Triple Loss for Hard Face Detection

Zhenyu Fang[a], Jinchang Ren[b,a,1], Stephen Marshall[a], Huimin Zhao[b], Zheng Wang[c], Kaizhu Huang[d], Bing Xiao[b]

[a] *Centre for Signal and Image Processing, University of Strathclyde, Glasgow, UK.*
[b] *School of Computer Sciences, Guangdong Polytechnic Normal University, China*
[c] *School of Computer Software, Tianjin University, Tianjin, China*
[d] *Department of Electrical and Electronical Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China*

Abstract

Although face detection has been well addressed in the last decades, despite the achievements in recent years, effective detection of small, blurred and partially occluded faces in the wild remains a challenging task. Meanwhile, the trade-off between computational cost and accuracy is also an open research problem in this context. To tackle these challenges, in this paper, a novel context enhanced approach is proposed with structural optimization and loss function optimization. For loss function optimization, we introduce a hierarchical loss, referring to "triple loss" in this paper, to optimize the feature pyramid network (FPN) [1] based face detector. Additional layers are only applied during the training process. As a result, the computational cost is the same as FPN during inference. For structural optimization, we propose a context sensitive structure to increase the capacity of the prediction network to improve the accuracy of the output. In details, a three-branch inception subnet [2] based feature fusion module is employed to refine the original FPN without increasing the computational cost significantly, further improving low-level semantic information, which is originally extracted from a single convolutional layer in the backward pathway of FPN. The proposed approach is evaluated on two publicly available face detection benchmarks, FDDB and WIDER FACE. By using a VGG-16 based

*Corresponding author
Email address: jinchang.ren@strath.ac.uk (Jinchang Ren)

detector, experimental results indicate that the proposed method achieves a good balance between the accuracy and computational cost of face detection.
Keywords: Face detection, Small face, Face feature fusion, Single shot detection, Efficiency-accuracy balance

1. Introduction

Face detection is a basic task in various computer vision and face related applications [3]. At the first beginning, handcrafted feature extractor played an important role, such as Haar-like features proposed by Viola-Jones [4], in which the face image is segmented to several patches via multi-scale sliding windows. For each patch, the classification work is conducted by a two-class cascade classifier. Based on this pipeline, the following subsequent works [5, 6, 7, 8] improve the accuracy by optimizing the cascade detectors. Limited by the complexity of Haar-like features, cascade classifier is only sensitive to the frontal face. To improve the robustness, deformable part models (DMP) [9, 10] build features by considering the relationship of deformable facial parts. However, handcrafted features are effective only on specific poses and angles, which are unable to handle multi-scale and multi-angle faces [11, 12] in the wild. Thanks to the breakthrough on the convolutional neural networks (CNN), which extract features automatically without manual work, a series of CNN based models are proposed on object detection. Faster-RCNN [13] finds that an end-to-end CNN is more robust and accurate than handcrafted object detector. To increase the inference speed, Single Shot Detector SSD [14] proposed a multi-stage prediction structure, which could predict objects from low-level to high-level feature maps. However, as the low-level feature maps are in lack of semantic information [1], SSD has difficulty in detecting small objects. To tackle this drawback, feature pyramids are proposed, with a "backward path", which can link the high-level feature map to the low-level feature map for more effective feature fusion. Due to different requirements of applications, Faster-RCNN, SSD and **Feature Pyramid Network(FPN)** are widely used in face detection

[15, 16, 17, 18, 19, 20, 21, 22, 23, 24].

Though previous CNN based face detectors have made a remarkable process, detection of hard faces is still a challenging task. Compared with easily detected faces, the resolution of hard faeces is always low, which are interfered by, such as, blurry, occlusion, illumination and makeup. Those interferences cause the lack of visual consistency [22]. Existing methods tackle this challenge from both structure and loss function optimizations in the deep learning framework. Structure optimization aims to improve the performance by enhancing the capability of feature extraction, which can be conducted in two ways. The first is to apply a deeper CNN feature extractor (backbone) [18, 25, 26], e.g. ResNet-101 [27], ResNet-152 [27], ResNeXt-101[28] and DenseNet [29]. The second is to assign a subnet in the backward pathway [21, 22, 23, 25] to enhance the merged feature extraction. As the scale of the network grows larger [18, 25, 30], the accuracy on the hard face detection improves, but the computational cost increases at the same time. Instead of increasing the scale of the model, the loss optimization strategy [31, 20, 30, 22, 21, 25] optimizes the weights of each layer by assigning multiple tasks during the training, such as key point [31], attention [20], segmentation [30], and head-body detection [22].

In this paper, we optimize the performance on hard face detection through optimization of both the structure and the loss function. For structure optimization, we propose a new feature fusion module (FFM) embedded in the backward pathway to make full use of the high-level and low-lever features. To avoid increasing the computational cost significantly, we utilize both dilated convolution and small-size-kernel convolution (1-by-N and N-by-1 kernels) in the FFM. For loss optimization, we propose a "triple loss" training strategy, which covers three resources in the training process: i.e. forward path (the first level), backward path (the second level) and the extended path (the third level). The first two paths are the same as FPN and the feature maps of the extended path are simply extracted from the features of the backward path, by an additionally proposed FFM. However, during inference, only results predicted from the second level will be considered, i.e. all the irrelevant layers will be

discarded. Through this training and inference strategy, the proposed network suppressed the increase of computational cost when compared with other FPN based methods [20, 22, 21, 30, 23, 25]. By taking VGG-16 [32] as the backbone, the proposed model achieves comparable results with other models which utilize much deeper backbones. When evaluated on the WIDER FACE database, compared with other VGG-16 based face detector, the proposed method reaches the state-of-the-art on the hard subset. Although accuracy and computational cost seem to conflict to each other in face detection, by using the proposed FFM and the triple loss training, we can reach a good balance between these two metrics. Experimental results show that, without considering the non-maximum-suppression (NMS), the proposed method can detect faces by taking $29.7ms$ for a VGA-resolution image.

In summary, the main contributions of this paper can be summarized as follows:

1. Based on FPN, we design a training strategy which calculates losses through different pathways, however, during inference, we only consider the backward pathway, which increases the accuracy without adding additional computation cost.
2. We introduce a feature fusion module, consisting of a mixed network structure to enhance the capability of feature extraction from the fused features.
3. When compared with other VGG-16 based face detector, we achieve superior performance over a number of state-of-the-art methods on the hard subset of WIDER FACE dataset and reach a balance between the accuracy and speed. By using an appropriate anchor setting, the proposed method can reach the state-of-the-art on the easy and medium subsets, while keeping the considerable performance on the hard subset.

The rest of the paper is organized as follows: Section 2 brings a brief introduction of the recent works. Section 3 details the proposed FFM and triple loss training strategy. Section 4 presents the experiment results, including the ablation learning, comparison analysis and further discussions. Finally, some

concluding remarks are given in Section 5.

2. Recent Work

Back to 1990s, face detection became increasingly important in computer vision, which has been wildly used in multiple applications such as face recognition, facial expression recognition, and face tracking [3]. At early stage, face detection mainly extracted feature using a hand-crafted feature extractor, such as Haar-like features [4], control point set [33] and the Deformable Part Model (DPM) [9, 10]. These detectors reached promising detection accuracy and high efficiency at the same time.

Recently, results in [13, 17] indicate that CNN can extract more powerful features than hand-crafted face detectors. As a result, CNN based face detectors become dominating in face detection in the last decade [16, 20, 21, 22, 23, 30, 31].

Structures of CNN face detector. According to the structure of CNN, we divide most of existing CNN face detectors into two categories, i.e. multi-step detectors (SSD-like [SSD], one stage only.) and single-step detectors (faster-RCNN-like [13], containing one stage [18] or two [17]). A single-step detector [14, 15, 16] produces a promising accuracy using the feature map, which is extracted from the deepest layer of its backbone. However, the stride of the deep layer is often quite large (usually 16 [17] or 32 [34]). As a result, the information of tiny faces may vanish. To tackle this issue, multi-step detectors detect faces on feature maps extracted from different depths of CNN, where shallow layers are for detecting small faces and deeper layer for large faces. However, due to the limitation caused by the insufficient capability of feature extraction, shallow layers are not rich enough for extracting semantic information as deep layers [1]. In order to enrich the semantic information on shallow layers, [1] proposed a top-down pathway, where the feature maps of deep layers and shallow layers are fused together, using addition [16], element-wise multiplication [25] or concatenation [34].

Feature extraction module. Faster-RCNN [13] firstly presented a convolutional subnet for face detection, in which the subnet contains a single 3×3 convolutional layer, followed by two sibling 1×1 convolutional layers (also called "detection head") for classification and box regression, respectively. To reduce the computational cost, SSD [14] replaced the two subnets in the Faster-RCNN with a subnet with two 3×3 convolutional layers. To further increase the capabilities of classification and regression, based on the SSD detection head, RetinaNet [35] inserts four additional 3×3 convolutional layers before the last two layers. However, a 4-layer subnet has significantly increased the computational cost: for a typical FPN-based face detector, there are in total 6 feature maps which means 12 additional feature extraction subnets will be added. Even though the weights of subnets are shared in between, the computational costs for each subnet are independent to each other. To balance the accuracy and the computational cost, in recently proposed methods, a series of inception based subnets [36] are introduced to replace the four-layer subnet. For example, FANet [21] and Pyramidbox [22] have found that a simple two-branch inception module can keep the accuracy as retina head when using the SSD head. SRN [23] introduced a four-branch residual-inception subnet [37], as a replacement of the first two layers of Retina head. DSFD [25] applied the dilation convolution into the subnet, which expands the receptive filed without increasing the computational cost significantly.

Loss function design. Imbalanced ratio of positive examples and negative samples during training impedes the performance significantly, especially for SSD-like detectors [35, 38, 39]. To address this issue, online hard example mining (OHEM) [38] automatically selects hard-negative examples as three times of positive examples. In order to further make use of easy-negative examples, Lin et al. [35] proposed a focal loss which weights the loss of examples according to the difficulty of learning. Another two applicable strategies are multi-task prediction and hierarchical learning. For multi-task prediction, detection head will be assigned additional face-related prediction tasks, such as key points de-

tection [31], had-body detection [22], face attention [20], and face segmentation [30]. Different from multi-task prediction, tasks of the hierarchical learning are the same as the ordinary object detection training yet predicting objects from different "pathways". FANet [21] applied the FPN [1] structure in evaluation but predicted faces from one forward and two backward pathways during training. DSFD [25] narrowed the range of prediction layers to two pathways and assigning different anchor sizes for each pathway. SRN [23] cascaded prediction results from both pathways, which reduced easy-false-positive examples significantly.

However, existing methods cannot well balance the accuracy and the computational cost. Large scale models, with multiple pathways and deep backbones [20, 23, 25, 30], improve the accuracy with a sacrifice of computational cost. On the other hand, the structures of high-efficiency face detectors are always shallow [34]. As a result, the accuracy is not high enough in some dense detection scenes [12]. To address this issue, we propose a novel context enhanced approach as detailed in the next section.

3. The Proposed Approach

In this section, we will present the proposed triple loss training strategy, as well as the feature fusion module for face detection. First, the whole network structure will be illustrated, followed by the structure of the proposed feature fusion module. After that, the triple loss training strategy will be detailed.

3.1. Overall Network Structure

Figure 1 illustrates the structure of the proposed network, which is composed of three levels according to the predicted outputs of triple loss. In the first level, feature maps are generated through a pre-trained backbone. As the triple loss is designed for generalized face detection, in this paper, we mainly consider VGG-16 [32] as used in [16, 21, 22]. As a result, following the structure in SFD [16], feature maps of the first layer are generated through "Conv3-3", "Conv4-3", "Conv5-3", "Conv-fc7", "Conv6-2", and "Conv7-2", where the first four are

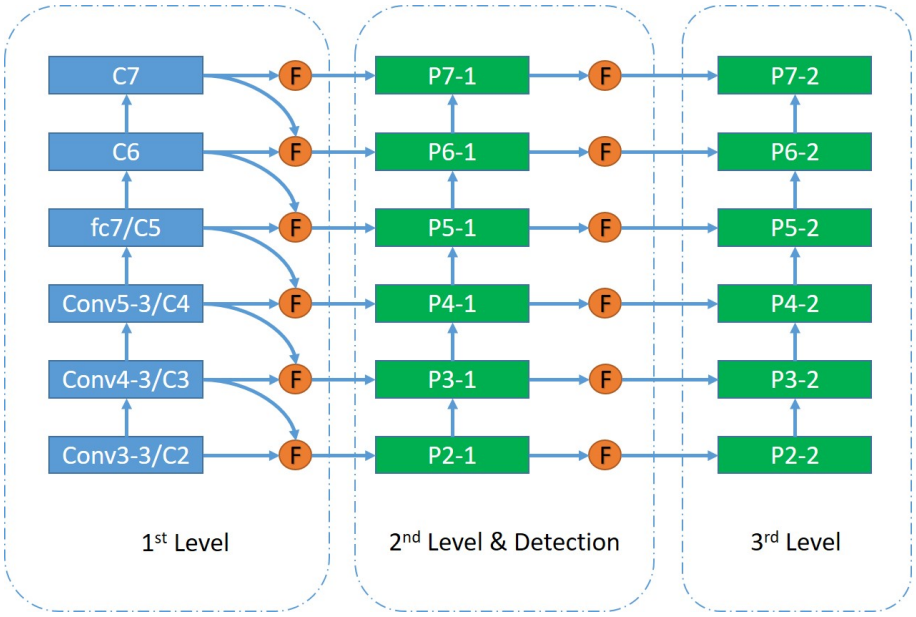


Figure 1: The proposed network structure trained with the "triple loss" training strategy.

from VGG-16, and the last two are from newly added layers. Two newly added layers are for detecting large scale faces, which are the same as used in SSD and SFD. The reduced sizes of feature maps related to the original image are 4, 8, 16, 32, 64, and 128.

Feature maps from the deeper layer have more semantic information extracted [1]. In order to obtain more semantic information for low-level feature maps, we normalize feature maps using 1×1 convolutional kernels in the top-down routine as suggested in [25]. From "Conv3-3" to "Conv6-2", we up-sample normalized feature map from up-layer and conduct elementwise product with the current one. After that, a feature fusion module is deployed to enhance the capability of feature extraction and increase the receptive field. Hence, feature maps extracted by FFM are used to form the second level. Results in [21, 22, 23, 25] show that such a module is helpful for improving the performance of the detector, and our experiments as detailed in Section 4 have also verified this point. The third level is simply extracted from the second level by

a proposed additional FFM without any top-down connection, whilst there are two top-down connections in FANet. However, our experimental result shows that our method outperforms FANet by 0.3% in detecting small faces without assigning the top-down connection at the third level.

Letting the feature map of the layer j ($j \in [1, 6]$) from the level i ($i \in [1, 3]$) be $\Phi_{(i,j)}$, the feature map of the next level $\Phi_{(i+1,j)}$, in FANet, can be mathematically defined as:

$$\begin{aligned} \Phi_{fusion} &= f_{eleprod}\left(f_{1\times1}\left(\Phi_{(i,j)}\right), f_{1\times1}\left(\Phi_{(i,j+1)}\right)\right) \\ \Phi_{(i+1,j)} &= f_{1\times1}\left(f_{\text{concat}}\ \left(f_{\text{inception}}\left(\Phi_{fusion}\right)\right)\right) \end{aligned} \tag{1}$$

where $f_{1\times1}$, $f_{eleprod}$ and f_{concat} indicate the operations of 1×1 convolution, element-wise production, and feature concatenation respectively; Φ_{fusion} is the fused feature map after elementwise production; and $f_{\text{inception}}$ indicates a inception subnet structure. On the other hand, in our proposed method, the feature map in level 2 and level 3 can be expressed by:

$$\begin{aligned} \Phi_{fusion} &= f_{eleprod}\left(f_{1\times1}\left(\Phi_{(1,j)}\right), f_{1\times1}\left(\Phi_{(1,j+1)}\right)\right) \\ \Phi_{(2,j)} &= f_{1\times1}\left(f_{\text{concat}}\ \left(f_{\text{inception}}\left(\Phi_{fusion}\right)\right)\right) \\ \Phi_{(3,j)} &= f_{1\times1}\left(f_{\text{concat}}\ \left(f_{\text{inception}}\left(\Phi_{(2,j)}\right)\right)\right) \end{aligned} \tag{2}$$

Similar to [14, 16, 21, 22, 23] feature maps of the first level are extracted from the forward path. For the second level, as given in Eq. 2, the initial feature map and the feature map derived from its upper layer are convolved by a $1\times1\times256$ kernel, respectively. The two normalized feature maps are fussed via elementwise product, which is taken as the inputs to the FFM of the second level. Afterwards, it will pass a three-branch inception subnet, which is the dominant part for FFM and will be detailed in the next section. The outputs from different branches are concatenated and the number of channels is normalized to 256. We denote the feature maps of the second level as {P2-1, P3-1, P4-1, P5-1, P6-1, P7-1}. For the third level, we simply assign a single FFM for each layer, where the input of each FFM is

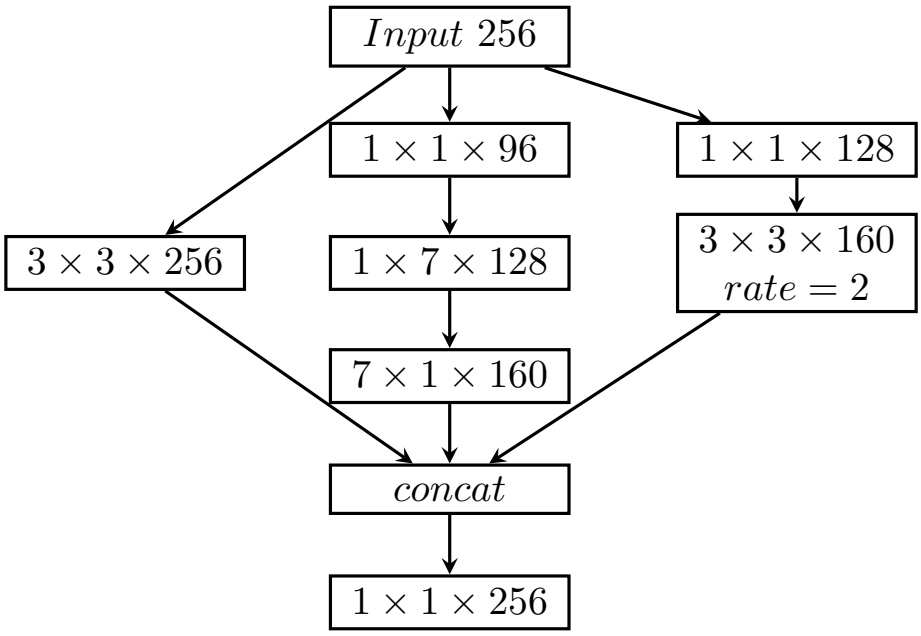


Figure 2: The proposed feature fusion module (F-block in Figure 1).

the feature map derived from the second level, to obtain the feature maps of the third level. Feature maps of the third level are labeled as {P2-2, P3-2, P4-2, P5-2, P6-2, P7-2}, see in Figure 1. The structure of FFM on the third level is the same as the second level without weight sharing.

3.2. Feature Fusion Module

In this paper, we propose a feature fusion module to enhance the capability of feature extraction, where the fused feature map is extracted through the backward pathways, as well as to increase the receptive field. At present, the mostly used backbone for face detection are VGG-16 [15, 16, 21, 22] and ResNet [20, 23, 25, 30], where the kernel shapes are 3×3 and $\times 1$. As a result, the effective receptive filed of each layer is in a square shape. However, experimental results in [40] indicate that for some non-square objects (i.e. aspect ratio is not 1), the shapes of effective receptive field may not be the typical shape

of squares. As illustrated in [23], this issue seems not crucial for frontal face detection, because the aspect ratio is about 1. However, this is important for multi-pose face detection as the aspect ratios can vary between 0.5 and 2. To tackle this challenging problem, we present a feature fusion module with multiple shapes of kernels. The structure of the proposed FFM is shown in Figure 2. Following the design in [21, 22, 23, 25], we use the inception structure [2] in the proposed feature fusion module, which consists of three branches. The first branch has a single 3×3 convolutional layer to smooth feature as in [1]. Inspired by RFB [40] and DSFD [25], the second branch consists of two dilation convolutional kernels, in order to further increase the receptive field sparsely. As illustrated in [23] a densely feature extractor is also important for refining the effectiveness of the receptive field. However, using $N \times N$ kernels will increase the computational cost significantly. Hence, to balance the computational cost and detection accuracy, in the third branch, we employ a $1 \times N$ and $N \times 1$ structure to extract dense features. At the end of the module, feature maps from the three sub-networks are concatenated together and then smoothed by a 1-by-1 convolutional layer. Experimental results show that the proposed FFM, using the combination of dilation convolution and $1 \times N$ (with $N \times 1$), performs better than the existing ones [21, 22, 23, 25].

3.3. Triple Loss Training Strategy

In this section, we will introduce the proposed triple loss training strategy in details. As described in Section 3.1, feature maps are splitted into three levels. During training, we assign a classification layer and a regression layer on each feature map in all three layers. To improve the inference efficiency, we only use two 3×3 convolutional layers for classification and regression (detection head) separately [20, 23, 25], without a retina head [35]. We define the triple loss function (TL) as follows:

$$\begin{aligned} &TL(\Phi_{(1,1)}, \Phi_{(1,2)}, ..., \Phi_{(2,j)}, ...\Phi_{(i,j)}, \mathbf{A}) \\ &= \sum_{i=1}^{3} \omega_i L(\Phi_{(i,1)}, \Phi_{(i,2)}, ..., \Phi_{(i,j)}, A_i) \end{aligned} \tag{3}$$

where A_i and ω_i denote respectively the anchor setting and the adjusting parameter with respect to level i ($i \in [1, 3]$) as there exist three levels in triple loss. Experimental result shows that the magnitudes of the loss from all three levels are the same, i.e. the contribution of each level to the loss is the same, which is similar to those reported in [25] and [21]. We use the cross-entropy loss (as in [20, 22, 25]) and the smooth L1 loss [13] to determine the classification loss and the regression loss, respectively. To be more specific, the total loss function of each level can be expressed as below:

$$\begin{aligned} TL(p, p^*, t, t^*) = \sum_{i=1}^{3} & \frac{1}{N_{conf}} L^i_{conf}(p_i, p^*_i) \\ & + \frac{1}{N_{loc}} [p^*_i = 1] L^i_{loc}(t_i, t^*_i) \end{aligned} \tag{4}$$

where, for level i, L^i_{conf} and L^i_{loc} are the confidence prediction loss and the localization loss terms, respectively; p_i, p^*_i, t_i and t^*_i refer respectively to the predicted probability, ground truth probability, predicted regression target and ground truth box regression target. The Iverson bracket indicates a function $[p^*_i = 1]$ outputs 1 when the condition holds true, i.e. only the regression loss of positive instances will be minimized during the training.

The anchor setting is another key factor that affects the performance of face detector. As suggested in [25], assigning small anchor size on the forward pathway improves the prediction performance, even it is not used during inference. Hence, the anchor sizes of the first level are halved compared with the following levels, as shown in Table 4. The aspect ratio is set as 1.25 for all three levels as suggested in [23].

During inference, we only use the second layer to conduct face detection. The detection heads of the first and the third levels, as well as the additional feature fusion modules in the third level, are discarded. Hence, the proposed face detector will not add additional parameters and computational cost compared to other FPN based methods [20, 21, 22, 25].

4. Experiment Results and Discussions

We first analyze the proposed method in detail to clarify the effectiveness of our contributions. We evaluate the final model on two commonly used face detection benchmark datasets, FDDB [11], and WIDER FACE [12].

4.1. Training Datasets and Hyperparameters

The ablation learning is conducted on the WIDER FACE dataset [12], which consists of 393,703 annotated face bounding boxes in 32,203 images. Images in WIDER FACE are splitted into three sub-datasets: training, validation and test dataset. Performance is evaluated in terms of average precision (AP) with the Interception-of-Union (IoU) set to 0.5. Instead of describing the result by a single mean average precision (mAP) over the whole validation dataset or test dataset, there are three subsets according to the detection difficulty levels: Easy, Medium, and Hard, based on the detection rate of EdgeBox [41]. The training dataset, which has 12,880 images, is applied as the only training dataset in this paper. Results of ablation learning are compared on the validation dataset with 3,226 images. In the end, we will evaluate the proposed model on the test datasets.

During training, we adopt the same data augmentation method as [22, 25]: At first, we conduct random flipping, colour distortion, etc., which is the same as in SSD [14]. For image resizing, with a probability of 0.6, we conduct the original image resize method as introduced in SSD. Otherwise, we resize the image using data-anchor-sampling as in Pyramidbox [22]. To balance the ratio of positive and negative training instances, we use the online hard example mining (OHEM) in a similar way as [14, 16, 22, 25] and assign the ratio of positive: negative is set as 1:3. In the end, a 640 $\times$ 640 patch will be resized from each cropped image patch. We also tried image expansion [16, 23] in augmentation but the results seemed quite poor. We deduce that expansion may not fit for low batch size training. As a result, we do not apply expansion in this paper.

The backbone network is initialized by the pretrained VGG on ImageNet. All newly added convolution layers' parameters are initialized by the 'xavier' method [42]. We use SGD with momentum and weight decay set as 0.9 and 0.0005 to train our models. The batch size is set to 12. The learning rate is initialized to **1e-3** and is decayed by 10 when at 80K and 100K steps, respectively. During inference, the settings of hyper parameters are the same as in [14, 16, 22, 25]. The second level predicts the top 5K high confident detections, followed by non-maximum suppression, with the Jaccard overlap of 0.3, to produce the top 750 high confident bounding boxes per image.

4.2. Model Analysis

In this section, a series of ablation experiments will be conducted on the WIDER FACE dataset to analyze how each contribution module improves the performance in detail. For a fair comparison, we use the same parameter settings, including anchor setting, training hyper parameters, data augmentations, etc., for all the experiments.

As the structures of recent proposed face detectors [20, 21, 22, 23, 25] contain both down-top and top-down pathway as FPN, we use FPN as a baseline to make a fair comparison. The anchor setting of the baseline is the same as SFD and PyramidBox, which is [16, 32, 64, 128, 256, 512], and the aspect ratio is 1.25 as in [23]. All models in this section are trained on the training set and evaluated on the validation set.

4.2.1. Feature Fusion Module

First, we will show how the proposed feature fusion module improves the performance of the baseline. In Table 1, we compare the performance of different feature fusion modules on the WIDER FACE validation dataset. As observed, with the same backbone (VGG-16) and the network structure (FPN), the proposed feature fusion module surpasses the baseline by 0.7%, 1.1% and 4% on the easy, medium and hard subsets, respectively. When compared with other feature fusion modules, the proposed module reaches the best on the medium

and the hard subsets, which leads the state-of-the-art method by 0.1% on the medium subset and 0.4% on the hard subset, respectively. We deduce that such improvement of increased accuracy is contributed by the combination of dilated convolution and the ordinary convolution. Under a similar computational cost, dilated convolution increases the receptive field of the feature map significantly [40]. However, a large receptive field may also harm the performance of small object detection [22].

To balance the performance on various scales, a concatenation of feature maps from both convolutional layers is needed. We notice that when compared with CPM [22], the proposed feature fusion module lags by 0.1% on easy subset. We deduce that this is caused by the larger output channel number of CPM: the number of the output channels from CPM is 512, which is the double of our proposed FFM. This large scale of the subnet will consume a huge amount of computational cost, which will be shown in Section 4.2.5 later. On the contrary, our FFM is more light-weighted, reaching the balance between accuracy and the inference efficiency. To validate the performance of the proposed FFM on a larger number of output channels, we keep the structure unchanged but expand the output channels to 512 and add batch normalization [36] before each convolutional layer as used in CPM, labelled as "FFM-512" in the table. As seen, when we double the number of output channels, the proposed FFM outperforms the CPM on all three subsets.

4.2.2. Triple Loss Training

In this section we evaluate the performance of the triple loss training strategy in detail. We will conduct an ablation learning to show how each level affects the model. We use SSD as the baseline, which calculates the loss only using the first level. We calculate the loss from the second and the third level separately, where the proposed feature fusion module will be applied. Finally, level by level, we combine the losses from different levels into training. During evaluation, for single level loss training, we obtain the result from that training level, while for multi-level training, we use the result from the deepest level. Experimental

Table 1: Effectiveness of various feature fusion approaches in terms of AP.

Component	Easy	Medium	Hard
Baseline [21]	94.3	92.9	83.8
+CEM [21]	94.8	93.6	84.4
+CPM [22]	**95.1**	93.9	87.4
+RFE [23]	94.9	93.8	87.2
+FEM [25]	94.9	93.9	87.5
+FFM (Ours)	**95.0**	**94.0**	**87.8**
+FFM-512 (Ours)	**95.2**	**94.0**	**87.9**

results are given in Table 2.

From the first three rows in Table 2, it is clear to see that the accuracy increases as the scale of the network increases, when using the single level loss training. However, the incensement between levels decreases at the same time. To balance the computational cost of training and the evaluation accuracy, we do not add the fourth level in the experiment. We deduce that the contribution from the fourth layer might be minor for face detection.

When using multi-level training, which is shown in the last two rows, we find out that the performance of the model is increased significantly. When compared with the models trained on a single level (on the third level), the performance of the model measured using detection accuracy, trained via triple loss, is increased by 0.6%, 0.5%, 1.7% on the three subsets, respectively. Experimental results indicate that when the scale of the model is fixed, the multi-level training strategy helps to increase the performance of the model, especially on the medium and the hard subsets.

Table 2: Results of the triple loss on the WIDER FACE validation subset.

Component	Easy	Medium	Hard
1st level	94.0	93.0	83.5
2nd level	95.0	94.0	87.8
3rd level	95.2	94.3	88.0
1+2 levels	95.6	94.7	89.1
1+2+3 levels	**95.8**	**94.8**	**89.7**

4.2.3. Prediction Level

Predicting using the third-level feature map increases the performance. However, it also increases the computational cost. As the anchor in the second and the third levels are identical, it is possible to predict via the second level. When evaluating through the second level, feature maps from the third level will be omitted hence the total computational cost is reduced. The result comparison of different prediction levels is presented in Table 3. In this test, after trained using triple loss, the model predicts result through the second and the third level separately. Compared the result predicted from the third level, the AP predicted through the second level is the same on the easy subset, and 0.1% better on the medium and hard subset. This indicates that the third level is essential during training but seems unnecessary during evaluation. In summary, the proposed triple loss training strategy improves the AP without increasing the computational cost during inference.

Furthermore, to validate the performance of multi-level prediction, we also collect the prediction results predicted from both the second and the third levels, which is shown in the last row of Table 3. Apparently, prediction from two levels does not bring an increase but a decrease on the prediction result. On the other hand, as prediction heads from both levels are applied, this will increase the computational cost. As a result, multi-pathways inference is not utilized in the

Table 3: Comparison of results on different prediction levels.

Prediction Level	Easy	Medium	Hard
3rd	95.8	94.8	89.7
2nd	**95.8**	**94.9**	**89.8**
2nd + 3rd	95.7	94.7	89.5

model.

4.2.4. Effect of Anchor Design

As anchor design is a key factor of the box size regression [21, 23, 25], we discuss how the anchor size affects the performance. In DSFD, experimental results show that a smaller anchor size on the forward pathway (first level), which is halved compared to the backward pathway, can further improve the performance. Motivated by this observation, we fix the anchor size by $[8, 16, 32, 64, 128, 256]$, as suggested in DSFD, on the first level and vary the anchor sizes on the second level and the third level. Based on the findings in Section 4.2.3, we use the second level as the prediction level during inference.

Experimental results are shown in Table 4. When we increase the anchor size progressively, the third level impedes the final prediction during inference. We then swap the anchor size between the second level and the third level. As a result, the model further improves the AP by 0.4% and 0.3% on the easy and medium subsets respectively (see row 2 of Table 4), when compared with the identity setting (row 1). It is not surprising to see the poor performance on the hard subset because the large anchor size is unsuitable for detecting small faces, which mainly belong to the hard subset. Consequently, we double the number of anchors in the second level, as shown in row 3, to gain benefits of both designs. In summary, the identity setting is important for hard face detection, while progressive setting brings increase on the other two subsets.

Table 4: Comparison of results on different anchor assignments.

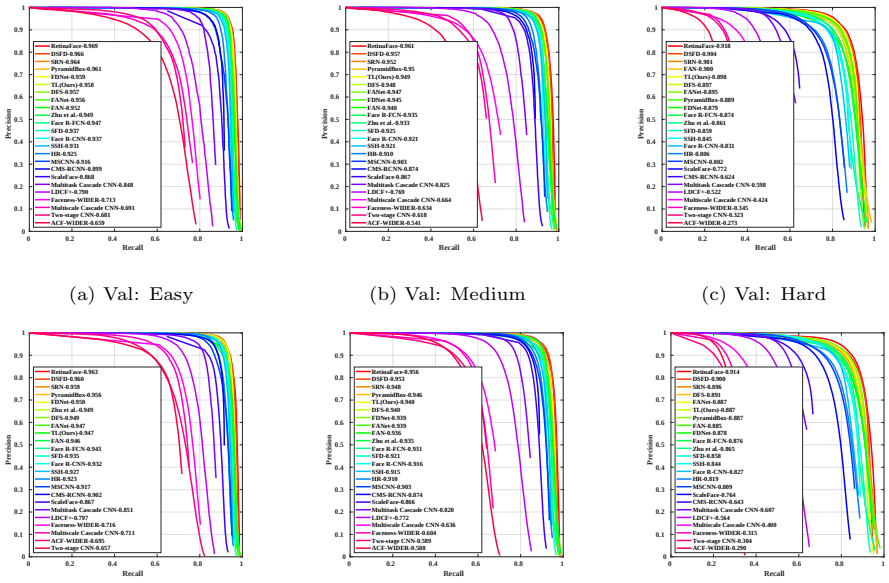
Predefined anchor sizes:			
A1: [16, 32, 64, 128, 256, 512]			
A2: [32, 64, 128, 256, 512, 1024]			
A3: [(16,32), (32,64), ..., (512,1024)]			
Anchor applied	Easy	Medium	Hard
A1 (2^{nd}, 3^{rd})	95.8	94.9	**89.8**
A1 (2^{nd}), A2 (3^{rd})	**96.2**	**95.2**	82.5
A3 (2^{nd}), A2 (3^{rd})	96.1	95.0	88.6

4.2.5. Comparison with Other Face Detectors

In this section, we compare the proposed method with other algorithms. The APs of three subsets on the WIDER Face are given in Figure 3, of which the model uses the identity anchor setting on the second and the third levels. As the accuracy relates to the scale to backbone, we also summarize the backbones of the state-of-the-art methods in Table 5. As observed, the proposed model reaches the best performance on the hard subset, when compared with other VGG-16 based models; it also attains the best AP on the easy and medium subsets, when using the progressively anchor setting (labelled by "TL-LA"). Even when compared with the state-of-the-art methods on the hard subset, as shown in Figure 4, the proposed method only sacrifices the accuracy by about 0.6% but with much more computational saving.

4.2.6. Effects of Backbone

To evaluate the robustness of the proposed detector, we also validate the performance on the Resnet-50, which is shown in the last two rows of Table 5. As most of the ResNet-based face detectors [20, 23, 25, 30] apply retinanet prediction head as in [35], for a fair comparison, we deploy both SSD predic-



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

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

Figure 3: Precision-recall curves on WIDER FACE validation and test sets.

tion head and retinanet prediction head in the model. During the training, we increase the batch size to 16 as DSFD did. Limited by the computational resource, we can only use a batch size of 12 when training on the retinanet head. Compared with the retinanet, we increase the performance by more than 0.6% on all three subsets. By using the retinanet head, the performance has been further improved by 0.3% and 0.2% on the easy and medium subset, respectively. As for the decrease on the hard subset, we deduce it is caused by the decrease of batch size because both batch size and training image size are crucial for the final performance [48]. We will solve such issues when more computational resource is available in the future.

When compared with VGG-16, ResNet-50 outperforms on the easy subset without using the large anchor setting. However, the hard set AP is lower by 1.2%. We deduce that this is caused by the nature of ResNet: the scales of the hard set are always small, which are detected by the low-level feature maps. As

Table 5: Result comparison on the WIDER FACE validation set.

Methods	Backbone	Easy	Medium	Hard
ScaleFace [43]	ResNet-50	86.8	86.7	77.2
HR [18]	ResNet-101	92.5	91.0	80.6
Face R-FCN [44]	ResNet-101	94.7	93.5	87.4
Zhu [26]	ResNet-101	94.9	93.3	86.1
RetinaNet [23]	ResNet-50	95.1	93.9	88.0
SRN [23]	ResNet-50	96.4	95.2	90.1
DSFD [25]	ResNet-152	96.6	95.7	90.4
RetinaFace [45]	ResNet-50	**96.5**	**95.6**	**90.4**
RetinaFace [45]	ResNet-152	**96.9**	**96.1**	**91.8**
CMS-RCNN [46]	VGG16	89.9	87.4	62.4
MSCNN [47]	VGG16	91.6	90.3	80.2
Face R-CNN [17]	VGG19	93.7	92.1	83.1
SSH [15]	VGG16	93.1	92.1	84.5
S3FD [16]	VGG16	93.7	92.5	85.9
PyramidBox [22]	VGG16	**96.1**	**95.0**	88.9
FANet [21]	VGG16	95.6	94.7	**89.5**
TL (Ours)	VGG16	95.8	94.9	**89.8**
TL-LA (Ours)	VGG16	**96.2**	**95.2**	82.5
TL-res50 (Ours)	ResNet-50	95.7	94.7	88.6
TL-res152 (Ours)	ResNet-152	96.2	95.5	88.5
TL-res50-RH	ResNet-50	96.0	94.9	88.4

suggested in [48], ResNet reduces the feature map size earlier than VGG, consequently losing the information of small objects. On the other hand, however, it will be more efficient because of the early reduction of size. In summary, ResNet based methods are more suitable for speed-prioritized applications, while VGG backbone can be applied for detection of small faces.

When compared with other ResNet-50 based detectors, the proposed method outperforms most of them on easy and medium subsets, except for SRN and RetinaFace [45]. In SRN, the prediction results from the first and the second levels are cascaded to reduce the number of false positives and refine the positions of boxes, which improves the AP but also sacrifices the computational efficiency. However, with a slight decrease of the AP, the proposed method can achieve a good balance between the AP and the computation cost, as discussed in the next section. **RetinaFace [45] achieves the state-of-the-art performance on the hard subset with the ResNet-152 used as backbone. For a fair comparison, RetinaFace with ResNet-50 as backbone is benchmarked with our proposed approach. As seen in Table 5, the proposed method lags by 1.8% on hard subset than RetinaFace when using ResNet-50 as backbone. The small difference is caused mainly by extra information such as facial landmarks and 3D positions used in RetinaFace, in addition to 2D face bounding box, which is the only information required in our proposed triple loss training model. On one hand additional features have led to significantly increased dimension of the prediction layer (from 6 to 160) and the associated computational cost. On the other hand, this inspires us to combine 3D information to further improve our model in the future. Furthermore, the face landmark prediction in RetinaFace relies on supervised training, which requires additional work for manual labelling the samples. In contrast, such extra labelling is avoided in our approach. We also conduct the training using ResNet-152 as backbone, limited by computational resources, we further reduce the batch size and the learning rate to 8 and 5e-4, respectively, whilst doubling the training**

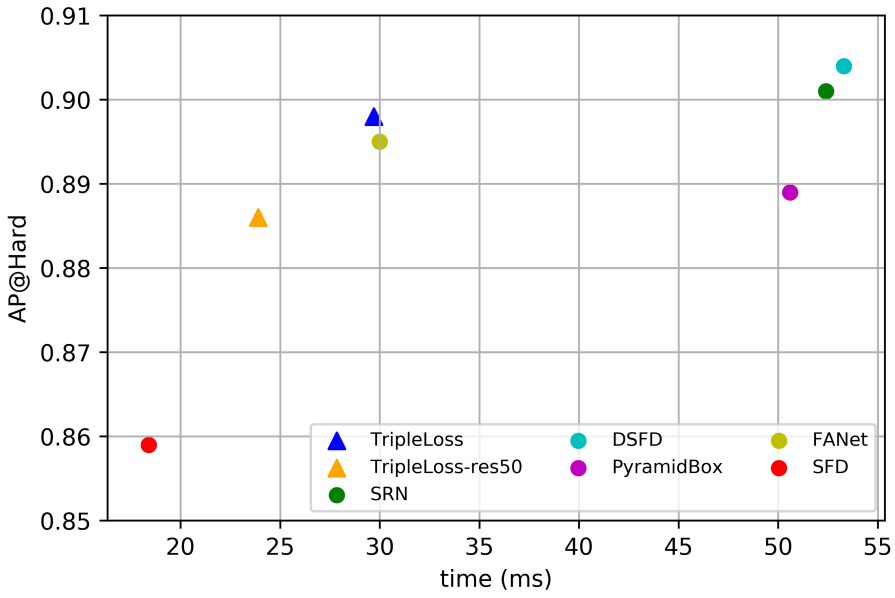


Figure 4: Time consumption among the state-of-the-art methods.

epochs. As shown in Table 5, even the model is affected by the low batch size in training, the proposed method slightly lags RetinaFace (ResNet-152) by 0.7% and 0.6%, on easy and medium subset without using any additional labelled samples. This has further validated the efficacy of the proposed approach.

4.2.7. Inference Speed

As described in the last section, the proposed triple training strategy and FFM can balance between the detection accuracy and the computational cost. Figure 4 illustrates the inference speed, accompany with the accuracy, among the-state-of-the-art methods. For a fair comparison, we deploy all the methods on Pytorch [49] without conducting the non-maximum suppression. All the tests are conducted on a single GTX1080Ti. As seen, a deep backbone [25] or a heavy subnet for feature extraction [21, 22, 23] improves the detection accuracy by sacrificing the speed. However, a light-weight network structure [16] seems insufficient. As a result, the proposed FFM can improve the accuracy by

slightly increasing the computational cost during inference. Different from the existing training methods [21, 23], the proposed triple loss training strategy only increases the computational cost during training without affecting the inference efficiency.

4.2.8. Evaluation on FDDB

To validate the performance on multiple datasets, we also evaluate the model on the FDDB dataset [11], where the proposed model is trained on the WIDER FACE dataset. In FDDB, there are 5,171 faces in 2,845 images taken from the faces in the wild dataset. Different from the WIDER FACE, faces in FDDB are labelled by ellipses. To show the robustness of the proposed method, we did not train a repressor offline. Instead, we use the ellipses regressor in [16] to transform the final prediction results from rectangle to ellipse. There exist unlabelled faces in the original dataset. For a fair comparison, we add additional annotations as in [16, 21, 22, 25] and report our results on discontinuous ROC curves [11], as shown in Figure 5. As seen, the proposed method achieves 98.4% when the number of false positives equals to 1,000. When compared with other state-of-the-art methods [16, 21, 22, 25], the proposed method lags by no more than 0.8% by applying a relatively light-weight backbone. **By applying a deeper backbone, FPN-based face detectors [16, 21, 23, 48, 25, 45] all show a certain degree of improved performance. To verify the performance of our approach under a deeper backbone, additional experiments are conducted on the FDDB dataset using the ResNet-152 as backbone, which is trained on the WIDER FACE dataset. Due to limited available computational resources, we have selected a low batch size of 8 for comparison (learning rate = 5e-4, training steps = 240k), where the accuracy achieved from ResNet-152 and VGG-16 became 98.0% and 97.6%, respectively. This on one hand has shown that a low batch size indeed leads to degraded accuracy, as a larger batch size with VGG-16 can produce an accuracy of 98.4%. On the other hand, it validates that a deeper backbone can**

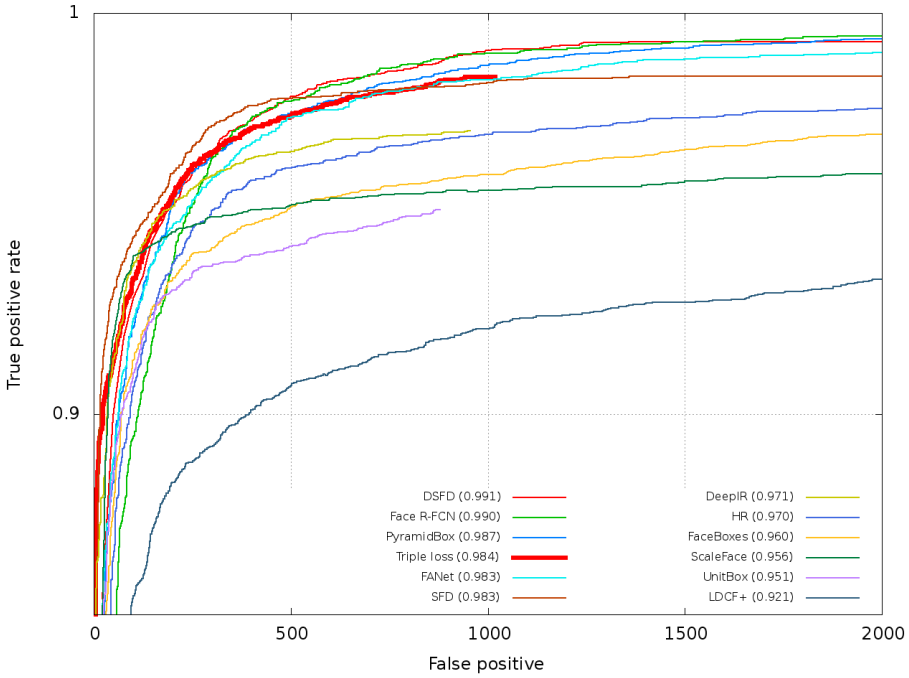


Figure 5: FDDB Discrete ROC Curves.

further improve the classification accuracy. As the proposed method is also FPN-based, we deduce, by using the same batch size training, its performance can also be further improved when using a deeper backbone.

5. Conclusion

In this paper, we proposed a novel training strategy, as well as an accuracy-computational cost balanced feature fusion strategy for single shot face detector, which is applied on the problem of unconstrained face detection.

We designed a feature fusion module to balance between the computational cost and the accuracy of the face detector. We combine both dilated convolution and the small-kernel-size convolution in the module, which marginally improves the accuracy, especially on small objects. Furthermore, we proposed

a training strategy, which refers to triple loss training, for FPN based face detector. During training, it takes the advantage of hierarchical loss from both forward and backward paths. During the evaluation module, however, only feature maps from the second level will be utilized, which improves the accuracy without affecting the inference efficiency.

Experimental results indicate that the proposed FFM and the triple loss training strategy are effective for identifying hard faces. Taking VGG-16 as the backbone, the proposed model achieves the state-of-the-art on the hard subset of the WIDER FACE validation dataset, when compared with other VGG-16 based face detectors. By assigning a larger anchor size, the performance can be further improved on the easy and medium subset. Without bells and whistles, the proposed method achieves comparable results on multiple common face detection benchmarks, when compared with other large-scale face detectors.

As the performance of the proposed network relies heavily on the scales of anchor setting, we will focus on the removal of anchor prior, i.e. anchor free [50, 51], to the model, in the future.

6. Acknowledgments

Zhenyu Fang acknowledges financial support from the Faculty of Engineering International Scholarships and the project of Teaching Space Utilisation System. This work was supported by Guangdong Provincial Key Laboratory of Intellectual Property and Big Data under Grant (2018B030322016), the Innovation Team Project of the Education Department of Guangdong Province (NO.2017KCXTD021). Results were obtained using the ARCHIE-WeSt High Performance Computer (www.archie-west.ac.uk) based at the University of Strathclyde.

References

[1] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE con-

ference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[3] S. Roy, S. Podder, Face detection and its applications, International Journal of Research in Engineering & Advanced Technology 1 (2) (2013) 1–10.

[4] P. Viola, M. Jones, et al., Rapid object detection using a boosted cascade of simple features, CVPR (1) 1 (511-518) (2001) 3.

[5] S. C. Brubaker, J. Wu, J. Sun, M. D. Mullin, J. M. Rehg, On the design of cascades of boosted ensembles for face detection, International Journal of Computer Vision 77 (1-3) (2008) 65–86.

[6] M.-T. Pham, T.-J. Cham, Fast training and selection of haar features using statistics in boosting-based face detection, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–7.

[7] S. Liao, A. K. Jain, S. Z. Li, A fast and accurate unconstrained face detector, IEEE transactions on pattern analysis and machine intelligence 38 (2) (2015) 211–223.

[8] B. Yang, J. Yan, Z. Lei, S. Z. Li, Aggregate channel features for multi-view face detection, in: IEEE international joint conference on biometrics, IEEE, 2014, pp. 1–8.

[9] M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool, Face detection without bells and whistles, in: European conference on computer vision, Springer, 2014, pp. 720–735.

[10] J. Yan, Z. Lei, L. Wen, S. Z. Li, The fastest deformable part model for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2497–2504.

[11] V. Jain, E. Learned-Miller, Fddb: A benchmark for face detection in unconstrained settings, Tech. Rep. UM-CS-2010-009, University of Massachusetts, Amherst (2010).

[12] S. Yang, P. Luo, C. C. Loy, X. Tang, Wider face: A face detection benchmark, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[13] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[15] M. Najibi, P. Samangouei, R. Chellappa, L. S. Davis, Ssh: Single stage headless face detector, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4875–4884.

[16] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S. Z. Li, S3fd: Single shot scale-invariant face detector, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 192–201.

[17] H. Jiang, E. Learned-Miller, Face detection with the faster r-cnn, in: 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), IEEE, 2017, pp. 650–657.

[18] P. Hu, D. Ramanan, Finding tiny faces, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 951–959.

[19] S. Zhang, X. Wang, Z. Lei, S. Z. Li, Faceboxes: A cpu real-time and accurate unconstrained face detector, Neurocomputing (2019).

[20] J. Wang, Y. Yuan, G. Yu, Face attention network: An effective face detector for the occluded faces, arXiv preprint arXiv:1711.07246 (2017).

[21] J. Zhang, X. Wu, J. Zhu, S. C. Hoi, Feature agglomeration networks for single stage face detection, arXiv preprint arXiv:1712.00721 (2017).

[22] X. Tang, D. K. Du, Z. He, J. Liu, Pyramidbox: A context-assisted single shot face detector, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 797–813.

[23] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, X. Zou, Selective refinement network for high performance face detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8231–8238.

[24] S. Qu, K. Huang, A. Hussain, Y. Goulermas, Mpssd: Multi-path fusion single shot detector, in: 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–6. doi:10.1109/IJCNN.2019.8852053.

[25] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, F. Huang, Dsfd: dual shot face detector, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5060–5069.

[26] C. Zhu, R. Tao, K. Luu, M. Savvides, Seeing small faces from robust anchor's perspective, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5127–5136.

[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[28] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1492–1500.

[29] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[30] W. Tian, Z. Wang, H. Shen, W. Deng, B. Chen, X. Zhang, Learning better features for face detection with feature fusion and segmentation supervision, arXiv preprint arXiv:1811.08557 (2018).

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

[32] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[33] Y. Abramson, B. Steux, H. Ghorayeb, Yef real-time object detection, in: International Workshop on Automatic Learning and Real-Time, Vol. 5, 2005, p. 7.

[34] S. Zhang, X. Zhu, Z. Lei, X. Wang, H. Shi, S. Z. Li, Detecting face with densely connected face proposal network, Neurocomputing 284 (2018) 119–127.

[35] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

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

[37] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[38] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 761–769.

[39] K. Huang, H. Yang, I. King, M. R. Lyu, Learning classifiers from imbalanced data based on biased minimax probability machine, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., Vol. 2, IEEE, 2004, pp. II–II.

[40] S. Liu, D. Huang, et al., Receptive field block net for accurate and fast object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 385–400.

[41] C. L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: European conference on computer vision, Springer, 2014, pp. 391–405.

[42] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249–256.

[43] S. Yang, Y. Xiong, C. C. Loy, X. Tang, Face detection through scale-friendly deep convolutional networks, arXiv preprint arXiv:1706.02863 (2017).

[44] Y. Wang, X. Ji, Z. Zhou, H. Wang, Z. Li, Detecting faces using region-based fully convolutional networks, arXiv preprint arXiv:1709.05256 (2017).

[45] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, S. Zafeiriou, Retinaface: Single-stage dense face localisation in the wild, arXiv preprint arXiv:1905.00641 (2019).

[46] C. Zhu, Y. Zheng, K. Luu, M. Savvides, Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection, in: Deep Learning for Biometrics, Springer, 2017, pp. 57–79.

[47] Z. Cai, Q. Fan, R. S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: european conference on computer vision, Springer, 2016, pp. 354–370.

[48] S. Zhang, R. Zhu, X. Wang, H. Shi, T. Fu, S. Wang, T. Mei, Improved selective refinement network for face detection, arXiv preprint arXiv:1901.06651 (2019).

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

[50] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, arXiv preprint arXiv:1904.01355 (2019).

[51] W. Liu, S. Liao, W. Ren, W. Hu, Y. Yu, High-level semantic feature detection: A new perspective for pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5187–5196.