DBCFace: Towards Pure Convolutional Neural Network Face Detection

Xin Li, Shenqi Lai, and Xueming Qian, *Member, IEEE*

***Abstract*— Face detection generally requires prior boxes and an extra non-maximum suppression(NMS) post-processing in modern deep learning methods. However, anchor design and anchor matching strategy significantly affect the performance of face detectors, so we have to spend a lot of time on anchor designing for different business scenarios. The other issue is that NMS cannot be easily parallelized and it may become a bottleneck of detection speed. In this paper, we propose a simple yet efficient pure convolutional neural network face detection method, named dual-branch center face detector(DBCFace for short), which solve face detection via a dual branch fully convolutional framework without extra anchor design and NMS. Extensive experiments are conducted on four popular face detection benchmarks, including AFW, PASCAL face, FDDB, and WIDER FACE, demonstrating that our method is comparable with state-of-the-art methods while the speed is faster.**

***Index Terms*— Face detection, anchor-free, feature aggregation, Gauss.**

I. Introduction

FACE detection aims to locate and predict boundary of human faces, which is an essential step for some important applications such as face recognition [1], [2], liveness detection [3] and facial age estimation [4], [5]. In the past decades, face detection has attracted more and more research interest. Recently, convolutional neural networks have revolutionized many areas, such as object detection [6], [7], image retrieval [8] and data compression [9], [10]. Naturally, convolutional neural networks have also emerged as the master algorithm in face detection. A series of excellent algorithms [11]–[18] based on deep learning have been proposed to dramatically improve the performance. Some methods [19], [20] focus on the acceleration and practical application of face detection methods. [21]–[23] provide new face detection benchmarks.

Manuscript received October 11, 2020; revised February 8, 2021 and April 13, 2021; accepted May 15, 2021. Date of publication May 21, 2021; date of current version April 5, 2022. This work was supported in part by the NSFC under Grant 61772407 and Grant 61732008. This article was recommended by Associate Editor J. Wu. *(Xin Li and Shenqi Lai contributed equally to this work.) (Corresponding author: Xueming Qian.)*

Xin Li is with the Faculty of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: lingfengyueguang@stu.xjtu.edu.cn).

Shenqi Lai was with the Faculty of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China. He is now with Meituan Dianping, Beijing 100102, China (e-mail: laishenqi@stu.xjtu.edu.cn).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, Xi'an Jiaotong University, Xi'an 710049, China, and also with SMILES Laboratory, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: qianxm@mail.xjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3082635.

Digital Object Identifier 10.1109/TCSVT.2021.3082635

Unlike image classification and semantic segmentation task which can generate the final result directly through a pure neural network, face detection usually needs to employ some pre-defined anchor boxes as the bridge to encode the location information, and then use non-maximum suppression (NMS) operation as a post-processing step to generate final detection results. Anchor box is a milestone work in the development of detection task, which leads to the trend of vision detection algorithm based on deep learning. In training phase, mainstream detection algorithms allocate positive and negative samples with the overlap between anchors and ground truth bounding boxes, and employ anchor as a reference to encode location information. In prediction phase, anchor-based methods still need to use anchor as a reference to reconstruct the prediction bounding boxes. Non-maximum suppression is another essential block independent of neural network. It removes duplicate detection boxes and reduces false positives. Almost all face detection methods use it to filter redundant detection boxes in post-processing step.

However, these operations outside neural networks bring some inconvenience and limitation. Most significantly, when we deploy a face detection model to an application program with neural network inference computing framework (such as TensoRT[1] or NCNN[2]), the most non-trivial work is always to write the anchor generation code and the corresponding post-processing operations code. It is not elegant to write the corresponding code of anchor generation and post-processing for different algorithms of face detection. Specifically, the pre-defined anchor boxes will make network only accept input images with fixed size. As the resolution of different image acquisition equipment is often different, the application of these models that only accept input images with fixed size will be limited. The alternative is to generate anchors dynamically according to the size of feature map in CNN inference stage, but this online dynamic generating anchors will reduce face detection speed. NMS is an essential module for traditional detection methods as it can remove redundant bounding boxes. As the NMS cannot be easily parallelized, it becomes a bottleneck in real time face detection. Usually, the time spent on convolutional operations will reduce significantly by GPU and NPU, while the time spent on NMS does not reduce.

In addition to these defects mentioned above, the anchor mechanism in existing object detection frameworks will introduce many hyper-parameters that need to be carefully tuned

[1]https://developer.nvidia.com/tensorrt

[2]https://github.com/Tencent/ncnn

in model design and training phase. Also, in order to achieve a high recall rate, anchor-based detectors usually need to densely place anchor boxes on the input image, e.g. there are more 30k anchor boxes uitilized in S3FD [12] and 100k in RetinaFace [24]. A great number of these anchors will bring heavy computing pressure to the post-processing.

In contrast, human vision system can easily locate and recognize faces at a glance without any predefined templates and complex post-processing operations like NMS. In other words, we human directly locate faces and predict the boundaries without enumerating the fixed rectangular regions and post-processing for selecting an appropriate from lots of candidate boxes. Similar to human visual system, some early works [25], [26] have attempted to leverage the fully convolutional network(FCN) framework that directly predicted bounding boxes without anchor boxes. However, these methods do not achieve satisfactory face detection results. Based on these observations, we raise a question. Is anchor necessary for face detection? Can we design a face detection algorithm that does not depend on the operations outside neural network such as NMS. In this work, we propose a simple and fast face detection framework which is compared with the prior state-of-the-art face detection performance.

To this end, we present our dual-branch center face (DBCFace), a completely pure convolutional neural network framework for face detection. Inspired by the recent anchor free general object detection methods [27], [28], we adopt keypoint estimation to locate face and parallel regression to predict face scale. Considering the large variations of faces (from several pixels to thousands of pixels), directly predicting faces of all scales in a fixed feature map is not stable. Instead, we develop a dual-branch parallel architecture, in which each branch is responsible for detecting faces in a fixed range. In previous works [11]–[17], the multi-branch detectors rely NMS operation to suppress the repeatedly detected bounding boxes. Duplicate detection candidate bounding boxes of the same object can be divided into two categories: one is that there are duplicated predictions around the same location within a single detector branch, and the other is adjacent detectors respond to the same object lying in the middle scale. In our method, we can accurately locate faces by extracting the peak of heatmap. This is the first step to avoid duplicated detection boxes within a single branch. Moreover, we propose a novel and effective technology, called branch route(BR) module to further avoid duplicated detection boxes between different branch detectors. As show in Fig. 1, the BR module generates a mask to predict the scale of face (small face or big face) for each position. The repeatable detection results are suppressed by BR module, thus it guarantees that the right result can be selected when both branches respond to the same face. To further enhance modeling capability for objects different scale, we propose a dual branch feature pyramid aggregation (FPA) module that fuse multi-level features into two single-scale feature maps. Our face detection framework is based on widely used feature pyramid network(FPN), since our DBCFace does not rely on the anchors and NMS operation, it can be implemented by a pure neural network and make full utilize of computing power of GPU. Compared to the existing state-of-the-art anchor-based face detection approaches [14], [15] at several common resolutions with single scale testing strategy, the proposed DBCFace shows speed advantage by a large margin. When using the same multi-scale testing strategy, our DBCFace achieves similar detection performances with anchor-based approaches.

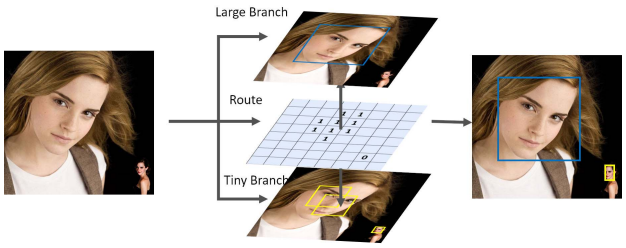


Fig. 1. Branch route module generates a binary mask to determine the detection result of which branch should be selected in every location. A value of 1 indicates that the outputs of large branch should be selected at this position, and value 0 indicates small branch.

The contributions of this paper are summarized as follows:

- We propose a dual-branch center face detector that conducts face detection in a pure convolutional neural network. It simplifies the processing steps of face detection and improves the detection speed.
- To reduce the influence of large variations of scales, we also propose feature pyramid aggregation (FPA) module and scale adaptive Gauss mask. Both of them can significantly improve the performance of face detection.
- We evaluate the proposed method on popular face detection benchmarks, such as AFW [29], PASCAL face [30], FDDB [22] and WIDER FACE [21]. Extensive experimental results show comparable performance to other state-of-the-art face detection methods and demonstrate the capability and competitiveness of the proposed method.

II. Related Work

Face detection is a classic task in computer vision, and has been extensively studied over the past few decades. Early face detectors are based on sliding-window and hand-crafted features. Similar to [20], it extracts the local binary pattern of the image and uses Bayesian classifier to recognize face. [18] is another traditional face detection framework that applies CNN as the feature extractor and employs blur-aware bi-channel CNN for further face versus non-face classification. The modern methods are most based on convolutional neural networks. CNN based face detection approaches can be roughly divided into two categories: anchor-based detectors [11]–[15], [17], [24] and anchor-free detectors [25], [26]. Considering deep learning approaches have achieved state-of-the-art results on all open datasets and far better than traditional detection methods, here we mainly introduce the related deep learning methods.

A. Anchor-Based Face Detector

Anchor-based detectors, which use multi-scale anchors at each cell of image to replace different sliding windows and

predict multiple candidate regions at the same time, inherit the concept of sliding window. Anchor was first proposed by Faster R-CNN [6], and then SSD [7] assigned different anchors to different level feature maps. Since then, anchor shows impressive performance in detection task and rapidly become a key component of modern detectors. Similar to general object detection [6], [7], [31], anchor is widely used in face detection, and even considered as the standard configuration for multi-scale face detection. In recent works [11]–[15], [17], [24], [32], [33], researchers focus on how to effectively design and make better use of anchor to improve the recall efficiency and location accuracy of face detection. FaceBoxes [17] makes anchors of different scales have same density on the input image to improve the recall rate of tiny faces. S3FD [12] introduces a scale compensation anchor matching strategy to ensure that faces of every scales match enough anchors through scale compensation. SRN [15] selectively applies two-step classification and regression to specific detection layers to significantly improve the performance of face detection. In [13], [24], RetinaNet [34] is used as a base framework to detect different scale faces and achieves satisfactory results. DEFace [32] expands the range of P layer in feature pyramid network to detect small faces.

B. Anchor-Free Face Detector

In contrast to anchor-based methods, anchor-free detectors directly predict bounding boxes without pre-defined anchor boxes. Previous works [25], [26] utilize fully convolution neural networks to regress a 4-D vector at every pixel of the feature map to locate the object. Recently, some works [27], [35], [36] adopt keypoint estimation to locate the objects. CornerNet [35] detected objects as paired keypoints, top-left corner and the bottom-right corner keypoints. Triplts [27] introduced the visual patterns within objects into the keypoint detection process by using center pooling and cascade corner pooling. ExtremeNet [36] detected four extreme points (top-most, leftmost, bottom-most, right-most) and one center point of objects to generate bounding boxes. These methods detect all scale objects on a single fixed feature map. Our method is designed for face detection. We assign faces with diverse scales into two different scales detectors to overcome the influence of scale variations.

C. Multi-Scale Face Detector

The multi-scale detection algorithms detect objects with diverse sizes by using multiple detectors in parallel. In general, detectors in the lower layers are used to detect small objects, and the upper are more likely to find large objects. Inspired by [7], [16], the multi-scale face detectors employ multi-branch detectors, followed by NMS step to produce the final detection results. However, this strategy increases the computational complexity and causes a large amount of duplicated detection boxes because each scale detector will try to detect objects lying in the intermediate scales. In this paper, we proposed a novel anchor-free multi-scale face detector, which only uses two parallel detectors and do not rely on NMS, but achieves the similar performance as those anchor-based methods with multi branches.

III. Proposed Method

In this section, we first give the overall architecture of DBCFace and then we provide specific description of each component, including anchor free detector, feature pyramid aggregation, dual branch prediction, scale adaptive Guass mask and training strategy.

A. Overall Architecture

Our proposed method is inspired by keypoint-based object detection method [37]. Considering that the scale of different faces varies greatly, we develop a dual-branch parallel network architecture, as show in Fig. 2. In our framework, the space of possible output bounding-box is divided into two subspaces by two parallel detectors, which are responsible for large and tiny face respectively. For each branch, the pyramid multi-scale features will be aggregated into a single scale feature map by FPA module, and an anchor free detector is employed to produce candidate bounding boxes upon the fused feature map. At last, the BR module is used to suppress repeatable detection results between two branches.

According to our statistics, about 90% of faces are distributed in the range of 16 to 512 in WIDER FACE [21] training set. We extracted five pyramid feature maps from $C3$ to $C7$. Specifically, these multi-scale feature maps extracted from backbone are denoted as $C3$, $C4$, and $C5$ (from stride 8 to stride 32), respectively, and then we add two down-sample 3×3 convolution layers after $C5$ to generate $C6$ and $C7$. We introduce high-level semantic features to the bottom layer in each branch (as shown in Fig. 3) respectively by following formula:

$$P_i = \Phi(Up(P_{i+1}) \oplus \Psi(C_i)). \tag{1}$$

where P_i and P_{i+1} represent the current output feature map and the deeper one respectively, Ψ is a 1×1 convolution layer to reduce the channel of original feature map C_i to 256, Up is up-sampling operation with factor of 2, $\oplus$ is the element-wise addition, and Φ is a 3×3 convolution to reduce the aliasing effect of up-sampling. It should be noticed that the merge starts from $P7$ and $P5\_1$ for two branches separately, and the feature $P7$ and $P5\_1$ computed by:

$$P_i = \Phi(\Psi(C_i)). \tag{2}$$

After that, we assign $P3$, P4 and $P5\_1$ to tiny branch, $P5\_2$, $P6$ and $P7$ to big branch as show in Fig. 2. Each branch employs FPA module to generate a single scale feature map ($H0$ for small branch and $H1$ for big branch) with multi-level features fusion. Next, an anchor-free detector is employed to generate detection results in each branch. At last, we employ BR module upon feature H (a concatenation of $H0$ and $H1$ in channel-wise) to filter these high overlap bounding boxes between two branches at every position.

B. Anchor Free Detector

We decompose face detection into two tasks: localization and scale prediction. Given a feature map $H_{det} \in R^{N \times C \times H \times W}$, the anchor free detector separately predicts two

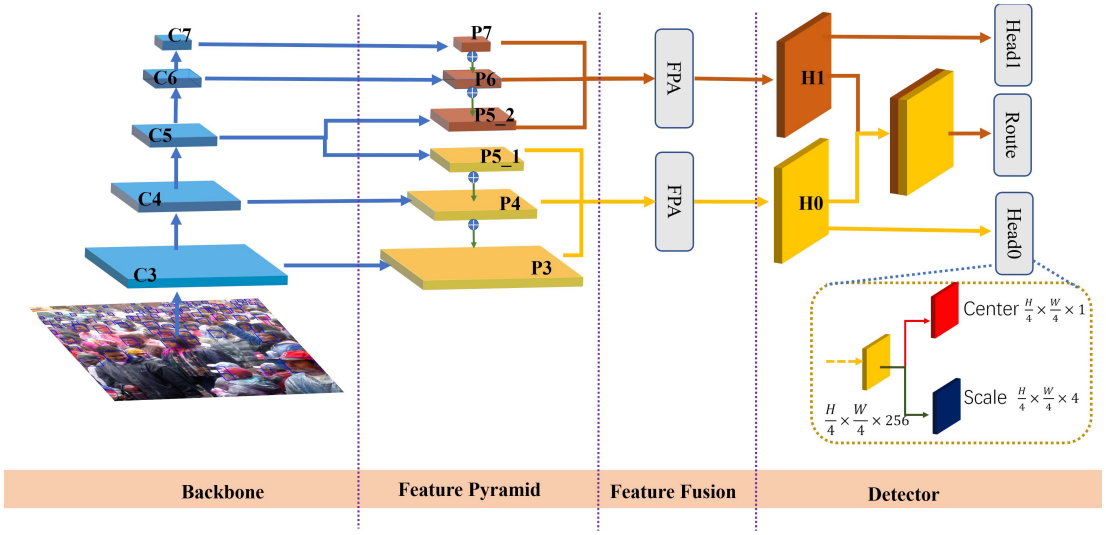


Fig. 2. An overview of the proposed DBCFace. It is designed on feature pyramid network, the features of which are divided into two groups. FPA module is used to aggregate multiple features into a single scale feature map for each group. An anchor free detector is used on each fused feature map to generate the candidate bounding boxes. Then, BR module is used to suppress the repeatable detection results between two branches.

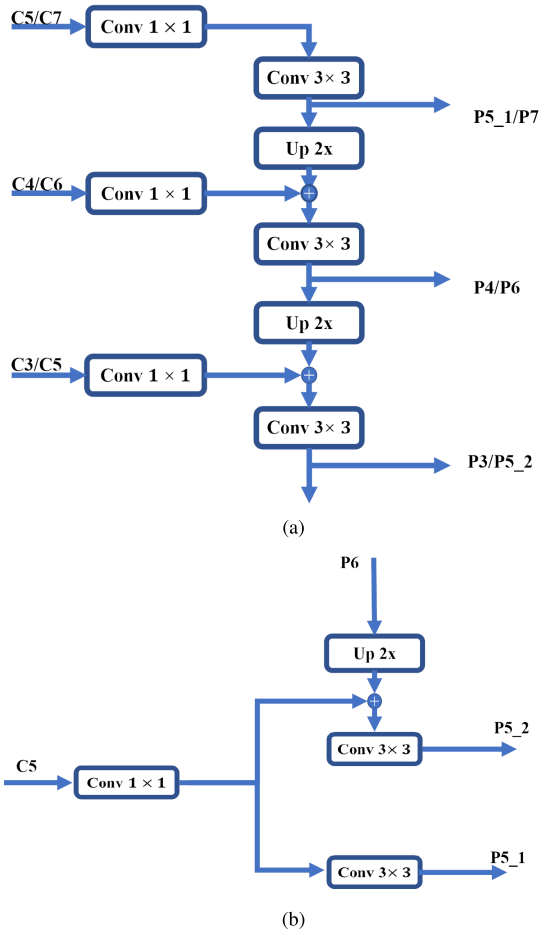


Fig. 3. (a) is the struct of additional layers for generating pyramid features. (b) is detailed struct for generating P5_1 and P5_2.

heatmaps $\widehat{C} \in R^{N\times 1\times H\times W}$ and $\widehat{S} \in R^{N\times 4\times H\times W}$. The first is used to predict where the face may exist, and the second is used to predict the scale of face at every position. In our work, H and W is 1/4 size of input images.

1) Localization: In localization task, we expect to detect the center point coordinates of face. We formulate it as a binary classification task (center or non-center) at pixel level. The box center is treated as positive sample while others are treated as the negative. However, the hard-definition brings more difficulties for training because it is hard to decide an exact center point. Similar to CenterNet [37], a 2D Gauss mask $G(.)$ centered at the location of each face is used to reduce the penalty for negative samples near the center point. Given the prediction center heatmap $\widehat{C}$ and corresponding Gauss mask G, the training objective with modified focal loss as follows:

$$L = -\frac{1}{N}\sum_{xyz}\begin{cases}\left(1-\hat{C}_{ijc}\right)^{\alpha} log\left(\hat{C}_{ijc}\right) & G_{ijc}=1\\ \left(1-G_{ijc}\right)^{\beta}\hat{C}_{ijc}^{\alpha} log\left(1-\hat{C}_{ijc}\right) & G_{ijc}\neq 1\end{cases} \tag{3}$$

where α and β are hyper-parameters in focal loss, N stands for the number of ground truth boxes. We use $\alpha = 2$ and $\beta = 4$ in all our experiments.

2) Scale Prediction: We formulate scale prediction as a regression task. For each center point, we predict a 4-D vector that is the distances from the location to four sides of bounding box. Due to the large scale variance of faces, we transform scale by $log(.)$ function instead directly regress it. If the location (x, y) is the center point of one face and $(x1, y1, x2, y2)$ is ground truth bounding box, the scale regression target for the location can be formulated as:

$$\begin{aligned} l_* &= log(x - x1), t_* = log(y - y1)\\ r_* &= log(x2 - x), b_* = log(y2 - y)\end{aligned} \tag{4}$$

where (l_*, t_*, r_*, b_*) is scale regression target for one face. We apply D-IOU Loss [38] to optimize it.

C. Feature Pyramid Aggregation

The feature pyramid will be divided into two groups and assigned to two branches. Our goal is to apply feature pyramid

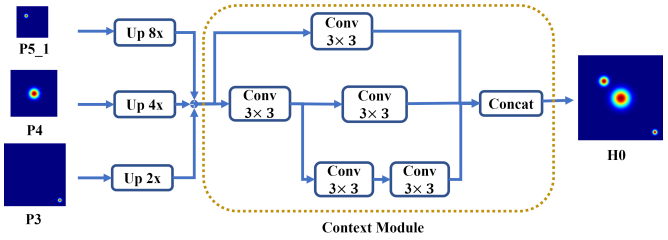


Fig. 4. The detial struct of FPA module.

aggregation module to fuse those multi-level features into a single scale feature map within each branch.

Taking tiny branch as an example, we will explain FPA module in detail. For small branch, the input of FPA is multi-level features P3, P4 and P5_1 (the stride is 8, 16, 32 separately), and output is the feature map H0(the stride is 4). We first rescale all inputs to the same size (the stride is 4). Then, an element-wise addition operation is used for merging those multi-level features with different receptive fields. After merging, we can get a feature map which integrates multi-level features. However, this simple fusion is rough, it cannot smoothly aggregate the multi-scale information of different layers into a feature map. In order to obtain better fusion effect, inspired by SSH [11], we apply a context module after add operation to reduce the aliasing effect of up-sampling and increase receptive field. Our FPA module can aggregate multiple features with multi-scale information into a single-scale feature map, as shown in Fig. 4.

D. Dual-Branch Prediction

Like most anchor-based methods, our two branches are responsible for different scale faces while the difference is that we only use two branches instead of multi branches to achieve similar performance. In our method, for each branch, upon the fused feature map H0 or H1, an anchor free detection head is employed. As mentioned in Section I, duplicated detection candidate bounding boxes of the same object comes from two aspects: the duplicated detection boxes within the same branch and between different branches. Next, we describe the detection within branch and merging detection boxes between different branches.

1) Prediction Within Branch: In inference, each branch will generate a localization heatmap where each value represents the probability whether this position is a face center. For each face, the response on heatmap is approximately Gaussian distribution where the score is attenuated from the center to the periphery. We regard the local peak of $n \times n$ grid in the heatmap as the face position and those objects around the local peak will be abandoned. Thus, the redundancy detection within the same branch will be suppressed. The local peak can be extracted by a simple max pooling operation as follows:

$$\hat{hm} = \max_{n} pooling(hm) \tag{5}$$

$$peak(x, y) = \begin{cases} hm(x, y), & hm(x, y) = \hat{hm}(x, y) \\ 0, & other \end{cases} \tag{6}$$

where n is the kernel size of max pooling and it is set to 3 in small branch and 9 in big branch.

Each peak is regarded as the center point of a face and the key-point value as its detection confidence.The location of center point is an integer coordinates (x_i, y_i). Then, we decode the detection result at each peak position as follows:

$$\begin{aligned} l_i &= e^{\hat{l}_i}, t_i = e^{\hat{t}_i} \\ r_i &= e^{\hat{r}_i}, b_i = e^{\hat{b}_i} \end{aligned} \tag{7}$$

where $\hat{l}_i$, $\hat{t}_i$, $\hat{r}_i$, and $\hat{b}_i$ are the scale information predicted by model at the i-th peak, and $(\hat{x}_i, \hat{y}_i)$ is integer coordinates of i-th center point.

The predicted bounding boxes are produced by:

$$(x_i - l_i, y_i - t_i, x_i + r_i, y_i + b_i) \tag{8}$$

2) Merging Between Branches: After using two branches to detect different sizes of faces, we get two candidate sets from two branches. We observed that some faces lying in the intermediate scales may be detected in two branches at the same time, but the regressed scales are not all correct (e.g., a huge face is detected in both branches, but the scale regressed by small branch is not accurate enough due to poor receptive filed). BR module is proposed to decide the output of which branch should be selected at each position of feature map. It is a pixel level binary classifier. During inference, BR module produces a mask that indicates whether this location is small face or big face. Thus, BR module can filter incorrect results, and suppress repeatable detection results of those faces lying intermediate scales between two branches.

E. Scale Adaptive Gauss Mask

In the previous works [27], [35], [37] based on keypoint estimation, they first determine a radius, and then use an unnormalized 2D Gauss kernel to produce Gauss mask $G(.)$. This Gauss mask ignores the shape information of objects because the mask produced by it is circle for any aspect ratio objects.

During training process, the circular mask will impose the same constraint in horizontal and vertical directions, which will increase the training difficulty. In training process, the heatmap will naturally adapt to training sample and become a similar ellipse. As a result, the circular mask will cause larger loss for these objects with large aspect ratio. The excessive optimization will make the model confused and reduce the accuracy of center point estimation. As for the response of the small target on the feature map is relatively weak and is more difficult to optimize in an inappropriate direction, the defect of circular mask is more serious in small face detection.

To conduct this problem, we modify it to the scale adaptive Gauss mask, whose variances in the x and y directions are related to the width and length of the object respectively. Thus, the final Gauss mask is a ellipse which is more like to the shape of input face, instead of circle. The modified Gauss kernel as follows:

$$G(x, y) = exp\left(-\frac{(x - x_o)^2}{2\sigma_x^2} - \frac{(y - y_o)^2}{2\sigma_y^2}\right) \tag{9}$$

where x_o and y_o are the coordinates of face center point on feature map, $\sigma_x = w$ and $\sigma_y = h$ (w and h are the corresponding width and height of the bounding box on the feature map).

F. Training

1) Loss Function: The final detector contains two detection branches and BR module, as shown in the right part of Fig. 2. Each detection branch consists of two subnetworks for center point prediction and scale regression respectively. Thus, the loss function is a multi-task loss, which can be represented as:

$$L = \sum_{i \in \{t,b\}} \left(\lambda_c \cdot L^i_{center} + \lambda_s \cdot L^i_{scale} \right) + \lambda_r \cdot L_{route} \quad (10)$$

where t and b represent tiny and big branches, respectively. L_{center}, and L_{scale} are the loss of center estimation and scale regression respectively, which is introduced in Part B of Section III. These two loss functions are the same in two branches, the superscript of which represents different branches. L_{route} is the loss of BR module, and we employ a standard binary cross entropy in this paper. λ_c, λ_s and λ_r are set to 1, 2 and 1 in experiment.

2) Scale-Aware Training Scheme: Unlike anchor-based methods, which assign training samples to different levels with the IOU between ground-truth and anchor boxes, we directly assign ground truth faces to the two branches with their scales. Instead of dividing the training samples into two parts by a hard threshold, we define an ignore range as follows:

$$lb < \sqrt{w * h} < ub. \quad (11)$$

where w and h are the corresponding width and height of the bounding box on the feature map. lb and ub are lower bound and upper bound of ignore range, respectively. The ground-truth faces are assigned to one of the three categories: scales are less than the lower bound, scales are bigger than upper bound, scales are between lower and upper bound, which are corresponded as {1, 0, -1}. During training, those samples labeled 0 and 1 are used to train BR module, and the other is ignored. For detection task, these samples labeled 0 and -1 are used to train small branch, while those labeled 1 and -1 are used for big branch.

IV. Experiments

In this section, we evaluate the proposed method on four common face detection datasets, including Annotated in the Wild(AFW) [29], PASCAL Face [30], Benchmark(FDDB) [22] and WIDER FACE [21]. Following standard practice, all models are trained on the WIDER FACE dataset while other datasets are only used to evaluate the final performance of our method. We also compare our approach with state-of-the-art face detection methods, such as [11], [12], [14], [15], [39]. To show the effectiveness of the proposed DBCFace detector, comprehensive ablation study and discussions are given.

A. Datasets

1) AFW dataset: It consists of 205 high-resolution images collected from Flickr. There are 473 labeled faces with large variations in apperance and viewpoint in this dataset.

2) PASCAL face dataset: This dataset is a subset of PASCAL VOC dataset. It selects the 851 images from PASCAL VOC and manually labels the 1335 face annotations.

3) FDDB: The images of FDDB are collected from unconstrained natural scenes. It has 2845 images with 5,171 annotated faces. These images have a wide range of difficulties, such as low images resolutions, make-ups, occlusions.

4) WIDER FACE: Due to dramatic variability in scale, pose and occlusion, WIDER FACE is the most challenging public face detection dataset. This dataset contains 32, 203 images with 393, 703 labeled faces, and those images are split into training (40%), validation (10%) and testing (50%) set. For testing and validation sets, the images are divided into three levels (Easy, Medium, Hard subset) according to the difficulties of detection. Ablation studies are performed on the validation set.

B. Experimental Setup

1) Data Augmentation: To make the model more robust to input face sizes and prevent over-fitting, random crop data augmentation strategy is adopted. More specifically, we random crop a square patch for each training image with a random size between [0.3,1] of the original image's short edge. Then we rescale this patch to 640×640. Besides random crop, random horizontal flip and photometric color distortion [12] are also employed.

2) Training Details: We use ResNet-50 [40] pre-trained on ImageNet [41] as the backbone for experiments. All models are trained by Adam optimizer with the batch size of 24. The learning rate is set to 1.5×10^{-4} for the first 100 epochs, and divided by 10 at 100 and 120 epochs. We use a gradual warm-up [42] strategy that increases the learning rate from 0 to the initial learning rate (1.5×10^{-4}) linearly at the first 4 epochs.

3) Testing Details: For these four datasets covered in this article, WIDER FACE and the other three datasets will perform different test strategies. For WIDER FACE dataset, we follow standard strategy [14], [15] that multi-scale testing and box voting are used to produce 750 best scoring results. On AFW, PASCAL face and FDDB datasets, we use single scale test strategy that we keep the aspect ratio of input image and rescale the shortest side of input image to 400 pixels. It should be noticed that NMS or boxes voting is not needed in single scale testing.

4) Evaluation Metrics: Similar to most publications, we use average precision (AP) to evaluate the performance of our method. AP is the area of precision/recall curve(as shown in Formula 12). Following the official guidelines of WIDER FACE [21], we calculate AP at Interception-of-Union (IOU) threshold at 0.5.

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} P(r) \quad (12)$$

where P is precision when the recall is r.

C. Ablation Study

To better understand the proposed DBCFace, we conduct extensive ablation experiments to examine how each proposed

TABLE I
THE ABLATION STUDY OF DUAL-BRANCH CENTER FACE ON THE WIDER FACE VALIDATION SET

Method	Multi-Scale Testing	Dual-Branch	FPA	Adaptive Gauss	Easy	Medium	Hard
Baseline					89.96	89.23	84.01
Ours		✓			92.14	90.84	84.73
Ours	✓	✓			93.88	92.47	87.84
Ours	✓	✓	✓		94.53	93.40	88.56
Ours	✓	✓		✓	94.68	93.34	89.30
Ours	✓	✓	✓	✓	95.84	94.96	90.34

component affects the final performance. For a fair comparison, we use the same parameter settings for all experiments, except for specified changes to the components. First, we implemented a baseline model by removing dual-branch, FPA module and adaptive gauss mask. Then, we evaluate the contribution of several important elements to our detector, including the FPA, Scale adaptive gauss mask and BR module. The results of ablation experiments are reported in Table I. The baseline model can reach an AP for 89.96, 89.23 and 84.01 on the easy, medium and hard subsets, respectively. Then we gradually add dual-branch, FPA and adaptive gauss mask into the baseline to evaluate the performance. The result shows all three components can improve the performance of face detection.

1) How import is the Dual-branch: As illustrated the first two lines in Table I, Dual-branch improves the AP of easy, medium, hard subsets by 2.18, 1.61, 0.92 compared to baseline (single branch). This result proves that dual-branch is more powerful than single branch. The baseline detect faces of all scales within single branch, which doesn't perform well in this large scale variations task, i.e, there are faces less than 10 pixels and larger than 1000 pixels in one image. As described in Part A of Section III, our method uses two branches to detect large face and tiny face respectively. Hence, this design reduces the scale variations within each branch and increases the detection ability for multi-scale faces. It is worth noting that the performance has a huge improvement by 2.18 and 1.61 on easy and medium task while 0.92 on hard task. As the ratio of large-scale faces in easy and medium subset is higher than hard subset, this significant improvement proves that the dual-branch structure is crucial for large faces.

2) How important is the FPA module: In the proposed method, FPA module is used to aggregate multiple features into a single-scale feature map within each branch. As described in Part C of Section III, element-wise addition operation is used for merging multi-level features, while context module is used to smooth the fused feature and increases the receptive field. We first train the model with FPA module, and then remove context module or feature fusion operation of FPA separately. When removing context module, we only use the fused features after element-wise addition to detect. If the feature fusion operation is ablated, we only use P3 and P5_1 followed by up-sample operation and context module for each branch separately. The results are shown in Table II. It turns out that the AP of easy, medium and hard subset drop significantly by 1.16, 1.62 and 1.04 by removing the context module, while the AP drops 0.89, 0.99 and 0.74 respectively by removing the feature fusion. Clearly, our FPA module plays a useful role.

TABLE II
FPA ANALYSIS. WE INVESTIGATE THE EFFECTIVENESS OF FEATURE FUSION AND CONTEXT MODULE, TWO COMPONENTS OF FPA MODULE

Method	Easy	Medium	Hard
w/o context	94.68	93.34	89.30
w/o fusion	94.93	93.97	89.60
w/ all	95.84	94.96	90.34

TABLE III
THE PERFORMANCE OF NMS AND BR MODULE ON WIDER FACE

Method	Easy	Medium	Hard
NMS	92.36	91.01	84.79
BR module fusion	92.14	90.84	84.73

3) How import is the Scale adaptive Gauss mask: As stated in Part B of Section III, we use Gauss mask to reduce the contribution of negative samples around the center point. In order to understand the influence of the form of Gauss mask, we train two models with Scale adaptive Gauss and original Gauss proposed in [35], [37]. By comparing with the fourth and last line in Table I, one can see that Scale adaptive Gauss mask effectively improves the performance, i.e., 1.31, 1.56, 1.78 on easy, medium and hard, respectively. This dramatic improvement shows that the mask adapted to the target shape will make the model easier to optimize and increase the accuracy of object center point estimation.

4) NMS vs BR Module: To verify the iou-based NMS is not needed for our DBCFace, we use NMS to replace BR module and peak extraction. The results are reported in Table III. It should be noticed that the performance is evaluated at single scale original images. As we expected, the performance of BR module only drops 0.22, 0.17 and 0.06 on the easy, medium and hard set of WIDER FACE than NMS. The impact is very minor, so we can replace NMS with BR module. As shown in Table IV, NMS spends more time with the number of candidate bounding boxes increasing. However, BR module nearly takes no time.

D. Evaluation On Benchmark

In this section, we compare our DBCFace with other different methods on all the common face detection benchmarks,including WIDER FACE, AFW, PASCAL Face, FDDB.

TABLE IV
TIME COST OF NMS WITH RESPECT TO THE NUMBER OF BOUNDING BOXES

number	500	5000	10000	20000	50000	100000
time(ms)	2.4	12.6	22.3	40.1	88.6	197.2

TABLE V
PERFORMANCE RESULTS ON THE VALIDATION SET OF WIDER FACE. ⋆ MEANS THAT THIS METHOD IS ORIGINALLY DESIGNED FOR GENERAL OBJECT DETECTION, AND WE REIMPLEMENT IT FOR FACE DETECTION

Method	Anchor-based	Easy	Medium	Hard
SSH [11]	✓	93.1	92.1	84.5
S3FD [12]	✓	93.7	92.5	85.9
EMO [39]	✓	94.9	93.1	86.9
PyramidBox [14]	✓	96.1	95.0	88.9
SFDNet [47]	✓	95.4	94.5	88.8
SRN [15]	✓	**96.4**	**95.2**	90.1
AFN [43]		87.4	85.1	69.6
FoveaBox⋆ [46]		95.59	93.50	67.83
FSAF⋆ [44]		94.51	92.35	68.97
FCOS⋆ [45]		95.01	90.61	55.02
CenterNet⋆ [37]		92.09	91.31	87.01
Ours		95.84	94.96	**90.34**

The comparison with the state-of-the-art methods is mainly on the WIDER FACE, and the other three datasets are only used to verify final performance and generalization of the model. The model is only trained on the WIDER FACE training dataset and evaluates on all four datasets.

1) WIDER FACE: We compare the proposed DBCFace with the state-of-the-art face detection methods [11], [12], [14], [15], [39], [43] and the state-of-the-art object detection methods [37], [44]–[46] on WIDER FACE val subsets. For a more comprehensive comparison, we evaluate the performance of several typical state-of-the-art anchor free methods [37], [44]–[46] on face detection datasets. The results are shown in Table V. Comparing with already published papers, DBCFace achieves competitive performance.

The scores of our method are very competitive compared to all the published methods in the literature. For example, the AP of SSH are 93.1, 92.1 and 84.5 for three level tasks, respectively. Our proposed method outperforms it with a large margin of 3.3, 3.1 and 5.6 in easy, medium and hard subset, respectively. Compared to recent anchor free methods, [37] is the most competitive one to us in terms of accuracy and speed because it also does not need anchor and NMS operation. However, [37] directly predicts all scales of objects on one fixed feature map, so it is not suitable for the task with large scale variation. Although [44]–[46] work well in easy and medium tasks, the detection results for small face are not ideal due to imbalance of training samples. In the general anchor free detection methods [44]–[46], they regard the area mapped by the object annotation box on the feature map as positive samples. In the training process, big face will be mapped to several positive samples, while those small faces less than 10 pixels in the hard subset will only be regarded as one positive sample. This imbalance between big and small objects makes these methods tend to detect large targets. In our method, any object is only regarded as one positive sample in the training process, so it will not cause the network prefer to detect big object. As for classical anchor-based methods [11], [12], [14], [15], [39], our DBCFace surpasses most of them in all level tasks, such as SSH,S3FD and SFDNet. Especially, it produces the best AP(90.34) in the hard set which contains a large number of tiny faces, surpassing all published approaches. This result shows that our method is superior to all other methods mentioned in above in small face detection task.As SRN adopts two-step regression strategy, the performance of our method is slightly lower than SRN on easy and medium tasks (as shown in Table V). However, our method has two obvious advantages over SRN. First, SRN needs anchor and NMS post-processing while our method does not rely on these operations. It can be implemented by a pure neural network and simplifies the processing steps of face detection to improve the detection speed. Second, our method is better than SRN in hard subset, which means our method has better performance in small face detection. Therefore, our DBCFace can accept input images with smaller size to reduce the computation compared with large input images. In a word, our DBCFace shows a feasible solution that it also can achieve excellent performance without the complicated anchor design and NMS operation.

Fig. 5 shows several qualitative results of detected faces in the WIDER FACE dataset by the proposed DBCFace. For challenging faces such as rotated, occluded and blur, our DBCFace achieve a satisfactory performance. It is worth noting that those very tiny faces (e.g. the first image in row 2, the second image in row 3 and the second image in row 4 in Fig. 5) can still be successfully detected. These results indicate that DBCFace is robust to varying scales, heavy occlusions, and severe blur degradations that are prevalent in unconstrained real-life scenarios.

2) FDDB: We evaluate the proposed DBCFace detector on the FDDB dataset and compare it with several state-of-the-art methods. The discrete ROC curves are shown in Fig. 6 with the results of other methods download from FDDB official site. Our method outperforms most of the state-of-art methods, which indicates our DBCFace can robustly detect unconstrained faces. Fig. 7 shows some qualitative results on the FDDB.

3) AFW and PASCAL face: In these two datasets, we compare the proposed DBCFace with some well-known works [28]–[30], [48], [49] and three commercial face detectors (Face.com, Face++ and Picasa). Considering that these two datasets are a little old and we only use them to verify the generalization of the model, we do not add more recent algorithms to compare. The precision-recall curves on AFW is shown in Fig. 8 and on PASCAL face is shown in Fig. 9. We reported the best average precision (AP) 99.87 on AFW dataset and 99.23 on PASCAL face. It turns out that our method outperforms others by a large margin and the AP on these two datasets tends to be saturated. The model trained on the WIDER FACE dataset still performs well on the AFW dataset and PASCAL face dataset, which shows that our method has good generalization.

Fig. 5. Examples on the WIDER FACE dataset. Our method works very well in complex scene where objects are small and highly occluded.

E. The Influence of Input Size

The input size can affect performance, especially for small faces. In practical application scene, users may pay more attention to the performance of single scale testing instead of multi-scale. Therefore, we evaluate the proposed DBCFace on the hard set of WIDER FACE validation set with different size. Similar with FAN [13], we keep the original aspect ratio and rescale the shortest side of input images to specific size. The results are show in Table VI. It is noticing that our method has more advantages in tiny face detection. When the min size is reduced to 600 pixels, it still reaches the AP of 82.23.

TABLE VI

THE EFFECT OF INPUT SIZE ON THE AVERAGE PRECISION(AP) IN WIDER FACE VALIDATION HARD SET

Min size	400	600	800	1000	1200	1400	Multi-scale
AP	68.23	82.23	86.25	87.48	87.9	87.86	90.34

F. Inference Time

We analyse the running speed of our method on a single NVIDIA GTX 1080Ti. The running speed is measured by the real detection time (including forward time and post-process time). We use batchsize 1 and a few common resolutions for testing. We regard average time on the WIDER FACE validation set as the real detection time. For comparison, we also test the two famous state-of-the-art anchor based methods PyramidBox [14] and SRN [15] under the same conditions. The final results are presented in Table VII.

As can be seen, our method obtains the highest speed at all three resolutions. For a VGA-resolution image (640×480)

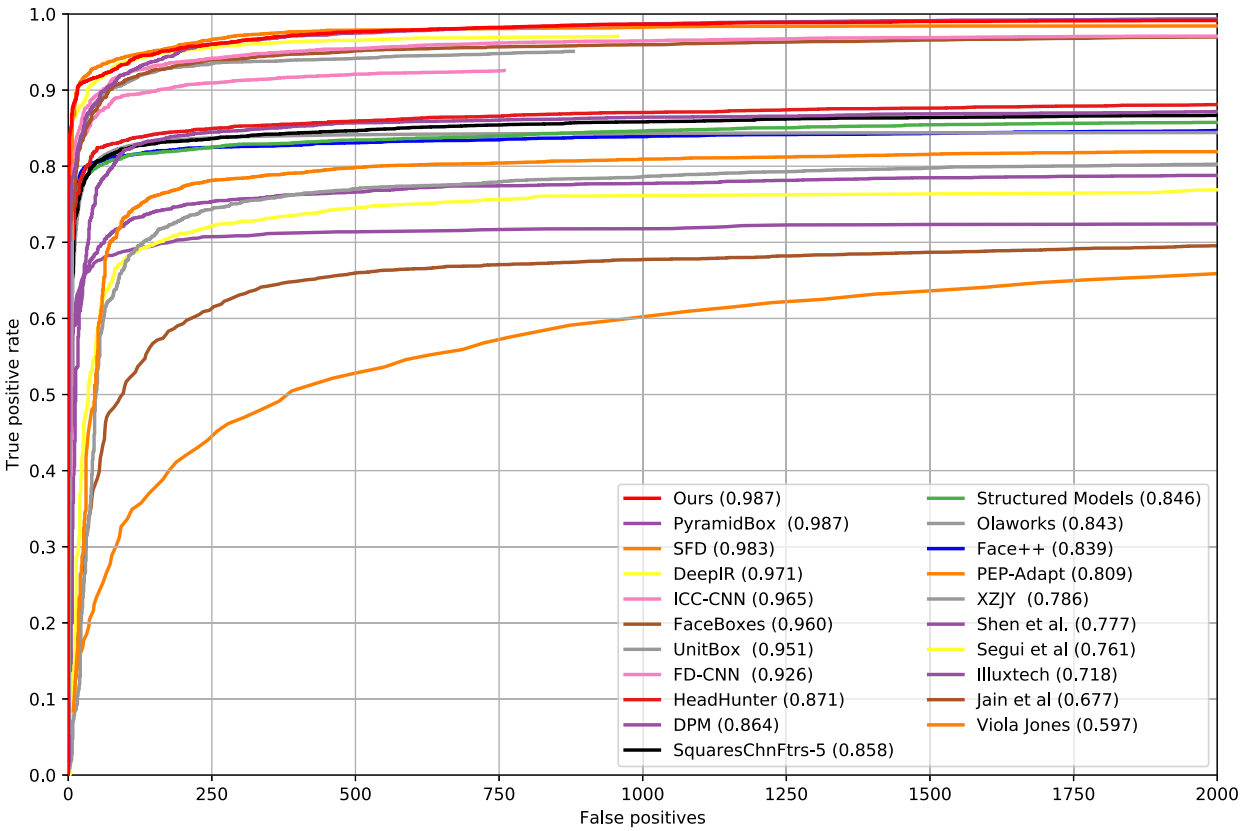
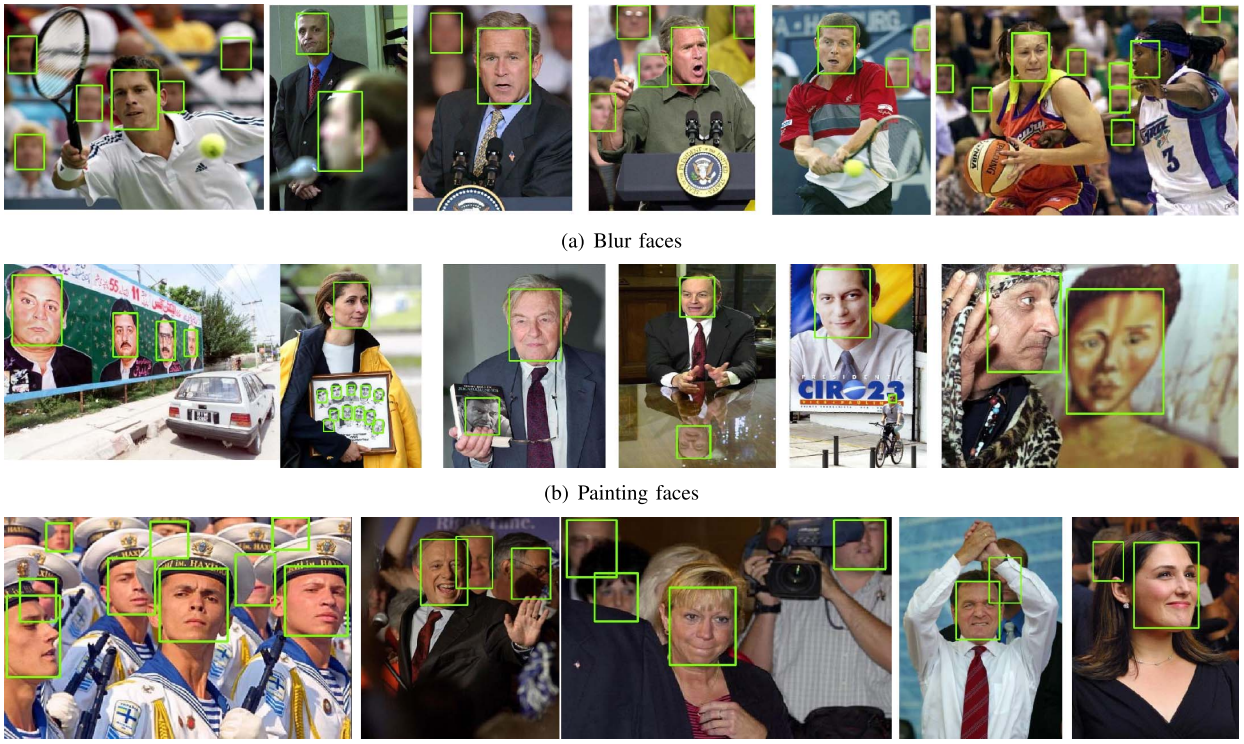


Fig. 6. The ROC curve on FDDB dataset by using the "discrete score" evaluation criterias.



(a) Blur faces

(b) Painting faces

(c) Occluded faces

Fig. 7. Qualitative results on the FDDB dataset. We display some examples for three special situation.

with batchsize 1, our method can run at 34.35 fps and satisfies high speed detection requirement. SRN and PyramidBox can achieve better accuracy, but they spent time over twice as ours. Especially, PyramidBox is much slower due to additional complex modules, although based on VGG16. Comparing with the anchor-based methods, our method performs on a single fully

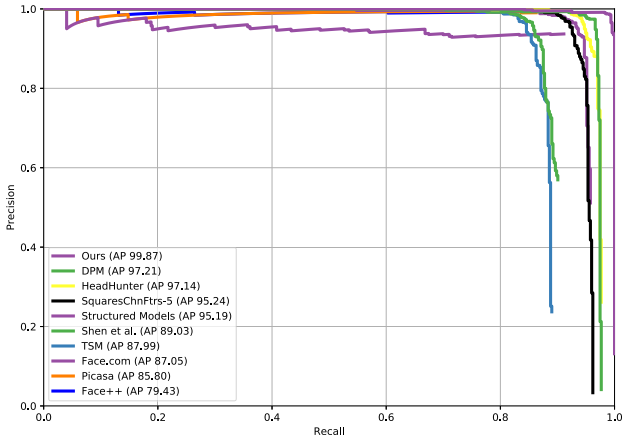


Fig. 8. Precision-recall curves on the AFW dataset.

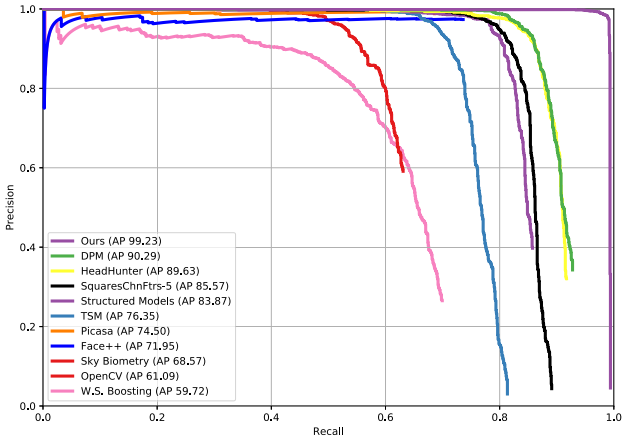


Fig. 9. Precision-recall curves on the PASCAL face dataset.

TABLE VII
DETECTION TIME WITH RESPECT TO DIFFERENT INPUT SIZES

Method	640 × 480	1280 × 720	1920 × 1080
SRN [15]	88.45ms	158.11ms	309.95ms
PyramidBox [14]	61.72ms	166.21ms	410.22ms
Ours	29.11ms	63.73ms	141.46ms

TABLE VIII
EFFECTIVENESS OF DIFFERENT BACKBONES

Backbone	easy	medium	hard
MobileNet V1x0.25	92.6	90.31	80.05
MobileNet V1	93.25	91.46	83.6
MobileNet V2	93.77	92.63	87.82
ResNet-18	93.68	92.32	87.57
ResNet-50	95.84	94.96	90.34

convolution network without complicated steps such as anchor generation and NMS. In our method, the post-processing step only includes the extraction of the local maximum in heatmap and the results selection between dual branches by BR module. About 95% of the detection time is spent on the inference stage of the network and the post-processing stage only takes about 2ms. Therefore, our approach is not limited by additional steps and will greatly improve the detection speed with a lighter network, especially for edge devices.

G. The Influence of Backbone

Faced with different application scenarios, we often need to trade-off between speed and performance. To better understand and easier use our DBCFace, we further conduct experiments to examine how different backbones affect the detection performance and running speed. Specifically, we use the same setting except for the feature extraction network. Following the setting in ResNet50-DBCFace, we implement MobileNet V1 × 0.25, MobileNet V1, MobileNet V2 and ResNet-18 in our DBCFace. The results are reported in Table VIII. It can be seen that our method can still achieve a high performance (92.6, 90.31 and 80.08 in three levels, respectively) even if we use Mobilenet v1 × 0.25, a lightweight network with 40M FLOPS.

TABLE IX
THE PERFORMANCE ON DIFFERENT FEATURE MAPS

Feature map	Easy	Medium	Hard
C3	69.13	74.04	71.56
C4	87.49	86.92	68.73
C5	90.52	85.57	41.44
C6	67.18	43.91	11.64

H. The Performance on Different Level Feature Maps

Different feature maps have different representation abilities and different effective receptive field. We take MobileNet V1 x 0.25 (a lightweight network) as backbone to explore the performance of different feature maps in different layers. We attend anchor free detector on different feature maps, respectively. The results are shown in Table IX. There are many small faces in hard subset, so the performance on hard subset is higher and the detection ability for small object is better. It can be seen the shallower feature map is suitable for detecting small objects, while the deeper feature map is better for big objects. As the downsampling rate of C6 is too big (64), only some large objects can be detected.

V. CONCLUSION

As graphics processing unit or neural-network processing units become increasingly powerful, those operations outside neural networks such as anchor generation and NMS may become the bottleneck of face detection speed. In this work, we propose an elegant and effective approach, referred as dual-branch center face detector, which can be implemented on a pure convolution neural network. By taking our FPA module and adaptive Gauss mask, DBCFace achieves comparable performance to state-of-the-art anchor-based methods on AFW, PASCAL face, FDDB and WIDER FACE datasets and the speed is even faster. Experiments show that those complex components such as anchor and NMS are unnecessary for face detection, and this pure convolutional neural method will obtain higher speedup ratio from the increase of computing units. We hope that the proposed DBCFace framework can be used as a starting point and baseline for pure convolution neural network face detection method.

REFERENCES

[1] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[2] S. Ge, C. Li, S. Zhao, and D. Zeng, "Occluded face recognition in the wild by identity-diversity inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3387–3397, Oct. 2020.

[3] X. Song, X. Zhao, L. Fang, and T. Lin, "Discriminative representation combinations for accurate face spoofing detection," *Pattern Recognit.*, vol. 85, pp. 220–231, Jan. 2019.

[4] A. Akbari, M. Awais, Z. Feng, A. Farooq, and J. Kittler, "Distribution cognisant loss for cross-database facial age estimation with sensitivity analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Oct. 7, 2020, doi: 10.1109/TPAMI.2020.3029486.

[5] F. Dornaika, I. Arganda-Carreras, and C. Belver, "Age estimation in facial images through transfer learning," *Mach. Vis. Appl.*, vol. 30, no. 1, pp. 177–187, Feb. 2019.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

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

[8] L. Wang, X. Qian, X. Zhang, and X. Hou, "Sketch-based image retrieval with multi-clustering re-ranking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4929–4943, Dec. 2020.

[9] Y. Wang, D. Liu, S. Ma, F. Wu, and W. Gao, "Ensemble learning-based rate-distortion optimization for end-to-end image compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1193–1207, Mar. 2021.

[10] D. Liu, Z. Chen, S. Liu, and F. Wu, "Deep learning-based technology in responses to the joint call for proposals on video compression with capability beyond HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1267–1280, May 2020.

[11] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "SSH: Single stage headless face detector," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4875–4884.

[12] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3FD: Single shot scale-invariant face detector," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 192–201.

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

[14] X. Tang, D. K. Du, Z. He, and J. Liu, "Pyramidbox: A context-assisted single shot face detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 797–813.

[15] C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou, "Selective refinement network for high performance face detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8231–8238.

[16] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

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

[18] J. Li, L. Liu, J. Li, J. Feng, S. Yan, and T. Sim, "Toward a comprehensive face detector in the wild," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 104–114, Jan. 2019.

[19] H. Mo *et al.*, "A multi-task hardwired accelerator for face detection and alignment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4284–4298, Nov. 2020.

[20] K.-Y. Chou and Y.-P. Chen, "Real-time and low-memory multi-faces detection system design with naive bayes classifier implemented on FPGA," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4380–4389, Nov. 2020.

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

[22] V. Jain and E. Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," Univ. Massachusetts, Amherst, Amherst, MA, USA, Tech. Rep. UM-CS-2010-009, 2010. [Online]. Available: http://vis-www.cs.umass.edu/fddb/

[23] G.-S. Hsu and T.-Y. Chu, "A framework for making face detection benchmark databases," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 2, pp. 230–241, Feb. 2014.

[24] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-shot multi-level face localisation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5203–5212.

[25] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "DenseBox: Unifying landmark localization with end to end object detection," 2015, *arXiv:1509.04874*. [Online]. Available: https://arxiv.org/abs/1509.04874

[26] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 516–520.

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

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

[29] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2879–2886.

[30] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image Vis. Comput.*, vol. 32, no. 10, pp. 790–799, Oct. 2014.

[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[32] T. M. Hoang, G. P. Nam, J. Cho, and I.-J. Kim, "DEFace: Deep efficient face network for small scale variations," *IEEE Access*, vol. 8, pp. 142423–142433, 2020.

[33] S. Hou, Y. Li, Y. Pan, X. Yang, and G. Yin, "A face detection algorithm based on two information flow block and retinal receptive field block," *IEEE Access*, vol. 8, pp. 30682–30691, 2020.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

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

[36] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 850–859.

[37] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.0785*. [Online]. Available: https://arxiv.org/abs/1904.0785

[38] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, Apr. 2020, pp. 12993–13000.

[39] C. Zhu, R. Tao, K. Luu, and M. Savvides, "Seeing small faces from robust anchor's perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5127–5136.

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

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

[42] P. Goyal *et al.*, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*. [Online]. Available: https://arxiv.org/abs/1706.02677

[43] C. Wang, Z. Luo, S. Lian, and S. Li, "Anchor free network for multi-scale face detection," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1554–1559.

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

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

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

[47] S. Zhang, L. Wen, H. Shi, Z. Lei, S. Lyu, and S. Z. Li, "Single-shot scale-aware network for real-time face detection," *Int. J. Comput. Vis.*, vol. 127, nos. 6–7, pp. 537–559, Jun. 2019.

[48] A. Ospina and F. Torres, "Countor: Count without bells and whistles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 720–735.

[49] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Detecting and aligning faces by image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3460–3467.

Xin Li received the B.S. degree in information engineering from Xi'an Jiaotong University in 2018, where he is currently pursuing the M.S. degree in information and communication engineering. His research interests include object detection, face detection, and image processing.

Shenqi Lai received the B.S. and M.S. degrees from Xi'an Jiaotong University, Xi'an, China, in 2014 and 2018, respectively. His research interests include multimedia retrieval, neural network acceleration, and computational aesthetics.

Xueming Qian (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree in electronics and information engineering from Xi'an Jiaotong University, Xi'an, in 2008. From 2010 to 2011, he was a Visiting Scholar with Microsoft Research Asia, Beijing, China. From 2011 to 2014, he was an Assistant Professor with Xi'an Jiaotong University, where he was an Associate Professor, and currently a Full Professor. He is currently the Director of the SMILES Laboratory, Xi'an Jiaotong University. His research interests include social media big data mining and search.