Group Sampling for Scale Invariant Face Detection

Xiang Ming, Fangyun Wei, Ting Zhang, Dong Chen, Nanning Zheng, *Fellow, IEEE*, and Fang Wen

Abstract—Detectors based on deep learning tend to detect multi-scale objects on a single input image for efficiency. Recent works, such as FPN and SSD, generally use feature maps from multiple layers with different spatial resolutions to detect objects at different scales, e.g., high-resolution feature maps for small objects. However, we find that objects at all scales can also be well detected with features from a single layer of the network. In this paper, we carefully examine the factors affecting detection performance across a large range of scales, and conclude that the balance of training samples, including both positive and negative ones, at different scales is the key. We propose a group sampling method which divides the anchors into several groups according to the scale, and ensure that the number of samples for each group is the same during training. Our approach using only one single layer of FPN as features is able to advance the state-of-the-arts. Comprehensive analysis and extensive experiments have been conducted to show the effectiveness of the proposed method. Moreover, we show that our approach is favorably applicable to other tasks, such as object detection on COCO dataset, and to other detection pipelines, such as YOLOv3, SSD and R-FCN. Our approach, evaluated on face detection benchmarks including FDDB and WIDER FACE datasets, achieves state-of-the-art results without bells and whistles.

Index Terms—Object detection, convolution neural network, sampling

1 Introduction

Face detection is the key step of many subsequent face related applications, such as face alignment [1], [2], [3], [4], [5], face synthesis [6], [7], [8], [9], [10], [11], [12] and face recognition [13], [14], [15], [16], [17]. Among the various factors that confront real-world face detection, extreme scale variations and small faces remain to be a big challenge.

Previous deep learning detectors detect multi-scale faces on a single feature map, e.g., Fast R-CNN [18] and Faster R-CNN [19]. They offer a good trade-off between accuracy and speed. However, these methods tend to miss faces at small scale because of the large stride size of the anchor (e.g., 16 pixels in [19]), making small faces difficult to match the appropriate anchors and thus have few positive samples during training.

To alleviate these problems arising from scale variation and small object instances, multiple solutions have been proposed, including: 1) using image pyramid for training and inference [20], [21]; 2) combining features from shallow and deep layers for prediction [22], [23], [24]; 3) using top-down and skip connections to produce a single high-level feature map with fine resolution [25], [26], [27]; 4) using multiple layers of different resolutions to predict object instances of different scales [28], [29], [30], [31], [32], [33]. All of these solutions have significantly improved the performance of detectors. Among them, adopting several layers with different resolution for prediction is the most popular one, since it achieves better performance, especially for detecting small objects.

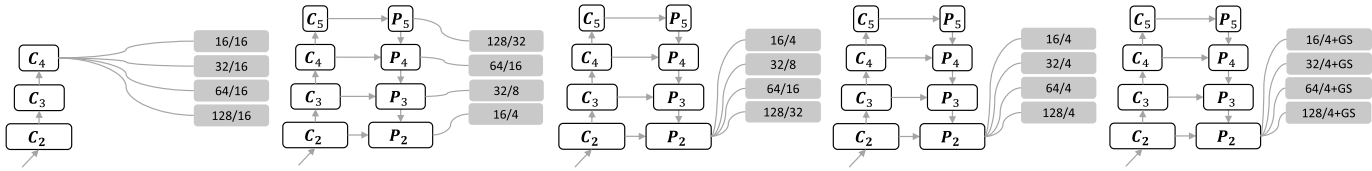
It was generally believed that the advantage of prediction over multiple layers stems from the multi-scale feature representation, which is more robust to scale variation than the feature from single layer. However, we find that multi-scale feature representation is just one of the reasons. We observe that making predictions over multiple layers will produce different number of anchors at different scale,[1] which is also the reason why pyramid features outperform single layer feature, and this factor is overlooked in the comparison experiment between pyramid features and single layer feature conducted in FPN [33]. Empirically we show that single layer predictions, if imposed with the same number of anchors as that in FPN, will achieve almost the same accuracy.

Motivated by this observation, we carefully examine the factors affecting detection performance through extensive empirical analysis and identify a key issue in existing anchor based detectors, i.e., *the number of sampled anchors at different scales are imbalanced during training*. To show this, we use two representative single shot detection architectures, the Region Proposal Network (RPN) in Faster R-CNN [19] and FPN [33], as examples. The network architectures are illustrated in Fig. 1. We calculate the number of training samples received

- *Xiang Ming is with the School of Electrical and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: xjtustu.mx@stu.xjtu.edu.cn.*
- *Fangyun Wei is with the Innovation Engineering Group, Microsoft Research Asia, Beijing 100080, China. E-mail: fawe@microsoft.com.*
- *Ting Zhang, Dong Chen, and Fang Wen are with the Visual Computing Group, Microsoft Research Asia, Beijing 100080, China. E-mail: {tinzhan, doch, fangwen}@microsoft.com.*
- *Nanning Zheng is with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China. E-mail: nnzheng@mail.xjtu.edu.cn.*

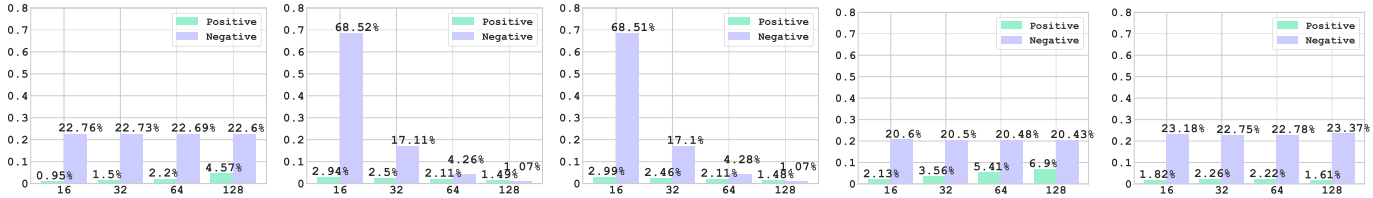
Manuscript received 22 July 2019; revised 15 June 2020; accepted 9 July 2020. Date of publication 28 July 2020; date of current version 7 Jan. 2022. (Corresponding author: Xiang Ming.) Recommended for acceptance by T. Hassner. Digital Object Identifier no. 10.1109/TPAMI.2020.3012414

1. The scale of a bounding box with size (w, h) is defined as $\sqrt{wh}$.



(a) RPN (b) FPN (c) FPN-finest-stride (d) FPN-finest (e) FPN-finest-sampling

Fig. 1. Illustration of network architectures of five different detectors used for WIDER FACE. The term '16/4' associated with the feature map denotes that anchors with scale 16 and anchor stride 4 (with respect to the input image) have been placed on that feature map. The difference between (d) FPN-finest and (e) FPN-finest-smapling which have the same network architecture, is that (e) uses the proposed group sampling in (d).



(a) RPN (b) FPN (c) FPN-finest-stride (d) FPN-finest (e) FPN-finest-sampling

Fig. 2. The distribution of positive and negative training samples at different scales on WIDER FACE training set with different network architectures. The number is normalized by the total amount of training samples.

by anchors at each scale and report them in Fig. 2a and 2b for RPN and FPN, respectively.

Conventional matching procedure for selecting positive and negative samples for training usually consists of two steps: 1) finding the best matching ground-truth bounding box for each anchor according to Intersection-over-Union (IoU); 2) associating anchors with the unmatched ground-truth boxes according to pre-defined thresholds.[2] As a result, given fixed matching thresholds, if large anchors and small anchors are associated with the same feature map with the same stride, the large ground-truth bounding boxes will be much more likely to match anchors than the small ones.

For detectors which rely on single layer feature map, e.g., RPN, as the strides of anchors for different scales are the same, the number of anchors for different scales is the same either. Since the small anchors are more difficult to match an appropriate labeled object, such detectors will have few positive samples and thus low accuracy for detecting small objects.

On the other hand, detectors using feature pyramid, e.g., FPN, adopt high-resolution feature maps for small anchors and low-resolution feature maps for large ones. Hence the number of anchors with small scale is several times more than the large ones, resulting in more negative training samples for small objects than large objects. Such trained classifiers would get higher accuracy for detecting small objects, which might be the reason why FPN based methods perform better on the WIDER FACE database [34], where small objects are nearly dominated.

We further report the number of training samples when only using the last layer of FPN, which are shown in Fig. 2c with different strides for different scales and Fig. 2d with the same stride for different scales. The distribution shown in Fig. 2c is similar to FPN, and the distribution shown in Fig. 2d is similar to RPN, though the absolute values are different. Empirically we show that imbalanced training data across scales leads to worse (better) performance for the minority (majority).

To handle this issue, we propose a group sampling method, which is simple and straightforward. The idea is to randomly sample the same number of anchors at each scale during each iteration of the training process. Thus the classifier is fed with balanced training samples at different scales. Fig. 2e shows the anchor distribution of our method, where the distribution becomes more balanced. Empirically we show that FPN-finest-sampling, which only uses the feature map on the finest level of FPN with group sampling, is able to achieve better performance than FPN.

In addition, we notice that the second stage of the Faster R-CNN [19] also suffers from sample imbalance issue. The conventional sampling procedure used for Fast R-CNN is to keep top K samples for Non Maximum Suppression (NMS). However, the proposals with small scale usually get lower score than the large ones, resulting in insufficient samples for refining those small objects. We show that our proposed method can be used here after RoIAlign [35] for different scales and thus further improve the detection accuracy.

In summary, our main contribution lies in three aspects, (1) we observe that anchor distribution across scales instead of the multi-scale feature representation is the key factor when making predictions over multiple layers; (2) we further carefully examine the factors affecting detection performance and identify the key issue in existing anchor-based detectors, i.e., the number of sampled anchors during training are imbalanced at different scales; (3) we present a simple and straightforward solution to handle this issue and the proposed solution is verified on various settings including different datasets and detection pipelines, and our method can make significant improvements under these settings.

Our single model based on ResNet-101 [36] using the proposed method has achieved the 2nd place on the Face Detection track of WIDER Challenge held in ECCV 2018.[3] This paper, based on our previous conference paper [37], presents a substantial extension with following improvements, (1) we

2. A detailed formulation of this procedure is described in Section 4.1.

3. http://wider-challenge.org/2018.html

introduce more comprehensive experiments analyzing the performance using various feature maps as well as various anchor strides on COCO [38] to show that our observation, anchor distribution is the key factor affecting detection performance, still holds on general object detection; (2) we apply our proposed group sampling method on COCO [38] under different settings to show the effectiveness of our method on general object detection; (3) we provide experiments to show that our approach is applicable to other detection pipelines including YOLOv3, SSD and R-FCN, achieving remarkable improvements without any extra inference cost; (4) we introduce empirical analysis about the effect of independently training sub-detectors and receptive field size to get a thorough understanding about the factors affecting detection performance; (5) we make some improvements to further push the state-of-the-art on WIDER FACE [34].

2 Related Work

Single Scale Feature for Multi-Scale Detection. Modern detectors, such as Fast R-CNN [18] and Faster R-CNN [19], use single scale feature for multi-scale detection by extracting scale-invariant features through RoI operation. They offer a good trade-off between accuracy and speed, while still perform not well on small objects. One of the reasons might be the insufficient positive training samples for small objects, and we show that the proposed group sampling alleviates this issue and improves the detection accuracy.

Top-Down Architecture With Lateral Connections. Top-down structure with lateral connections is getting popular ever since proposed and has been widely used in various computer vision tasks, e.g., U-Net [25] and SharpMask [27] for semantic segmentation, Recombinator Network [39] for face detection, and Stacked Hourglass Network [40] for human pose estimation. The advantage of such an architecture is that a single high-resolution feature map which captures both semantic information and fine grained details can be obtained through the combination of the low-level feature maps and the high-level feature maps. In our experiments, we show that top-down architecture with lateral connections is indeed very helpful for face detection.

Multi-Scale Feature for Multi-Scale Detection. Some recent works adopt the feature pyramid for object detection, in which features at different levels are used to handle objects at different scales, e.g., SSD [30], MS-CNN [29], FPN [33], and SSH [32], S^3FD [41], DSFD [42] for face detection, some of which also exploit the top-down architecture with lateral connections for strong feature representation. Such multi-scale feature representation is now widely adopted as the default backbone for recent top methods, such as RetinaFace [43], DSFD [42], PyramidBox [44], SRN [45] on WIDER FACE [34], and SNIP [21], RetinaNet [46], DCR [47] on COCO [38]. They predict boxes on multiple feature map layers with different resolutions in order to achieve better performance. However, we show that the improvement comes from the anchor distribution towarding small objects rather than the multi-level feature representation.

Data Imbalance. % Learning from class imbalanced data, in which the distribution of training data across different object classes is significantly skewed, is a long-standing problem. A common approach to address class imbalance issue in machine learning is re-sampling the training data [48], [49], [50], [51], [52], e.g., under-sampling instances of the majority classes [53] or over-sampling the instances of the minority classes with generative or discriminative models [54], [55], [56]. Another common approach is cost-sensitive learning, which reformulates existing learning algorithms by penalizing the misclassifications of the minority classes more heavily than the misclassifications of the majority class [57], [58]. As sampling implicitly defines the weights of samples and cost-sensitive learning explicitly assigns weights, both actually change the overall loss function considered. This paper adopts sampling to handle the observed imbalance issue. The specific sampling technique may seem simple, but it is nontrivial to discover the unnoticed yet influential imbalance issue that affects the final performance. For example, In [59], the authors identify the pairwise distance distribution is imbalanced in embedding learning research area and proposes distance weighted sampling to improve the performance. For detection area, there also exists data imbalance issue. For instance, online hard example mining [60] or carefully designed loss functions [46] aim to handle the imbalance between easy examples (majority) and hard examples (minority) by assigning larger weights for hard samples. S^3FD [41] has observed that the positive training samples is insufficient for small objects and propose to increase the number of small positive training samples by decreasing the IoU threshold. In this work, we discover a novel unnoticed imbalance issue that the anchor distribution across different scales of not only the positive samples but also the negative samples is imbalanced and crucial in detection task, and propose a simple group sampling method to explicitly handle it, which can achieve better detection accuracy.

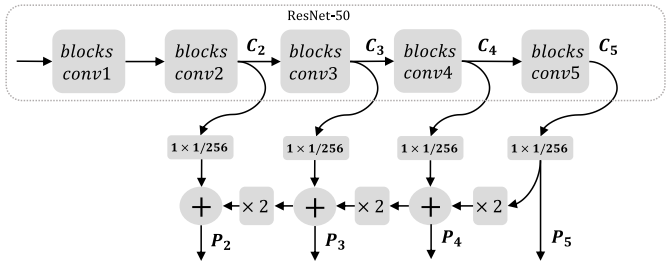


Fig. 3. Network architecture used in our experiments, $\times 2$ is bilinear upsampling and $\oplus$ is element wise summation.

3 Motivation: Scale Imbalance Distribution

In this section, we provide an in-depth analysis of two factors that might affect detection accuracy: multi-scale feature representation and scale imbalance distribution. We use ResNet-50 [36] combined with top-down lateral connections as the backbone. Fig. 3 briefly illustrates the network structure. Following the notation used in [33], the output features from the last residual block of conv2, conv3, conv4 and conv5 are denoted as C_2, C_3, C_4, C_5 respectively. The bottom-up feature map first undergoes a 1×1 convolution layer to reduce the channel dimensions and then is merged with the up-sampled feature map by element-wise summation. This procedure is repeated for three times. We denote the final output feature maps as $\{P_2, P_3, P_4, P_5\}$, and P_i has the same spatial size with C_i. The anchor scales are {16, 32, 64, 128}

TABLE 1
Average Precision (AP) of Face Detection on WIDER FACE Validation Set With Different Feature Map

Detector	Anchor Stride	Feature	Easy	Med	Hard	All	@16	@32	@64	@128
FPN	{4, 8, 16, 32}	$\{P_2, P_3, P_4, P_5\}$	90.9	91.3	87.6	82.1	72.3	67.3	43.2	21.2
FPN-finest-stride	{4, 8, 16, 32}	P_2	90.4	91.0	87.1	81.6	72.2	66.6	43.3	21.8
FPN-single	4	P_2	94.1	93.0	86.6	80.2	65.6	66.8	43.9	22.4
	8	P_2	94.2	92.8	86.0	79.5	62.5	66.5	44.1	22.8
		P_3	93.9	92.6	85.9	79.6	62.5	66.4	44.2	22.4
	16	P_2	93.3	91.7	83.1	73.7	46.1	64.7	44.2	22.0
		P_3	93.2	91.8	83.5	74.2	48.1	65.0	44.5	22.1
		P_4	93.1	91.6	83.5	74.4	47.9	65.6	44.2	22.1

@16 represents the AP on the 'All' subset when only using anchors of scale 16 for detection. So dose the same for @32, @64 and @128.

with aspect ratio 1 on WIDER FACE. For object detection on COCO dataset, we add one more output feature map P_6 by applying max pooling and then two 3×3 convolutions over the feature map P_5. The anchor scales are therefore {32, 64, 128, 256, 512} with 3 aspect ratios {0.5, 1, 2}. Based on this network architecture, we compare three types of detectors:

1) FPN: $\{P_2, P_3, P_4, P_5\}$ ($\{P_2, P_3, P_4, P_5, P_6\}$) are used as the detection layers with the anchor scales {16, 32, 64, 128} ({32, 64, 128, 256, 512}) corresponding to feature stride {4, 8, 16, 32} ({4, 8, 16, 32, 64}) pixels respectively on WIDER FACE (COCO).
2) FPN-finest-stride: all anchors are tiled on the finest layer of the feature pyramid, i.e., P_2. The stride is {4, 8, 16, 32} ({4, 8, 16, 32, 64}) pixels for anchors with scale {16, 32, 64, 128} ({32, 64, 128, 256, 512}), respectively. This is implemented by sub-sampling P_2 for larger strides.
3) FPN-single: only one single layer of the feature pyramid is used for detection. The stride is the same for all anchors with different scales. In our experiments, we also consider applying sub-sampling on a single feature map for a large stride, e.g., sub-sample the feature map P_2 with stride 2 to get feature map with stride 8.

To ensure fair comparison, all detectors use the same setting for both training and inference except that we keep different number of proposals before NMS in the inference process in order to maintain enough proposals after NMS. The detailed experiment settings can be found in Section 5. And the results on WIDER FACE dataset [34] and the challenging Microsoft COCO dataset [38] are shown in Tables 1 and 2 respectively. We have following observations.

Using single layer feature is able to achieve comparable performance compared with the counterpart using multiple layer features. The only difference between FPN and FPN-finest-stride lies in that whether the features used for detection come from one single layer or come from multiple layers. From Table 1, we can see that on WIDER FACE dataset, the Average Precision (AP) of FPN is 90.9, 91.3, 87.6 percent for easy, medium and hard subsets respectively. In contrast, the results for FPN-finest-stride are 90.4, 91.0, 87.1 percent. From Table 2 showing the results on COCO dataset, the AR^{100} and AR^{1k} for FPN are 46.1 and 58.7 percent, while the results for FPN-finest-stride are 46.7 and 58.6 percent. The results are comparable, showing that the differences between single and multi-scale features are marginal.

Scale Imbalance Distribution Matters. We further compare FPN-finest-stride with FPN-finest (equivalent to FPN-single using feature map P_2 and stride 4 for all different anchors). We observe that 1) since more training examples for large anchors are selected than FPN-finest-stride, FPN-finest gets better performance on easy and medium subsets on WIDER FACE and large subset on COCO; and 2) performance drops on hard subset of WIDER FACE and small subset as well as medium subset of COCO, even though FPN-finest has the same number of anchors for the smallest scale with FPN-finest-stride. To find out the reason behind, we take WIDER FACE as an example and plot the proportions of positive training samples and negative ones at different scales for all the compared detectors in Fig. 2.

First, the performance of FPN and FPN-finest-stride are almost the same and their anchor distribution at different scales are also similar as shown in Fig. 2b and 2c, suggesting that similar distribution, when the total number of anchors is the same, gives rise to similar performance.

Second, as shown in Fig. 2c and 2d, the sample distribution at different scales for both FPN-finest-stride and FPN-

TABLE 2
Average Recall (AR) of Object Detection on MS COCO val2017 Dataset

Detector	Anchor Stride	Feature	AR^{100}	AR^{100}_s	AR^{100}_m	AR^{100}_l	AR^{1k}	AR^{1k}_s	AR^{1k}_m	AR^{1k}_l
FPN	{4, 8, 16, 32, 64}	$\{P_2, P_3, P_4, P_5, P_6\}$	46.1	32.2	54.7	58.4	58.7	48.9	65.9	65.5
FPN-finest-sride	{4, 8, 16, 32, 64}	P_2	46.7	31.2	54.4	62.9	58.6	48.4	65.3	66.9
FPN-single	4	P_2	41.6	22.6	49.3	64.3	54.9	40.7	62.8	68.6
	8	P_2	42.5	23.3	50.7	64.8	55.2	40.7	63.6	68.7
		P_3	42.4	23.4	50.3	64.5	55.2	41.0	63.1	68.6
	16	P_2	42.8	23.2	51.1	65.2	53.9	37.9	63.1	68.6
		P_3	42.9	23.5	51.2	65.3	54.0	38.2	63.2	68.8
		P_4	42.8	23.8	50.8	64.8	53.9	38.3	63.0	68.5

We report the AR^{100} and AR^{1k}, respectively.

finest is imbalanced and quite different. It seems that FPN-finest-stride has more small negative anchors and achieves higher accuracy on hard set, while FPN-finest has more large positive anchors and achieves higher accuracy on easy set. This leads to our hypothesis that scale imbalance distribution is a key factor affecting the detection accuracy.

In addition, we report the results of using the same anchor stride on different feature maps, e.g., FPN-single using same stride 8 on feature map P_2 or P_3, and FPN-single using same stride 16 on feature map P_2, P_3, or P_4. Same anchor stride leads to similar anchor distribution, and the resulting performance shown in Tables 1 and 2 is also comparable though different feature maps are used, showing that anchor distribution is the key issue.

Motivated by above observations, we propose a group sampling method to handle the scale imbalance distribution, and show that it is effective on various kinds of network architectures as well as on other detection pipelines. Fig. 2e shows that the anchor distribution of FPN-finest-sampling which equips FPN-finest with the proposed group sampling method during training, is more balanced, and as a result, FPN-finest-sampling achieves the best performance among the models in Fig. 2.

4 Group Sampling Method

For anchor based face detection, there is an important step which is to match ground-truth boxes with anchors and assign those anchors with labels based on their IoU ratios. Therefore the classifiers are optimized based on these assigned positive and negative anchors. In this section, we first introduce the anchor matching strategy that we adopt and then present the proposed group sampling method followed by discussion.

4.1 Anchor Matching Strategy

Current anchor matching strategies usually follow a *two-pass scan* policy, which has been widely used in detection works [19], [30]. In the first pass, each anchor is matched with all ground-truth boxes and it is assigned with a positive/negative label if its highest IoU is above/below a predefined threshold. However, some ground-truth boxes may be unmatched in this step. The second pass is to further associate those unmatched ground-truth boxes with anchors. We also adopt such policy and the details are described below.

Formally, the set of anchors is denoted as $\{p_i\}_{i=1}^{n}$, where i is the index of the anchor and n is the number of the anchors for all scales. Similarly, the ground-truth boxes are denoted as $\{g_j\}_{j=1}^{m}$, where j is the index of the ground-truth and m is the number of ground-truth boxes. Before the matching step, a matching matrix $\mathbf{M} \in \mathcal{R}^{n\times m}$ is first constructed, representing the IoUs between anchors and ground-truth boxes, i.e., $\mathbf{M}(i,j) = \mathrm{IoU}(p_i, g_j)$.

In the first pass, each anchor p_i is matched with all the ground-truth boxes to find the highest IoU, denoted as $C(i) = \max_{1\le j\le m}\mathbf{M}(i,j)$. Hence p_i is assigned with a label according to the following equation:

$$L(i) = \begin{cases} 1, & \lambda_1 \le C(i) \\ -1, & \lambda_2 \le C(i) < \lambda_1 \\ 0, & C(i) < \lambda_2 \end{cases}, \tag{1}$$

where λ_1 and λ_2 are two preset thresholds, the label 1, 0 represents the positive and negative samples respectively. -1 means that p_i will be ignored during training.

It is likely that some ground-truth bounding boxes are not matched to any anchor in the first pass, especially for small objects. So the second pass often aims to make full use of all ground-truth boxes to increase the number of positive training samples. Specifically, for each unmatched ground-truth box, say g_j, we match it with the anchor p_i which satisfies three conditions: 1) this anchor is not matched to any other ground-truth boxes; 2) $\mathrm{IoU}(p_i, g_j) \ge \lambda_2$; 3) $j = \arg\max_{1\le u\le m} \mathrm{IoU}(p_i, g_u)$.

4.2 Group Sampling

After each anchor is associated with a label, we find that there exist two kinds of imbalance in the training samples.

- Positive and negative samples are not balanced, i.e., the number of negative samples in the image is much larger than the number of positive samples due to the nature of object detection task.
- Samples at different scales are not balanced, i.e., small objects are more difficult to find a suitable anchor than large objects due to the IoU based matching policy.

Previous methods often focus on the first issue and usually handle it by controlling the number of positive and negative training samples, e.g., the positive and negative sample ratio is set to 1:3 for sampling. But the second issue does not get enough attention.

To handle above two issues, we propose a scale aware sampling strategy called *group sampling*. We first divide all the anchors into several groups according to their scale, e.g., all anchors in each group have the same scale. Then we randomly sample the same number of training samples for each group, and ensure that the ratio of positive and negative samples in each sampled group is 1:3. If there is a shortage of positive samples in a group, we will increase the number of negative samples in this group to make sure that the total number of samples for each group is the same during training.

Formally, let $\mathcal{P}_s$ and $\mathcal{N}_s$ be the set of randomly sampled positive and negative anchors with scale s, that is $\mathcal{P}_s \subseteq \{p_i|\mathcal{L}(i) = 1, \mathcal{S}(i) = s\}$ and $\mathcal{N}_s \subseteq \{p_i|\mathcal{L}(i) = 0, \mathcal{S}(i) = s\}$, where $\mathcal{S}(i)$ denotes the scale of anchor p_i. Thus our proposed approach is to first guarantee that $|\mathcal{P}_s| + |\mathcal{N}_s| = N$ where N is a constant, and then ensure that $3|\mathcal{P}_s| = |\mathcal{N}_s|$ for scale s. Therefore, for all the scales, each classifier would have sufficient and balanced positive and negative samples for training.

The loss function used for training is

$$\mathcal{L} = \sum_s \left(\frac{1}{|\mathcal{P}_s| + |\mathcal{N}_s|}\mathcal{L}_{cls} + \frac{\eta}{|\mathcal{P}_s|}\mathcal{L}_{reg} \right), \tag{2}$$

in which $\mathcal{L}_{cls}$ is softmax loss for binary classification as used in [19], $\mathcal{L}_{reg}$ is our proposed least square IoU loss as described in Section 5, and η is the loss weight to balance the classification loss and the regression loss. In our experiments, we set η as 1.

Group Sampling With Hard Examples Mining. One intuitive extension for group sampling is to sample hard examples in each group instead of adopting random sampling (RS) as described above. The first approach is to calculate all the examples' training losses and then sample K examples

which have the highest losses, this approach is equivalent to apply OHEM [60] for each group. The results are shown in Table 9. OHEM [60] typically relies on Non-Maximum Suppression (NMS) to prune proposals which are highly overlapped, but NMS is actually the chain process which will consume lots of training time, especially in the cases that the amount of proposals is very large. Apply OHEM directly will introduce non-negligible computational cost, which is not desired. And from the perspective of sampling, OHEM [60] only takes the samples with highest losses into consideration, which will sometimes result in bad performance when the annotations in a dataset are in low-quality. To resolve these issues, we also explore another approach, i.e., Weighted Random Sampling(WRS) [61] to mine hard examples. Following is the formulation of WRS used in our paper.

Given a proposal set $\{p_1, p_2, \ldots, p_n\}$ of size N, we first calculate the classification loss for each proposal, which will be denoted as $\{l_1, l_2, \ldots, l_n\}$. Second, we will assign a weight w_i for p_i as following:

$$w_i = r_i^{1/(l_i+\epsilon)}, \tag{3}$$

in which, r_i is a random variable drawn from a uniform distribution in $(0, 1)$, i.e, $r_i \sim U(0, 1)$ and ϵ is a constant value to avoid numeric error when $l_i = 0$, which is 10^{-6} in our setting.[4] Finally, we will select K proposals with highest weight as training samples on positive and negative samples separately as what we have mentioned for random sampling.

Compared with OHEM, WRS does not realy on NMS and the additional cost to compute classification loss is negligible. And it will not omit the easy samples during training. As shown in Table 9, WRS outperforms RS and OHEM by a large margin % on both WIDER FACE and COCO dataset.

Grouped Fast R-CNN. It is known that after obtaining the candidate regions, using Region-of-Interest (RoI) operation to extract features for each proposal and then feeding these features into another network to further improve the detection accuracy. However, directly applying Fast R-CNN brings a little performance improvement (about 1 percent). Considering the huge computation cost introduced, this practice is quite cost-ineffective. Interestingly, we notice that the scale distribution of the training samples for Fast R-CNN is also unbalanced, where the proposed group sampling method can be used again.

Specifically, before sampling for Fast R-CNN, Non-Maximum-Suppression(NMS) is usually applied first to keep top K scored proposals. However, there exists an issue that the proposals at small scale usually have lower score than the large ones, which will result in insufficient small proposals for further refinement. To resolve this issue, we apply the proposed group sampling method in this step. Instead of grouping proposals according to the anchor scale, we reassign a reference scale for each proposal from a predefined scale set $\mathcal{R} = \{16, 32, 64, 128\}$, and the reference scale r_s for a proposal with width w and height h is obtained as

$$r_s = \arg\min_{s \in \mathcal{R}}(|\log_2(\sqrt{wh}/s)|). \tag{4}$$

4. a detailed discussion of this formulation and strict proof is beyond the scope of this paper.

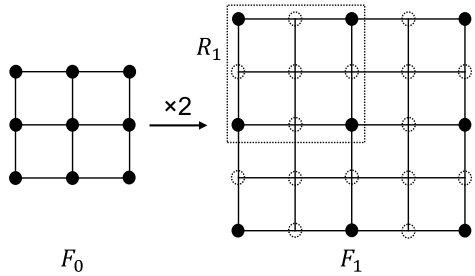


Fig. 4. Illustrating that the receptive field size of applying a 3×3 convolution on an up-sampled feature map is equivalent to that of applying a 2×2 convolution on the original feature map.

We show that this can effectively improve the accuracy of Fast R-CNN. We denote Fast R-CNN with group sampling as *Grouped Fast R-CNN*.

4.3 Discussion

Relation to OHEM and Focal Loss. Online hard example mining (OHEM) [30] and Focal Loss [46] are two widely adopted approaches to increase the weight of the hard examples. Online hard example mining (OHEM) [30] is to keep the top K samples with highest training loss for back propagation. Focal Loss [46] proposes giving each sample a specific weight according to the corresponding score (the weight is larger if the score is lower for positive samples). Both can be viewed as to assign weights for different training samples,[5] similar to the cost-sensitive learning by penalizing the misclassifications of the minority class more heavily. Specifically, the data imbalance they addressed here is the imbalance between easy examples (majority) and hard examples (minority). Differently, our approach aims to handle the imbalance of anchors of not only the positives but also the negatives across different scales, which is not considered in OHEM and Focal Loss as both focus on mining hard examples regardless of either scale or whether the sample is positive or not. On the other hand, we notice that we can combine OHEM and Focal Loss in group sampling so that the positive examples and negative ones in each group are sampled according to the loss instead randomly. We present experiments comparison in Table 9, showing that using group sampling achieves better performance than OHEM and Focal Loss, and can further boost the performance of detecting small faces when combined with OHEM / Focal Loss. Note that we randomly sample 2,048 samples and apply the Focal Loss on them when combing GS and Focal Loss on WIDER FACE.

Analysis of the Receptive Field Size. Another thing that might be ignored when comparing using multi-scale features or single-scale features for detection is the receptive field size. We first show that the receptive field size after applying a 3×3 convolution on the up-sampled feature maps is equivalent to that after applying a 2×2 convolution on the original feature maps. It can be easily derived and we present a simple illustration in Fig. 4.

In FPN [33], feature maps from different levels of the pyramid are followed by independent 3×3 convolutions before being fed into the shared RPN head network. Therefore,

5. OHEM assigns the loss weight 1 for the selected samples and 0 for others.

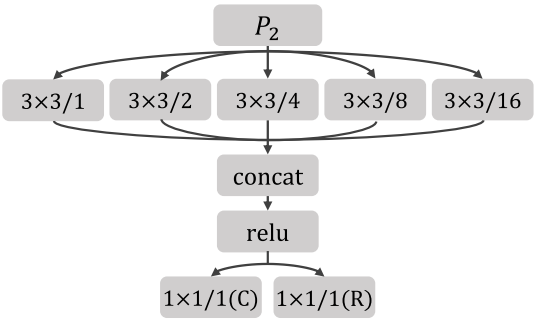


Fig. 5. The network architecture of RPN head in FPN-finest-DC, in which we adopt dilated convolution for enlarging the receptive field. '$k \times k/d$' means a convolution with kernel size k and dilation size d, all these convolutions are with stride 1. 'C' and 'R' are short for classification and regression, respectively.

anchors associated with different feature maps get different receptive field size and more further, larger anchors get larger receptive field size. In contrast, only using the finest feature maps, i.e., P_2, results in same receptive field size for anchors across different scales. Specifically, larger anchors get smaller receptive field size compared to FPN. To enlarge the receptive filed, we adopt dilated convolutions, which has been widely used in lots of tasks [62], [63], [64]. The resulting model for FPN-finest is denoted as FPN-finest-DC, whose RPN head is illustrated in Fig. 5. Detailed empirical analysis is presented in Section 6.2.

5 Training Process

In this section, we introduce the training dataset, loss function and other implementation details. Note that we propose a new IoU based loss function for regression to get better performance compared with Smooth-L1 loss [19].

Training Dataset. For face detection, as with previous works [30], [41], we train our models on the WIDER FACE training set which contains 12,880 images and test on the WIDER FACE validation and test set, as well as the FDDB dataset. For object detection, we use the Microsoft COCO object detection dataset [38]. This dataset contains 80 object categories. We train models on the training set including 118,287 images (train2017) and test on the validation set including $5k$ images (val2017).

Loss Function. We use softmax loss for binary classification. For regression, we propose a new IoU based loss, denoted as least square IoU loss

$$\mathcal{L}_{reg} = \frac{1}{N_{reg}} \sum_{(p,g)} \|1 - \mathrm{IoU}(p,g)\|_2^2, \tag{5}$$

where (p, g) is a matching pair of a proposal p and a ground-truth g. Compared with the Smooth-L1 loss [19], this loss function directly optimizes the IoU ratio, which is consistent with the evaluation metric, as with some IoU based loss function proposed in previous works [65], [66], e.g., $-\ln(\mathrm{IoU})$ proposed in [66]. However, there is an situation for previous regression loss that they will get non-zero gradient when IoU equals to 1.

To show this, let the coordinates of the ground-truth bounding box g be $(x_1^g, y_1^g, x_2^g, y_2^g)$ and the coordinates of the matched proposal p be $(x_1^p, y_1^p, x_2^p, y_2^p)$. The partial gradient of our loss function with respect to x_1^p is

$$\frac{\partial \mathcal{L}_{reg}}{\partial x_1^p} = -\frac{2(1 - \mathrm{IoU}(p,g))}{N_{reg}} \frac{\partial \mathrm{IoU}(p,g)}{\partial x_1^p}. \tag{6}$$

While the partial gradient of previous loss function $\mathcal{L}_{reg}^{pre} = -\sum_{(p,g)} \ln(\mathrm{IoU}(p,g))$ with respect to x_1^p is

$$\frac{\partial \mathcal{L}_{reg}^{pre}}{\partial x_1^p} = -\frac{1}{N_{reg}} \frac{1}{\mathrm{IoU}(p,g)} \frac{\partial \mathrm{IoU}(p,g)}{\partial x_1^p}. \tag{7}$$

To calculate $\frac{\partial \mathrm{IoU}(p,g)}{\partial x_1^p}$, we first formalize how IoU is obtained. Let

$$x_1^* = \max(x_1^p, x_1^g), \quad y_1^* = \max(y_1^p, y_1^g), \tag{8}$$

$$x_2^* = \min(x_2^p, x_2^g), \quad y_2^* = \min(y_2^p, y_2^g). \tag{9}$$

The intersection area $\mathcal{I}$ and the union area $\mathcal{U}$ are given as

$$\mathcal{I} = (x_2^* - x_1^*)(y_2^* - y_1^*), \tag{10}$$

$$\mathcal{U} = (x_2^p - x_1^p)(y_2^p - y_1^p) + (x_2^g - x_1^g)(y_2^g - y_1^g) - \mathcal{I}. \tag{11}$$

As $\mathrm{IoU} = \mathcal{I}/\mathcal{U}$, we have

$$\begin{aligned} \frac{\partial \mathrm{IoU}(p,g)}{\partial x_1^p} &= \frac{\frac{\partial \mathcal{I}}{\partial x_1^p}\mathcal{U} - \mathcal{I}\frac{\partial \mathcal{U}}{\partial x_1^p}}{\mathcal{U}^2} \\ &= \begin{cases} \frac{\mathcal{I}(y_2^p - y_1^p)}{\mathcal{U}^2} & \text{if } x_1^p < x_1^g \\ \frac{-(\mathcal{U}+\mathcal{I})(y_2^* - y_1^*) + \mathcal{I}(y_2^p - y_1^p)}{\mathcal{U}^2} & \text{if } x_1^p \geq x_1^g \end{cases}. \end{aligned} \tag{12}$$

When $\mathrm{IoU}(p,g)$ equals to 1, which is the ideal case, it can be easily seen that previous IoU loss gets non-zero gradient according to Equations (7) and (12). Moreover, the left-side limit and the right-side limit are different, i.e., $\lim_{x_1^p \to x_1^{g-}} \frac{\partial \mathrm{IoU}(p,g)}{\partial x_1^p} \neq \lim_{x_1^p \to x_1^{g+}} \frac{\partial \mathrm{IoU}(p,g)}{\partial x_1^p}$. On the other hand, our proposed IoU least square loss gets zero gradient according to Equation (6), allowing the network to converge stably. Empirically we show that the proposed IoU loss achieves better performance.

Following the approach used in [18], [19], [67], [68], we do not predict the positions of a proposal directly and adopt the parameterizations of the 4 coordinates following:

$$\begin{aligned} t_x &= (x_c^p - x_c^a)/w^a, \; t_y = (y_c^p - y_c^a)/h^a, \\ t_w &= \log(w^p/w^a), \; t_h = \log(h^p/h^a), \end{aligned} \tag{13}$$

in which (x_c^a, y_c^a, w^a, h^a) ((x_c^p, y_c^p, w^p, h^p)) are the center coordinates, width and height of the anchor (the predicted box) respectively.

Optimization Details. All models are initialized with the ImageNet [69] pre-trained weights of ResNet-50 provided by `torchvision`[6] and fine-tuned on the WIDER FACE training set or COCO train2017 dataset. Our code is based on `PyTorch` [70].

WIDER FACE. Each training iteration contains one image per GPU for an 8 × NVIDIA Tesla M40 GPUs server. all the

6. https://github.com/pytorch/vision

TABLE 3
Average Precision (AP) of Face Detection on WIDER FACE Validation Set

Methods	Anchor Stride	Feature	GS	Easy	Medium	Hard	All	@16	@32	@64	@128
RPN	16	C_4		92.5	91.0	83.0	74.0	48.6	65.1	43.8	21.5
FPN-finest	4	P_2		94.1	93.0	86.6	80.2	65.6	66.8	43.9	22.4
FPN-finest-DC	4	P_2		94.4	93.3	87.3	81.1	66.5	67.9	44.5	23.5
FPN	{4, 8, 16, 32}	$\{P_2, P_3, P_4, P_5\}$		90.9	91.3	87.6	82.1	72.3	67.3	43.2	21.2
FPN-finest-stride	{4, 8, 16, 32}	P_2		90.4	91.0	87.1	81.6	72.2	66.6	43.3	21.8
FPN-finest-sampling	4	P_2	✓	94.7	93.8	88.7	82.8	74.1	72.9	47.8	24.5
FPN-finest-stride-sampling	{4, 8, 16, 32}	P_2	✓	87.6	89.1	88.2	83.5	75.9	**74.2**	48.5	25.1
FPN-finest-DC-sampling	4	P_2	✓	94.9	94.0	89.3	83.7	74.8	73.4	48.8	25.8
FPN-finest-DC-WRS	4	P_2	✓	**95.1**	**94.9**	**89.5**	**84.5**	**75.9**	73.8	**49.3**	**26.3**

GS represents the proposed group sampling method. @16 represents the AP on 'All' subset when only using outputs from the sub-detector at scale 16 for detection. So dose @32, @64, and @128.

models are trained for 100 epochs by synchronized SGD. We set the initial learning rate to 0.01 and decrease the learning rate by 0.1 at the 60th and 80th epoch respectively. The momentum and weight decay is set to 0.9 and 5×10^{-5} respectively. During training, we use scale jitter and random horizontal flip for data augmentation. For scale jitter, each image will be resized with a scale $0.25\times n$, and n is randomly chosen from [1, 8], where $n \in \mathbb{N}$. We also randomly crop a patch from the resized image to ensure that each side of the image does not exceed 1,200 pixels due to the GPU memory limitation and also for efficiency. We set λ_1=0.6, λ_2=0.4 and λ_1=0.6, λ_2=0.5 for the *two-pass scan* matching policy in RPN and Fast R-CNN respectively. At the inference stage, we build an image pyramid with the scales used in scale jitter for multi-scale test. The proposals from each level of the image pyramid will be merged by Non-Maximum Suppression (NMS) with threshold 0.5. We keep $50k$ proposals before NMS in the inference stage. Due to the GPU memory limitation, each side of the test image will not exceed 3,000 pixels, which means that some level of the image pyramid will not be involved for inference.

COCO. We follow the setting used in [33] to finetune the model on COCO train2017 for 12 epochs (nearly $90k$ iterations). The initial learning rate is set to 0.005 and decreased at the 8th and 10th epoch by 0.1, the momentum and weight decay is set to 0.9 and 5×10^{-5} respectively. During training stage, we resize the short side of each image to 800 pixels and random crop a patch to ensure that the long side will not exceed 1,200 pixels. We also conduct single scale test as described in [33]. In our implementation, we keep $12k$ and $200k$ proposals before NMS in the inference stage for FPN and FPN-single respectively in order to maintain enough proposals after NMS. The class-agnostic detector is adopted for COCO for single shot detection.

6 Experiments

In this section, we first examine the factors affecting detection accuracy and then present extensive ablation experiments to demonstrate the effectiveness of our approach. Finally, we introduce our approach that using single layer predictions advances the state-of-the-arts on several challenging benchmarks, including WIDER FACE [34] and FDDB [71] datasets.

6.1 Factors Affecting Detection Accuracy

We further present a thorough analysis about the five detectors: RPN, FPN, FPN-finest, FPN-finest-stride and FPN-finest-sampling, which have been introduced in Fig. 1. There are two differences among them: 1) the feature map on which the anchors are tiled on; 2) the stride used for different anchors. The stride of an anchor indicates the number of anchors and smaller stride gives rise to more anchors than the large ones.

We adopt Average Precision (AP) for WIDER FACE [34] and Average Recall (AR) for MS COCO [38] as the evaluation metric. Previous methods usually report the results on each dataset and also their subset, e.g., the easy, medium and hard subsets of WIDER FACE, or the small, medium and large objects of COCO. Note that the authors of [34] split WIDER FACE into three subsets according to the recall rate when using EdgeBox [72] to extract proposals, and previous works evaluated the performance on these three subsets. In this paper, besides the results on these subsets, we also report the results on the "All" set provided in WIDER FACE, which takes all the annotated faces into consideration. However, these results cannot show the ability of sub-detector which is used to handle objects in a specific scale range. This is because a large object (e.g., 128 $\times$ 128 pixels) which is usually detected by the sub-detector with scale 128 in original image, is possible to be actually detected by the sub-detector with scale 16 in the down-sampled image due to the multi-scale inference mechanism. Therefore, to clearly show the ability of each sub-detector, we also report the performance of each sub-detector in our model on WIDER FACE dataset, which will be denoted as $@S$ and S is the scale of each sub-detector. The performance of each sub-detector is evaluated by the AP on the 'All' set of WIDER FACE with the IoU threshold 0.5. The performance comparison is shown in Tables 3 and 4. We have following observations:

Imbalanced training data at scales leads to worse (better) accuracy for the minority (majority). The only difference between FPN-finest and FPN-finest-stride is the anchor stride, i.e., the number of anchors for different scales are different. Considering the experiments conducted on WIDER FACE in this case, for scale 16, its stride is the same in this two models. Therefore, the number of anchors at scale 16 is also the same. However, this is not the case for performance over @16. FPN-finest-stride achieves 72.2 percent, which is

TABLE 4
Average Recall (AR) on MS COCO val2017 Set

Methods	Anchor Stride	Feature	GS	AR^{100}	AR_s^{100}	AR_m^{100}	AR_l^{100}	AR^{1k}	AR_s^{1k}	AR_m^{1k}	AR_l^{1k}
FPN-finest	4	P_2		41.6	22.6	49.3	64.3	54.9	40.7	62.8	68.6
FPN-finest-DC	4	P_2		43.7	23.9	51.9	**67.0**	56.4	41.9	64.3	**70.9**
FPN	{4, 8, 16, 32, 64}	$\{P_2, P_3, P_4, P_5, P_6\}$		46.1	32.2	54.7	58.4	58.7	**48.9**	**65.9**	65.5
FPN-finest-stride	{4, 8, 16, 32, 64}	P_2		46.7	31.2	54.4	62.9	58.6	48.4	65.3	66.9
FPN-finest-sampling	4	P_2	✓	45.6	22.6	49.3	64.3	54.9	40.7	62.8	68.6
FPN-finest-DC-sampling	4	P_2	✓	47.2	**33.0**	53.2	63.8	**59.2**	48.4	64.9	70.1
FPN-finest-DC-WRS	4	P_2	✓	**47.9**	32.2	**55.0**	65.3	58.6	47.3	65.1	69.4

GS represents the proposed group sampling method.

6.6 percent higher than that in FPN-finest. This is because that in FPN-finest, the number of positive samples at scale 16 is fewer than that at other scales, but sampling is applied on the collection of training samples from all the scales, which means that the positive samples of scale 16 is less likely to be chosen, resulting in lower accuracy. On the contrary, in FPN-finest-stride, the number of positive as well as negative samples at scale 16 is greater than other scales, resulting in higher accuracy. On COCO dataset, we have similar observation that FPN-finest-stride performs better than FPN-finest on small objects.

Similar Anchor Distribution, Similar Performance. As we can see, the results of FPN and FPN-finest-stride are very close. The only difference between the two models is that FPN uses features from multiple layers for detection but FPN-finest-stride only uses a single layer feature map. This suggests that using multi-level feature representation is of little help for improving detection accuracy. Therefore, we ask a question: *does similar anchor distribution leads to similar performance* ? Considering another comparison on WIDER FACE between RPN and FPN-finest, whose sample distributions are similar: both have more large positive examples, the two models have the same tendency of achieving lower accuracy for @16 and higher accuracy for @128 compared with FPN (or FPN-finest-stride), suggesting that similar anchor distribution leads to similar performance.

Data Balance Achieves Better Result. All the above discussed four detectors have imbalanced anchor distributions. Comparing FPN-finest with FPN-finest-sampling, which adopts the proposed group sampling method for FPN-finest, the only difference is the distribution of training data at different scale. We can see that using evenly distributed training data can significantly improve the results, e.g., increasing from 80.2 to 82.8 percent on the 'All' subset of WIDER FACE validation set, increasing from 41.6 to 45.6 percent for AR^{100} on COCO. We will further demonstrate the effectiveness of data balance under various settings in the ablation study.

Enlarging the Effective Receptive Field Helps. In the discussion, we adopt dilated convolution to enlarge the receptive field size. Here we compare the results of FPN-finest and FPN-finest-DC as well as the results of FPN-finest-sampling and FPN-finest-DC-sampling on WIDER FACE and COCO datasets. We can see that with or without the proposed group sampling method, enlarging the receptive field size helps to achieve better performance on WIDER FACE as well as COCO. The reason might be that large receptive field may provide more context information when performing detection and thus improve the accuracy of detectors.

*Group Sampling With Hard Examples Mining Helps.*We further adopt group sampling with hard examples mining. Comparing FPN-finest-DC-WRS with FPN-finest-DC-sampling, the only difference is that WRS samples the hard examples in each group instead of adopting random sampling. We can see that using WRS achieves better performance on both WIDER FACE and COCO datasets. It suggests that group sampling and hard examples mining are complementary.

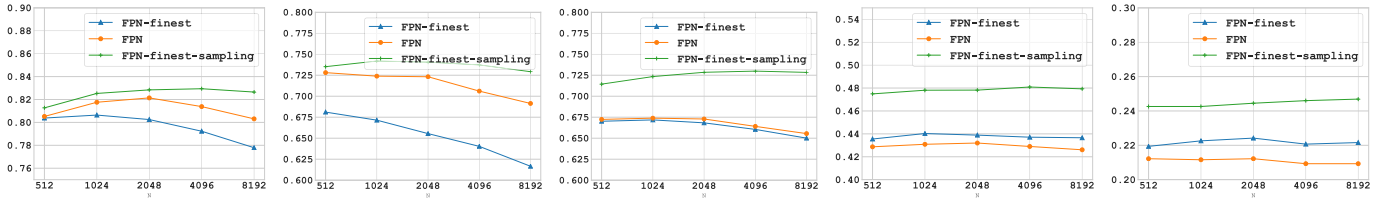
6.2 Ablation Experiments

The Effect of Feature Map. We first compare detection accuracy with and without group sampling when using different feature maps. Table 5 shows the detection performance when using $\{P_2, P_3, P_4, P_5\}$, P_2, and other feature maps. We have following observations, 1) using top-down lateral connections to provide semantic information always helps, the performance of P_n is superior to C_n under all these settings; 2) using high resolution feature map produces more small training samples and helps to detect small faces; 3) regardless of the feature maps, using group sampling can always improve the performance. For the sake of simplicity, we use P_2 as the feature for following experiments.

The Effect of the Number of Training Samples N. As introduced in Section 4.2, we randomly choose N training samples for each group during training. The performance under

TABLE 5
Comparison of Models With/Without Group Sampling Using Different Feature Maps on WIDER FACE

Feature	GS	@16	@32	@64	@128	All
$\{P_2, P_3, P_4, P_5\}$		72.3	67.3	43.2	21.2	82.1
	✓	75.7	73.4	48.2	24.9	83.6
P_2		65.6	66.8	43.9	22.4	80.2
	✓	74.1	72.9	47.8	24.5	82.8
P_3		62.5	66.4	44.2	22.4	79.6
	✓	72.1	73.2	48.7	25.3	83.7
P_4		47.9	65.6	44.2	22.1	74.4
	✓	57.8	71.0	48.4	25.2	79.6
C_3		59.8	61.8	39.2	18.0	71.0
	✓	68.8	68.9	44.3	21.2	75.4
C_4		48.6	65.1	43.8	21.5	74.0
	✓	58.0	70.8	48.2	24.6	78.9



(a) All (b) @16 (c) @32 (d) @64 (e) @128

Fig. 6. Illustrating the effect of the number of training samples N. Our approach (FPN-finest-sampling) gets better performance when N increases, benefiting from more training examples. The performance of FPN and FPN-finest decreases as N get larger, suffering from more imbalanced data.

different N is shown in Fig. 6. It can be seen that, 1) the performance gets better when N gets larger; 2) the accuracy gets saturated when N is greater than 2,048. Besides, we also plot the results of FPN and FPN-finest under different values of N. We can see that the performance of both models degrades when N increases, because the distribution of training examples become more imbalanced, i.e., negative training samples are nearly dominated.

Application to Other Detection Pipelines. To further verify the effectiveness of our proposed method, we apply the proposed group sampling on several other detection pipelines, including YOLOv3 [73], SSD [30], RetinaFace [43] and R-FCN [74]. Note that YOLOv3 is well known by its inference speed at the sacrifice of accuracy. We adopt darknet-59 as the backbone for YOLOv3. For SSD, RetinaFace and R-FCN, we adopt ResNet-50 as the backbone. The comparison results are shown in Table 6. Other details for training and inference are the same as described in Section 5. We can see that using group sampling achieves large improvement, which verifies the effectiveness of our approach. The reason of YOLOv3 not performing well on easy subset may be its anchor matching strategy, which results in insufficient positive training samples for large scale object. Note that the SSD in our previous version adopts lateral connections for feature extraction. But we remove these lateral connections in this paper for fair comparison. The group sampling method for R-FCN is only applied in the first stage in this table.

The Effect of the Proposed Loss. We propose a new IoU based loss for regression, namely least square IoU loss, to allow the network converge stably. Here we compare different loss functions, including Smooth-L1, $-\ln(\text{IoU})$ and the proposed $\|1-\text{IoU}\|_2^2$. The detector we used is FPN-finest-sampling. The comparison results are shown in Table 8. We can see that the two IoU based loss functions perform better than Smooth-L1, as they directly optimize the evaluation metric. Compared with $-\ln(\text{IoU})$, our proposed least square IoU loss achieves marginally better performance.

The Effect of Independently Training Sub-Detectors. Another way to handle the data imbalance across different scales is that we could train only one sub-detector for the whole training process. In that case, there will be no imbalance at all since there's only one scale during training. Table 7 shows the performance of each sub-detector, which is evaluated in terms of AP only using the outputs of the sub-detector on WIDER FACE. We denote the training using only one scale as 'S' and the baseline using multiple scales as 'M' (which is equivalent to FPN-finest). Our approach using multiple scales along with the proposed group sampling is denoted as 'M+GS' (which is equivalent to FPN-finest-sampling).

It can be seen that results from 'S' are much better than the results from 'M', suggesting that focusing on one scale largely improves the performance of the corresponding sub-detector. However, it needs several networks with each focusing on one scale. In contrast, compared with 'S', our approach achieves better performance using one single model. The key difference is the usage of multi-scale learning, which might suggest that multi-task learning is helpful for generalizing better compared with the single task.

TABLE 6
Comparison of YOLOv3, SSD, RetinaFace and R-FCN on WIDER FACE Validation Set

Method	GS	Easy	Medium	Hard	All
YOLOv3		90.8	89.6	81.0	74.0
	✓	92.1	91.5	83.7	76.0
SSD		93.0	91.3	81.2	74.2
	✓	93.9	92.4	83.4	75.4
RetinaFace		93.8	92.8	88.7	83.5
	✓	95.0	94.0	89.7	84.3
R-FCN		95.9	94.9	90.3	85.1
	✓	95.9	95.0	90.6	85.4

Note that deformable convolution and self-supervision is not applied in RetinaFace for our implementation.

TABLE 7
Performance Comparison of Each Sub-Detector in Terms of AP on WIDER FACE Validation Set Over Three Models: (1) 'S' Denotes Training Using One Single Scale; (2) 'M' Denotes the Baseline Using Multiple Scales; (3) 'M+GS' Denotes Training Using Multiple Scales Along With the Proposed Group Sampling Method

Scale	S	M	M+GS	Easy	Med	Hard	All
16	✓			34.2	56.8	75.5	73.7
		✓		13.5	36.9	66.1	65.6
			✓	32.9	56.8	76.2	74.1
32	✓			83.4	88.6	86.0	72.3
		✓		75.7	84.1	81.5	66.8
			✓	85.0	89.6	86.7	72.9
64	✓			95.1	94.8	58.5	47.1
		✓		94.7	94.6	54.5	43.9
			✓	95.6	95.3	59.4	47.8
128	✓			95.1	71.3	29.8	24.0
		✓		95.3	66.8	27.9	22.4
			✓	95.9	72.6	30.4	24.5

TABLE 8
Comparison of Different Loss Function for the Regression Task

Loss	@16	@32	@64	@128	All
Smooth-L1	74.1	72.9	47.8	24.5	82.8
$-\ln(\text{IoU})$	74.6	73.1	48.0	**25.1**	83.5
$\|1-\text{IoU}\|_2^2$	**75.0**	**73.2**	**48.2**	24.9	**83.7**

The proposed loss function performs better.

Comparison With OHEM and Focal Loss. Here we compare our approach with two closely related methods, OHEM [60] and Focal Loss [46], both adopting hard example mining, which can be regarded as a way of cost sensitive learning handling data imbalance between the easy examples (the majority) and the hard examples (the minority). Specifically, OHEM dynamically select B samples with highest loss among all samples during training. We experiment with different values of B and find that using a relatively smaller B is important to make OHEM work. Hence we set $B = 1024$ in our experiment. For Focal Loss(FL), we adopt the same setting with [46], in which α=0.25 and γ=2. But note that we use a different method to normalize the FL during training, using the approach to normalize the FL with the number of positive samples used in [46] will not converge. We use the weight assigned by FL for each item to normalize FL instead, which is denoted as Focal* in Table 9.

We adopt the same network architecture for the proposed group sampling, OHEM and Focal Loss to ensure fair comparison. The performance comparison is shown in Table 9, where we adopt FPN and FPN-finest as the baselines. Both OHEM and Focal Loss can effectively improve the performance of detecting small faces. For example, the sub-detector @16 of FPN using OHEM and Focal Loss achieve 77.6 and 78.2 percent, about 5 percent higher than FPN; and the sub-detector @16 of FPN-finest using OHEM and Focal Loss get 76.0 and 75.8 percent respectively, about 10 percent higher than FPN-finest. However, the performance of sub-detectors for large scales keeps comparable or even decreases. In contrast, our approach gets improvement for all the sub-detectors compared with the baseline by simply using the proposed group sampling, and also achieves better performance on the 'All' subset compared with OHEM and Focal Loss. Meanwhile, we notice that we can combine OHEM and Focal Loss to enable hard example mining while keeping the samples balanced in each group. The results show that using OHEM and Focal Loss, the performance of detecting small faces can be further improved.

For the experiments on the COCO dataset, we adopt the RetinaNet [46] as the baseline. For fair comparison, the settings for both training and inference are the same as described in [46], which uses the feature pyramid from P_3 to P_7 and tiles 9 anchors on different feature maps. We select 256 samples which have highest training loss for OHEM, and the number of training samples are the same for each group of group sampling. For group sampling, we use $\{32, 64, 128, 256\}$ as the scale set of 4 different group. The results are shown in Table 9b. With the proposed group sampling, GS with WRS can outperform Focal Loss by 1.0 percent in terms of mAP with RetinaNet, which shows the effectiveness of our proposed method. And the comparison of choosing different number of training samples for GS with WRS for FPN is shown in Table 12. In another network architecture which only uses P_3 with dilated convolutions as shown in Fig. 5 to extract features for single stage object detection, which is denoted as FPN-P3-DC in Table 9b, Focal Loss does not converge with the same setting. But WRS still works well in this case. And applying the GS with WRS achieves the best performance.

TABLE 9
Performance Comparison of the Proposed Group Sampling, OHEM and Focal Loss, Showing That Our Approach Achieves Better Performance

(a) WIDER FACE

Method	@16	@32	@64	@128	All
FPN	72.3	67.3	43.2	21.2	82.1
+Focal*	78.2	69.6	44.0	21.2	83.1
+OHEM	77.6	69.9	44.3	21.6	82.7
+WRS	78.8	69.5	44.0	20.5	83.4
+GS	75.7	73.4	48.2	24.9	83.6
+GS +OHEM	79.4	74.1	48.2	24.6	83.6
+GS +WRS	81.2	74.3	48.5	24.2	84.1
+GS +Focal*	77.1	73.6	47.8	24.0	83.7
FPN-finest	65.6	66.8	43.9	22.4	80.2
+Focal*	75.8	68.5	44.2	21.5	81.2
+OHEM	76.0	68.9	43.9	22.0	81.5
+WRS	77.6	69.1	43.7	21.5	82.9
+GS	74.1	72.9	47.8	24.5	82.8
+GS +OHEM	77.9	72.7	47.4	24.2	82.6
+GS +WRS	78.2	73.1	46.8	23.7	83.1
+GS +Focal*	75.9	72.9	47.4	23.7	82.5

(b) COCO

Method	AP@0.5	AP	AP_s	AP_m	AP_l
FPN	47.4	27.1	16.3	31.1	33.3
+Focal	55.8	34.0	20.0	37.4	44.7
+OHEM	53.3	32.0	17.4	35.1	42.3
+WRS	55.4	33.7	19.3	36.9	43.9
+GS	51.4	31.2	17.9	33.4	43.0
+GS +OHEM	54.8	33.7	18.6	36.6	43.7
+GS +WRS	55.9	35.0	19.9	37.8	46.4
FPN-P3-DC	40.1	17.8	9.6	22.3	26.1
+Focal	n/a	n/a	n/a	n/a	n/a
+OHEM	45.3	20.5	13.0	25.3	24.5
+WRS	44.9	20.6	11.3	25.4	28.7
+GS	44.3	21.0	12.9	25.9	26.9
+GS +OHEM	43.5	20.8	11.1	25.9	25.9
+GS +WRS	48.0	23.5	13.7	29.5	29.0

Comparison With Anchor Free Approach. Besides anchor based approaches, another trend for detection is the anchor free approaches [75], [76], [77], [78], [79] introduced in recent years. These methods usually detect an object as a set of points instead of using anchor proposals so that there is no need to design a large number of anchors, or choose appropriate hyper-parameters and so on. Here, we provide the comparison on WIDER FACE with CenterNet [79], one of the recent representative anchor free approahces, in terms of both accuracy and computational complexity. We tune the hyper-parameters in CenterNet to get its best performance on WIDER FACE. For our model, we use FPN-

TABLE 10
Performance Comparison With CenterNet

Method	Easy	Medium	Hard	All	GFLOPs
CenterNet	81.9	81.8	76.5	70.1	98.61
FPN-finest-sampling	92.2	91.3	85.1	76.0	29.73

Our model and CenterNet are all built on ResNet-18 in this table. The huge computation cost for CenterNet is caused by the upsampling modules which adopt Hourglass like structure and the non-shared 3×3 convolutions for different outputs. The input size for GFLOPS computation is 800×800.

TABLE 11
Complexity Comparison

Method	GFLOPs		
	Feature Extraction	RPN head	Total
FPN	52.42	36.50	88.92
FPN-finest-sampling	52.42	28.91	81.33

The input size for GFLOPS computation is 800 × 800.

finest-sampling which adopts ResNet-18 as the backbone. The results are shown in Table 10. It can be seen that our approach achieves better performance than CenterNet. We also show the complexity comparison in Table 10 in terms of FLOPs. We can see that the FLOPs of CenterNet is much larger than FPN-finest-sampling. This is because CenterNet adopts deconvolutions when upsampling feature maps, leading to huge computation cost, while FPN-finest-sampling adopts bilinear upsampling which is much faster than deconvolutions. We use `thop` [7] for GFLOPS computation in our paper.

Complexity Analysis. We have demonstrated that the proposed group sampling is effective in a variety of scenarios including different feature maps as well as different detection pipelines, and such sampling operation does not bring any extra computational cost. In addition, as in the experiments we adopt FPN-finest-sampling that tiles all anchors on the finest feature map, here we present a detailed analysis about the complexity of FPN-finest-sampling, showing that it actually takes even fewer FLOPs than FPN. There are two parts in FPN-finest-sampling, 1) the feature extraction network which is the same with FPN, and 2) the RPN head that is different from FPN. The RPN head is composed of a 3×3 convolution and then two 1×1 convolutions for classification and regression. Say the size of the finest feature map P_2 is $h \times w \times c$, accordingly the size of P_3, P_4 and P_5 are $\frac{h}{2} \times \frac{w}{2} \times c$, $\frac{h}{4} \times \frac{w}{4} \times c$, and $\frac{h}{8} \times \frac{w}{8} \times c$ respectively. the number of output channel of the 3×3 convolutional layer is c' and the number of output channel of the two 1×1 convolutional layers are 4 for regression and 2 for classification. For FPN-finest-sampling, we tile 4 anchors on P_2, and the FLOPs of RPN head is $S_1 = 9c'chw + 4 \times (4c'hw + 2c'hw)$. For FPN, we tile 4 anchors respectively on P_2, P_3, P_4, and P_5, the FLOPs of RPN head is $S_2 = (9c'chw + 4c'hw + 2c'hw) \times (1 + \frac{1}{4} + \frac{1}{16} + \frac{1}{64})$. The difference is $S_1 - S_2 = (1026 - 189c)c'hw/64$, which is smaller than 0 as long as $c > 5$. We have $S_1 < S_2$ given $c > 5$, which is usually satisfied. In our experiments, c is set as 256. Hence, the FLOPs of RPN head for FPN-finest-sampling is actually fewer than that for FPN. Table 11 shows the detailed FLOPS comparison, where we use ResNet-50 as backbone for both models. It can be seen that the total number of FLOPS are comparable.

7. https://github.com/Lyken17/pytorch-OpCounter

TABLE 12
The Performance of FPN With GS and WRS When Choosing Different Number of Training Samples on COCO val2017 Dataset

N	AP@0.5	AP	AP_s	AP_m	AP_l
128	53.5	33.4	17.3	35.8	45.0
256	54.6	34.4	18.8	36.7	47.3
512	54.4	34.4	18.9	36.8	47.1
1024	55.2	35.1	19.1	38.0	46.1

6.3 Grouped Fast R-CNN

We show that the proposed group sampling method can be applied to Fast R-CNN in order to further improve the detection accuracy. We use FPN-finest-sampling as the baseline model. The results on WIDER FACE are given in Table 13. We simply ensemble the scores from the first stage and the second stage as the final score for each bounding box. As shown in the table, the AP is increased from 82.8 to 83.9 percent through directly using Fast R-CNN. After using the proposed least square IoU loss as described in Equation (5), we can increase the AP for easy, medium and hard subsets by 0.2, 0.2 and 0.6 percent respectively. Next, we further adopt the group sampling method and the AP for easy, medium and hard subsets are 96.2, 95.5 and 91.1 percent, which advances the state-of-the-art on WIDER FACE validation set. It is worth noting that using group sampling gets 0.8 percent improvement on hard set, which is significant as the accuracy after using the least square IoU loss is already 90.3 percent. In addition, we also compare group sampling with OHEM and Focal Loss in the second stage, and the results indicate the consistent conclusion that group sampling gets better performance. The model shown in the last line of Table 13 is our final model, which is used to compare with other methods. Besides, we also provide the AP results of the second stage on COCO dataset in Table 14, demonstrating that group sampling is also effective on COCO dataset.

7 Comparison With State-of-the-Art

We compare our approach on four benchmark datasets: WIDER FACE, FDDB, AFW and PASCAL Face datasets with

TABLE 13
The Results of Using Group Sampling Method in Fast R-CNN on WIDER FACE, Showing That the Proposed Method is Also Effective in Fast R-CNN

Method	Easy	Medium	Hard	All
FPN-finest-sampling	94.7	93.8	88.7	82.8
+Fast R-CNN	96.2	95.1	89.7	83.9
+IoU loss	96.4	95.3	90.3	84.6
+OHEM	96.1	95.0	90.4	85.3
+Focal Loss	95.9	95.0	90.7	85.5
+GS	96.2	95.5	91.1	85.7

TABLE 14
The Results of Using Group Sampling Method in Fast R-CNN on COCO, Showing That the Proposed Method is Also Effective in Fast R-CNN

Method	AP@0.5	AP	AP_s	AP_m	AP_l
Faster R-CNN	56.4	33.4	17.7	36.1	45.4
+IoU loss	56.4	33.5	18.1	36.4	45.2
+OHEM	54.9	32.0	17.0	35.4	44.3
+WRS	56.1	33.6	17.9	36.7	45.3
+GS	57.3	34.4	18.9	37.0	46.3
+GS +WRS	57.5	34.8	19.0	38.5	47.1

We adopte P_3 to extract features for detection. Note that the IoU loss is only applied in RPN because the proposals from RPN have a very small IoU loss for the second stage, which will results in performance drop compared with Smooth-L1 and L1 loss.

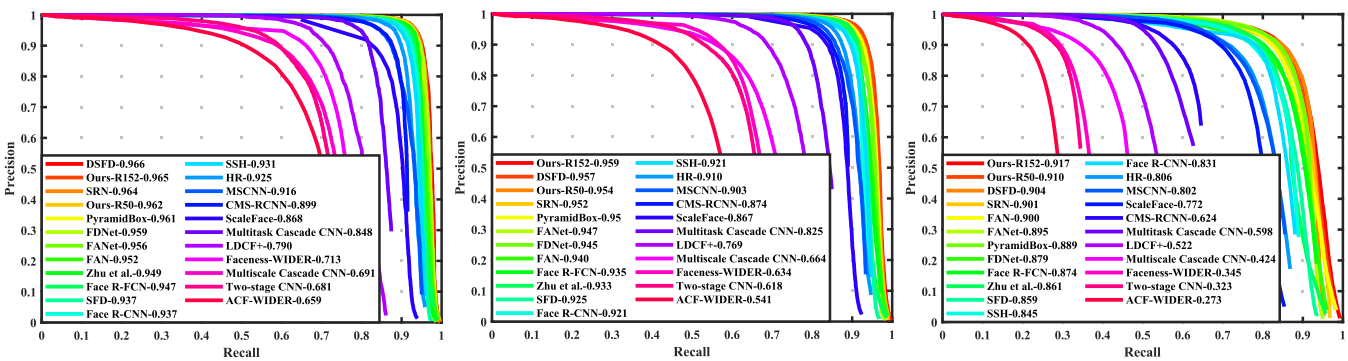
TABLE 15
Complexity Comparison of Our Model With Other Top Methods

Method	GFLOPs	#Params
HR [20]	93.0	30.0M
SSH [32]	208.5	19.8M
S^3FD [41]	201.0	22.5M
PyramidBox* [44]	229.1	67.3M
SRN [45]	394.6	53.4M
DSFD [42]	540.5	120.0M
RetinaFace [43]	164.5	61.9M
Ours(R18)	29.8	12.5M
Ours(R50)	81.3	27.1M
Ours(R152)	155.7	61.7M

We calculate the FLOPs by regarding the input image size as 800 × 800. Note that the PyramidBox [44] is built on ResNet-50 in this table. Other models use the default setting which is reported in their paper. Our model which based on ResNet-152 is a single stage detector only using the P_3 for detection and so does the same for our model based on ResNet-18.

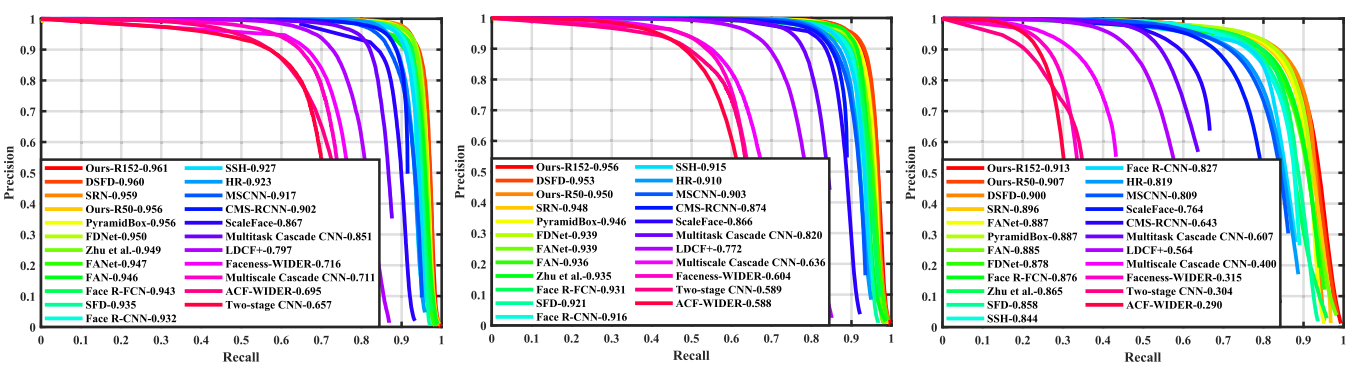
other face detection methods [20], [34], [41], [42], [44], [45], [57], [66], [80], [81], [82], [83], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], [100]. Besides ResNet-50, we also adopt ResNet-152 as backbone to further improve the performance, which is denoted as 'Ours-R152' in the figures. The implementation of using ResNet-152 as backbone differs from ResNet-50 in three aspects, 1) we adopt the feature map from P_3 for efficiency; 2) the initial learning rate is set to 0.02 with 20 epochs warming-up to speedup training; 3) we flip the test image in the inference process.

Results on WIDER FACE Dataset. WIDER FACE [34] has 393,703 faces in 32,203 images. The faces have high degree of variability in scale, pose and occlusion. All faces are divided into three subsets, i.e., Easy, Medium and Hard according to the difficulties of the detection. Our model based on group sampling is trained only on the training set and tested on the validation set and test set. The comparison results in terms of precision-recall curves and AP values are shown in Fig. 7. It can be seen on the validation set that our method based on ResNet-152 achieves 96.5, 95.9 and 91.7 percent on the three subsets, and outperforms all other methods on Hard subset by a large margin. On the test set, our method achieves best performance on all subsets as shown in Fig. 8. In addition, we show the complexity



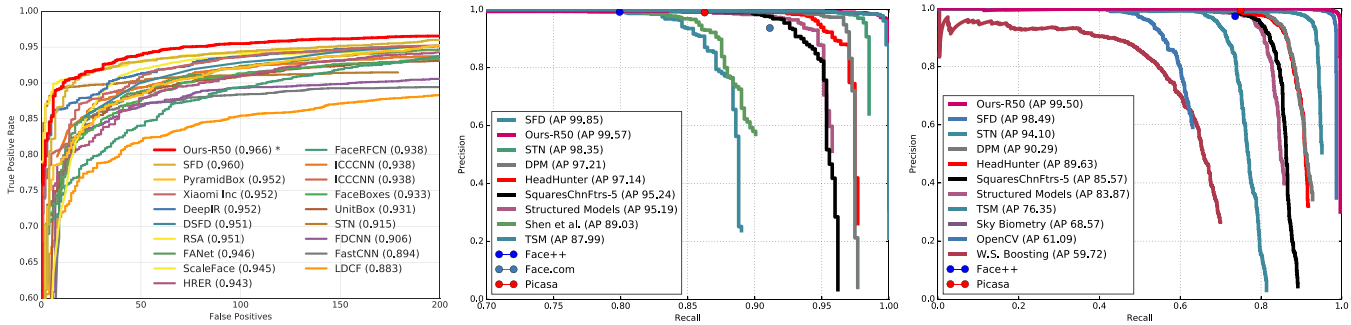
(a) Easy (b) Medium (c) Hard

Fig. 7. Performance comparison with state-of-the-arts in terms of precision-recall curves on WIDER FACE validation set.



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

Fig. 8. Performance comparison with state-of-the-arts in terms of precision-recall curves on WIDER FACE test set.



(a) FDDB (b) AFW (c) PASCAL Face

Fig. 9. Performance comparison with state-of-the-arts on FDDB, AFW and PASCAL Face datasets. Note that the values are calculated by setting the maximum number of false positives to 200 for FDDB.

Fig. 10. Illustration of the detection results of our approach for faces with a high degree of variability in scale, pose, and occlusion.

comparison with the top methods in terms of FLOPs and the number of parameters in Table 15. Fig. 10 illustrates the detected faces on some images sampled from WIDER FACE validation and test set.

Results on FDDB Dataset. FDDB [71] has 5,171 faces in 2,845 images. FDDB adopts the bounding ellipse for evaluation, while our method only outputs rectangle bounding boxes. Hence we adopt the regressor provided by S^3FD [41] to generate a bounding ellipse from our rectangle output. The performance comparison under discontinuous score is shown in Fig. 9a. We can see that our method achieves the best performance in terms of ROC curve.

Results on AFW. This dataset consists of 205 images with 473 annotated faces. The performance comparison between our approach and other detectors is presented in Fig. 9b. It can be seen that our approach achieves comparable performance with other state-of-the-art detectors.

Results on PASCAL Face. This dataset contains 851 images with 1,335 labeled faces. It is a subset of the PASCAL VOC testing set. Fig. 9c shows the comparison in terms of Precision-Recall curves. We can see that our approach advances the state-of-the-art performance.

8 Conclusion

In this paper, we examine the factors affecting detection performance and identify that the scale imbalance distribution is the key factor. Motivated by this observation, we propose a simple group sampling method to handle this issue. We show that the proposed method is effective in several existing frameworks, achieving better performance without extra computational cost. On several challenging face detection benchmarks, our method achieves state-of-the-art performance.

REFERENCES

[1] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem?(and a dataset of 230,000 3D facial landmarks)," in *Proc. Int. Conf. Comput. Vis.*, 2017, Art. no. 4.

[2] S. Zhu, C. Li, C.-C. Loy, and X. Tang, "Unconstrained face alignment via cascaded compositional learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3409–3417.

[3] A. Jourabloo and X. Liu, "Pose-invariant face alignment via CNN-based dense 3D model fitting," *Int. J. Comput. Vis.*, vol. 124, no. 2, pp. 187–203, 2017.

[4] A. Jourabloo, M. Ye, X. Liu, and L. Ren, "Pose-invariant face alignment with a single CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3219–3228.

[5] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at boundary: A boundary-aware face alignment algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2129–2138.

[6] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang, "FaceID-GAN: Learning a symmetry three-player GAN for identity-preserving face synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 821–830.

[7] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[8] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Towards open-set identity preserving face synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6713–6722.

[9] S. Zhu, S. Liu, C. C. Loy, and X. Tang, "Deep cascaded bi-network for face hallucination," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 614–630.

[10] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang, "FSRNet: End-to-end learning face super-resolution with facial priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2492–2501.

[11] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[12] H. Yang, D. Huang, Y. Wang, and A. K. Jain, "Learning face age progression: A pyramid architecture of GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[13] J. Yang *et al.*, "Neural aggregation network for video face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, Art. no. 7.

[14] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy, "Pose-robust face recognition via deep residual equivariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5187–5196.

[15] W. Wan, Y. Zhong, T. Li, and J. Chen, "Rethinking feature distribution for loss functions in image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9117–9126.

[16] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6738–6746.

[17] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," *IEEE Signal Process. Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[18] R. B. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[19] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[20] P. Hu and D. Ramanan, "Finding tiny faces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1522–1530.

[21] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection–SNIP," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3578–3587.

[22] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 447–456.

[23] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 845–853.

[24] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2874–2883.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[26] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, "Beyond skip connections: Top-down modulation for object detection," 2016, *arXiv:1612.06851*. [Online]. Available: https://arxiv.org/abs/1612.06851

[27] P. O. Pinheiro, T. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 75–91. [Online]. Available: https://doi.org/10.1007/978–3-319-46448-0_5

[28] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2129–2137.

[29] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 354–370.

[30] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[31] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 985–996, Apr. 2018.

[32] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "SSH: Single stage headless face detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4885–4894.

[33] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944. [Online]. Available: https://doi.org/10.1109/CVPR.2017.106

[34] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5525–5533.

[35] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[37] X. Ming, F. Wei, T. Zhang, D. Chen, and F. Wen, "Group sampling for scale invariant face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3441–3451.

[38] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[39] S. Honari, J. Yosinski, P. Vincent, and C. Pal, "Recombinator networks: Learning coarse-to-fine feature aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5743–5752.

[40] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.

[41] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S^3FD: Single shot scale-invariant face detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 192–201.

[42] J. Li *et al.*, "DSFD: Dual shot face detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5060–5069.

[43] J. Deng, J. Guo, Z. Yuxiang, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage dense face localisation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.

[44] X. Tang, D. K. Du, Z. He, and J. Liu, "PyramidBox: A context-assisted single shot face detector," in *Proc. 15th Eur. Conf. Comput. Vis.*, 2018, pp. 812–828. [Online]. Available: https://doi.org/10.1007/978–3-030-01240-3_49

[45] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "Selective refinement network for high performance face detection," *CoRR*, vol. abs/1809.02693, 2018. [Online]. Available: http://arxiv.org/abs/1809.02693

[46] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[47] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, "Revisiting RCNN: On awakening the classification power of faster RCNN," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 453–468.

[48] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[49] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, 2005, pp. 878–887.

[50] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[51] S.-J. Yen and Y.-S. Lee, "Cluster-based under-sampling approaches for imbalanced data distributions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5718–5727, 2009.

[52] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 20–29, 2004.

[53] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[54] T. M. Hospedales, S. Gong, and T. Xiang, "Finding rare classes: Active learning with generative and discriminative models," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 374–386, Feb. 2013.

[55] X. Zhu, Y. Liu, J. Li, T. Wan, and Z. Qin, "Emotion classification with data augmentation using generative adversarial networks," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2018, pp. 349–360.

[56] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Syst. Appl.*, vol. 91, pp. 464–471, 2018.

[57] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5325–5334.

[58] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, Jan. 2006.

[59] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl, "Sampling matters in deep embedding learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2840–2848.

[60] A. Shrivastava, A. Gupta, and R. B. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 761–769.

[61] P. Efraimidis and P. P. Spirakis, *Weighted Random Sampling*. New York, NY, USA: Springer, 2016, pp. 2365–2367. [Online]. Available: https://doi.org/10.1007/978–1-4939-2864-4_478

[62] W. Shi, F. Jiang, and D. Zhao, "Single image super-resolution with dilated convolution based multi-scale information learning inception module," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 977–981.

[63] S. Yang, G. Lin, Q. Jiang, and W. Lin, "A dilated inception network for visual saliency prediction," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 2163–2176, 2020.

[64] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3047–3055.

[65] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.

[66] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. S. Huang, "UnitBox: An advanced object detection network," in *Proc. ACM Conf. Multimedia Conf.*, 2016, pp. 516–520. [Online]. Available: https://doi.org/10.1145/2964284.2967274

[67] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[68] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[69] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[70] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Adv. Neural Inform. Process. Syst. 32*, 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[71] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. UM-CS-2010–009, 2010.

[72] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.

[73] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[74] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[75] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 840–849.

[76] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9627–9636.

[77] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "FoveaBox: Beyond anchor-based object detector," *IEEE Trans. Image Process.*, vol. 29, pp. 7389–7398, 2020.

[78] H. Law and J. Deng, "CornerNet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 734–750.

[79] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6569–6578.

[80] J. Zhang, X. Wu, J. Zhu, and S. C. H. Hoi, "Feature agglomeration networks for single stage face detection," *CoRR*, vol. abs/1712.00721, 2017. [Online]. Available: http://arxiv.org/abs/1712.00721

[81] J. Wang, Y. Yuan, and G. Yu, "Face attention network: An effective face detector for the occluded faces," *CoRR*, vol. abs/1711.07246, 2017. [Online]. Available: http://arxiv.org/abs/1711.07246

[82] Y. Wang, X. Ji, Z. Zhou, H. Wang, and Z. Li, "Detecting faces using region-based fully convolutional networks," *CoRR*, vol. abs/1709.05256, 2017. [Online]. Available: http://arxiv.org/abs/1709.05256

[83] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: An improved faster RCNN approach," *Neurocomputing*, vol. 299, pp. 42–50, 2018. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.03.030

[84] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu, "Detecting faces using inside cascaded contextual CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3190–3198.

[85] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: A CPU real-time face detector with high accuracy," in *Proc. IEEE Int. Joint Conf. Biometrics*, 2017, pp. 1–9.

[86] S. Yang, Y. Xiong, C. C. Loy, and X. Tang, "Face detection through scale-friendly deep convolutional networks," *CoRR*, vol. abs/1706.02863, 2017. [Online]. Available: http://arxiv.org/abs/1706.02863

[87] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016. [Online]. Available: https://doi.org/10.1109/LSP.2016.2603342

[88] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3676–3684.

[89] Y. Li, B. Sun, T. Wu, and Y. Wang, "Face detection with end-to-end integration of a ConvNet and a 3D model," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 420–436. [Online]. Available: https://doi.org/10.1007/978–3-319-46487-9_26

[90] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, pp. 109–122. [Online]. Available: https://doi.org/10.1007/978–3-319-10599-4_8

[91] 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," *CoRR*, vol. abs/1603.01249, 2016. [Online]. Available: http://arxiv.org/abs/1603.01249

[92] H. Jiang and E. G. Learned-Miller, "Face detection with the faster R-CNN," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit.*, 2017, pp. 650–657.

[93] S. S. Farfade, M. J. Saberian, and L. Li, "Multi-view face detection using deep convolutional neural networks," in *Proc. 5th ACM Int. Conf. Multimedia Retrieval*, 2015, pp. 643–650. [Online]. Available: http://doi.acm.org/10.1145/2671188.2749408

[94] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Detecting and localizing occluded faces," *CoRR*, vol. abs/1506.08347, 2015. [Online]. Available: http://arxiv.org/abs/1506.08347

[95] J. Li and Y. Zhang, "Learning SURF cascade for fast and accurate object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3468–3475.
[96] H. Wang, Z. Li, X. Ji, and Y. Wang, "Face R-CNN," *CoRR*, vol. abs/1706.01061, 2017. [Online]. Available: http://arxiv.org/abs/1706.01061
[97] C. Zhu, R. Tao, K. Luu, and M. Savvides, "Seeing small faces from robust anchor's perspective," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5127–5136.
[98] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "CMS-RCNN: Contextual multi-scale region-based CNN for unconstrained face detection," *CoRR*, vol. abs/1606.05413, 2016. [Online]. Available: http://arxiv.org/abs/1606.05413
[99] E. Ohn-Bar and M. M. Trivedi, "To boost or not to boost? On the limits of boosted trees for object detection," in *Proc. 23rd Int. Conf. Pattern Recognit.*, 2016, pp. 3350–3355.
[100] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Proc. IEEE Int. Joint Conf. Biometrics*, 2014, pp. 1–8.

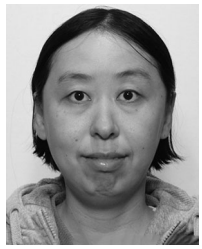
Xiang Ming received the BS degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2013. He is currently working toward the PhD degree in computer vision.

Fangyun Wei received the BS degree from Shandong University, Jinan, China, in 2014, and the MS degree from Peking University, Beijing, China, in 2017. In July 2017, he joined Microsoft Research working on face detection and recognition. His research interest includes computer vision.

Ting Zhang received the BS and PhD degrees from the University of Science and Technology of China, Hefei, China, in 2012 and 2017. She joined Visual Computing Group, Microsoft Research in July 2017. Her research interests include computer vision and machine learning.

Dong Chen received the BS and PhD degrees from the University of Science and Technology of China, Hefei, China, in 2010 and 2015. He joined Microsoft Research in July 2015 as a lead researcher. His research interests include face detection, recognition, and generative models.

Nanning Zheng (Fellow, IEEE) received the graduate degree from the Department of Electrical Engineering, Xi'an Jiaotong University (XJTU), Xi'an, China, in 1975, the ME degree in information and control engineering from Xi'an Jiaotong University, Xi'an, China, in 1981, and the PhD degree in electrical engineering from Keio University, Tokyo, Japan, in 1985. He is currently a professor and the director with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, autonomous vehicle, and brain-inspired computing. Since 2000, he has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He became a member of the Chinese Academy Engineering, in 1999. He is also the president of the Chinese Association of Automation(CAA).



Fang Wen received the BS degree in automation from Tsinghua University, Beijing, China, and the MS and PhD degrees in pattern recognition and intelligent system, in 1997 and 2003, respectively. Currently, she is a senior researcher of Visual Computing Group, Microsoft Research. Her research interests include computer vision, pattern recognition, and multimedia search.