tend to miss small and medium faces. Firstly, the stride size of the lowest anchor-associated layer is too large (*e.g.*, 8 pixels in [26] and 16 pixels in [38]), therefore small and medium faces have been highly squeezed on these layers and have few features for detection, see Fig. 1(a). Secondly, small face, anchor scale and receptive field are mutual mismatch: anchor scale mismatches receptive field and both are too large to fit small face, see Fig. 1(b). To address the above problems, we propose a scale-equitable face detection framework. We tile anchors on a wide range of layers whose stride size vary from 4 to 128 pixels, which guarantees that various scales of faces have enough features for detection. Besides, we design anchors with scales from 16 to 512 pixels over different layers according to the effective receptive field [29] and a new equal-proportion interval principle, which ensures that anchors at different layers match their corresponding effective receptive field and different scales of anchors evenly distribute on the image.

Anchor matching strategy. In the anchor-based detection frameworks, anchor scales are discrete (*i.e.*, $16, 32, 64, 128, 256, 512$ in our method) but face scale is continuous. Consequently, those faces whose scale distribute away from anchor scales can not match enough anchors, such as tiny and outer face in Fig. 1(c), leading to their low recall rate. To improve the recall rate of these ignored faces, we propose a scale compensation anchor matching strategy with two stages. The first stage follows current anchor matching method but adjusts a more reasonable threshold. The second stage ensures that every scale of faces match enough anchors through scale compensation.

Background from small anchors. To detect small faces well, plenty of small anchors have to be densely tiled on the image. As illustrated in Fig. 1(d), these small anchors lead to a sharp increase in the number of negative anchors on the background, bringing about many false positive faces. For example, in our scale-equitable framework, over 75% of negative anchors come from the lowest conv3_3 layer, which is used to detect small faces. In this paper, we propose a max-out background label for the lowest detection layer to reduce the false positive rate of small faces.

For clarity, the main contributions of this paper can be summarized as:

- Proposing a scale-equitable face detection framework with a wide range of anchor-associated layers and a series of reasonable anchor scales so as to handle different scales of faces well.
- Presenting a scale compensation anchor matching strategy to improve the recall rate of small faces.
- Introducing a max-out background label to reduce the high false positive rate of small faces.
- Achieving state-of-the-art results on AFW, PASCAL face, FDDB and WIDER FACE with real-time speed.

2. Related work

Face detection has attracted extensive research attention in past decades. The milestone work of Viola-Jones [48] uses Haar feature and AdaBoost to train a cascade of face/non-face classifiers that achieves a good accuracy with real-time efficiency. After that, lots of works have focused on improving the performance with more sophisticated hand-crafted features [25, 28, 53, 60] and more powerful classifiers [2, 33]. Besides the cascade structure, [30, 51, 63] introduce deformable part models (DPM) into face detection tasks and achieve remarkable performance. However, these methods highly depend on the robustness of hand-crafted features and optimize each component separately, making face detection pipeline sub-optimal.

Recent years have witnessed the advance of CNN-based face detectors. CascadeCNN [21] develops a cascade architecture built on CNNs with powerful discriminative capability and high performance. Qin et al. [34] proposes to jointly train CascadeCNN to realize end-to-end optimization. Faceness [55] trains a series of CNNs for facial attribute recognition to detect partially occluded faces. MTCNN [58] proposes to jointly solve face detection and alignment using several multi-task CNNs. UnitBox [57] introduces a new intersection-over-union loss function.

Additionally, face detection has inherited some achievements from generic object detection tasks. Jiang et al. [15] applies Faster R-CNN in face detection and achieves promising results. CMS-RCNN [59] uses Faster R-CNN in face detection with body contextual information. Convnet [24] integrates CNN with 3D face model in an end-to-end multi-task learning framework. Wan et al. [49] combines Faster R-CNN with hard negative mining and achieves significant boosts in face detection performance. STN [3] proposes a new supervised transformer network and a ROI convolution with RPN for face detection. Sun et al. [43] presents several effective strategies to improve Faster RCNN for resolving face detection tasks. In this paper, inspired by the RPN in Faster RCNN [38] and the multi-scale mechanism in SSD [26], we develop a state-of-the-art face detector with real-time speed.

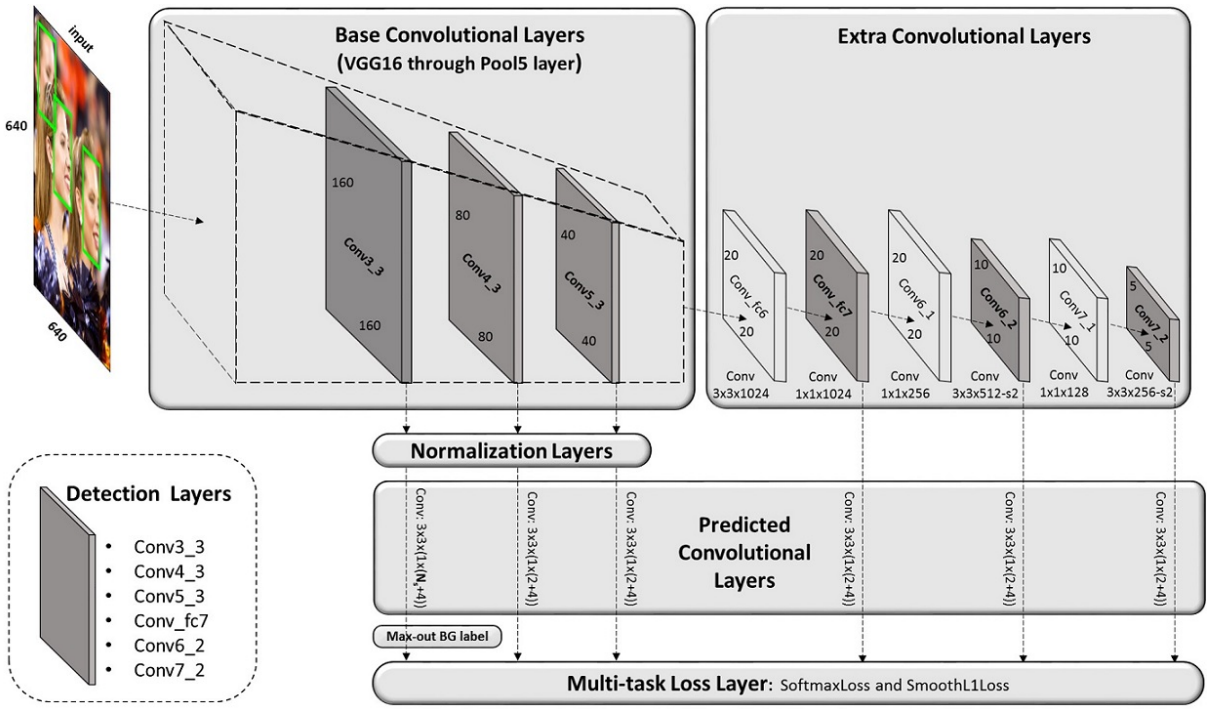


Figure 2. Architecture of Single Shot Scale-invariant Face Detector (S^{3}FD). It consists of **Base Convolutional Layers**, **Extra Convolutional Layers**, **Detection Convolutional Layers**, **Normalization Layers**, **Predicted Convolutional Layers** and **Multi-task Loss Layer**.

3. Single shot scale-invariant face detector

This section introduces our single shot scale-invariant face detector, including the scale-equitable framework (Sec. 3.1), the scale compensation anchor matching strategy (Sec. 3.2), the max-out background label (Sec. 3.3) and the associated training methodology (Sec. 3.4).

3.1. Scale-equitable framework

Our scale-equitable framework is based on the anchor-based detection framework, such as RPN [38] and SSD [26]. Despite its great achievement, the main drawback of the framework is that the performance drops dramatically as the face becomes smaller [12]. To improve the robustness to face scales, we develop a network architecture with a wide range of anchor-associated layers, whose stride size gradually double from 4 to 128 pixels. Hence, our architecture ensures that different scales of faces have adequate features for detection at corresponding anchor-associated layers. After determining the location of anchors, we design the scales of anchors from 16 to 512 pixels based on the effective receptive field and our equal-proportion interval principle. The former guarantees that each scale of anchors matches the corresponding effective receptive field well, and the latter makes different scales of anchors have the same density on the image.

Constructing architecture. Our architecture (see Fig.2) is based on the VGG16 [42] network (truncated before any classification layers) with some auxiliary structures:

- *Base Convolutional Layers*: We keep layers of VGG16 from conv1_1 to pool5, and remove all the other layers.
- *Extra Convolutional Layers*: We convert fc6 and fc7 of VGG16 to convolutional layers by subsampling their parameters [4], then add extra convolutional layers behind them. These layers decrease in size progressively and form the multi-scale feature maps.
- *Detection Convolutional Layers*: We select **conv3_3, conv4_3, conv5_3, conv_fc7, conv6_2** and **conv7_2** as the detection layers, which are associated with different scales of anchor to predict detections.
- *Normalization Layers*: Comparing to other detection layers, conv3_3, conv4_3 and conv5_3 have different feature scales. Hence we use L2 normalization [27] to rescale their norm to 10, 8 and 5 respectively, then learn the scale during the back propagation.
- *Predicted Convolutional Layers*: Each detection layer is followed by a $p\times3\times3\times q$ convolutional layer, where p and q are the channel number of input and output, and 3×3 is the kernel size. For each anchor, we predict 4 offsets relative to its coordinates and N_s scores for classification, where $N_s = N_m + 1$ (N_m is the max-out background label) for conv3_3 detection layer and $N_s = 2$ for other detection layers.
- *Multi-task Loss Layer*: We use softmax loss for classification and smooth L1 loss for regression.

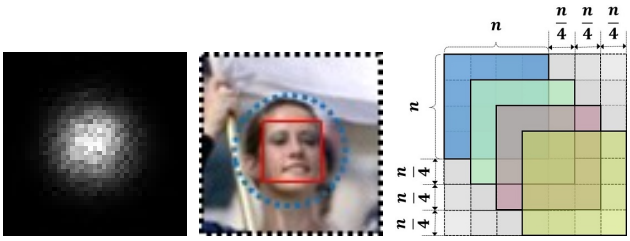
Detection Layer	Stride	Anchor	RF
conv3_3	4	16	48
conv4_3	8	32	108
conv5_3	16	64	228
conv_fc7	32	128	340
conv6_2	64	256	468
conv7_2	128	512	724

Table 1. The stride size, anchor scale and receptive field (RF) of the six detection layers. The receptive field here is related to 3×3 units on the detection layer, since it is followed by a 3×3 predicted convolutional layer to predict detections.

Designing scales for anchors. Each of the six detection layers is associated with a specific scale anchor (*i.e.*, the third column in Tab. 1) to detect corresponding scale faces. Our anchors are 1:1 aspect ratio (*i.e.*, square anchor), because the bounding box of face is approximately square. As listed in the second and fourth column of Tab. 1, the stride size and the receptive field of each detection layer are fixed, which are two base points when we design the anchor scales:

- *Effective receptive field*: As pointed out in [29], a unit in the CNN has two types of receptive fields. One is the *theoretical receptive field*, which indicates the input region that can theoretically affect the value of this unit. However, not every pixel in the theoretical receptive field contributes equally to the final output. In general, center pixels have much larger impacts than outer pixels, as shown in Fig. 3(a). In other words, only a fraction of the area has effective influence on the output value, which is another type of receptive field, named the *effective receptive field*. According to this theory, the anchor should be significantly smaller than theoretical receptive field in order to match the effective receptive field (see the specific example in Fig. 3(b)).
- *Equal-proportion interval principle*: The stride size of a detection layer determines the interval of its anchor on the input image. For example, the stride size of conv3_3 is 4 pixels and its anchor is 16×16, indicating that there is a 16×16 anchor for every 4 pixels on the input image. As shown in the second and third column in Tab. 1, the scales of our anchors are 4 times its interval. We call it equal-proportion interval principle (illustrated in Fig. 3(c)), which guarantees that different scales of anchor have the same density on the image, so that various scales face can approximately match the same number of anchors.

Benefits from the scale-equitable framework, our face detector can handle various scales of faces better, especially for small faces.



(a) (b) (c)

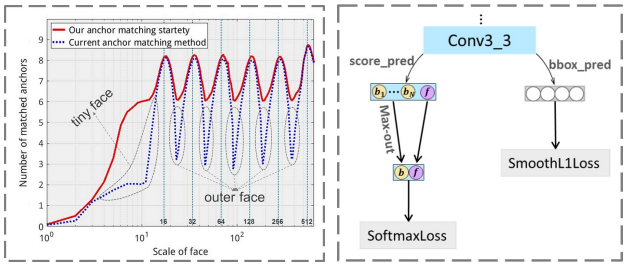
Figure 3. **(a) Effective receptive field:** The whole black box is the theoretical receptive field (TRF) and the white point cloud with Gaussian distribution is the effective receptive field (ERF). ERF only occupies a fraction of TRF. The figure is from [29]. **(b) A specific example**: In our framework, conv3_3's TRF is 48×48 (the black dotted box) and ERF is the blue dotted circle estimated by (a). Its anchor is 16×16 (the red solid line box), which is much smaller than TRF but matches ERF. **(c) Equal-proportion interval principle:** Assuming n is the anchor scale, so $n/4$ is the interval of this scale anchor. $n/4$ also corresponds to the stride size of the layer associated with this anchor. Best viewed in color.

3.2. Scale compensation anchor matching strategy

During training, we need to determine which anchors correspond to a face bounding box. Current anchor matching method firstly matches each face to the anchors with the best jaccard overlap [5] and then matches anchors to any face with jaccard overlap higher than a threshold (usually 0.5). However, **anchor scales are discrete while face scales are continuous**, these faces whose scales distribute away from anchor scales can not match enough anchors, leading to their low recall rate. As shown in Fig. 1(c), we count the average number of matched anchors for different scales of faces. There are two observations: 1) the average number of matched anchors is about 3 which is not enough to recall faces with high scores; 2) the number of matched anchors is highly related to the anchor scales. The faces away from anchor scales tend to be ignored, leading to their low recall rate. To solve these problems, we propose a scale compensation anchor matching strategy with two stages:

- *Stage one*: We follow current anchor matching method but decrease threshold from 0.5 to 0.35 in order to increase the average number of matched anchors.
- *Stage Two*: After stage one, some faces still do not match enough anchors, such as tiny and outer faces marked with the gray dotted curve in Fig. 4(a). We deal with each of these faces as follow: firstly picking out anchors whose jaccard overlap with this face are higher than 0.1, then sorting them to select top-N as matched anchors of this face. We set N as the average number from stage one.

As shown in Fig. 4(a), our anchor matching strategy greatly increases the matched anchors of tiny and outer faces, which notably improve the recall rate of these faces.



(a) (b)

Figure 4. (a) The matched number for different scales of faces are compared between current anchor matching method and our scale compensation anchor matching strategy. (b) The illustration of the max-out background label.

3.3. Max-out background label

Anchor-based face detection methods can be regarded as a binary classification problem, which determines if an anchor is face or background. In our method, it is an extremely unbalanced binary classification problem: according to our statistical results, over 99.8% of the pre-set anchors belong to negative anchors (*i.e.*, background) and only a few of anchors are positive anchors (*i.e.*, face). This extreme imbalance is mainly caused by the detection of small faces. Specifically, we have to densely tile plenty of small anchors on the image to detect small faces, which causes a sharp increase in the number of negative anchors. For example, as listed in Tab. 2, a 640×640 image has totally $34,125$ anchors, while about 75% of them come from conv3_3 detection layer which is associated with the smallest anchor (16×16). These smallest anchors contribute most to the false positive faces. As a result, improving the detection rate of small faces by tiling small anchors will inevitably lead to the high false positive rate of small faces.

Position	Scale	Number	Percentage (%)
conv3_3	16	25600	**75.02**
conv4_3	32	6400	18.76
conv5_3	64	1600	4.69
conv_fc7	128	400	1.17
conv6_2	256	100	0.29
conv7_2	512	25	0.07

Table 2. Detailed information about anchors in a 640×640 image.

To address this issue, we propose to apply a more sophisticated classification strategy on the lowest layer to handle the complicated background from small anchors. We apply the max-out background label for the conv3_3 detection layer. For each of the smallest anchors, we predict N_m scores for background label and then choose the highest as its final score, as illustrated in Fig. 4(b). Max-out operation integrates some local optimal solutions into our S^{3}FD model so as to reduce the false positive rate of small faces.

3.4. Training

In this subsection, we introduce the training dataset, data augmentation, loss function, hard negative mining and other implementation details.

Training dataset and data augmentation. Our model is trained on $12,880$ images of the WIDER FACE training set with the following data augmentation strategies:

- Color distort: Applying some photo-metric distortions similar to [11].
- Random crop: We apply a zoom in operation to generate larger faces since there are too many small faces in the WIDER FACE training set. Specifically, each image is randomly selected from five square patches, which are randomly cropped from the original image: one is the biggest square patch, and the size of the other four square patches range between $[0.3, 1]$ of the short size of the original image. We keep the overlapped part of the face box if its center is in the sampled patch.
- Horizontal flip: After random cropping, the selected square patch is resized to 640×640 and horizontally flipped with probability of 0.5.

Loss function. We employ the multi-task loss defined in RPN [**?**] to jointly optimize model parameters:

$$L(\{p_i\},\{t_i\}) = \frac{\lambda}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*),$$

where i is the index of an anchor and p_i is the predicted probability that anchor i is a face. The ground-truth label p_i^* is 1 if the anchor is positive, 0 otherwise. As defined in [**?**], t_i is a vector representing the 4 parameterized coordinates of the predicted bounding box, and t_i^* is that of the ground-truth box associated with a positive anchor. The classification loss $L_{cls}(p_i, p_i^*)$ is softmax loss over two classes (face vs. background). The regression loss $L_{reg}(t_i, t_i^*)$ is the smooth L1 loss defined in [8] and $p_i^* L_{reg}$ means the regression loss is activated only for positive anchors and disabled otherwise. The two terms are normalized by N_{cls} and N_{reg}, and weighted by a balancing parameter λ. In our implementation, the *cls* term is normalized by the number of positive and negative anchors, and the *reg* term is normalized by the number of positive anchors. Because of the imbalance between the number of positive and negative anchors, λ is used to balance these two loss terms.

Hard negative mining. After anchor matching step, most of the anchors are negative, which introduces a significant imbalance between the positive and negative training examples. For faster optimization and stable training, instead of using all or randomly select some negative samples, we sort them by the loss values and pick the top ones so that

the ratio between the negatives and positives is at most 3:1. With hard negative mining, we set above background label $N_m = 3$, and $\lambda = 4$ to balance the loss of classification and regression.

Other implementation details. As for the parameter initialization, the base convolutional layers have the same architecture as VGG16 and their parameters are initialized from the pre-trained [39] VGG16. The parameters of conv_fc6 and conv_fc7 are initialized by subsampling parameters from fc6 and fc7 of VGG16 and the other additonal layers are randomly initialized with the "xavier" method [9]. We fine-tune the resulting model using SGD with 0.9 momentum, 0.0005 weight decay and batch size 32. The maximum number of iterations is $120k$ and we use 10^{-3} learning rate for the first $80k$ iterations, then continue training for $20k$ iterations with 10^{-4} and 10^{-5}. Our method is implemented in Caffe [14] and the code is available at https://github.com/sfzhang15/SFD.

4. Experiments

In this section, we firstly analyze the effectiveness of our scale-equitable framework, scale compensation anchor matching strategy and max-out background label, then evaluate the final model on common face detection benchmarks, finally introduce the inference time.

4.1. Model analysis

We analyze our model on the WIDER FACE validation set by extensive experiments. The WIDER FACE validation set has easy, medium and hard subsets, which roughly correspond to large, medium and small faces, respectively. Hence it is suitable to evaluate our model.

Baseline. To evaluate our contributions, we carry out comparative experiments with our baselines. Our S^{3}FD is inspired by RPN [38] and SSD [26], so we directly use them to train two face detectors as the baselines, marked as RPN-face and SSD-face, respectively. Different from [38], the RPN-face tiles six scales of the square anchor (16, 32, 64, 128, 256, 512) on the conv5_3 layer of VGG16 to make the comparison more substantial. The SSD-face inherits the architecture and anchor-setting of SSD. The remainder is set as the same with our S^{3}FD.

Ablative Setting. To understand S^{3}FD better, we conduct ablation experiments to examine how each proposed component affects the final performance. We evaluate the performance of our method under three different settings: (i) *S^{3}FD(F)*: it only uses the scale-equitable framework (*i.e.*, constructed architecture and designed anchors) and ablates another two components; (ii) *S^{3}FD(F+S)*: it applies the scale-equitable framework and the scale compensation anchor matching strategy; (iii) *S^{3}FD(F+S+M)*: it is our complete model, consisting of the scale-equitable framework, the scale compensation anchor matching strategy and the max-out background label.

mAP(%) Subsets / Methods	Easy	Medium	Hard
RPN-face	91.0	88.2	73.7
SSD-face	92.1	89.5	71.6
S^{3}FD(F)	92.6	91.6	82.3
S^{3}FD(F+S)	93.5	92.0	84.5
S^{3}FD(F+S+M)	**93.7**	**92.4**	**85.2**

Table 3. The comparative and ablative results of our model on WIDER FACE validation subset. The precision-recall curves of these methods are in the supplementary materials.

From the results listed in Tab. 3, some promising conclusions can be summed up as follows:

Scale-equitable framework is crucial. Comparing with S^{3}FD(F), the only difference with RPN-face and SSD-face are their framework. RPN-face has the same choice of anchors as ours but only tiles on the last convolutional layer of VGG16. Not only its stride size (16 pixels) is too large for small faces, but also different scales of anchors have the same receptive field. SSD-face tiles anchors over several convolutional layers, while its smallest stride size (8 pixels) and smallest anchors are still slightly large for small faces. Besides, its anchors do not match the effective receptive field. The result of S^{3}FD(F) in Tab. 3 shows that our framework greatly outperforms SSD-face and RPN-face, especially on the hard subsets (rising by 8.6%), which mainly consists of small faces. Comparing the results between different subsets, our S^{3}FD(F) handles various scales of faces well, and deteriorates slightly as the faces become smaller, which demonstrates the robustness to face scales.

Scale compensation anchor matching strategy is better. The comparison between the third and fourth rows in Tab. 3 indicates that our scale compensation anchor matching strategy effectively improves the performance, especially for small faces. The mAP is increased by 0.9%, 0.4%, 2.2% on easy, medium and hard subset, respectively. The increases mainly come from the higher recall rate of various scales of faces, especially for those faces that are ignored by the current anchor matching method.

Max-out background label is promising. The last contribution of S^{3}FD is the max-out background label. It deals with the massive small negative anchors (*i.e.*, background) from the conv3_3 detection layer which is designed to detect small faces. As reported in Tab. 3, the improvements on easy, medium and hard subsets are 0.2%, 0.4%, 0.7%, respectively. It demonstrates that the effectiveness of the max-out background label is positively related to the difficulty of the input image. Since the harder images will generate the more difficult small backgrounds.

4.2. Evaluation on benchmark

We evaluate our S^{3}FD method on all the common face detection benchmarks, including Annotated Faces in the Wild (AFW)[63], PASCAL Face[52], Face Detection Data Set and Benchmark (FDDB)[13] and WIDER FACE [56]. Due to the limited space, some qualitative results on these dataset will be shown in the supplementary materials.

AFW dataset. It contains 205 images with 473 labeled faces. We evaluate our model against the well-known works [3, 25, 30, 41, 52, 55, 63] and commercial face detectors (*e.g.*, Face.com, Face++ and Picasa). As illustrated in Fig.5, our S^{3}FD outperforms all others by a large margin.

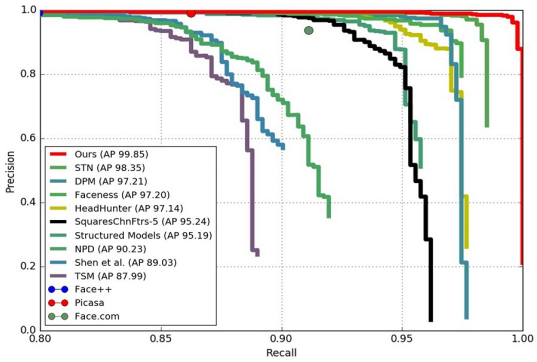


Figure 5. Precision-recall curves on AFW dataset.

PASCAL face dataset. It has 1,335 labeled faces in 851 images with large face appearance and pose variations. It is collected from PASCAL person layout test subset. Fig.6 shows the precision-recall curves on this dataset, our method significantly outperforms all other methods [3, 16, 30, 52, 55, 63] and commercial face detectors (*e.g.*, Sky-Biometry, Face++ and Picasa).

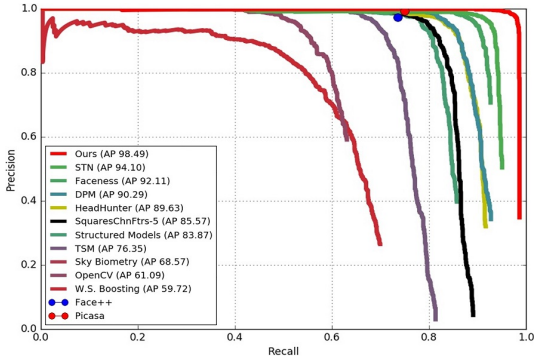
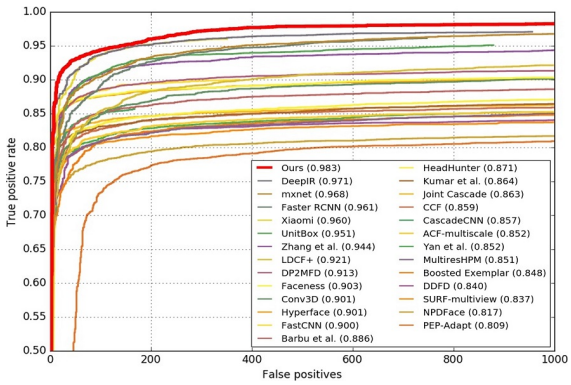
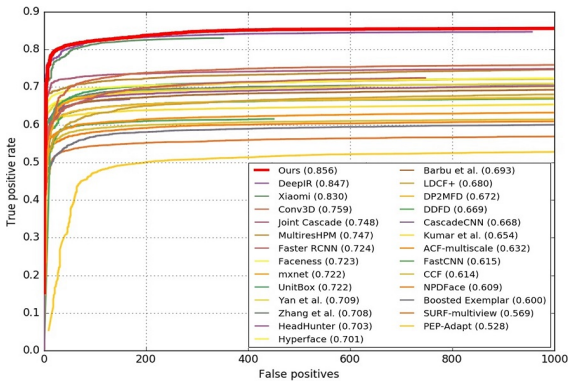


Figure 6. Precision-recall curves on PASCAL face dataset.

FDDB dataset. It contains 5,171 faces in 2,845 images. There are two problems for evaluation: 1) FDDB adopts the bounding ellipse while our S^{3}FD outputs rectangle bounding box. This inconsistency has a great impact on the continuous score, so we train an elliptical regressor to transform our predicted bounding boxes to bounding ellipses. 2) FDDB has lots of unlabelled faces, which results in many false positive faces with high scores. Hence, we manually review the results and add 238 unlabelled faces (annotations will be released later and some examples are shown in the supplementary materials). Finally, we evaluate our face detector on FDDB against the state-of-the-art methods [1, 6, 7, 15, 18, 19, 20, 22, 24, 25, 31, 35, 36, 43, 47, 49, 55, 57, 58]. The results are shown in Fig. 7(a) and Fig.7(b). Our S^{3}FD achieves the state-of-the-art performance and outperforms all others by a large margin on discontinuous and continuous ROC curves. These results indicate that our S^{3}FD can robustly detect unconstrained faces.



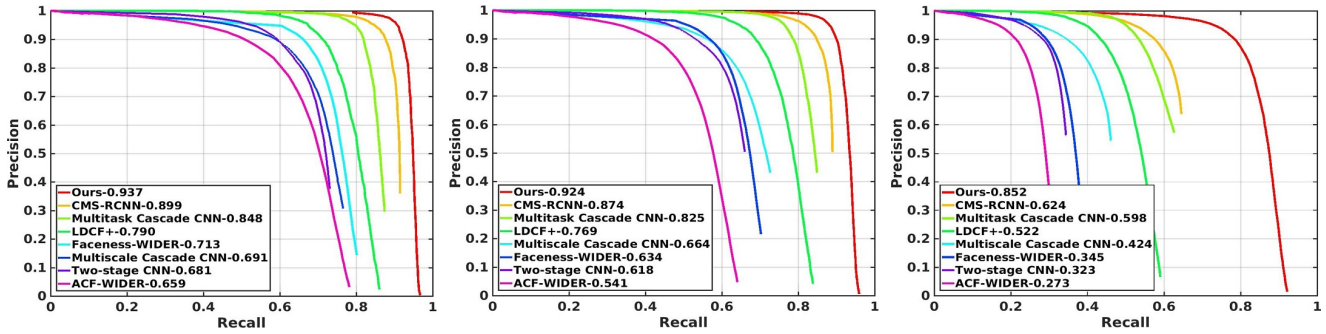
(a) Discontinuous ROC curves



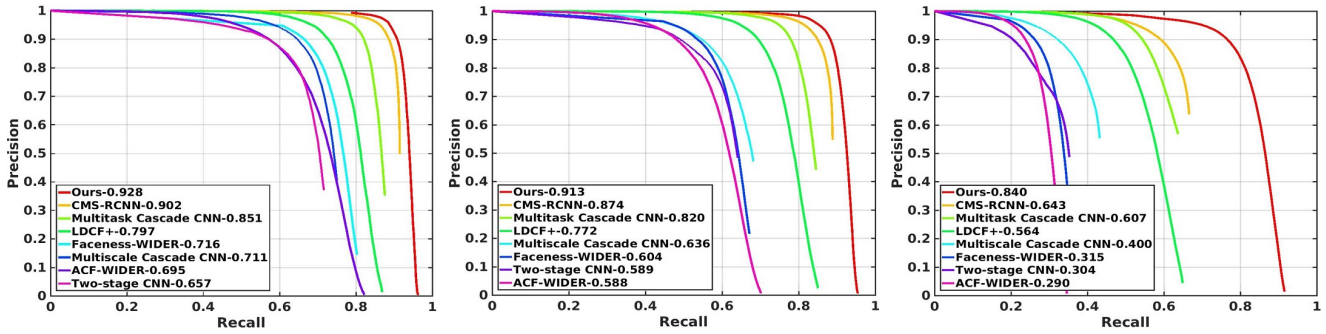
(b) Continuous ROC curves

Figure 7. Evaluation on the FDDB dataset.

WIDER FACE dataset. It has 32,203 images and labels 393,703 faces with a high degree of variability in scale, pose and occlusion. The database is split into training (40%), validation (10%) and testing (50%) set. Besides, the images are divided into three levels (Easy, Medium and Hard subset) according to the difficulties of the detection. The images and annotations of training and validation set are available online, while the annotations of testing set are not released and the results are sent to the database server for receiving the precision-recall curves. Our S^{3}FD is trained only on the training set and tested on both validation and testing set against recent face detection methods [31, 53, 55, 56, 58, 59]. The precision-recall curves and



(a) Val: Easy (b) Val: Medium (c) Val: Hard



(d) Test: Easy (e) Test: Medium (f) Test: Hard

Figure 8. Precision-recall curves on WIDER FACE validation and test sets.[1]

mAP values are shown in Fig. 8. Our model outperforms others by a large margin across the three subsets, especially on the hard subset which mainly contains small faces. It achieves the best average precision in all level faces, i.e. 0.937 (Easy), 0.924 (Medium) and 0.852 (Hard) for validation set, and 0.928 (Easy), 0.913 (Medium) and 0.840 (Hard) for testing set.[1] These results not only demonstrate the effectiveness of the proposed method but also strongly show the superiority of the proposed model in detecting small and hard faces.

4.3. Inference time

During inference, our method outputs a large number of boxes (*e.g.*, $25,600$ boxes for a VGA-resolution image). To speed up the inference time, we first filter out most boxes by a confidence threshold of 0.05 and keep the top 400 boxes before applying NMS, then we perform NMS with jaccard overlap of 0.3 and keep the top 200 boxes. We measure the speed using Titan X (Pascal) and cuDNN v5.1 with Intel Xeon E5-2683v3@2.00GHz. For the VGA-resolution image with batch size 1 using a single GPU, our face detector can run at 36 FPS and achieve the real-time speed. Besides, about 80% of the forward time is spent on the VGG16 network, hence using a faster base network could further improve the speed.

[1]Our latest results on WIDER FACE are shown in Fig. 16

5. Conclusion

This paper introduces a novel face detector by solving the common problem of anchor-based detection methods whose performance decrease sharply as the objects becoming smaller. We analyze the reasons behind this problem, and propose a scale-equitable framework with a wide range of anchor-associated layers and a series of reasonable anchor scales in order to well handle different scales of faces. Besides, we propose the scale compensation anchor matching strategy to improve the recall rate of small faces, and the max-out background label to reduce the false positive rate of small faces. The experiments demonstrate that our three contributions lead S^{3}FD to the state-of-the-art performance on all the common face detection benchmarks, especially for small faces. In our future work, we intend to further improve the classification strategy of background patches. We believe that explicitly dividing the background class into some sub-categories is worthy of further study.

Acknowledgments

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

References

[1] A. Barbu, N. Lay, and G. Gramajo. Face detection with a 3d model. *arXiv:1404.3596*, 2014. 7

[2] S. C. Brubaker, J. Wu, J. Sun, M. D. Mullin, and J. M. Rehg. On the design of cascades of boosted ensembles for face detection. *IJCV*, 77(1-3):65–86, 2008. 2

[3] D. Chen, G. Hua, F. Wen, and J. Sun. Supervised transformer network for efficient face detection. In *ECCV*, pages 122–138, 2016. 1, 2, 7

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 3

[5] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, pages 2155–2162, 2014. 4

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

[7] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Detecting and localizing occluded faces. *arXiv:1506.08347*, 2015. 7

[8] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 1, 5

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010. 6

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[11] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013. 5

[12] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv:1611.10012*, 2016. 2, 3

[13] V. Jain and E. G. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. *UMass Amherst Technical Report*, 2010. 7

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMMM*, 2014. 6

[15] H. Jiang and E. Learned-Miller. Face detection with the faster r-cnn. *arXiv:1606.03473*, 2016. 1, 2, 7

[16] Z. Kalal, J. Matas, and K. Mikolajczyk. Weighted sampling for large-scale boosting. In *BMVC*, pages 1–10, 2008. 7

[17] M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, pages 1–8, 2008. 1

[18] V. Kumar, A. Namboodiri, and C. Jawahar. Visual phrases for exemplar face detection. In *ICCV*, pages 1994–2002, 2015. 7

[19] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic part model for unsupervised face detector adaptation. In *ICCV*, pages 793–800, 2013. 7

[20] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua. Efficient boosted exemplar-based face detection. In *CVPR*, pages 1843–1850, 2014. 7

[21] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *CVPR*, pages 5325–5334, 2015. 1, 2

[22] J. Li and Y. Zhang. Learning surf cascade for fast and accurate object detection. In *CVPR*, pages 3468–3475, 2013. 7

[23] Y. Li, K. He, J. Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016. 1

[24] Y. Li, B. Sun, T. Wu, and Y. Wang. Face detection with end-to-end integration of a convnet and a 3d model. In *ECCV*, pages 420–436, 2016. 2, 7

[25] S. Liao, A. K. Jain, and S. Z. Li. A fast and accurate unconstrained face detector. *PAMI*, 38:211–223, 2016. 2, 7

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 1, 2, 3, 6

[27] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. In *ICLR workshop*, 2016. 3

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[29] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *NIPS*, pages 4898–4906, 2016. 2, 4

[30] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *ECCV*, pages 720–735, 2014. 2, 7

[31] E. Ohn-Bar and M. M. Trivedi. To boost or not to boost? on the limits of boosted trees for object detection. In *ICPR*, 2016. 1, 7, 8

[32] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 1

[33] M.-T. Pham and T.-J. Cham. Fast training and selection of haar features using statistics in boosting-based face detection. In *ICCV*, pages 1–7, 2007. 2

[34] H. Qin, J. Yan, X. Li, and X. Hu. Joint training of cascaded cnn for face detection. In *CVPR*, pages 3456–3465, 2016. 2

[35] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. In *BTAS*, pages 1–8, 2015. 7

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

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[38] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 2, 3, 6

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6

[40] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 1

[41] X. Shen, Z. Lin, J. Brandt, and Y. Wu. Detecting and aligning faces by image retrieval. *CVPR*, pages 3460–3467, 2013. 7

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 3

[43] X. Sun, P. Wu, and S. C. Hoi. Face detection using deep learning: An improved faster rcnn approach. *arXiv:1701.08289*, 2017. 1, 2, 7

[44] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, pages 1891–1898, 2014. 1

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 1

[46] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014. 1

[47] D. Triantafyllidou and A. Tefas. A fast deep convolutional neural network for face detection in big visual data. In *INNS Conference on Big Data*, pages 61–70, 2016. 7

[48] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 1, 2

[49] S. Wan, Z. Chen, T. Zhang, B. Zhang, and K.-k. Wong. Bootstrapping face detection with hard negative examples. *arXiv:1608.02236*, 2016. 1, 2, 7

[50] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, pages 532–539, 2013. 1

[51] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *CVPR*, pages 2497–2504, 2014. 2

[52] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. Face detection by structural models. *Image and Vision Computing*, 32(10):790–799, 2014. 7

[53] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In *IJCB*, pages 1–8, 2014. 2, 8

[54] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. In *ICCV*, pages 82–90, 2015. 1

[55] S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, pages 3676–3684, 2015. 1, 2, 7, 8

[56] S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *CVPR*, pages 5525–5533, 2016. 7, 8

[57] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *ACMMM*, pages 516–520, 2016. 2, 7

[58] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 23(10):1499–1503, 2016. 1, 2, 7, 8

[59] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. *arXiv:1606.05413*, 2016. 1, 2, 8

[60] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, volume 2, pages 1491–1498, 2006. 2

[61] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li. Face alignment across large poses: A 3d solution. In *CVPR*, pages 146–155, 2016. 1

[62] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li. High-fidelity pose and expression normalization for face recognition in the wild. In *CVPR*, pages 787–796, 2015. 1

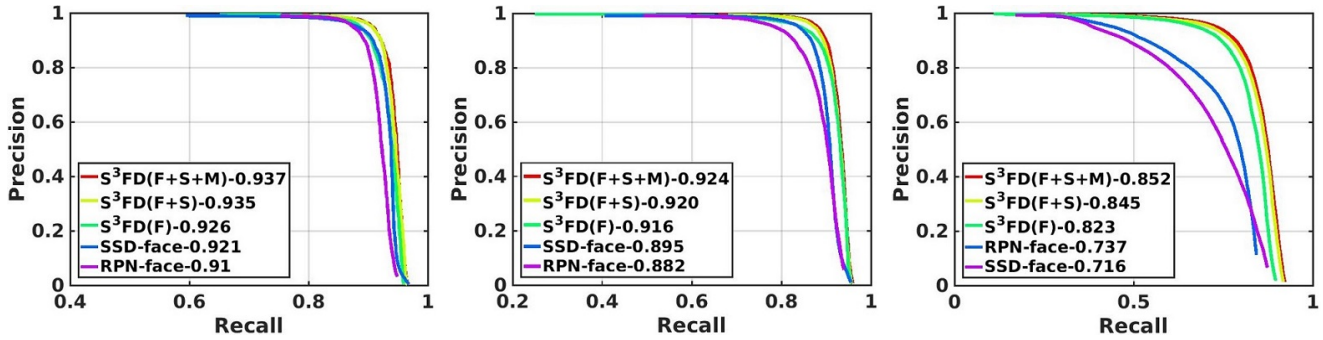
[63] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012. 2, 7

S^{3}FD: Single Shot Scale-invariant Face Detector
-Supplementary Material-

Shifeng Zhang Xiangyu Zhu Zhen Lei Hailin Shi Xiaobo Wang Stan Z. Li
CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{shifeng.zhang,xiangyu.zhu,zlei,hailin.shi,xiaobo.wang,szli}@nlpr.ia.ac.cn

A. Precision-recall curves

In our submitted paper, Tab. 3 in subsection 4.1 only provides the mAP of RPN-face, SSD-face, S^{3}FD(F), S^{3}FD(F+S) and S^{3}FD(F+S+M). Their precision-recall curves on the WIDER FACE validation set are shown in Fig. 9 for details.



(a) Easy (b) Medium (c) Hard

Figure 9. Precision-recall curves on WIDER FACE validation set.

B. Qualitative results

In this section, we demonstrate some qualitative results on common face detection benchmarks, including AFW (Fig. 10), PASCAL face (Fig. 11), FDDB (Fig. 12) and WIDER FACE (Fig. 13). Besides, another impressive result is shown in Fig. 14.

Figure 10. Qualitative results on AFW. The faces in these results have a high degree of variability in scale, pose and occlusion. Our S^{3}FD is able to detect these faces with a high confidence, especially for small faces. Please zoom in to see some small detections.

Figure 11. Qualitative results on PASCAL face. Most faces in these results are small faces, because the image in PASCAL face has a low resolution. Our S^{3}FD is able to handle small faces well. Please zoom in to see some small detections.

Figure 12. Qualitative results on FDDB. These results indicate that our S^{3}FD is robust to large appearance, heavy occlusion, scale variance and heavy blur. Please zoom in to see some small detections.

(a) Scale attribute. Our S^{3}FD is able to detect faces at a continuous range of scales.

(b) Our S^{3}FD is robust to pose, occlusion, expression, makeup, illumination and blur.

Figure 13. Qualitative results on WIDER FACE. We visualize some examples for each attribute. Please zoom in to see small detections.

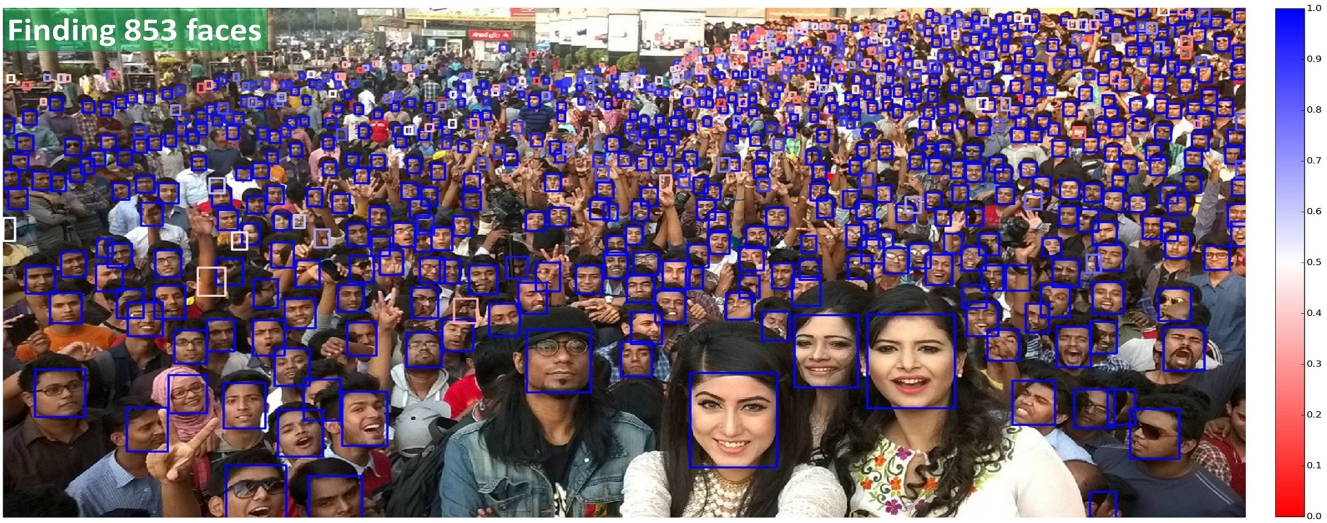


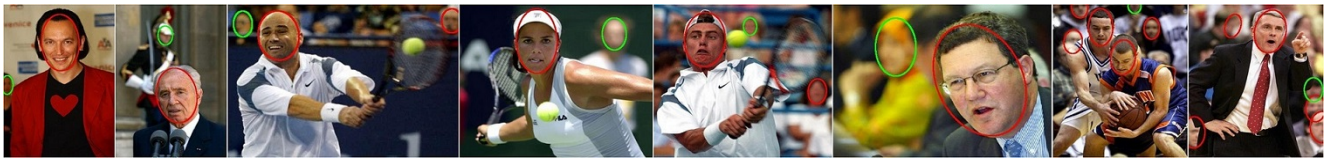
Figure 14. Another qualitative result. Our S^{3}FD can find 853 faces out of the reportedly 1000 present in the above image. Detector confidence is given by the colorbar on the right. Please zoom in to see some small detections.

C. Examples of manually labelled faces on FDDB

We add 238 unlabelled faces whose height and width are more than 20 pixels. Some examples are shown in Fig. 15.

(a) Profile faces

(b) Occluded faces



(c) Blur faces

(d) Statue faces

(e) Miscellaneous faces

Figure 15. Examples of our manually labelled faces on the FDDB dataset. Red ellipses are the faces that FDDB has already labelled, green ellipses are the newly added faces.

D. Ablative analysis of each detection layers

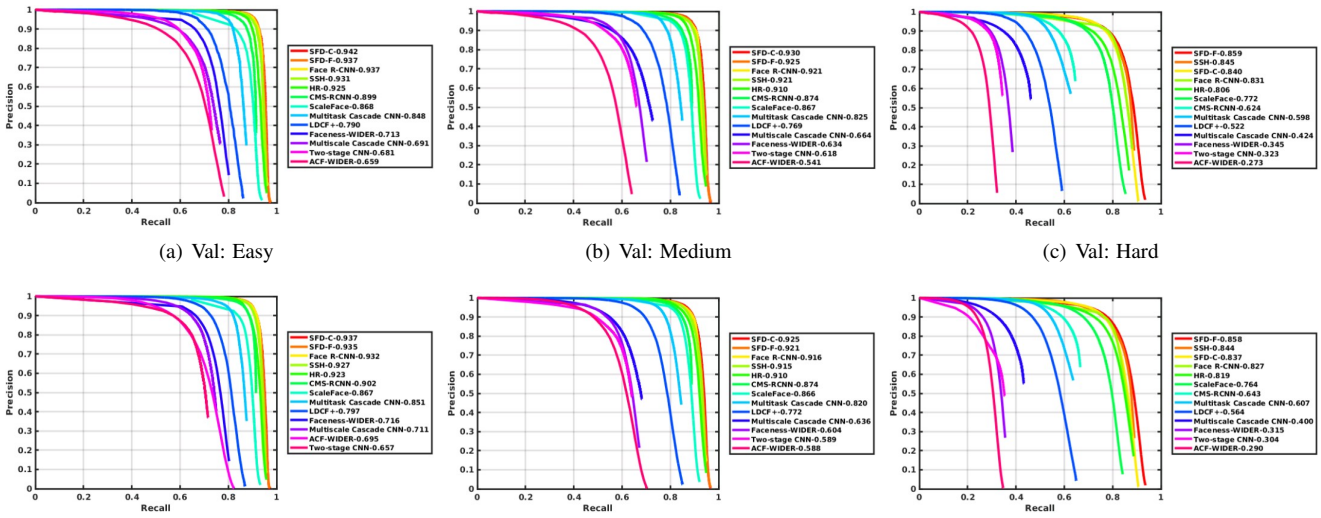
To examine the contribution of each detection layers on the mAP performance, we progressively remove the detection layers to test their contribution on the WIDER FACE Val set. The detailed experiment results are listed in Tab. 4. After removing Conv3_3 layer, the mAP changes are +0.3%(Easy), +0.5%(Medium) and -24.7%(Hard), showing Conv3_3 is crucial to detect small faces, but tiling plenty of smallest anchors also slightly hurts medium and large face detection performance. Besides, the most contribution of Easy and Medium subset are Conv5_3 (25.8%) and Conv4_3 (20.6%), respectively.

Detection layers	Ablative analysis					
Conv3_3	×					
Conv4_3		×				
Conv5_3			×			
Conv_fc7				×		
Conv6_2					×	
Conv7_2						×
mAP changes on Easy subset (%)	+0.3	-0.6	**-25.8**	-10.2	-3.2	-1.4
mAP changes on Medium subset (%)	+0.5	**-20.6**	-12.2	-5.0	-1.5	-0.7
mAP changes on Hard subset (%)	**-24.7**	-8.7	-4.1	-1.8	-0.6	-0.2

Table 4. The ablative results of each detection layers on the WIDER FACE Val set.

E. Latest results on the WIDER FACE dataset

Fig. 16 shows the latest precision-recall (PR) curves of our S^{3}FD (*i.e.,* SFD-F and SFD-C) on WIDER FACE validation and test sets. SFD-F and SFD-C are our upgraded detectors. SFD-F further improves the detection ability of small faces and SFD-C focuses more on big and medium faces. The RP curves of SFD-F and SFD-C can be downloaded from the official website of WIDER FACE dataset[2], which plots only the RP curves of SFD-F on its figure with the legend "SFD". Our S^{3}FD achieves the best average precision on all subsets, i.e. **0.942** (Easy), **0.930** (Medium) and **0.859** (Hard) for validation set, and **0.937** (Easy), **0.925** (Medium) and **0.858** (Hard) for testing set.



(a) Val: Easy (b) Val: Medium (c) Val: Hard

(d) Test: Easy (e) Test: Medium (f) Test: Hard

Figure 16. The latest precision-recall curves on WIDER FACE validation and test sets.[3]

[2] http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/WiderFace_Results.html

[3] Note-worthily, the latest evaluation code and annotation are used to generate these PR curves, while the results of WIDER FACE reported in our above paper are generated from the previous version of evaluation code or annotation.