Group Sampling for Scale Invariant Face Detection

Xiang Ming[1*] Fangyun Wei[2] Ting Zhang[2] Dong Chen[2] Fang Wen[2]
Xi'an Jiaotong University[1] Microsoft Research Asia[2]
xjtustu.mx@stu.xjtu.edu.cn {fawe,tinzhan,doch,fangwen}@microsoft.com

Abstract

Detectors based on deep learning tend to detect multi-scale faces 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 such multi-layer prediction is not necessary. Faces at all scales can be well detected with features from a single layer of the network. In this paper, we carefully examine the factors affecting face detection 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 the last 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. Our approach, evaluated on face detection benchmarks including FDDB and WIDER FACE datasets, achieves state-of-the-art results without bells and whistles.

1. Introduction

Face detection is the key step of many subsequent face related applications, such as face alignment [5, 77, 27, 28, 60], face synthesis [48, 1, 2, 78, 10, 24, 62] and face recognition [63, 7, 55, 39, 56]. Among the various factors that confront real-world face detection, extreme scale variations and small facial remain a big challenge.

Previous deep learning detectors detect multi-scale faces on a single feature map, e.g., Fast R-CNN [15] and Faster R-CNN [46]. 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 [46]), making small faces difficult to match the appropriate anchor and thus have few positive samples during training.

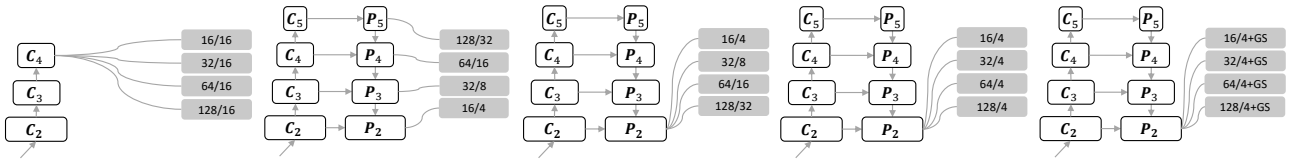
To alleviate these problems arising from scale variation and small object instances, multiple solutions have been proposed, including: 1) using image pyramid for training and inference [22, 51]; 2) combining features from shallow and deep layers for prediction [17, 29, 4]; 3) using top-down and skip connections to produce a single high-level feature map with fine resolution [47, 50, 44]; 4) using multiple layers with different resolutions to predict object instances of different scales [61, 6, 38, 31, 41, 35]. All of these solutions significantly improve the performance of detectors. Among them, adopting several layers with different resolutions for prediction is the most popular one, since it achieves better performance, especially for detecting small objects.

It is 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 a single layer. However, we find that this is not the case, at least for face detection. We observe that making predictions on multiple layers will produce different numbers of anchors for different scales[1], which is the reason why pyramid features outperform single layer feature instead of the pyramid representation, and this factor is overlooked in the comparison between pyramid features and single layer feature conducted in FPN [35]. Empirically we show that single layer predictions, if imposed with the same number of anchors as that in FPN [35], achieve almost the same accuracy.

Motivated by this observation, we carefully examine the factors affecting face detection performance through extensive empirical analysis and identify a key issue in existing anchor based face detectors, i.e., *the anchors sampled at different scales are imbalanced*. To show this, we use two representative detection architectures, the Region Proposal Network (RPN) in Faster R-CNN [46] and FPN [35], as examples. Figure 1 illustrates the network architectures. We calculate the number of training samples received by anchors at each scale during the training process and report them in Figure 2 (a) and (b) for RPN and FPN, respectively.

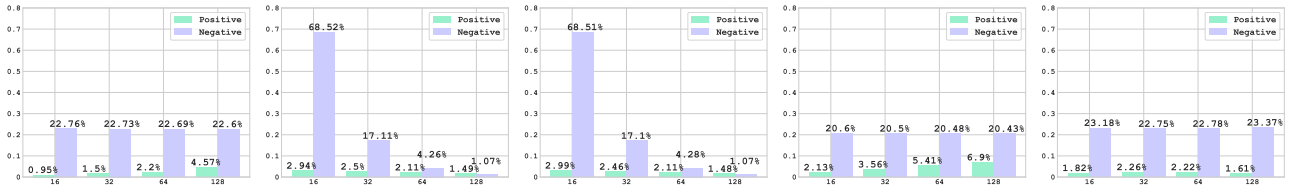
*Work done during the internship at Microsoft Research Asia.

[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

Figure 1: Illustration of network architectures of five different detectors described in Section 3. The term "$16/4$" associated with the feature map denote 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) used the proposed group sampling in (d).



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

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

For RPN, since the strides of anchors for different scales are the same, the small anchors are more difficult to match an appropriate labeled object, resulting in fewer positive samples and thus lower accuracy for small objects. On the other hand, FPN uses high-resolution feature maps for small object prediction, this gives rise to several more times negative training samples for small objects than large objects.

Such trained classifiers would get higher accuracy for small objects than RPN like detectors, which might be the reason why FPN based methods perform better on the COCO detection benchmark [37] and the WIDER FACE database [65], where small objects are dominant. We further report the number of training samples when only using the last layer of FPN, which is shown in Figure 2 (d). The distribution is similar to RPN, though the absolute values are different. Empirically we show that similar to RPN, predictions made only on the last layer of FPN get lower accuracy for small objects due to the insufficient positive samples.

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 positive samples as well as the same number of negative samples at each scale during each iteration of the training process. Thus the classifier is fed with balanced training samples at different scales. Figure 2 (e) shows the anchor distribution of our method, where the distribution becomes more balanced. With our approach, only using the feature map on the finest level of FPN with group sampling is able to achieve better performance than FPN on WIDER FACE [65].

In addition, we notice that the second stage of the Faster R-CNN [46] also suffers from data imbalance issue, since the scale distribution of the objects in the dataset is imbalanced. We show that our proposed method can be used here by evenly sampling the features after RoI Pooling 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, which challenges our understanding of FPN; (2) We further carefully examine the factors affecting detection performance and identify the key issue in existing anchor-based detectors: the anchors are imbalanced at different scales; (3) We present a simple and straightforward solution to handle that issue and the proposed solution achieves noticeable effect on detection performance.

2. Related Work

Single scale feature for multi-scale detection. Modern detectors, such as Fast R-CNN [15] and Faster R-CNN [46], 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 [47] and SharpMask [44] for semantic segmentation, Recombinator Network [20] for face detection, and Stacked Hourglass Network [42] 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 [38], MS-CNN [6] and FPN [35]. FPN also exploits the top-down architecture with lateral connections for strong feature representation. Such multi-scale feature representation is also widely used for face detection to improve accuracy, e.g., SSH [41] and S^{3}FD [75]. However, we show that the improvement comes from the anchor distribution toward 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 [8, 16, 18, 68, 3], e.g., under-sampling instances of the majority classes [40] or over-sampling the instances of the minority classes with generative and discriminative models [21, 79, 12]. 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 [54, 76]. For detection, the imbalanced scale distribution can also be regarded as one kind of class imbalance issue. Previous works using hard example mining [49] or carefully designed loss functions [36] have implicitly mitigated this issue to some degree. For instance, S^{3}FD [75] 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 point out that the scale imbalance distribution over not only the positive samples but also the negative samples is a key issue and propose a simple group sampling method to explicitly handle it, resulting in better detection accuracy.

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 [19] combined with top down and skip connections. Figure 3 briefly illustrates the network structure. 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 addition. This process is repeated 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 $\{16, 32, 64, 128\}$ and the aspect ratio is set as 1. Based on this network architecture, we compare five types of detectors:

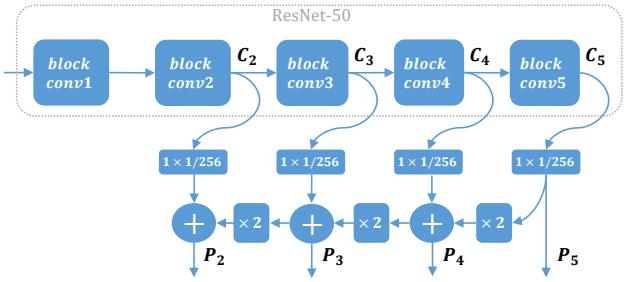


Figure 3: Network architecture used in our experiments, $\times 2$ is bilinear upsampling and $\bigoplus$ is element wise summation.

1. RPN: The feature map C_4 is used as the detection layer where all anchors are tiled with stride 16 pixels.

2. FPN: $\{P_2, P_3, P_4, P_5\}$ are used as the detection layers with the anchor scales $\{16, 32, 64, 128\}$ corresponding to feature stride $\{4, 8, 16, 32\}$ pixels respectively.

3. FPN-finest-stride: All anchors are tiled on the finest layer of the feature pyramid, i.e., P_2. The stride is $\{4, 8, 16, 32\}$ pixels for anchors with scale $\{16, 32, 64, 128\}$, respectively. This is implemented by sub-sampling P_2 for larger strides.

4. FPN-finest: All anchors are also tiled on P_2. The stride for each anchor is 4 pixels.

5. FPN-finest-sampling: This adopts the same setting with FPN-finest. Additionally, we use the proposed group sampling method to balance the training samples for different scales.

To ensure fair comparison, all detectors use the same setting for both training and inference on the challenging WIDER FACE dataset [65]. The results are evaluated on the WIDER FACE validation dataset. we have following observations.

Using multiple layer features is little helpful. 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. The Average Precision (AP) of FPN is 90.9%, 91.3%, 87.6% for easy, medium and hard subsets respectively. In contrast, the results for FPN-finest-stride are 90.4%, 91.0%, 87.1%. The results are comparable, showing that using single layer feature is sufficient for face detection.

Scale imbalance distribution matters. We further compare FPN-finest-stride with FPN-finest, which set the stride for all different anchors as 4 pixels. We observe that 1) FPN-finest gets better performance on easy and medium subsets as expected, since more training examples for large anchors are selected than FPN-finest-stride, and 2) loses 1.1% AP on hard subset, even though FPN-finest has the same number of anchors for scale 16. To find out the reason behind, we plot the proportions of training positive samples and negative ones at different scales for all the compared detectors in Figure 2 and show the AP results in Table 1.

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 Figure 2 (b) and (c), suggesting that similar distribution, when the total number of anchors is the same, gives rise to similar performance.

Second, as shown in Figure 2 (c) and (d), the sample distribution at different scales for both FPN-finest-stride and FPN-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. RPN as shown in Figure 2 (a) also has more large positive anchors and gets 2.0% higher on easy subset compared with FPN-finest-stride, future supporting our hypothesis.

Motivated by above observations, we propose a group sampling method to handle the scale imbalance distribution. Figure 2 (e) shows that the anchor distribution of FPN-finest-sampling which uses the proposed group sampling method during training is more balanced, and as a result, FPN-finest-sampling achieves the best performance.

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.

4.1. Anchor Matching Strategy

Current anchor matching strategies usually follow a *two-pass* policy, which has been widely used in detection works [46, 38]. 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 he 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., $\mathbf{M}(i,j) = \text{IoU}(p_i, g_j)$.

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 represents the positive samples, 0 represents the negative samples, and -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) $\text{IoU}(p_i, g_j) \ge \lambda_2$; 3) $j = \operatorname{argmax}_{1\le u\le m} \text{IoU}(p_i, g_u)$.

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: the number of negative samples in the image is much greater than the number of positive samples due to the nature of object detection task.

- Samples at different scales are not balanced: small objects are more difficult to find a suitable anchor than large objects due to IoU based matching policy.

Previous methods often notice the first point and usually handle it by hard negative example mining, e.g., the positive and negative sample ratio is set to 1:3 when sampling training examples. But they all ignored the second point.

To handle above two issues, we propose a scale aware sampling strategy called *group sampling*. We first divide all the training samples into several groups according to the anchor scale, i.e., 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 remains the same.

Formally, let $\mathcal{P}_s$ and $\mathcal{N}_s$ represent 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 $\mathcal{N}_s \subseteq \{p_i|\mathcal{L}(i) = 0, \mathcal{S}(i) = s\}$. 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.

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%). 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. Therefore, we use group sampling here to ensure that the number of training samples in each group is the same, and the ratio for positive and negative sample remains 1:3. 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*.

Relation to OHEM and Focal Loss. Online hard example mining (OHEM) [38] is to keep the top K samples with highest training loss for back propagation. Focal Loss [36] proposes giving each sample a specific weight. Both seem similar to the cost-sensitive learning often used in addressing data imbalance by penalizing the misclassifications of the minority class more heavily. However, the weight for each sample in OHEM and Focal Loss is set with respect to the sample's loss in a hard/soft manner, which can be viewed as an implicit and dynamic way of handling data imbalance. Our approach, on the other hand, is able to explicitly handle the data imbalance for different scales and achieve better performance as shown in Table 4.

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 [46].

Training dataset. As with previous works [38, 75], we train our models on the WIDER FACE training set which contains $12,880$ images and test on the WIDER FACE validation and testing set, as well as the FDDB dataset.

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

$$\mathcal{L}_{reg} = \frac{1}{N_{reg}} \sum_{(p_i, g_j)} \|1 - \text{IoU}(p_i, g_j)\|_2^2, \qquad (2)$$

where (p_i, g_j) is a matching pair of an anchor p_i and a ground-truth g_j. Compared to smooth-L1 loss, this loss function directly optimizes the IoU ratio, which is consistent with the evaluation metric. Another IoU based loss function is $-\ln(\text{IoU})$ proposed in [69]. It is clear that when IoU equals to 1, which is the ideal case, previous IoU loss will get non-zero gradient, while our IoU least square loss gets zero gradient, allowing the network to converge stably. Empirically we show that the proposed IoU loss achieves better performance.

Optimization details. All models are initialized with the pre-trained weights of ResNet-50 provided by `torchvision`[2] and fine-tuned on the WIDER FACE training set. Each training iteration contains one image per GPU for an 8 NVIDIA Tesla M40 GPUs server. We set the initial learning rate to 0.01 and decrease the learning rate by 0.1 at 60th and 80th epoch. All the models are trained for 100 epochs by synchronized SGD. The momentum and weight decay is set to 0.9 and 5×10^{-5} respectively. Our code is based on PyTorch [43].

During training, we use scale jitter and random horizontal flip for data augmentation. For scale jitter, each image will be resized by $0.25 \times n$, and n is randomly chosen from $[1, 8]$. Then we 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. We set $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$ for the two-pass anchor matching policy. At the inference stage, we build the image pyramid for multi-scale test. The proposals from each level of the image pyramid will be merged by Non-Maximum Suppression (NMS). Due to the GPU memory limitation, each side of the test image will not exceed 3,000 pixels.

[2] https://github.com/pytorch/vision

Table 1: Average Precision (AP) of face detection on WIDER FACE validation set. GS represents the proposed group sampling method. @16 represents the AP on all data when only using outputs from the sub-dector at scale 16 for detection. So dose @32, @64 and @128.

Methods	Feature	Anchor Stride	GS	Easy	Medium	Hard	All	@16	@32	@64	@128
RPN	C_4	16		92.5	91.0	83.0	74.0	48.6	65.1	43.8	21.5
FPN-finest	P_2	4		94.1	93.0	86.6	80.2	65.6	66.8	43.9	22.4
FPN	$\{P_2, P_3, P_4, P_5\}$	{4, 8, 16, 32}		90.9	91.3	87.6	82.1	72.3	67.3	43.2	21.2
FPN-finest-stride	P_2	{4, 8, 16, 32}		90.4	91.0	87.1	81.6	72.2	66.6	43.3	21.8
FPN-finest-sampling	P_2	4	✓	**94.7**	**93.8**	**88.7**	**82.8**	**74.1**	**72.9**	**47.8**	**24.5**

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 that our approach using single layer predictions advances the state-of-the-arts on WIDER FACE [65] and FDDB [25] 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 Section 3. There are two differences among them: 1) the feature map on which the anchors are tiled; 2) the stride for different anchors. The stride of an anchor indicates the number of anchors and smaller stride gives rise to more anchors. Usually, the size of the feature map will have a corresponding anchor stride with respect to the original image. For FPN-finest-stride, we tile the anchors of scale $\{16, 32, 64, 128\}$ with the stride of $\{1, 2, 4, 8\}$ on the feature map P_2, equivalent to stride of $\{4, 8, 16, 32\}$ on the original image.

We adopt Average Precision (AP) as the evaluation metric. Previous methods usually report AP on the easy, medium and hard subsets for evaluation. However, the results cannot reflect the ability of the sub-detector which is used to handle objects in a specific scale range. This is because a large face (e.g., 128×128 pixels) which is usually detected by the anchor with scale 128, is possible to be actually detected by the anchor with scale 16 due to the multi-scale test. Therefore, to clearly show the ability of each sub-detector, we also report the performance of 4 sub-detectors in our model on the 'All' subsets and they are denoted as @16, @32, @64 and @128. The performance comparison is shown in Table 1. We have following observations:

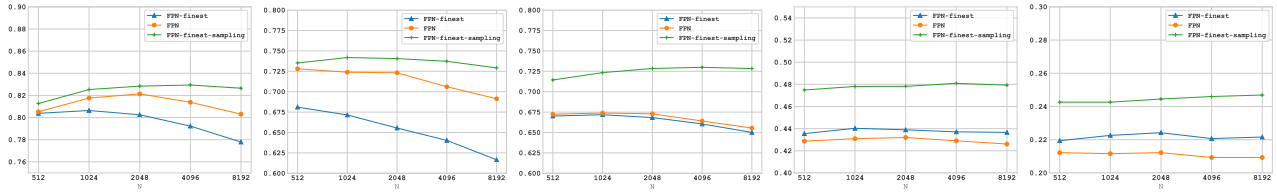
Imbalanced training data at scales leads to worse (better) accuracy for the minority (majority). The only difference between FPN-finest and FPN-finest-stride is the anchor stride, i.e., the number of anchors for different scales are different. For scale 16, its stride is the same in the two models. Therefore, the number of anchors at scale 16 is also the same. However, this is not the case for performance over @16. FPN-finest-tride achieves 72.3%, 6.7% 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, 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.

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 the features used for detection coming from multiple layers or a single layer. 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 ?* Consider another comparison 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 and FPN-finest-sampling, which adopts the proposed group sampling method in FPN-finest, the only difference is the distribution of the training data at each scale. We can see that using more evenly distributed training data can significantly improve the results, increasing from 80.2% to 82.8% on the whole dataset.

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 2 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 and lateral connections to provide more 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 detecting small faces; 3) regardless of the feature maps, using group sampling can always improve the results. For the sake of simplicity, we use P_2 as the feature in our final model.



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

Figure 4: 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.

Table 2: Comparison of models with/without group sampling using different feature maps.

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	80.3
	✓	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

Table 3: Comparison of different loss function for the regression task. The proposed loss function performs better.

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**

Table 4: Performance comparison of the proposed group sampling, OHEM and Focal Loss, showing that our approach achieves better performance.

Method	@16	@32	@64	@128	All
FPN-finest (baseline)	65.6	66.8	43.9	22.4	80.2
OHEM	**76.0**	68.9	43.9	22.0	81.5
Focal Loss	75.8	68.5	44.2	21.5	81.2
Group Sampling	74.1	**72.9**	**47.8**	**24.5**	**82.8**

The effect of the number of training samples N. As introduced in Section 4.2, we randomly choose N training samples for each scale during training. The performance under different N is shown in Figure 4. It can be seen that: 1) the performance gets better when N increases; 2) the accuracy gets saturated when N is greater than 2048. 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.

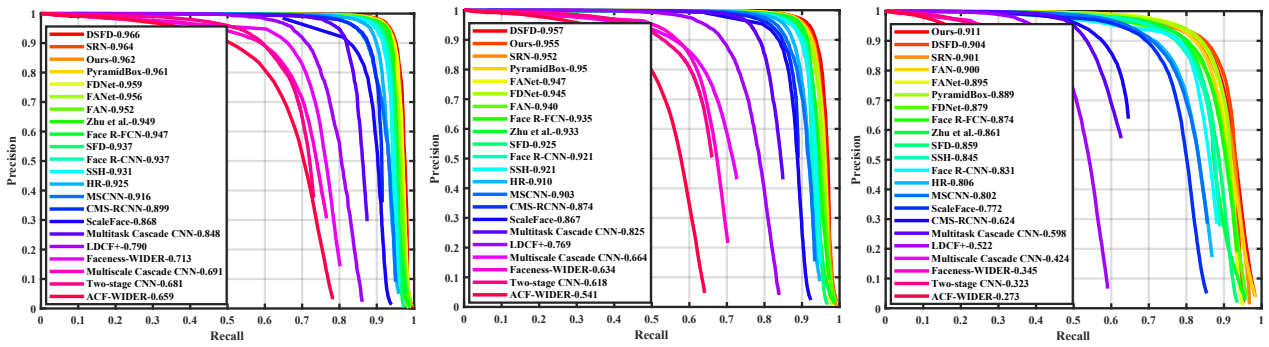
The effect of the proposed loss. We propose a new IoU based loss for regression, namely least square IoU loss, to allow the network converge stably. Here we compare different loss functions, including Smooth-L1, $-\ln(\text{IoU})$ and $\|1-\text{IoU}\|_2^2$. The detector we used is FPN-finest-sampling. The comparison results are shown in Table 3. We can see that the two IoU based loss functions perform better then Smooth-L1, as they directly optimize the evaluation metric. Compared with $-\ln(\text{IoU})$, our proposed least square IoU loss achieves better performance.

Comparison with OHEM and Focal Loss. Here we compare our approach with two methods: OHEM [49] and Focal loss [36], both adopting hard example mining, which can be regarded as a way of handling data imbalance. 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, we adopt the same setting in [36], in which $\alpha = 0.25$ and $\gamma = 2$.

The performance comparison is shown in Table 4. Both OHEM and Focal Loss can effectively improve the performance of detecting small faces. Take sub-detector @16 as an example, OHEM and Focal Loss achieve 76.0% and 75.8% respectively, about 10% higher than the baseline model. However, the performance of sub-detectors for large scales decrease. For example, the performance of sub-detector @128 is worse than the baseline. In contrast, our approach gets improvement for all the sub-detectors compared with the baseline by simply using the proposed group sampling method, and also achieves better performance on the whole dataset compared with OHEM and Focal Loss.

6.3. Grouped Fast R-CNN

We show that the proposed group sampling method can be applied to Fast R-CNN to further improve the detection accuracy. We use FPN-finest-sampling as the baseline model. The AP is increased from 82.8% to 83.9% through



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

Figure 5: Performance comparison with state-of-the-arts in terms of precision-recall curves on WIDER FACE validation set.

Table 5: The results of using group sampling method in Fast R-CNN, showing that the proposed method is also effective in Fast R-CNN.

Method	Easy	Medium	Hard	All
FPN-finest-sampling	95.1	94.1	88.8	82.7
+Fast R-CNN	96.2	95.1	89.7	83.9
+IoU loss	96.4	95.3	90.3	84.6
+ Group Sampling	96.2	95.5	91.1	85.7

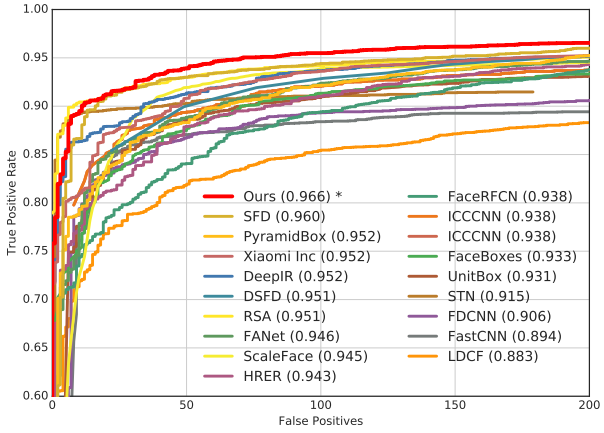


Figure 6: Performance comparison with state-of-the-arts on FDDB dataset. The values are calculated by setting the maximum number of false positives to 200.

directly using Fast R-CNN as shown in Table 5. After using the proposed least square IoU loss as described in Equation 2, we can increase the AP for easy, medium and hard subsets by 0.2%, 0.2% and 0.6% respectively. Next, we further adopt the group sampling method and the AP for easy, medium and hard subsets are 96.3%, 95.6% and 91.2%, which advances the state-of-the-art on WIDER FACE validation set. It is worth noting that using group sampling gets 0.9% improvement on hard set, which is significant as the accuracy after using the least square IoU loss is already 90.3%. The model shown in the last line of Table 5 is our final model, which is used to compare with other methods.

7. Comparison with State-of-the-Art

We compare our approach on two benchmark datasets: WIDER FACE and FDDB datasets with other face detection methods [53, 11, 71, 58, 32, 59, 75, 52, 23, 73, 74, 67, 70, 72, 64, 66, 34, 9, 45, 26, 13, 14, 30, 33, 57].

Results on WIDER FACE dataset. WIDER FACE [65] 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. The comparison results in terms of precision-recall curves and AP values are shown in Figure 5. It can be seen that our method achieves 96.2%, 95.7% and 91.1% on the three subsets, and outperforms all other methods on Hard subset by a large margin.

Results on FDDB dataset. FDDB [25] 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 [75] to generate a bounding ellipse from our rectangle output. The performance comparison under discontinuous score is shown in Figure 6. We can see that our method achieves the best performance in terms of ROC curve.

8. Conclusion

In this paper, we examine the factors affecting detection accuracy and identify that the scale imbalance distribution is the key factor. Motivated by this observation, we propose a simple group sampling method to handle the sample imbalance across different scales. We show that the proposed method is effective in existing frameworks, e.g., Faster R-CNN and FPN, achieving better performance without extra computational cost. On the challenging benchmarks like WIDER FACE and FDDB, our method achieves state-of-the-art performance. In future work, we tend to verify the effectiveness of group sampling in general object detection.

References

[1] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Cvae-gan: fine-grained image generation through asymmetric training. *CoRR, abs/1703.10155*, 5, 2017. 1

[2] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Towards open-set identity preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6713–6722, 2018. 1

[3] 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 newsletter*, 6(1):20–29, 2004. 3

[4] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016. 1

[5] 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 *International Conference on Computer Vision*, volume 1, page 4, 2017. 1

[6] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016. 1, 3

[7] K. Cao, Y. Rong, C. Li, X. Tang, and C. C. Loy. Pose-robust face recognition via deep residual equivariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5187–5196, 2018. 1

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 3

[9] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, pages 109–122, 2014. 8

[10] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2492–2501, 2018. 1

[11] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou. Selective refinement network for high performance face detection. *CoRR*, abs/1809.02693, 2018. 8

[12] G. Douzas and F. Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with applications*, 91:464–471, 2018. 3

[13] S. S. Farfade, M. J. Saberian, and L. Li. Multi-view face detection using deep convolutional neural networks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, June 23-26, 2015*, pages 643–650, 2015. 8

[14] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Detecting and localizing occluded faces. *CoRR*, abs/1506.08347, 2015. 8

[15] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 1, 2

[16] H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005. 3

[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015. 1

[18] H. He, Y. Bai, E. Garcia, and S. A. Li. Adaptive synthetic sampling approach for imbalanced learning. ieee international joint conference on neural networks, 2008 (ieee world congress on computational intelligence), 2008. 3

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[20] S. Honari, J. Yosinski, P. Vincent, and C. Pal. Recombinator networks: Learning coarse-to-fine feature aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5743–5752, 2016. 3

[21] T. M. Hospedales, S. Gong, and T. Xiang. Finding rare classes: Active learning with generative and discriminative models. *IEEE transactions on knowledge and data engineering*, 25(2):374–386, 2013. 3

[22] P. Hu and D. Ramanan. Finding tiny faces. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1522–1530. IEEE, 2017. 1

[23] P. Hu and D. Ramanan. Finding tiny faces. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1522–1530, 2017. 8

[24] R. Huang, S. Zhang, T. Li, R. He, et al. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. *arXiv preprint arXiv:1704.04086*, 2017. 1

[25] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010. 6, 8

[26] H. Jiang and E. G. Learned-Miller. Face detection with the faster R-CNN. In *12th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2017, Washington, DC, USA, May 30 - June 3, 2017*, pages 650–657, 2017. 8

[27] A. Jourabloo and X. Liu. Pose-invariant face alignment via cnn-based dense 3d model fitting. *International Journal of Computer Vision*, 124(2):187–203, 2017. 1

[28] A. Jourabloo, M. Ye, X. Liu, and L. Ren. Pose-invariant face alignment with a single cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 3219–3228. IEEE, 2017. 1

[29] T. Kong, A. Yao, Y. Chen, and F. Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 845–853, 2016. 1

[30] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5325–5334, 2015. 8

[31] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan. Scale-aware fast r-cnn for pedestrian detection. *IEEE Transactions on Multimedia*, 20(4):985–996, 2018. 1

[32] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang. DSFD: dual shot face detector. *CoRR*, abs/1810.10220, 2018. 8

[33] J. Li and Y. Zhang. Learning SURF cascade for fast and accurate object detection. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 3468–3475, 2013. 8

[34] Y. Li, B. Sun, T. Wu, and Y. Wang. Face detection with end-to-end integration of a convnet and a 3d model. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, pages 420–436, 2016. 8

[35] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 1, 3

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 3, 5, 7

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 3, 4, 5

[39] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2017. 1

[40] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009. 3

[41] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis. Ssh: Single stage headless face detector. In *ICCV*, pages 4885–4894, 2017. 1, 3

[42] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 3

[43] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017. 5

[44] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 1, 3

[45] 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*, abs/1603.01249, 2016. 8

[46] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 4, 5

[47] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1, 3

[48] Y. Shen, P. Luo, J. Yan, X. Wang, and X. Tang. Faceidgan: Learning a symmetry three-player gan for identity-preserving face synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–830, 2018. 1

[49] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016. 3, 7

[50] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*, 2016. 1

[51] B. Singh and L. S. Davis. An analysis of scale invariance in object detection–snip. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3578–3587, 2018. 1

[52] X. Sun, P. Wu, and S. C. H. Hoi. Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing*, 299:42–50, 2018. 8

[53] X. Tang, D. K. Du, Z. He, and J. Liu. Pyramidbox: A context-assisted single shot face detector. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IX*, pages 812–828, 2018. 8

[54] K. M. Ting. A comparative study of cost-sensitive boosting algorithms. In *In Proceedings of the 17th International Conference on Machine Learning*. Citeseer, 2000. 3

[55] W. Wan, Y. Zhong, T. Li, and J. Chen. Rethinking feature distribution for loss functions in image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9117–9126, 2018. 1

[56] F. Wang, W. Liu, H. Liu, and J. Cheng. Additive margin softmax for face verification. *arXiv preprint arXiv:1801.05599*, 2018. 1

[57] H. Wang, Z. Li, X. Ji, and Y. Wang. Face R-CNN. *CoRR*, abs/1706.01061, 2017. 8

[58] J. Wang, Y. Yuan, and G. Yu. Face attention network: An effective face detector for the occluded faces. *CoRR*, abs/1711.07246, 2017. 8

[59] Y. Wang, X. Ji, Z. Zhou, H. Wang, and Z. Li. Detecting faces using region-based fully convolutional networks. *CoRR*, abs/1709.05256, 2017. 8

[60] W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2138, 2018. 1

[61] 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 *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 2129–2137, 2016. 1

[62] H. Yang, D. Huang, Y. Wang, and A. K. Jain. Learning face age progression: A pyramid architecture of gans. *arXiv preprint arXiv:1711.10352*, 2017. 1

[63] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *CVPR*, volume 4, page 7, 2017. 1

[64] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3676–3684, 2015. 8

[65] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5533, 2016. 2, 3, 6, 8

[66] S. Yang, P. Luo, C. C. Loy, and X. Tang. WIDER FACE: A face detection benchmark. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5525–5533, 2016. 8

[67] S. Yang, Y. Xiong, C. C. Loy, and X. Tang. Face detection through scale-friendly deep convolutional networks. *CoRR*, abs/1706.02863, 2017. 8

[68] S.-J. Yen and Y.-S. Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3):5718–5727, 2009. 3

[69] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 516–520. ACM, 2016. 5

[70] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. S. Huang. Unitbox: An advanced object detection network. In *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*, pages 516–520, 2016. 8

[71] J. Zhang, X. Wu, J. Zhu, and S. C. H. Hoi. Feature agglomeration networks for single stage face detection. *CoRR*, abs/1712.00721, 2017. 8

[72] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.*, 23(10):1499–1503, 2016. 8

[73] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu. Detecting faces using inside cascaded contextual CNN. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3190–3198, 2017. 8

[74] 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 *2017 IEEE International Joint Conference on Biometrics, IJCB 2017, Denver, CO, USA, October 1-4, 2017*, pages 1–9, 2017. 8

[75] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. Sˆ 3fd: Single shot scale-invariant face detector. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 192–201. IEEE, 2017. 3, 5, 8

[76] Z.-H. Zhou and X.-Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006. 3

[77] S. Zhu, C. Li, C.-C. Loy, and X. Tang. Unconstrained face alignment via cascaded compositional learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3409–3417, 2016. 1

[78] S. Zhu, S. Liu, C. C. Loy, and X. Tang. Deep cascaded bi-network for face hallucination. In *European Conference on Computer Vision*, pages 614–630. Springer, 2016. 1

[79] X. Zhu, Y. Liu, J. Li, T. Wan, and Z. Qin. Emotion classification with data augmentation using generative adversarial networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 349–360. Springer, 2018. 3