

PyramidBox: A Context-assisted Single Shot Face Detector.

Xu Tang^{*}[0000-0003-1375-0778], Daniel K. Du^{*}[0000-0001-5313-0413], Zeqiang He^[0000-0002-7192-0096], and Jingtuo Liu[†][0000-0003-0566-0780]

Baidu Inc.

tangxu02@baidu.com, daniel.kang.du@gmail.com, {hezeqiang,liujingtuo}@baidu.com

Abstract. Face detection has been well studied for many years and one of remaining challenges is to detect small, blurred and partially occluded faces in uncontrolled environment. This paper proposes a novel context-assisted single shot face detector, named *PyramidBox* to handle the hard face detection problem. Observing the importance of the context, we improve the utilization of contextual information in the following three aspects. First, we design a novel context anchor to supervise high-level contextual feature learning by a semi-supervised method, which we call it PyramidAnchors. Second, we propose the Low-level Feature Pyramid Network to combine adequate high-level context semantic feature and Low-level facial feature together, which also allows the PyramidBox to predict faces of all scales in a single shot. Third, we introduce a context-sensitive structure to increase the capacity of prediction network to improve the final accuracy of output. In addition, we use the method of Data-anchor-sampling to augment the training samples across different scales, which increases the diversity of training data for smaller faces. By exploiting the value of context, PyramidBox achieves superior performance among the state-of-the-art over the two common face detection benchmarks, Fddb and WIDER FACE. Our code is available in PaddlePaddle: https://github.com/PaddlePaddle/models/tree/develop/fluid/face_detection.

Keywords: face detection, context, single shot, PyramidBox

1 Introduction

Face detection is a fundamental and essential task in various face applications. The breakthrough work by Viola-Jones [1] utilizes AdaBoost algorithm with Haar-Like features to train a cascade of face vs. non-face classifiers. Since that, numerous of subsequent works [2-7] are proposed for improving the cascade detectors. Then, [8-10] introduce deformable part models (DPM) into face detection tasks by modeling the relationship of deformable facial parts. These

^{*} Equal contribution.

[†] Corresponding author.

methods are mainly based on designed features which are less representable and trained by separated steps.

With the great breakthrough of convolutional neural networks(CNN), a lot of progress for face detection has been made in recent years due to utilizing modern CNN-based object detectors, including R-CNN [11–14], SSD [15], YOLO [16], FocalLoss [17] and their extensions [18]. Benefiting from the powerful deep learning approach and end-to-end optimization, the CNN-based face detectors have achieved much better performance and provided a new baseline for later methods.

Recent anchor-based detection frameworks aim at detecting hard faces in uncontrolled environment such as WIDER FACE [19], SSH [20] and S³FD [21] develop scale-invariant networks to detect faces with different scales from different layers in a single network. Face R-FCN [22] re-weights embedding responses on score maps and eliminates the effect of non-uniformed contribution in each facial part using a position-sensitive average pooling. FAN [23] proposes an anchor-level attention by highlighting the features from the face region to detect the occluded faces.

Though these works give an effective way to design anchors and related networks to detect faces with different scales, how to use the contextual information in face detection has not been paid enough attention, which should play a significant role in detection of hard faces. Actually, as shown in Fig. 1, it is clear that faces never occur isolated in the real world, usually with shoulders or bodies, providing a rich source of contextual associations to be exploited especially when the facial texture is not distinguishable for the sake of low-resolution, blur and occlusion. We address this issue by introducing a novel framework of context assisted network to make full use of contextual signals as the following steps.

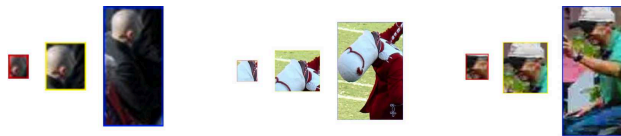


Fig. 1: Hard faces are difficult to be located and classified due to the lack of visual consistency, while the larger regions which give hints to the position of face are easier to be located and classified, such as head and body.

Firstly, the network should be able to learn features for not only faces, but also contextual parts such as heads and bodies. To achieve this goal, extra labels are needed and the anchors matched to these parts should be designed. In this work, we use a semi-supervised solution to generate approximate labels for contextual parts related to faces and a series of anchors called PyramidAnchors are invented to be easily added to general anchor-based architectures.

Secondly, high-level contextual features should be adequately combined with the low-level ones. The appearances of hard and easy faces can be quite differ-

ent, which implies that not all high-level semantic features are really helpful to smaller targets. We investigate the performance of Feature Pyramid Networks (FPN) [24] and modify it into a *Low-level Feature Pyramid Network (LFPN)* to join mutually helpful features together.

Thirdly, the predict branch network should make full use of the joint feature. We introduce the *Context-sensitive prediction module (CPM)* to incorporate context information around the target face with a wider and deeper network. Meanwhile, we propose a max-in-out layer for the prediction module to further improve the capability of classification network.

In addition, we propose a training strategy named as *Data-anchor-sampling* to make an adjustment on the distribution of the training dataset. In order to learn more representable features, the diversity of hard-set samples is important and can be gained by data augmentation across samples.

For clarity, the main contributions of this work can be summarized as five-fold:

1. We propose an anchor-based context assisted method, called PyramidAnchors, to introduce supervised information on learning contextual features for small, blurred and partially occluded faces.
2. We design the Low-level Feature Pyramid Networks (LFPN) to merge contextual features and facial features better. Meanwhile, the proposed method can handle faces with different scales well in a single shot.
3. We introduce a context-sensitive prediction module, consisting of a mixed network structure and max-in-out layer to learn accurate location and classification from the merged features.
4. We propose the scale aware Data-anchor-sampling strategy to change the distribution of training samples to put emphasis on smaller faces.
5. We achieve superior performance over state-of-the-art on the common face detection benchmarks FDDB and WIDER FACE.

The rest of the paper is organized as follows. Section 2 provides an overview of the related works. Section 3 introduces the proposed method. Section 4 presents the experiments and Section 5 concludes the paper.

2 Related Work

Anchor-based Face Detectors. Anchor was first proposed by Faster R-CNN [14], and then it was widely used in both two-stage and one single shot object detectors. Then anchor-based object detectors [15, 16] have achieved remarkable progress in recent years. Similar to FPN [24], Lin [17] uses translation-invariant anchor boxes, and Zhang [21] designs scales of anchors to ensure that the detector can handle various scales of faces well. FaceBoxes [25] introduces anchor densification to ensure different types of anchors have the same density on the image. S³FD [21] proposed anchor matching strategy to improve the recall rate of tiny faces.

Scale-invariant Face Detectors. To improve the performance of face detector to handle faces of different scales, many state-of-the-art works [20, 21, 23, 26] construct different structures in the same framework to detect faces with variant size, where the high-level features are designed to detect large faces while low-level features for small faces. In order to integrate high-level semantic feature into low-level layers with higher resolution, FPN [24] proposed a top-down architecture to use high-level semantic feature maps at all scales. Recently, FPN-style framework achieves great performance on both objection detection [17] and face detection [23].

Context-associated Face Detectors. Recently, some works show the importance of contextual information for face detection, especially for finding small, blurred and occluded faces. CMS-RCNN [27] used Faster R-CNN in face detection with body contextual information. Hu et al. [28] trained separate detectors for different scales. SSH [20] modeled the context information by large filters on each prediction module. FAN [23] proposed an anchor-level attention, by highlighting the features from the face region, to detect the occluded faces.

3 PyramidBox

This section introduces the context-assisted single shot face detector, *PyramidBox*. We first briefly introduce the network architecture in Sec. 3.1. Then we present a context-sensitive prediction module in Sec. 3.2, and propose a novel anchor method, named *PyramidAnchors*, in Sec. 3.3. Finally, Sec. 3.4 presents the associated training methodology including data-anchor-sampling and max-in-out.

3.1 Network Architecture

Anchor-based object detection frameworks with sophisticated design of anchors have been proved effective to handle faces of variable scales when predictions are made at different levels of feature map [14, 15, 20, 21, 23]. Meanwhile, FPN structures showed strength on merging high-level features with the lower ones. The architecture of PyramidBox(Fig. 2) uses the same extended VGG16 backbone and anchor scale design as S³FD [21], which can generate feature maps at different levels and anchors with equal-proportion interval. Low-level FPN is added on this backbone and a Context-sensitive Predict Module is used as a branch network from each pyramid detection layer to get the final output. The key is that we design a novel pyramid anchor method which generates a series of anchors for each face at different levels. The details of each component in the architecture are as follows:

Scale-equitable Backbone Layers. We use the base convolution layers and extra convolutional layers in S³FD [21] as our backbone layers, which keep layers of VGG16 from *conv1_1* to *pool5*, then convert *fc6* and *fc7* of VGG16 to *conv.fc* layers, and then add more convolutional layers to make it deeper.

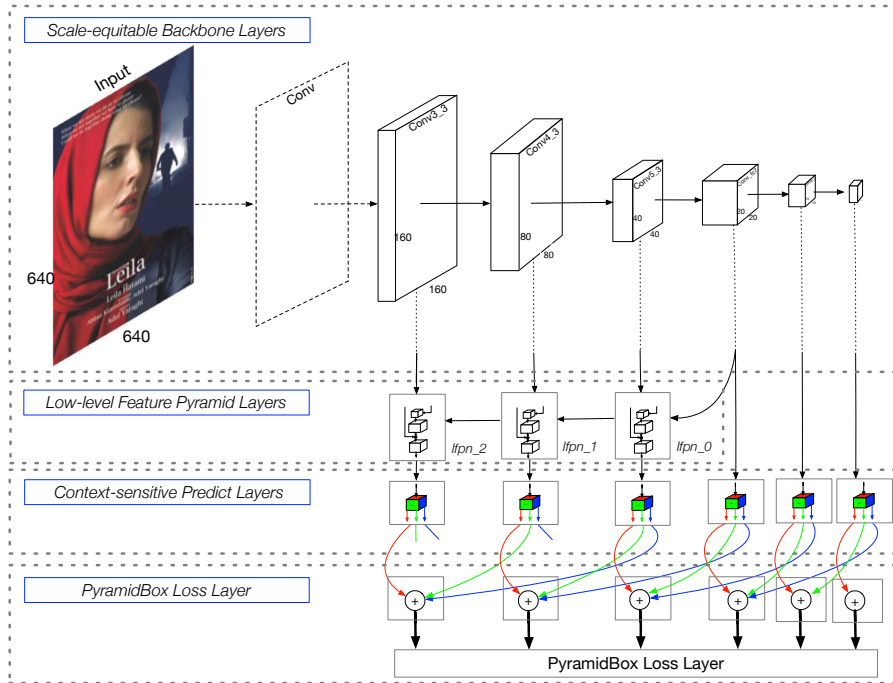


Fig. 2: Architecture of PyramidBox. It consists of **Scale-equitable Backbone Layers**, **Low-level Feature Pyramid Layers (LFPN)**, **Context-sensitive Predict Layers** and **PyramidBox Loss Layer**.

Low-level Feature Pyramid Layers. To improve the performance of face detector to handle faces of different scales, the low-level feature with high-resolution plays a key role. Hence, many state-of-the-art works [20, 21, 23, 26] construct different structures in the same framework to detect faces with variant size, where the high-level features are designed to detect large faces while low-level features for small faces. In order to integrate high-level semantic feature into low-level layers with higher resolution, FPN [24] proposed a top-down architecture to use high-level semantic feature maps at all scales. Recently, FPN-style framework achieves great performance on both objection detection [17] and face detection [23].

As we know, all of these works build FPN start from the top layer, which should be argued that not all high-level features are undoubtedly helpful to small faces. First, faces that are small, blurred and occluded have different texture feature from the large, clear and complete ones. So it is rude to directly use all high-level features to enhance the performance on small faces. Second, high-level features are extracted from regions with little face texture and may introduce noise information. For example, in the backbone layers of our PyramidBox, the receptive field [21] of the top two layers *conv7.2* and *conv6.2* are 724 and 468,

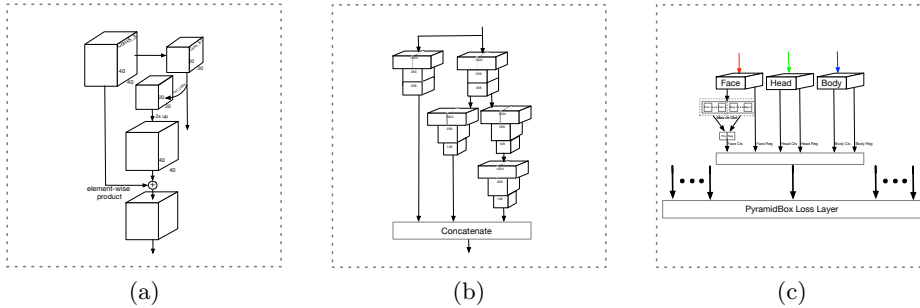


Fig. 3: (a) Feature Pyramid Net. (b) Context-sensitive Prediction Module. (c) PyramidBox Loss.

respectively. Notice that the input size of training image is 640, which means that the top two layers contain too much noisy context features, so they may not contribute to detecting medium and small faces.

Alternatively, we build the *Low-level Feature Pyramid Network (LFPN)* starting a top-down structure from a middle layer, whose receptive field should be close to the half of the input size, instead of the top layer. Also, the structure of each block of LFPN, as same as FPN [24], one can see Fig. 3(a) for details.

Pyramid Detection Layers. We select $lfpn_2$, $lfpn_1$, $lfpn_0$, $conv_fc7$, $conv6_2$ and $conv7_2$ as detection layers with anchor size of 16, 32, 64, 128, 256 and 512, respectively. Here $lfpn_2$, $lfpn_1$ and $lfpn_0$ are output layer of LFPN based on $conv3_3$, $conv4_3$ and $conv5_3$, respectively. Moreover, similar to other SSD-style methods, we use L2 normalization [29] to rescale the norm of LFPN layers.

Predict Layers. Each detection layer is followed by a *Context-sensitive Predict Module (CPM)*, see Sec 3.2. Notice that the outputs of CPM are used for supervising pyramid anchors, see Sec. 3.3, which approximately cover face, head and body region in our experiments. The output size of the l -th CPM is $w_l \times h_l \times c_l$, where $w_l = h_l = 640/2^{2+l}$ is the corresponding feature size and the channel size c_l equals to 20 for $l = 0, 1, \dots, 5$. Here the features of each channels are used for classification and regression of faces, heads and bodies, respectively, in which the classification of face need 4 ($= cp_l + cn_l$) channels, where cp_l and cn_l are max-in-out of foreground and background label respectively, satisfying

$$cp_l = \begin{cases} 1, & \text{if } l = 0, \\ 3, & \text{otherwise.} \end{cases}$$

Moreover, the classification of both head and body need two channels, while each of face, head and body have four channels to localize.

PyramidBox loss layers. For each target face, see in Sec. 3.3, we have a series of pyramid anchors to supervise the task of classification and regression

simultaneously. We design a *PyramidBox Loss*, see Sec. 3.4, in which we use softmax loss for classification and smooth L1 loss for regression.

3.2 Context-sensitive Predict Module

Predict Module. In original anchor-based detectors, such as SSD [15] and YOLO [16], the objective functions are applied to the selected feature maps directly. As proposed in MS-CNN [30], enlarging the sub-network of each task can improve accuracy. Recently, SSH [20] increases the receptive field by placing a wider convolutional prediction module on top of layers with different strides, and DSSD [31] adds residual blocks for each prediction module. Indeed, both SSH and DSSD make the prediction module deeper and wider separately, so that the prediction module get the better feature to classify and localize.

Inspired by the Inception-ResNet [32], it is quite clear that we can jointly enjoy the gain of wider and deeper network. We design the *Context-sensitive Predict Module (CPM)*, see Fig. 3(b), in which we replace the convolution layers of context module in SSH by the residual-free prediction module of DSSD. This would allow our CPM to reap all the benefits of the DSSD module approach while remaining rich contextual information from SSH context module.

Max-in-out. The conception of Maxout was first proposed by Goodfellow et al. [33]. Recently, S³FD [21] applied max-out background label to reduce the false positive rate of small negatives. In this work, we use this strategy on both positive and negative samples. Denote it as max-in-out, see Fig. 3(c). We first predict $c_p + c_n$ scores for each prediction module, and then select $\max c_p$ as the positive score. Similarly, we choose the max score of c_n to be the negative score. In our experiment, we set $c_p = 1$ and $c_n = 3$ for the first prediction module since that small anchors have more complicated background [25], while $c_p = 3$ and $c_n = 1$ for other prediction modules to recall more faces.

3.3 PyramidAnchors

Recently anchor-based object detectors [15–17, 24] and face detectors [21, 25] have achieved remarkable progress. It has been proved that balanced anchors for each scale are necessary to detect small faces [21]. But it still ignored the context feature at each scale because the anchors are all designed for face regions. To address this problem, we propose a novel alternatively anchor method, named *PyramidAnchors*.

For each target face, PyramidAnchors generate a series of anchors corresponding to larger regions related to a face that contains more contextual information, such as head, shoulder and body. We choose the layers to set such anchors by matching the region size to the anchor size, which will supervise higher-level layers to learn more representable features for lower-level scale faces. Given extra labels of head, shoulder or body, we can accurately match the anchors to ground truth to generate the loss. As it’s unfair to add additional labels, we implement it in a semi-supervised way under the assumption that regions with the same ratio and offset to different faces own similar contextual feature.

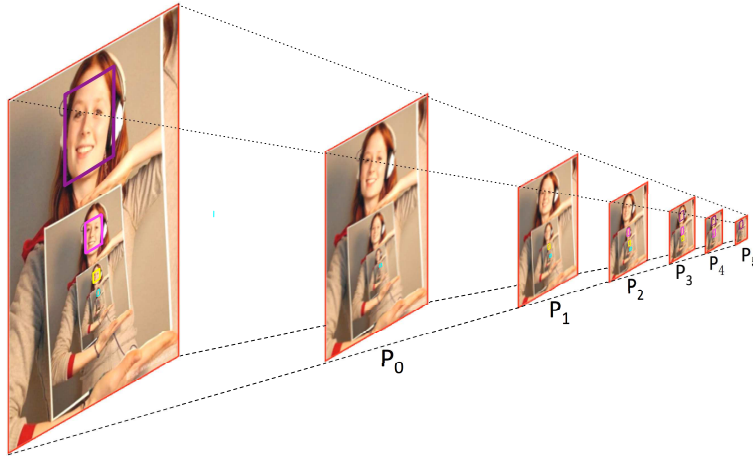


Fig. 4: Illustration of PyramidAnchors. For example, the largest purple face with size of 128 have pyramid-anchors at P_3 , P_4 and P_5 , where P_3 are anchors generated from *conv_fc7* labeled by the face-self, P_4 are anchors generated from *conv6_2* labeled by the head (of size about 256) of the target face, and P_5 are anchors generated from *conv7_2* labeled by the body (of size about 512) of the target face. Similarly, to detect the smallest cyan face with the size of 16, one can get a supervised feature from pyramid-anchors on P_0 which labeled by the original face, pyramid-anchors on P_1 which labeled by the corresponding head with size of 32, and pyramid-anchors on P_2 labeled by the corresponding body with size of 64.

Namely, we can use a set of uniform boxes to approximate the actual regions of head, shoulder and body, as long as features from these boxes are similar among different faces. For a target face localized at $region_{target}$ at original image, considering the $anchor_{i,j}$, which means the j -th anchor at the i -th feature layer with stride size s_i , we define the label of k -th pyramid-anchor by

$$label_k(anchor_{i,j}) = \begin{cases} 1, & \text{if } iou(anchor_{i,j} \cdot s_i / s_{pa}^k, region_{target}) > threshold, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

for $k = 0, 1, \dots, K$, respectively, where s_{pa} is the stride of pyramid anchors. $anchor_{i,j} \cdot s_i$ denotes the corresponding region in the original image of $anchor_{i,j}$, and $anchor_{i,j} \cdot s_i / s_{pa}^k$ represents the corresponding down-sampled region by stride s_{pa}^k . The *threshold* is the same as other anchor-based detectors. Besides, a PyramidBox Loss will be demonstrated in Sec. 3.4.

In our experiments, we set the hyper parameter $s_{pa} = 2$ since the stride of adjacent prediction modules is 2. Furthermore, let $threshold = 0.35$ and $K = 2$. Then $label_0$, $label_1$ and $label_2$ are labels of face, head and body respectively. One can see that a face would generate 3 targets in three continuous prediction

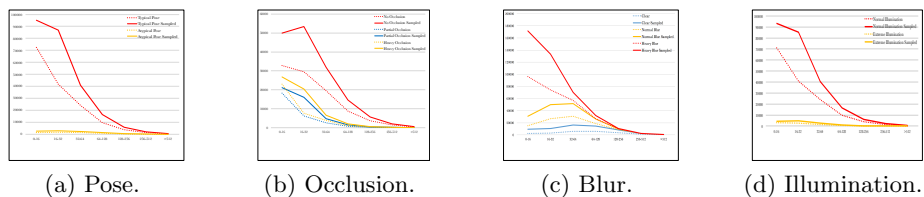


Fig. 5: Data-anchor-sampling changes the distribution of the train data. Dotted lines show the distribution of certain attribute, while solid lines represent the corresponding distribution of those attribute after the data-anchor-sampling.

modules, which represent for the face itself, the head and body corresponding to the face. Fig. 4 shows an example.

Benefited from the PyramidBox, our face detector can handle small, blurred and partially occluded faces better. Notice that the pyramid anchors are generated automatically without any extra label and this semi-supervised learning help PyramidAnchors extract approximate contextual features. In prediction process, we only use output of the face branch, so no additional computational cost is incurred at runtime, compared to standard anchor-based face detectors.

3.4 Training

In this section, we introduce the training dataset, data augmentation, loss function and other implementation details.

Train dataset. We trained PyramidBox on 12,880 images of the WIDER FACE training set with color distort, random crop and horizontal flip.

Data-anchor-sampling. Data sampling [34] is a classical subject in statistics, machine learning and pattern recognition, it achieves great development in recent years. For the task of objection detection, Focus Loss [17] address the class imbalance by reshaping the standard cross entropy loss.

Here we utilize a data augment sample method named Data-anchor-sampling. In short, data-anchor-sampling resizes train images by reshaping a random face in this image to a random smaller anchor size. More specifically, we first randomly select a face of size s_{face} in a sample. As previously mentioned that the scales of anchors in our PyramidBox, as shown in Sec. 3.1, are

$$s_i = 2^{4+i}, \text{ for } i = 0, 1, \dots, 5,$$

let

$$i_{anchor} = \operatorname{argmin}_i \operatorname{abs}(s_{anchor_i} - s_{face})$$

be the index of the nearest anchor scale from the selected face, then we choose a random index i_{target} in the set

$$\{0, 1, \dots, \min(5, i_{anchor} + 1)\},$$

finally, we resize the face of size of s_{face} to the size of

$$s_{target} = random(s_{i_{target}}/2, s_{i_{target}} * 2).$$

Thus, we got the image resize scale

$$s^* = s_{target}/s_{face}.$$

By resizing the original image with the scale s^* and cropping a standard size of 640×640 containing the selected face randomly, we get the anchor-sampled train data. For example, we first select a face randomly, suppose its size is 140, then its nearest anchor-size is 128, then we need to choose a target size from 16, 32, 64, 128 and 256. In general, assume that we select 32, then we resize the original image by scale of $32/140 = 0.2285$. Finally, by cropping a 640×640 sub-image from the last resized image containing the originally selected face, we get the sampled train data.

As shown in Fig. 5, data-anchor-sampling changes the distribution of the train data as follows: 1) the proportion of small faces is larger than the large ones. 2) generate smaller face samples through larger ones to increase the diversity of face samples of smaller scales.

PyramidBox loss. As a generalization of the multi-box loss in [13], we employ the *PyramidBox Loss* function for an image is defined as

$$L(\{p_{k,i}\}, \{t_{k,i}\}) = \sum_k \lambda_k L_k(\{p_{k,i}\}, \{t_{k,i}\}), \quad (2)$$

where the k -th pyramid-anchor loss is given by

$$L_k(\{p_{k,i}\}, \{t_{k,i}\}) = \frac{\lambda}{N_{k,cls}} \sum_{i_k} L_{k,cls}(p_{k,i}, p_{k,i}^*) + \frac{1}{N_{k,reg}} \sum_{i_k} p_{k,i}^* L_{k,reg}(t_{k,i}, t_{k,i}^*). \quad (3)$$

Here k is the index of pyramid-anchors ($k = 0, 1$, and 2 represents for face, head and body, respectively, in our experiments), and i is the index of an anchor and $p_{k,i}$ is the predicted probability of anchor i being the k -th object (face, head or body). The ground-truth label defined by

$$p_{k,i}^* = \begin{cases} 1, & \text{if the anchor down-sampled by stride } s_{pa}^k \text{ is positive,} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

For example, when $k = 0$, the ground-truth label is equal to the label in Fast R-CNN [13], otherwise, when $k \geq 1$, one can determine the corresponding label by matching between the down-sampled anchors and ground-truth faces. Moreover, $t_{k,i}$ is a vector representing the 4 parameterized coordinates of the predicted bounding box, and $t_{k,i}^*$ is that of ground-truth box associated with a positive anchor, we can define it by

$$t_{k,i}^* = \left(t_x^* + \frac{1-s_{pa}^k}{2} t_w^* s_{w,k} + \Delta_{x,k}, t_y^* + \frac{1-s_{pa}^k}{2} t_h^* s_{h,k} + \Delta_{y,k}, \right. \\ \left. s_{pa}^k t_w^* s_{w,k} - 2\Delta_{x,k}, s_{pa}^k t_h^* s_{h,k} - 2\Delta_{y,k} \right), \quad (5)$$

where $\Delta_{x,k}$ and $\Delta_{y,k}$ denote offset of shifts, $s_{w,k}$ and $s_{h,k}$ are scale factors respect to width and height respectively. In our experiments, we set $\Delta_{x,k} = \Delta_{y,k} = 0$, $s_{w,k} = s_{h,k} = 1$ for $k < 2$ and $\Delta_{x,2} = 0$, $\Delta_{y,2} = t_h^*$, $s_{w,2} = \frac{7}{8}$, $s_{h,2} = 1$ for $k = 2$. The classification loss $L_{k,cls}$ is log loss over two classes (face *vs.* not face) and the regression loss $L_{k,reg}$ is the smooth L_1 loss defined in [13]. The term $p_{k,i}^* L_{k,reg}$ means the regression loss is activated only for positive anchors and disabled otherwise. The two terms are normalized with $N_{k,cls}$, $N_{k,reg}$, and balancing weights λ and λ_k for $k = 0, 1, 2$.

Optimization. As for the parameter initialization, our PyramidBox use the pre-trained parameters from VGG16 [35]. The parameters of *conv_fc67* and *conv_fc7* are initialized by sub-sampling parameters from *fc6* and *fc7* of VGG16 and the other additional layers are randomly initialized with “xavier” in [36]. We use a learning rate of 10^{-3} for 80k iterations, and 10^{-4} for the next 20k iterations, and 10^{-5} for the last 20k iterations on the WIDER FACE training set with batch size 16. We also use a momentum of 0.9 and a weight decay of 0.0005 [37].

4 Experiments

In this section, we firstly analyze the effectiveness of our PyramidBox through a set of experiments, and then evaluate the final model on WIDER FACE and Fddb face detection benchmarks.

4.1 Model Analysis

We analyze our model on the WIDER FACE validation set by contrast experiments.

Baseline. Our PyramidBox shares the same architecture of S³FD, so we directly use it as a baseline.

Contrast Study. To better understand PyramidBox, we conduct contrast experiments to evaluate the contributions of each proposed component, from which we can get the following conclusions.

Low-level feature pyramid network (LFPN) is crucial for detecting hard faces. The results listed in Table 1 prove that LFPN started from a middle layer, using *conv_fc7* in our PyramidBox, is more powerful, which implies that features with large gap in scale may not help each other. The comparison between the first and forth column of Table 1 indicates that LFPN increases the mAP by 1.9% on hard subset. This significant improvement demonstrates the effectiveness of joining high-level semantic features with the low-level ones.

Data-anchor-sampling makes detector easier to train. We employ Data-anchor-sampling based on LFPN network and the result shows that our data-anchor-sampling effectively improves the performance. The mAP is increased by 0.4%, 0.4% and 0.6% on easy, medium and hard subset, respectively. One can see that Data-anchor-sampling works well not only for small hard faces, but also for easy and medium faces.

Table 1: Performances of LFPN starting from different layers.

Start layer	Baseline	<i>conv7_2</i> (FPN)	<i>conv6_2</i>	<i>conv_fc7</i> (LFPN)	<i>conv5_3</i>	<i>conv4_3</i>
RF/InputSize		1.13125	0.73125	0.53125	0.35625	0.16875
easy	94.0	93.9	94.1	94.3	94.1	93.6
mAP medium	92.7	92.9	93.1	93.3	93.1	92.5
hard	84.2	85.9	85.9	86.1	85.7	84.8

Table 2: The Parameters of PyramidAnchors.

Method	Baseline	(K, s_{pa})	(K, s_{pa})	(K, s_{pa})	(K, s_{pa})
		(1, 1.5)	(1, 2.0)	(1, 3.0)	(2, 2.0)
easy	94.0	93.8	94.2	94.3	94.7
mAP medium	92.7	92.7	93.0	93.1	93.3
hard	84.2	84.8	84.9	85.0	85.1

PyramidAnchor and PyramidBox loss is promising. By comparing the first and last column in Table 2, one can see that PyramidAnchor effectively improves the performance, i.e., 0.7%, 0.6% and 0.9% on easy, medium and hard, respectively. This dramatical improvement shows that learning contextual information is helpful to the task of detection, especially for hard faces.

Wider and deeper context prediction module is better. Table 3 shows that the performance of CPM is better than both DSSD module and SSH context module. Notice that the combination of SSH and DSSD gains very little compared to SSH alone, which indicates that large receptive field is more important to predict the accurate location and classification. In addition, by comparing the last two column of Table 4, one can find that the method of Max-in-out improves the mAP on WIDER FACE validation set about +0.2%(Easy), +0.3%(Medium) and +0.1%(Hard), respectively.

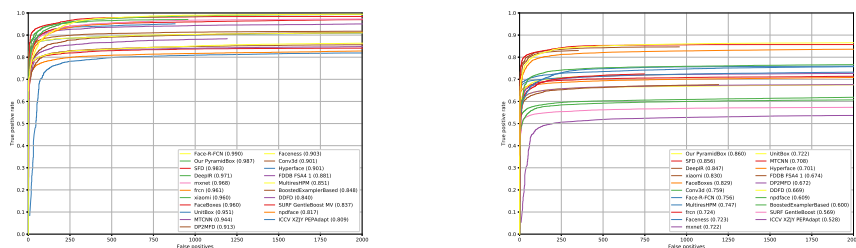
Table 3: Context-sensitive Predict Module.

Method	DSSD prediction module	SSH context module	CPM
easy	95.3	95.5	95.6
mAP medium	94.3	94.3	94.5
hard	88.2	88.4	88.5

To conclude this section, we summarize our results in Table 4, from which one can see that mAP increase 2.1%, 2.3% and **4.7%** on easy, medium and **hard** subset, respectively. This sharp increase demonstrates the effectiveness of proposed PyramidBox, especially for hard faces.

Table 4: Contrast results of the PyramidBox on WIDER FACE validation subset.

Contribution		Baseline	PyramidBox				
	lfpn?		✓	✓	✓	✓	
	data-anchor-sampling?			✓	✓	✓	
	pyramid-anchors?				✓	✓	
	context-predict-module?					✓	
	max-in-out?					✓	
mAP	easy	94.0	94.3	94.7	95.5	95.9	96.1
	medium	92.7	93.3	93.7	94.3	94.7	95.0
	hard	84.2	86.1	86.7	88.3	88.8	88.9



(a) Discontinuous ROC curves

(b) Continuous ROC curves

Fig. 6: Evaluation on the Fddb dataset.

4.2 Evaluation on Benchmark

We evaluate our PyramidBox on the most popular face detection benchmarks, including Face Detection Data Set and Benchmark (Fddb) [38] and WIDER FACE [39].

Fddb Dataset. It has 5,171 faces in 2,845 images collected from the Yahoo! news website. We evaluate our face detector on Fddb against the other state-of-art methods [4, 19, 21, 25, 30, 40–54]. The PyramidBox achieves state-of-art performance and the result is shown in Fig. 6(a) and Fig. 6(b).

WIDER FACE Dataset. It contains 32,203 images and 393,703 annotated faces with a high degree of variability in scale, pose and occlusion. The database is split into training (40%), validation (10%) and testing (50%) set, where both validation and test set are divided into “easy”, “medium” and “hard” subsets, regarding the difficulties of the detection. Our PyramidBox is trained only on the training set and evaluated on both validation set and testing set comparing with the state-of-the-art face detectors, such as [6, 20–23, 25–28, 30, 39, 40, 43, 51, 55, 56]. Fig. 7 presents the precision-recall curves and mAP values. Our PyramidBox outperforms others across all three subsets, i.e. 0.961 (easy), 0.950 (medium),

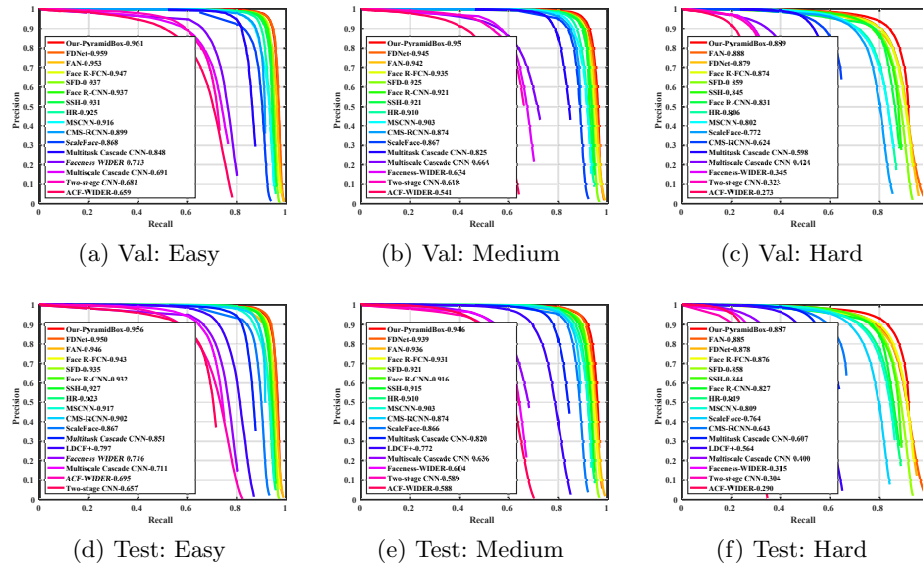


Fig. 7: Precision-recall curves on WIDER FACE validation and test sets.

0.889 (hard) for validation set, and 0.956 (easy), 0.946 (medium), 0.887 (hard) for testing set.

5 Conclusion

This paper proposed a novel context-assisted single shot face detector, denoted as PyramidBox, to handle the unconstrained face detection problem. We designed a novel context anchor, named PyramidAnchor, to supervise face detector to learn features from contextual parts around faces. Besides, we modified feature pyramid network into a low-level feature pyramid network to combine features from high-level and high-resolution, which are effective for finding small faces. We also proposed a wider and deeper prediction module to make full use of joint feature. In addition, we introduced Data-anchor-sampling to augment the train data to increase the diversity of train data for small faces. The experiments demonstrate that our contributions lead PyramidBox to the state-of-the-art performance on the common face detection benchmarks, especially for hard faces.

Acknowledgments. We wish to thank Dr. Shifeng Zhang and Dr. Yuguang Liu for many helpful discussions.

References

1. Viola, P., Jones, M.J.: Robust real-time face detection. *International Journal of Computer Vision* **57**(2) (2004) 137–154
2. Brubaker, S.C., Wu, J., Sun, J., Mullin, M.D., Rehg, J.M.: On the design of cascades of boosted ensembles for face detection. *International Journal of Computer Vision* **77**(1-3) (2008)
3. Pham, M.T., Cham, T.J.: Fast training and selection of haar features using statistics in boosting-based face detection. In: *ICCV* (2007)
4. Liao, S., Jain, A.K., Li, S.Z.: A fast and accurate unconstrained face detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **38** (2016)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* **60**(2) (2004)
6. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Aggregate channel features for multi-view face detection. In: *IJCB* (2014) 1–8
7. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. In: *CVPR* **2** (2006)
8. Mathias, M., Benenson, R., Pedersoli, M., Gool, L.V.: Face detection without bells and whistles. In: *ECCV* (2014)
9. Yan, J., Lei, Z., Wen, L., Li, S.Z.: The fastest deformable part model for object detection. In: *CVPR* (2014)
10. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: *CVPR* (2012)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: *CVPR* (2014)
12. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(3) (2016)
13. R.Girshick: Fastr-cnn. In: *ICCV* (2015)
14. Ren, S., Girshick, K.H.R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: *NIPS* (2015)
15. Liu, W., Anguelov, D., D.Erhan, Christian, S., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: single shot multibox detector. In: *ECCV* (2016)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: *CVPR* (2016)
17. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *ICCV* (2017)
18. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. *arXiv preprint* (2017)
19. Barbu, A., Gramajo, G.: Face detection with a 3d model. *arXiv preprint arXiv::1404.3596* (2014)
20. Najibi, M., Samangouei, P., Chellappa, R., Davis, L.S.: Ssh: Single stage headless face detector. In: *ICCV* (2017)
21. Zhang, S., Zhu, X., Lei, X., Shi, H., Wang, X., Li, S.Z.: S³fd: Single shot scale-invariant face detector. In: *ICCV* (2017)
22. Wang, Y., Ji, X., Zhou, Z., Wang, H., Li, Z.: Detecting faces using region-based fully convolutional networks. *arXiv preprint arXiv::1709.05256* (2017)
23. Wang, J., Y.Yuan, Yu, G.: Face attention network: An effective face detector for the occluded faces. *arXiv preprint arXiv::1711.07246* (2017)

24. Lin, T.Y., Dollár, P., Girshick, R.: Feature pyramid networks for object detection. In: CVPR (2017)
25. Zhang, S., Zhu, X., Lei, X., Shi, H., Wang, X., Li, S.Z.: Faceboxes: A cpu real-time face detector with high accuracy. arXiv preprint arXiv::1708.05234 (2017)
26. Yang, S., Xiong, Y., Loy, C.C., Tang, X.: Face detection through scale-friendly deep convolutional networks. arXiv preprint arXiv::1706.02863 (2017)
27. Zhu, C., Zheng, Y., Luu, K., Savvides, M.: Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. arXiv preprint arXiv::1606.05413 (2016)
28. Hu, P., Ramanan, D.: Finding tiny faces. In: CVPR (2017)
29. Liu, W., Rabinovich, A., Berg, A.C.: Parsenet: Looking wider to see better. ICLR (2016)
30. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multiscale deep convolutional neural network for fast object detection. In: ECCV (2016)
31. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv::1701.06659
32. Szegedy, C., Ioffe, S., Vanhoucke, V.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv::1602.07261 (2016)
33. Goodfellow, I.J., Farley, D.W., Mirza, M., Courville, A., Bengio, Y.: Maxout networks. (2013)
34. Thompson, S.K.: Sampling. WILEY (Mar. 2012)
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Internatinoal Journal of Computer Vision **115**(3) (2015)
36. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Aistats **9** (2010)
37. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
38. Jain, V., Learned-Miller, E.G.: Fddb: A benchmark for face detection in unconstrained settings. UMass Amherst Technical Report (2010)
39. S. Yang and, P.L.n.C.C.L., Tang, X.: Wider face: A face detection benchmark. In: CVPR (2016)
40. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. In: SPL **23**(10) (2016)
41. Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T.: Unitbox: An advanced object detection network. In: ACMMM (2016)
42. Triantafyllidou, D., Tefas, A.: A fast deep convolutional neural network for face detection in big visual data. In INNS Conference on Big Data (2016)
43. S. Yang and, P.L.n.C.C.L., Tang, X.: From facial parts responses to face detection: A deep learning approach. In: ICCV (2015)
44. Li, Y., Sun, B., Wu, T., Wang, Y.: Facedetection with end-to-end integration of a convnet and a 3d model. (2016)
45. Farfade, S.S., Saberian, M.J., Li, L.J.: Multi-view face detection using deep convolutional neural networks
46. Ghiasi, G., Fowlkes, C.: Occlusion coherence: Detecting and localizing occluded faces. (2015)
47. Kumar, V., Namboodiri, A., Jawahar, C.: Visual phrases for exemplar face detection. In: ICCV (2015)
48. Li, H., Hua, G., Lin, Z., Brandt, J., Yang, J.: Probabilistic elastic part model for unsupervised face detector adaptation. In: ICCV (2013)

49. Li, J., Zhang, Y.: Learning surf cascade for fast and accurate object detection. In: CVPR (2013)
50. Li, H., Lin, Z., Brandt, J., Shen, X., Hua, G.: Efficient boosted exemplar-based face detection. In: CVPR (2014)
51. Ohn-Bar, E., Trivedi, M.M.: To boost or not to boost? on the limits of boosted trees for object detection. In: ICPR (2016)
52. Ranjan, R., Patel, V.M., Chellappa, R.: A deep pyramid deformable part model for face detection. In: BTAS (2015)
53. Ranjan, R., Patel, V.M., Chellappa, R.: Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. arXiv preprint arXiv:1603.01249 (2016)
54. Wan, S., Chen, Z., Zhang, T., Zhang, B., Wong, K.K.: Bootstrapping face detection with hard negative examples. arXiv preprint arXiv: 1608.02236 (2016)
55. Zhang, C., Xu, X., Tu, D.: Face detection using improved faster rcnn. arXiv preprint arXiv: 1802.02142 (2018)
56. Wang, H., Li, Z., Ji, X., Wang, Y.: Face r-cnn. arxiv preprint. arXiv preprint arXiv:1706.01061 **7** (2017)