Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Faceboxes: A CPU real-time and accurate unconstrained face detector

Shifeng Zhang[a,b], Xiaobo Wang[a,b], Zhen Lei[a,b,*], Stan Z. Li[a,b]

[a] CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b] University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 26 January 2019
Revised 11 June 2019
Accepted 24 July 2019
Available online xxx

Communicated by Dr. Wang QI

Keywords:
Face detection
CPU real-time
Convolutional neural network

ABSTRACT

Although tremendous strides have been made in face detection, one of the remaining open issues is to achieve CPU real-time speed as well as maintain high performance, since effective models for face detection tend to be computationally prohibitive. To address this issue, we propose a novel face detector, named FaceBoxes, with superior performance on both speed and accuracy. Specifically, the proposed method has a lightweight yet powerful network that consists of the Rapidly Digested Convolution Layers (RDCL) and the Multiple Scale Convolution Layers (MSCL). The former is designed to enable FaceBoxes to achieve CPU real-time speed, while the latter aims to enrich the features and discretize anchors over different layers to handle faces of various scales. Besides, we propose a new anchor densification strategy to make different types of anchors have the same density on the image, which significantly improves the recall rate of small faces. Finally, we present a Divide and Conquer Head (DCH) to boost the prediction ability of the detection layer using above strategy. As a consequence, the proposed detector runs at 28 FPS on the CPU and 254 FPS using a GPU for VGA-resolution images. Moreover, the speed of FaceBoxes is invariant to the number of faces. We evaluate the proposed method on several face detection benchmarks including AFW, PASCAL face, FDDB, WIDER FACE and achieve state-of-the-art performance among CPU real-time methods.

1. Introduction

Face detection is a long-standing problem in computer vision and pattern recognition with extensive applications including face recognition, tracking, animation and expression analysis, to name a few. With the great progress over the past few decades, especially the breakthrough of deep convolutional neural network (CNN), face detection has been successfully applied in our daily life under the restricted scenarios.

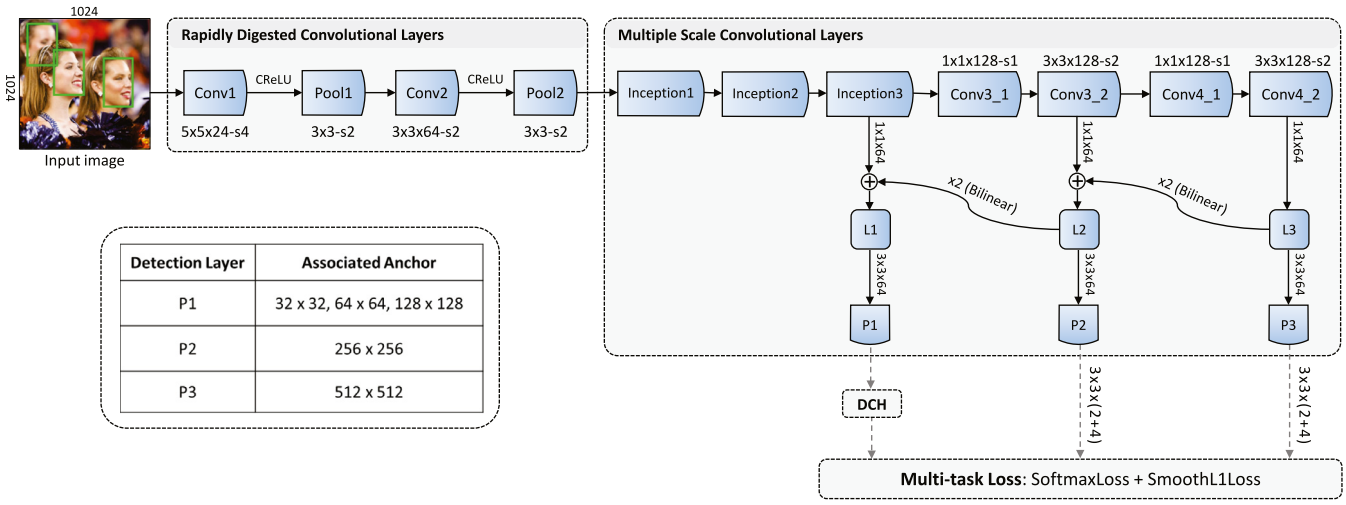
However, there are still some challenging issues in uncontrolled face detection, especially on the CPU devices. These challenges mainly come from two requirements for face detectors: (1) The large visual variation of faces in the cluttered backgrounds requires face detectors to accurately address a complicated face and non-face classification problem; (2) the large search space of possible face positions and sizes further imposes a time efficiency requirement. These two requirements are conflicting, since high-accuracy face detectors tend to be computationally expensive. Therefore, it is one of the remaining open issues for practical face detectors on the CPU devices to achieve real-time speed as well as maintain high performance.

In order to meet these two conflicting requirements, face detection has been studied mainly in two ways. The early way is based on hand-crafted features and classifiers. Following the pioneering work of Viola and Jones [1], most of early face detection methods have focused on designing robust features (*e.g.*, Haar [1], HOG [2]) and training effective classifiers (*e.g.*, Adaboost [3]). Besides the cascade structure, the deformable part model (DPM) [4] is introduced into face detection tasks and achieves remarkable performance. However, these methods highly depend on non-robust hand-crafted features and optimize each component separately, limiting the performance of these methods when they are deployed in real life complex scenarios. In brief, these early traditional methods are efficient on the CPU devices but not accurate enough against the large visual variation of faces.

The other way is based on the deep convolutional neural networks (CNNs), which have significantly improved the state-of-the-art performance and rapidly become the tool of choice [5–7]. On the one hand, some methods [8–11] use CNN as a feature extractor in traditional face detection framework to improve the performance. On the other hand, many works [12,13] regard face detection as a special case of generic object detection and solve it via CNN-based object detection algorithm relying on identifying

* Corresponding author.
E-mail addresses: shifeng.zhang@nlpr.ia.ac.cn (S. Zhang), xiaobo.wang@nlpr.ia.ac.cn (X. Wang), zlei@nlpr.ia.ac.cn (Z. Lei), szli@nlpr.ia.ac.cn (S.Z. Li).

https://doi.org/10.1016/j.neucom.2019.07.064
0925-2312/



Detection Layer	Associated Anchor
P1	32 x 32, 64 x 64, 128 x 128
P2	256 x 256
P3	512 x 512

Fig. 1. Architecture of FaceBoxes and information about anchor. The architecture consists of the Rapidly Digested Convolution Layers (RDCL), Multiple Scale Convolution Layers (MSCL) and Divide and Conquer Head (DCH). L1/L2/L3 and P1/P2/P3 are intermediate outputs and final outputs of the newly added FPN.

anchors in the image [14,15]. Specifically, these anchor-based algorithms detect objects by classifying and regressing a series of pre-set anchors, which are generated by regularly tiling a collection of boxes with different scales and aspect ratios on the image. These CNN-based face detectors are robust to the large variation of facial appearances and demonstrate state-of-the-art performance, but they are too time-consuming to achieve real-time speed, especially on the CPU devices.

In this paper, inspired by the anchor-based detectors [14,15], we develop an one-stage face detector with CPU real-time speed, named FaceBoxes, which only contains a single fully convolutional neural network and can be trained in an end-to-end way. Specifically, the proposed method has a lightweight yet powerful network structure shown in Fig. 1 that consists of the Rapidly Digested Convolution Layers (RDCL) and the Multiple Scale Convolution Layers (MSCL). The RDCL is designed to enable FaceBoxes to achieve real-time speed on the CPU, and the MSCL aims to enrich the features and discretize anchors over different layers to handle various scales of faces. Besides, we propose a new anchor densification strategy to make different types of anchors have the same density on the input image, which significantly improves the recall rate of small faces. Finally, we present a Divide and Conquer Head (DCH) to boost the prediction ability of the detection layer. Consequently, for VGA-resolution images, our face detector runs at 28 FPS on the CPU and 254 FPS using a GPU. More importantly, the speed of FaceBoxes is invariant to the number of faces on the image. Extensive experiments are conducted on several face detection benchmark datasets including AFW, PASCAL face, FDDB, WIDER FACE and we achieve state-of-the-art performance among CPU real-time detectors.

For clarity, the main contributions of this work can be summarized as five-fold: (1) We design the Rapidly Digested Convolution Layers (RDCL) to enable face detection to achieve real-time speed on the CPU devices. (2) We introduce the Multiple Scale Convolution Layers (MSCL) to handle various scales of face via enriching features and discretizing anchors over layers. (3) We propose a new anchor densification strategy to improve the recall rate of small faces. (4) We present a Divide and Conquer Head (DCH) to boost the prediction ability of the detection layer. (5) We achieve state-of-the-art performance on the AFW, PASCAL face, FDDB and WIDER FACE datasets among CPU real-time methods.

Preliminary results of this work have been published in [16]. The current work has been improved and extended to the conference version in several important aspects. (1) We simplify the network architecture (*i.e.*, using smaller convolution kernel size and fewer channels) to achieve faster speed and utilize the Feature Pyramid Network (FPN) [17] to improve the overall performance. (2) We introduce the Divide and Conquer Head (DCH) for the detection layer with anchor densification strategy to improve its prediction performance. (3) We noticeably improve the speed and accuracy of the detector in our previous work. (4) Some additional experiments are conducted to demonstrate the effectiveness of the proposed method and analyze the effects of different components in performance.

2. Related work

2.1. Generic object detection

Early generic object detectors apply the hand-crafted features and classifiers in the sliding-window paradigm to find objects of interest. After the arrival of deep convolutional neural network, the object detection task is quickly dominated by the CNN-based detectors, which can be roughly divided into two categories, *i.e.*, the two-stage approach and one-stage approach.

The two-stage approach first generates a pool of object proposals by a separate proposal generator (*e.g.*, Selective Search [18], EdgeBoxes [19], RPN [14]), then classifies each proposal to get the class label and estimate the accurate size and location. With the step by step development from R-CNN [20], SPPNet [21], Fast R-CNN [22] to Faster R-CNN [14], the two-stage approach achieves dominated performance on several challenging datasets. Recent improvements of the two-stage approach focus on redesigning architecture diagram [23,24] and training strategy [25,26], using contextual reasoning [27–30] and exploiting multiple layers for prediction [17,31,32].

The one-stage approach eliminates the proposal generation step and straight predicts class labels and regresses object bounding boxes from the regularly pre-tiled anchor boxes using deep CNNs. Comparing to the two-stage approach, the main advantage of the one-stage approach is its high computational efficiency, but its detection accuracy is usually inferior to that of the two-stage approach. OverFeat [33] is one of the first one-stage detectors and after that, several more efficient single-stage object detection methods have been proposed, most noticeably, YOLO [34] and SSD [15]. Since the one-stage approach has high running time efficiency, it has attracted more and more attention including the

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064

topics of training detector from scratch [35], designing different architecture [36,37] and addressing class imbalance issue [38–40].

Among them, PVANet [41] has some similarities with our algorithm in architecture design. It designs a thin and light network via elaborate adoption and combination of some existing technical innovations, such as C.ReLU and Inception, which make it achieve the state-of-the-art accuracy and minimize the computational cost. Similar to it, we also adapt and combine some exiting technical innovations. However, our approach is very different from PVANet. First, PVANet aims to achieve real-time on the powerful GPU, and it only runs at 1.3 FPS on the CPU device. In contrast, our FaceBoxes is designed to reach CPU real-time speed. Second, due to different goals, we only use several convolution layers with few channels and suitable kernel sizes, resulting in model size of different orders of magnitude (*i.e.*, 2.5MB for FaceBoxes *vs.* 368MB for PVANet). Third, we introduce a novel anchor densification strategy to improve the recall rate. Final, we present a Divide and Conquer Head (DCH) to boost the prediction ability of the detection layer using above strategy.

2.2. Face detection

Even as one of the long-standing problems in computer vision with an extensive literature, face detection still attracts much attention these days for its wide practical applications [42,43]. Previous face detection systems are mostly based on hand-crafted features. The pioneering work of Viola and Jones [1] is the first major breakthrough, which uses Adaboost with Haar features to train a series of cascaded classifiers to detect face and achieves satisfactory accuracy with high efficiency. Since then, several methods focus on designing new local features [44–48], new boosting algorithms [49–52] and new cascade structures [53–57]. DPM [4] is another popular method in face detection [58–61] by using mixtures of multi-scale deformable part models to represent multi-view faces. However, the aforementioned methods rely on hand-crafted features and classifiers, making them unreliable in complex scenarios.

Recently, face detection has been dominated by CNN-based methods. CascadeCNN [62] improves detection accuracy by training a serious of interleaved CNN models and the follwing work [5] realizes end-to-end optimization. Faceness [9] formulates face detection as scoring facial parts responses to detect faces under severe occlusion. MTCNN [63] proposes a joint face detection and alignment method using unified cascaded CNNs for multi-task learning. UnitBox [64] introduces an IoU loss for bounding box prediction. SAFD [65] develops a scale proposal stage which automatically normalizes face sizes prior to detection. Chu et al. [66] propose a deep neural network method to do manga face detection. S^2AP [67] pays attention to specific scales in image pyramid and valid locations in each scales layer. PCN [68] proposes a cascade-style structure to rotate faces in a coarse-to-fine manner. Bai et al. [69] design a novel network to directly generate a clear super-resolution face from a blurry small one.

Other face detection algorithms are inspired by generic object detection methods. Some works [70–74] use the improved Faster R-CNN [14] and R-FCN [23] to detect faces. CMS-RCNN [12] integrates contextual reasoning based on human body anatomy into the Faster R-CNN algorithm to help reduce the overall detection errors. Conv3D [75] combines a ConvNet model with a 3D mean face model into an end-to-end multi-task discriminative learning framework, which produces competitive results compared to the state-of-the-art methods. To handle the class imbalance issue in detector training, [76] uses the hard negative mining technique to improve the performance. STN [6] designs an end-to-end learning based supervised transformer network to deal with the large pose variation in face detection, where the supervised transformer layer enables learning the optimal canonical pose to best differentiate face/non-face images. Recent works [40,77–84] focus on tiny faces in crowd images, which is another remaining open challenge that needs to be solved.

3. FaceBoxes

This section presents four contributions that make FaceBoxes accurate and efficient on CPU devices: The Rapidly Digested Convolution Layers (RDCL), the Multiple Scale Convolution Layers (MSCL), the anchor densification strategy and the Divide and Conquer Head (DCH). Finally, we introduce the associated training methodology.

3.1. Rapidly digested convolution layers

Most of CNN-based face detection methods are usually limited by the heavy cost of time, especially on the CPU devices. More specifically, the convolution operation for CPU is extremely time-consuming when the size of input, kernel and output are large. Our RDCL is designed to fast shrink the input spatial size by suitable kernel size with reducing the number of output channels, enabling the FaceBoxes to reach real-time speed on the CPU devices, as follows.

Shrinking the spatial size of input: To rapidly shrink the spatial size of input, our RDCL sets a series of large stride sizes for its convolution and pooling layers. As illustrated in Fig. 1, the stride size of Conv1, Pool1, Conv2 and Pool2 are 4, 2, 2 and 2, respectively. The total stride size of RDCL is 32, which means the input spatial size is reduced by 32 times quickly.

Choosing suitable kernel size: The kernel size of the first few layers in one network should not be too large to make the network forward computation efficient. On the other hand, the kernel size is also supposed to be large enough to alleviate the information loss brought by the spatial size reducing. As shown in Fig. 1, to keep the network efficient as well as effective, we choose 5×5, 3×3 and 3×3 kernel size for Conv1, Conv2 and all Pool layers, respectively.

Reducing the number of output channels: We utilize the C.ReLU activation function (illustrated in Fig. 2(a)) to reduce the number of output channels of convolution layers. C.ReLU [85] is motivated from the observation in CNN that the filters in the lower layers form pairs (*i.e.*, filters with opposite phase). From this observation, C.ReLU can double the number of output channels by simply concatenating negated outputs before applying ReLU, which reduces the output channels of convolution layers and hence the computational efficiency is significantly improved.

3.2. Multiple scale convolution layers

The proposed detector belongs to the one-stage approach that straight predicts class labels and regresses face bounding boxes from the regularly pre-tiled anchor boxes using deep CNNs. When we design the MSCL, three key points are taken into consideration. Firstly, we use the Inception [86] module to enrich the receptive field, because each detection layer is responsible for detecting faces within a certain size range, so richer receptive field is helpful. Secondly, we utilize the Feature Pyramid Network (FPN) [17] to fuse different levels of features, in which both the abstract and detailed features are integrated to improve the detection performance. Thirdly, we follow SSD [15] to discretize different sizes of anchors over different layers to handle faces of various scales, because it is difficult for one-stage methods to use only one layer associated with all anchors (*i.e.*, RPN [14]) to detect different scales of faces. The MSCL is shown in Fig. 1 and here is the details as follows.

Inception: Enriching the receptive field. To learn visual patterns for different scales of faces, output features of the detection

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064

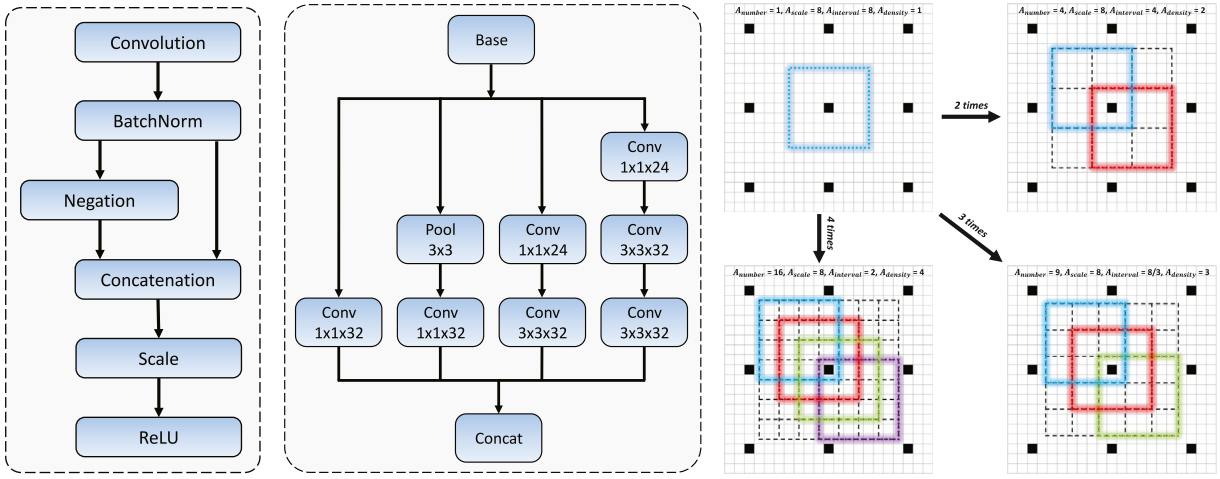


Fig. 2. (a) C.ReLU module, where Negation simply multiplies −1 to the output. (b) Inception module. (c) Examples of anchor densification. For clarity, we only densify anchors at one receptive field centre (*i.e.*, the central black cell), and only color the diagonal anchors.

layers should correspond to various sizes of receptive fields, which can be easily fulfilled via Inception modules. As shown in Fig. 1, the first three layers in the MSCL module are based on the Inception module, which is a cost-effective module to capture different scales of faces. Fig. 2(b) illustrates our Inception implementation that consists of multiple convolution branches with different kernels to enrich the receptive fields.

FPN: Fusing different levels of features. Inspired by FPN, we add the high-level feature maps to the low-level layers to improve the detection accuracy. As shown in Fig. 1, we expand the spatial resolution of a coarser-resolution feature map by a factor of 2 via the bilinear upsampling. Then, after the corresponding low-level map undergoes a 1×1 convolution layer to reduce channel dimensions, we use element-wise addition to merge it with the upsampled map to get the intermediate outputs (*e.g.*, {L1, L2, L3}). This process is iterated until the finest resolution map is generated. To start the iteration, we simply attach a 1×1 convolution layer on Conv4_2 to produce the coarsest resolution map. Finally, we append a 3×3 convolution on each merged map to generate the final feature map, which is to reduce the aliasing effect of upsampling. This final set of feature maps is called {P1, P2, P3}, corresponding to {inception3, Conv3_2, Conv4_2} that are respectively of the same spatial sizes.

SSD: Discretizing anchors over different layers. With the Inception module and FPN operation, our designed MSCL has several layers with different spatial sizes that form the multi-scale feature maps. These layers have rich features and receptive fields to detect various scales of faces. As shown in Fig. 1, we follow [15] to associate our default anchors with multi-scale feature maps. These layers discretize anchors over multiple layers with different resolutions to naturally handle faces of various sizes.

3.3. Anchor densification strategy

As illustrated in Fig. 1, we pre-set only 1:1 aspect ratio for the default anchors (*i.e.*, square anchor), because the face box is approximately square. The scales of anchor for the P1 layer are 32, 64 and 128 pixels, for the P2 layer and P3 layer are 256 and 512 pixels, respectively.

The tiling interval of anchor on the image is equal to the stride size of the corresponding detection layer. For example, the stride size of P2 is 64 pixels and its anchor is 256×256, indicating that there is a 256×256 anchor for every 64 pixels on the input image. We define the tiling density of anchor $A_{density}$ as follows:

$$A_{density} = A_{scale} \ / \ A_{interval} \tag{1}$$

Here, A_{scale} is the scale of anchor and $A_{interval}$ is the tiling interval of anchor. The tiling intervals for our 5 default anchors are 32, 32, 32, 64 and 128, respectively. According to Eq. (1), the corresponding densities are **1**, **2**, 4, 4 and 4, and there is a tiling density imbalance problem between anchors of different scales. Comparing with large anchors (*i.e.*, 128×128, 256×256 and 512×512), small anchors (*i.e.*, 32×32 and 64×64) are too sparse, which results in low recall rate of small faces.

To eliminate this imbalance, we propose a new anchor densification strategy. Specifically, to densify one type of anchors n times, we uniformly tile $A_{number} = n^2$ anchors around the center of one receptive field instead of only tiling one at the center of this receptive field to predict. Some examples are shown in Fig. 2(c). In this work, to improve the tiling density of the small anchor, our strategy is used to densify the 32×32 anchor 4 times and the 64×64 anchor 2 times, which guarantees that different scales of anchor have the same density (*i.e.*, 4) on the image, so that different scales of faces can match almost the same number of anchors.

3.4. Divide and conquer head

As shown in Fig. 3, after densifying the 32×32 scale anchor 4 times and the 64×64 scale anchor 2 times, each 1×1 cell on the P1 detection layer will be responsible for a total of 21 anchors, *i.e.*, 16 anchors for the 32×32 scale, 4 anchors for the 64×64 scale and 1 anchor for the 128×128 scale. It can be observed that there is an imbalance problem among different scales of anchors for each 1×1 cell on this detection layer. The prediction difficulty of one 128×128 anchor, four 64×64 anchors and sixteen 32×32 anchors is gradually increasing. In summary, an imbalance problem among different scales of anchors will be existing in each 1×1 cell after applying the anchor densification strategy on corresponding detection layers.

To solve this issue, we present a Divide and Conquer Head (DCH) based on the "divide and conquer" strategy. As described above, predicting 3 different scales of anchors (*i.e.*, 32×32, 64×64 and 128×128) on the P1 detection layer has varying degrees of difficulty. More anchors of one scale are associated with a 1×1 cell, it is more difficult to classify them. To this end, we divide the prediction task on the P1 layer into three sub-tasks with dif-

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064

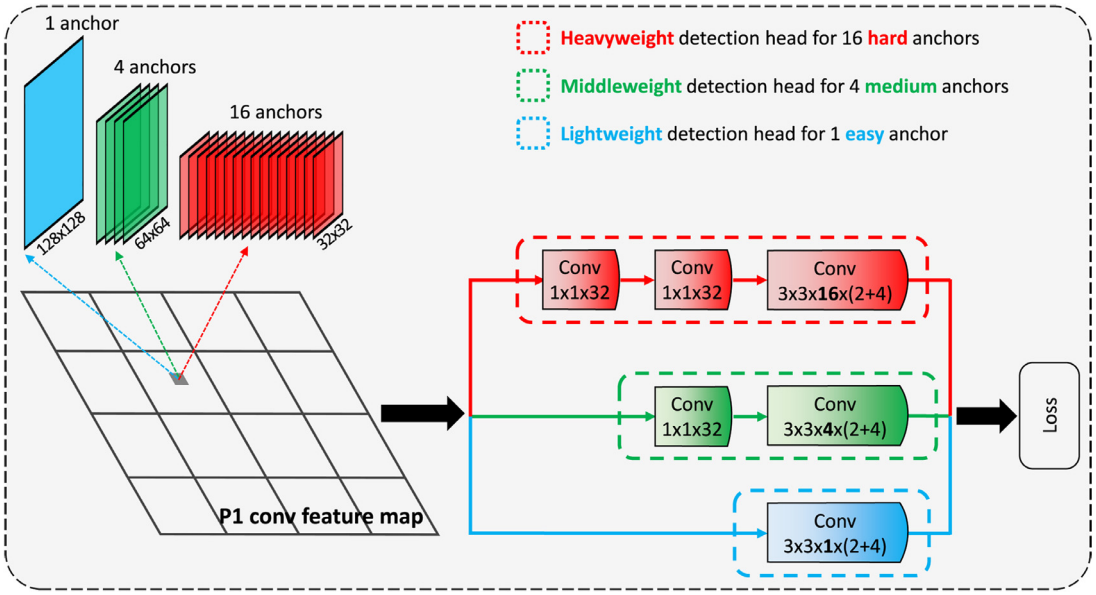
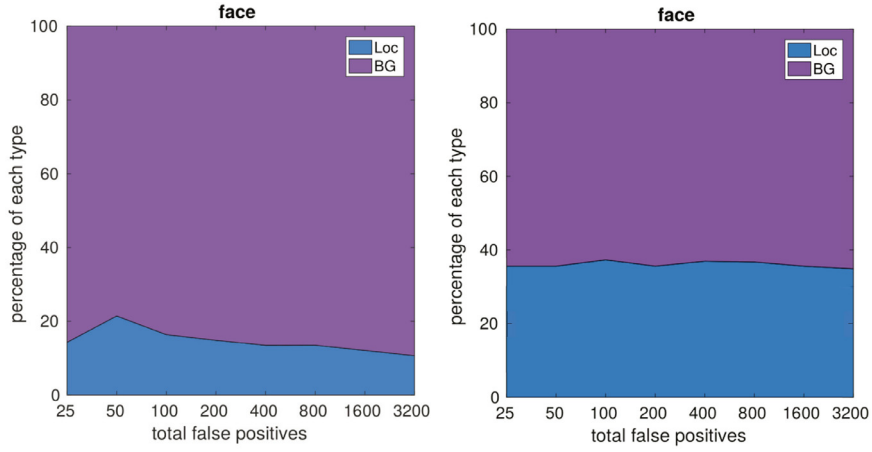


Fig. 3. The Divide and Conquer Head (DCH).



(a) FDDB (b) WIDER FACE

Fig. 4. Distribution of two error modes of false positives.

ferent difficulty levels, then use three different sub-heads to separately conquer them. As shown in Fig. 3, the DCH consists of three detection heads with different number of convolution layers to predict the corresponding scale of the anchor, *i.e.*, the heavyweight head with three layers for 16 hard 32×32 anchors, the middleweight head with two layers for 4 medium 64×64 anchors and the lightweight with one layer for 1 easy 128×128 anchor. Besides, the proposed DCH has fewer parameters than ordinary prediction head[1], resulting in less model size and faster speed.

3.5. Training details

Training dataset and data augmentation. Our model is trained on 12,880 images of the WIDER FACE training subset. To construct a robust model and prevent overfitting, each training image is sequentially processed by the following data augmentation strategies: (1) Color distortion: Applying some photo-metric distortions, similar to [87], to change the brightness, contrast, hue, or saturation of the original training images. (2) Random cropping: We randomly crop five square patches from the original image and select one for training. One patch is with the size of the image's shorter side and the others are with the size determined by multiplying a random number in the interval [0.3, 1.0] by the image's shorter side. (3) Scale transformation: After random cropping, we randomly flip the selected patch with probability of 0.5 and resize it to 1024×1024 to get the final training sample. (4) Face filtering: We keep the overlapped part of the face box if its center is in the above processed image, then filter out these face boxes whose height or width is less than 20 pixels.

Matching strategy. During the training phase, we need to determine which anchors correspond to a face bounding box. We first match the anchors to the faces with the largest Jaccard overlap [88], and then match the anchors to any face with Jaccard overlap larger than a preset threshold (*i.e.*, 0.35).

Hard negative mining. After the anchor matching step, most of the anchors are negatives, leading to extremely class imbalance of the training samples. Using all negative anchors or randomly selecting some of them will make the training process slow and

[1] Each anchor corresponds to $2+4=6$ output channels, so the parameter amount of the ordinary prediction head is $64 \times 3 \times 3 \times (21 \times 6) = 72,576$, while the parameter amount of the Divide and Conquer Head is $\left(64 \times 1 \times 1 \times 32 + 32 \times 1 \times 1 \times 32 + 32 \times 3 \times 3 \times (16 \times 6)\right) + \left(64 \times 1 \times 1 \times 32 + 32 \times 3 \times 3 \times (4 \times 6)\right) + \left(64 \times 3 \times 3 \times (1 \times 6)\right) = 43,136$.

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064

(a) FDDB

(b) WIDER FACE

Fig. 5. Top-N scoring false positives on the FDDB and WIDER FACE dataset. Error type is labeled at the left bottom of each image. "face(bg)" represents background confusion and "face(loc)" represents inaccurate localization. "v" represents overlap with ground truth bounding boxes, "1-r" represents the percentage of detections whose confidence is below the current one's.

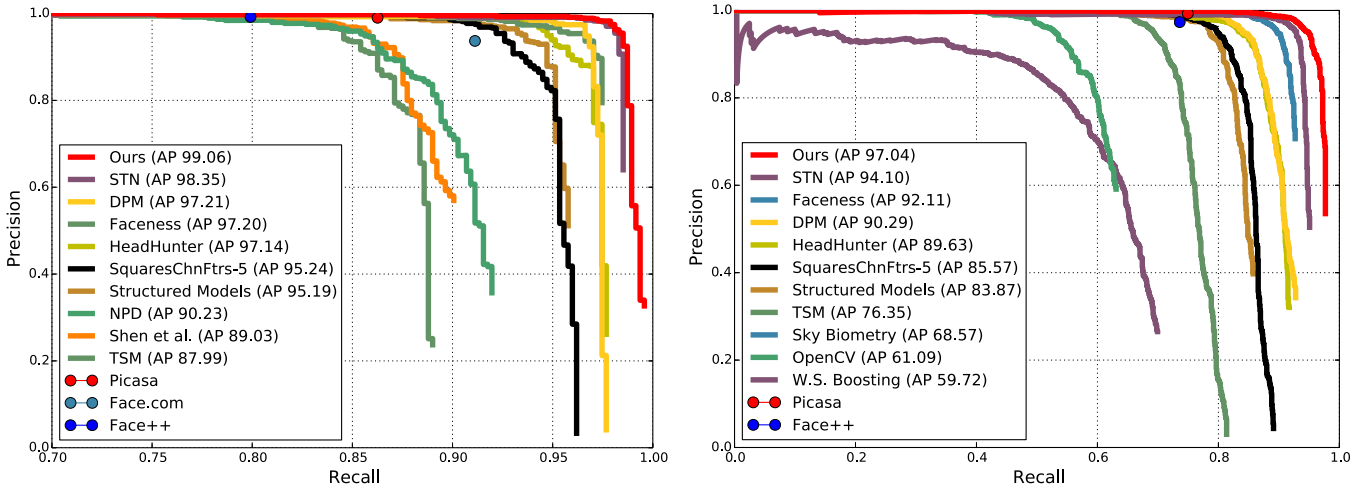
unstable. To mitigate this issue, we select some negative anchors with top loss values and make the ratio between the negative and positive anchors below 7:1.

Loss function. We use the loss function defined in RPN [14] to jointly optimize model parameters as,

$$L(p_i, t_i) = \frac{\lambda}{N_{\text{cls}}} \sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N_{\text{reg}}} \sum_i p_i^* \cdot L_{\text{reg}}(t_i, t_i^*)$$

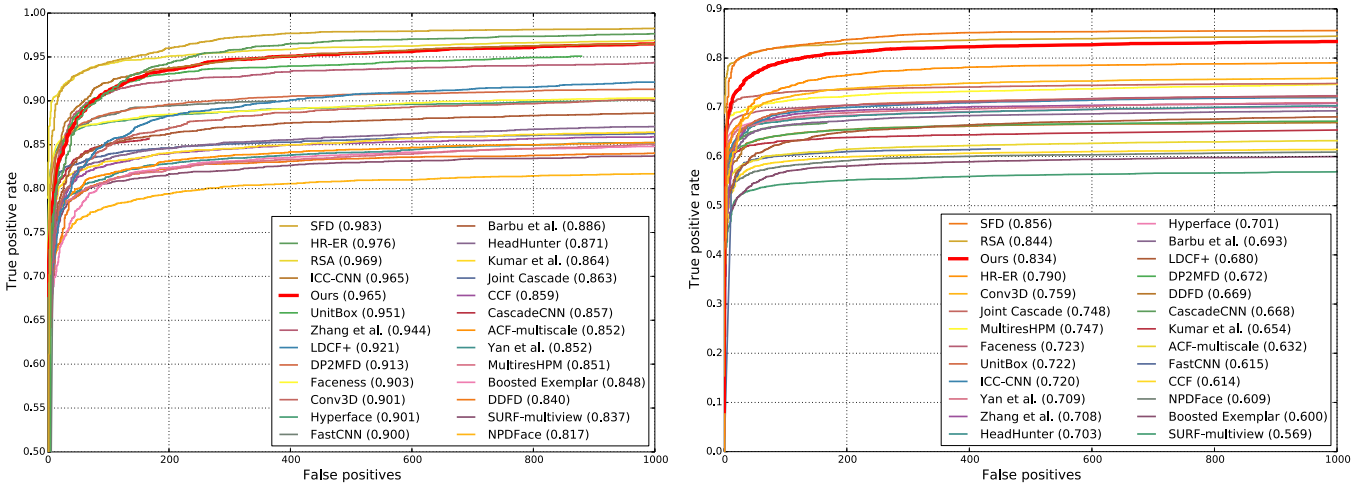
where i is the index of the anchor, p_i is the prediction score of the i-th anchor to be a face, t_i is the vector representing the four coordinates of the predicted face bounding box, t_i^* is the ground truth box matched with the i-th anchor, N_{cls} is the number of positive and negative anchors used to normalize the classification loss term, N_{reg} is the number of positive anchors used to normalize the regression loss term, λ is the hyper-parameter to balance the two

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064



(a) AFW (b) PASCAL face

Fig. 6. Precision-recall curves.



(a) Discontinuous ROC curves (b) Continuous ROC curves

Fig. 7. Evaluation on the FDDB dataset.

task losses.[2] The classification loss $L_{\text{cls}}(p_i, p_i^*)$ is a two-class (*i.e.*, face or non-face) softmax loss, and the regression loss $L_{\text{reg}}(t_i, t_i^*)$ is the smooth L1 loss defined in [22].

Optimization. All the parameters are randomly initialized by the "xavier" method. We fine-tune the final model using the adaptive moment estimation (Adam) algorithm with 0.9 momentum, 0.0005 weight decay and batch size 32. The maximum number of iterations is 120k and we use 10^{-3} learning rate for the first 80k iterations, then continue training for 20k iterations with 10^{-4} and 10^{-5}, respectively. Our method is implemented in the Caffe [89] library.

4. Experiments

We first examine the runtime efficiency of FaceBoxes, then analyze our model in an ablative way and the false positive errors. Finally, we report the performance of FaceBoxes compared with state-of-the-art methods on common face detection benchmarks.

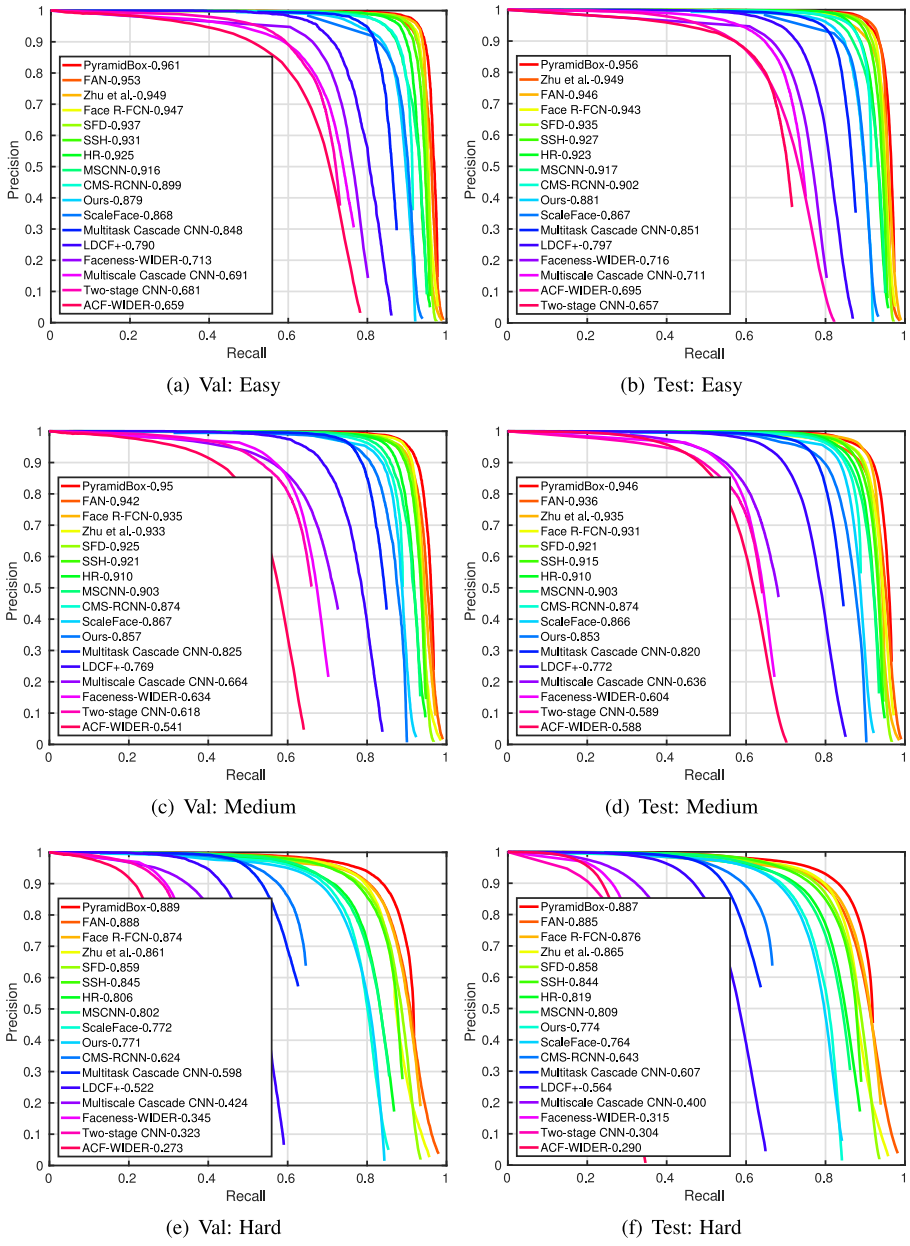
[2] Since the ratio between positive and negative anchors is set to 1: 7, we use $\lambda = 8$ to balance the classification and regression losses in training.

4.1. Runtime efficiency

CNN-based methods have always been accused of its runtime efficiency. Although the existing CNN face detectors can be accelerated via high-end GPUs, they are not fast enough in most practical applications, especially on CPU-based applications. As described below, our FaceBoxes is efficient enough to meet practical requirements.

At the inference phase, our method outputs a large number of detection boxes. For example, it produces 6,400 bounding boxes for a VGA-resolution image of 640×480 pixels. We first filter out most of the boxes with a confidence threshold 0.05 and only retain 200 boxes with top confidence score. After that, we apply non-maximum suppression (NMS) with Jaccard overlap of 0.3 to generate the final 100 high confident detection results per image. We measure the speed using Titan X (Pascal) and cuDNN v6.0 with Intel Xeon E5 – 2660v3@2.60 GHz. As listed in Table 1, comparing with recent CPU real-time detectors, our FaceBoxes can run at 26 FPS on the CPU with state-of-the-art accuracy. And it can run at 178 FPS using a single GPU and has only 2.5 MB in size. Besides, all batch normalization (BN) layers can be merged with convolution layer at the inference stage, which can further accelerate the speed to 28 and 254 FPS on the CPU and GPU, respectively.

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064



(a) Val: Easy (b) Test: Easy

(c) Val: Medium (d) Test: Medium

(e) Val: Hard (f) Test: Hard

Fig. 8. Precision-recall curves on the WIDER FACE `validation` and `testing` sets.

Table 1
CPU speed *v.s* AP for different methods. The **FPS** is for VGA-resolution images on CPU and the **AP** means the true positive rate at 1,000 false positives on the FDDB.

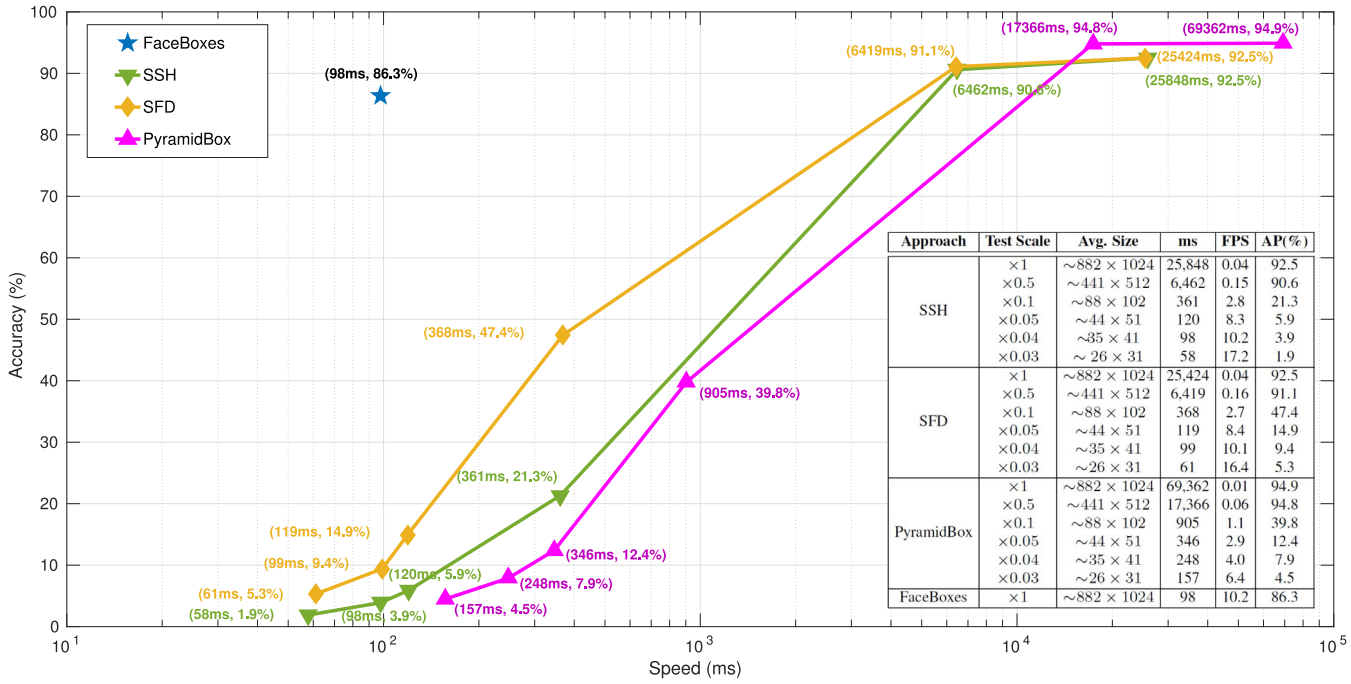
Approach	Resolution	CPU	GHz	AP(%)	FPS
ACF [91]	640 × 480	Intel I7-3770	3.40	85.2	20
CasCNN [62]	640 × 480	Intel E5-2620	2.00	85.7	14
FaceCraft [5]	640 × 480	N/A	N/A	90.8	10
STN [90]	640 × 480	Intel I7-4770K	3.50	91.5	10
MTCNN [63]	640 × 480	N/A	2.60	94.4	16
ICC-CNN [92]	640 × 480	N/A	N/A	96.5	12
Ours	640 × 480	Intel E5-2660v3	2.60	96.5	26
Ours*	640 × 480	Intel E5-2660v3	2.60	**96.5**	**28**

* indicates that all BN layers are merged with convolution layers at the inference stage. Notably, for STN [90], its AP is the true positive rate at 179 false positives and with ROI convolution, its FPS can be accelerated to 30 with 0.6% recall rate drop.

4.2. Model analysis

We carry out extensive ablation experiments on the FDDB dataset to analyze our model. For all the experiments, we use the same settings except for specified changes to the components.

Ablative setting. To better understand FaceBoxes, we ablate each component one after another to examine how each proposed component affects the final performance. Firstly, we use the ordinary prediction head instead of the proposed Divide and Conquer Head (DCH). Secondly, we ablate the anchor densification strategy. Thirdly, we replace MSCL with three convolution layers, which all have 3×3 kernel size and whose output number is the same as the first three Inception modules of MSCL. Meanwhile, we associate all anchors with the last convolution layer. 4) Finally, we take the place of C.ReLU with ReLU in RDCL. The ablative results are listed in Table 2 and some promising conclusions can be summed up as follows:



Approach	Test Scale	Avg. Size	ms	FPS	AP(%)
SSH	×1	∼882 × 1024	25,848	0.04	92.5
	×0.5	∼441 × 512	6,462	0.15	90.6
	×0.1	∼88 × 102	361	2.8	21.3
	×0.05	∼44 × 51	120	8.3	5.9
	×0.04	∼35 × 41	98	10.2	3.9
	×0.03	∼ 26 × 31	58	17.2	1.9
SFD	×1	∼882 × 1024	25,424	0.04	92.5
	×0.5	∼441 × 512	6,419	0.16	91.1
	×0.1	∼88 × 102	368	2.7	47.4
	×0.05	∼44 × 51	119	8.4	14.9
	×0.04	∼35 × 41	99	10.1	9.4
	×0.03	∼26 × 31	61	16.4	5.3
PyramidBox	×1	∼882 × 1024	69,362	0.01	94.9
	×0.5	∼441 × 512	17,366	0.06	94.8
	×0.1	∼88 × 102	905	1.1	39.8
	×0.05	∼44 × 51	346	2.9	12.4
	×0.04	∼35 × 41	248	4.0	7.9
	×0.03	∼26 × 31	157	6.4	4.5
FaceBoxes	×1	∼882 × 1024	98	10.2	86.3

Fig. 9. Trade-off on CPU for different methods. The $\times N$ in **Test Scale** means resizing the images N times for testing ($\times 1$ indicates using original images). The **Avg. Size** is the average size of images with corresponding test scales. The **ms** and **FPS** are measured on the Intel E5-2660v3@2.60 GHz CPU with the Caffe library. The **AP** means average precision on WIDER FACE validation Easy subset.

Table 2
Ablative results of the FaceBoxes on the FDDB dataset. Accuracy (AP) means the true positive rate at 1,000 false positives. Speed (ms) is for the VGA-resolution images on the CPU.

Contribution	FaceBoxes				
Rapidly Digested Convolution Layers					×
Multiple Scale Convolution Layers				×	×
Anchor Densification Strategy			×	×	×
Divide and Conquer Head		×	×	×	×
Accuracy (AP)	96.5	96.1	95.0	93.7	93.8
Speed (ms)	38.62	38.73	36.79	34.23	45.43

DCH is promising. From the comparison between the first and second columns in Table 2, it can be known that the DCH increases the AP by 0.4% as well as 0.11 ms faster, owning to using the divide and conquer mechanism to deal with the prediction issue raised by the anchor densification strategy.

Anchor densification strategy is crucial. Our anchor densification strategy is used to increase the density of small anchors (*i.e.*, 32×32 and 64×64) in order to improve the recall rate of small faces. From the results listed in Table 2, we can see that the AP on the FDDB is reduced from 96.1% to 95.0% after ablating the anchor densification strategy. The sharp decline (*i.e.*, 1.1%) demonstrates the effectiveness of the proposed anchor densification strategy.

MSCL is better. The comparison between the third and fourth columns in Table 2 indicates that MSCL effectively increases the AP by 1.3%, owning to the diverse features resulting from FPN and Inception as well as the multi-scale anchor tiling mechanism in SSD.

RDCL is efficient and accuracy-preserving. The design of RDCL enables our FaceBoxes to achieve CPU real-time speed. As reported in Table 2, RDCL leads to a negligible decline on accuracy but a significant improvement on speed. Specifically, the FDDB AP decreases by 0.1% in return for the about 11.2 ms speed improvement.

4.3. Error analysis

In this part, we utilize the detection analysis tool[3] to analyze the error of FaceBoxes on the FDDB and WIDER FACE datasets. There are two error modes of false positives in face detectors, *i.e.*, LOC and BG. LOC indicates the localization errors that occurs when a face is detected with a misaligned bounding box, and BG indicates that a background region is mistakenly detected as a face. Fig. 4(a) shows the distribution of two types of false positives on FDDB and BG seems the dominating error mode among top-scoring detection. However, as shown in Fig. 5(a), 13 out of top 14 scoring false positives are in fact due to missed annotation(*i.e.*, the predicted bounding box encloses a face while "ov" almost equals to zero in Fig. 5(a)). Since the FDDB dataset does not label the profile faces and these faces whose width or height are fewer than 20 pixels. The analysis of false positives on WIDER FACE dataset is shown in Fig. 4(b), BG is still the dominating error mode. Comparing with FDDB, the percentage of BG error mode on WIDER FACE has dropped a lot, since WIDER FACE aims to label all the faces, but it also has a little unlabelled faces as shown in Fig. 5(b).

4.4. Evaluation on benchmark

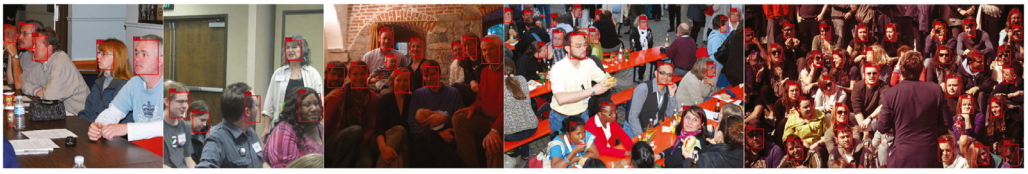
We evaluate the FaceBoxes on the common face detection benchmark datasets, including the Annotated Faces in the Wild (AFW) [60], PASCAL face [59], Face Detection Data Set and Benchmark (FDDB) [93] and WIDER FACE [94].

AFW dataset.[4] It has 205 images with 473 faces collected from Flickr images, which contain cluttered backgrounds with large

[3] http://web.engr.illinois.edu/~dhoiem/projects/detectionAnalysis.

[4] http://www.ics.uci.edu/~xzhu/face/.

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064



(a) AFW

(b) PASCAL face

(c) FDDB

(d) WIDER FACE

Fig. 10. Qualitative results on face detection benchmark datasets.

variations in both face viewpoint and appearance (*e.g.*, ages, sunglasses, make-ups, skin colors, expressions, etc.). We evaluate the FaceBoxes against 7 well-known works [6,9,59–61,95,96] and 3 commercial face detectors (*i.e.*, Face.com, Face++ and Picasa). As illustrated in Fig. 6(a), our method outperforms all others by a large margin. Fig. 10(a) shows some qualitative results on the AFW dataset.

PASCAL face dataset.[5] It is collected from the test set of PASCAL person layout dataset, consisting of 1,335 faces with large face appearance and pose variations from 851 images. Fig. 6(b) shows the precision-recall curves on this dataset. Our method significantly

[5] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/.

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064

outperforms all other methods [6,9,59–61,97] and 3 commercial face detectors (*i.e.*, SkyBiometry, Face++ and Picasa). Fig. 10(b) shows some qualitative results on the PASCAL face dataset.

FDDB dataset.[6] It has 5,171 faces annotated in 2,845 images taken from news articles on Yahoo websites with a wide range of difficulties, such as occlusions, large poses, and low image resolutions. FDDB adopts the bounding ellipse to represent the faces, while our FaceBoxes outputs rectangle bounding box. This inconsistency has a great impact on the continuous score for evaluation. For fair comparison, we train an elliptical regressor to transform the predicted bounding boxes to ellipses. We evaluate our face detector on the FDDB dataset against the other methods [7–9,11,13,40,58,61–64,75,77,90–92,95,98–104]. The discrete and continuous evaluation results are shown in Fig. 7(a) and Fig. 7(b), respectively. Among all the CPU real-time detectors, our FaceBoxes achieves the state-of-the-art performance and outperforms all others by a large margin on both the discrete and continuous metrics. Besides, our detector performs on-pair with some heavyweight detectors, such as RSA and HR-ER. These results indicate that FaceBoxes is robust to varying scales, large appearance changes, heavy occlusions, and severe blur degradations that are prevalent in detecting face in unconstrained real-life scenarios. Fig. 10(c) shows some qualitative results on the FDDB dataset.

WIDER FACE dataset.[7] It contains 393,703 faces annotated in 32,203 images with variations in pose, scale, facial expression, occlusion, and lighting condition. The dataset is divided into the *training* (40%), *validation* (10%) and *testing* (50%) sets. Besides, based on the detection rate of EdgeBox [19], it defines three levels of difficulty: *Easy, Medium, Hard*. Following the evaluation protocol in WIDER FACE, our FaceBoxes is trained only on the training set and tested on both the *validation* and the *testing* sets against state-of-the-art face detection methods [9,11,12,31,40,63,77,78,81,91,94]. As shown in Fig. 8, our FaceBoxes, with CPU real-time speed, achieves promising AP performance in all subsets of both *validation* and *testing* sets, *i.e.,* 88.5% (*Easy*), 86.2% (*Medium*) and 77.3% (*Hard*) for *validation* set, and 88.7% (*Easy*), 85.8% (*Medium*) and 77.6% (*Hard*) for *testing* set. Among CPU real-time detectors (*i.e.*, ACF, MTCNN), our FaceBoxes outperforms them by a large margin across the three subsets. Besides, our detector performs better than some detectors based on ResNet, such as CMS-RCNN and ScaleFace. Actually, there are some state-of-the-art heavyweight face detectors have higher AP than the proposed FaceBoxes including PyramidBox, FAN, Zhu et al., Face R-FCN, SFD, SSH, HR and MSCNN. However, we would like to emphasize that **they are too heavyweight to be used on the CPU devices** and the proposed FaceBoxes achieves better trade-off on CPU. To verify this statement, we measure the CPU speed under different single-scale test size for three open source algorithms (*i.e.*, SSH [78], SFD [40], PyramidBox [82]). As shown in Fig. 9, using the original images of WIDER FACE validation set for testing on our CPU, SSH achieves 92.5% AP with 25,848 ms/image, SFD obtains 92.5% AP with 25,424 ms/image and PyramidBox gets 94.9% AP with 69,362 ms/image. In contrast, the proposed FaceBoxes achieves 86.5% AP with 98 ms/image, which is 6.2 ∼ 8.6 points lower than above methods but 259 ∼ 708 times faster. Reducing test scale can speed up, but it also reduces accuracy. If the aforementioned methods want to reach the speed of FaceBoxes, then their APs are less than 10%. These results demonstrate that FaceBoxes achieves excellent trade-off between effectiveness and efficiency on CPU. Fig. 10(d) shows some qualitative results on the WIDER FACE dataset.

[6] http://vis-www.cs.umass.edu/fddb/index.html.
[7] http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/index.html .

5. Conclusion

Since highly accurate models for the face detection task tend to be computationally prohibitive, it is challenging for the CPU devices to achieve real-time speed as well as maintain high performance. In this work, we present a novel face detector with a good trade-off between speed and accuracy. The proposed method has a lightweight yet powerful network structure, which consists of RDCL and MSCL. The former enables FaceBoxes to achieve real-time speed, while the latter aims to enrich the features and discretize anchors over different layers to handle faces of various scales. Besides, a new anchor densification strategy is proposed to improve the recall rate of small faces. Finally, we present a Divide and Conquer Head (DCH) to boost the prediction ability of the detection layer using above strategy. The experiments demonstrate that our contributions lead FaceBoxes to the state-of-the-art performance among the lightweight detectors on the common face detection benchmarks. The proposed detector is very fast, achieving 28 FPS for VGA-resolution images on the CPU and can be accelerated to 254 FPS on the GPU.

Declaration of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the Chinese National Natural Science Foundation Projects #61876178, #61806196, #61872367, #61572501.

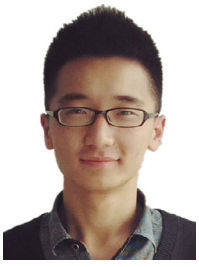
References

[1] P. Viola, M.J. Jones, Robust real-time face detection, IJCV 57 (2) (2004) 137–154, doi:10.1023/B:VISI.0000013087.49260.fb.
[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR, Vol. 1, 2005, pp. 886–893, doi:10.1109/CVPR.2005.177.
[3] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139, doi:10.1006/jcss.1997.1504.
[4] P.F. Felzenszwalb, R.B. Girshick, D.M. Allester, D. Ramanan, Object detection with discriminatively trained part-based models, TPAMI 32 (9) (2010) 1627–1645, doi:10.1109/TPAMI.2009.167.
[5] H. Qin, J. Yan, X. Li, X. Hu, Joint training of cascaded cnn for face detection, in: CVPR, 2016, pp. 3456–3465, doi:10.1109/CVPR.2016.376.
[6] D. Chen, G. Hua, F. Wen, J. Sun, Supervised transformer network for efficient face detection, in: ECCV, 2016, pp. 122–138, doi:10.1007/978-3-319-46454-1_8.
[7] R. Ranjan, V.M. Patel, R. Chellappa, Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition, in: TPAMI, doi:10.1109/TPAMI.2017.2781233.
[8] B. Yang, J. Yan, Z. Lei, S.Z. Li, Convolutional channel features, in: ICCV, 2015, pp. 82–90, doi:10.1109/ICCV.2015.18.
[9] S. Yang, P. Luo, C.C. Loy, X. Tang, From facial parts responses to face detection: a deep learning approach, in: ICCV, 2015, pp. 3676–3684, doi:10.1109/ICCV.2015.419.
[10] B. Yang, J. Yan, Z. Lei, S.Z. Li, Fine-grained evaluation on face detection in the wild, in: FG, Vol. 1, 2015, pp. 1–7, doi:10.1109/FG.2015.7163158.
[11] E. Ohn-Bar, M.M. Trivedi, To boost or not to boost? on the limits of boosted trees for object detection, in: ICPR, 2016, pp. 3350–3355, doi:10.1109/ICPR.2016.7900151.
[12] C. Zhu, Y. Zheng, K. Luu, M. Savvides, Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection, Deep Learn. Biometrics (2017) 57–79, doi:10.1007/978-3-319-61657-5_3.
[13] Y. Liu, H. Li, J. Yan, F. Wei, X. Wang, X. Tang, Recurrent scale approximation for object detection in cnn, in: ICCV, 2017, pp. 571–579, doi:10.1109/ICCV.2017.69.
[14] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: NIPS, 2015, pp. 91–99.
[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: ECCV, 2016, pp. 21–37, doi:10.1007/978-3-319-46448-0_2.
[16] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S.Z. Li, Faceboxes: a cpu real-time face detector with high accuracy, in: IJCB, 2017, pp. 1–9, doi:10.1109/BTAS.2017.8272675.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: CVPR, 2017, pp. 936–944, doi:10.1109/CVPR.2017.106.
[18] J.R. Uijlings, K.E.V. De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, in: IJCV, doi:10.1007/s11263-013-0620-5.
[19] C.L. Zitnick, P. Dollár, Edge boxes: locating object proposals from edges, in: ECCV, 2014, pp. 391–405, doi:10.1007/978-3-319-10602-1_26.
[20] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: CVPR, 2014, pp. 580–587, doi:10.1109/CVPR.2014.81.
[21] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: ECCV, 2014, pp. 346–361, doi:10.1007/978-3-319-10578-9_23.
[22] R. Girshick, Fast r-cnn, in: ICCV, 2015, pp. 1440–1448, doi:10.1109/ICCV.2015.169.
[23] J. Dai, Y. Li, K. He, J. Sun, R-fcn: object detection via region-based fully convolutional networks, in: NIPS, 2016, pp. 379–387.
[24] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, Couplenet: coupling global structure with local parts for object detection, in: ICCV, 2017, pp. 4146–4154, doi:10.1109/ICCV.2017.444.
[25] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: CVPR, 2016, pp. 761–769, doi:10.1109/CVPR.2016.89.
[26] X. Wang, A. Shrivastava, A. Gupta, A-fast-rcnn: hard positive generation via adversary for object detection, in: CVPR, 2017, pp. 3039–3048, doi:10.1109/CVPR.2017.324.
[27] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks, in: CVPR, 2016, pp. 2874–2883, doi:10.1109/CVPR.2016.314.
[28] S. Gidaris, N. Komodakis, Object detection via a multi-region and semantic segmentation-aware cnn model, in: ICCV, 2015, pp. 1134–1142, doi:10.1109/ICCV.2015.135.
[29] A. Shrivastava, A. Gupta, Contextual priming and feedback for faster r-cnn, in: ECCV, 2016, pp. 330–348, doi:10.1007/978-3-319-46448-0_20.
[30] X. Zeng, W. Ouyang, B. Yang, J. Yan, X. Wang, Gated bi-directional cnn for object detection, in: ECCV, 2016, pp. 354–369, doi:10.1007/978-3-319-46478-7_22.
[31] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: ECCV, 2016, pp. 354–370, doi:10.1007/978-3-319-46493-0_22.
[32] T. Kong, A. Yao, Y. Chen, F. Sun, Hypernet: towards accurate region proposal generation and joint object detection, in: CVPR, 2016, pp. 845–853, doi:10.1109/CVPR.2016.98.
[33] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: integrated recognition, localization and detection using convolutional networks, arXiv:1312.6229.
[34] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: CVPR, 2016, pp. 779–788, doi:10.1109/CVPR.2016.91.
[35] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, X. Xue, Dsod: learning deeply supervised object detectors from scratch, in: ICCV, 2017, pp. 1937–1945, doi:10.1109/ICCV.2017.212.
[36] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection, CVPR, 2018.
[37] Y. Yuan, Z. Xiong, Q. Wang, Vssa-net: vertical spatial sequence attention network for traffic sign detection, TIP.
[38] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, Y. Chen, Ron: reverse connection with objectness prior networks for object detection, in: CVPR, Vol. 1, 2017, p. 2, doi:10.1109/CVPR.2017.557.
[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: ICCV, 2017, pp. 2999–3007, doi:10.1109/ICCV.2017.324.
[40] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S.Z. Li, S^3FD: single shot scale-invariant face detector, in: ICCV, 2017, pp. 192–201, doi:10.1109/ICCV.2017.30.
[41] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, M. Park, Pvanet: deep but lightweight neural networks for real-time object detection, arXiv:1608.08021.
[42] Y. Deng, H. Li, Q. Wang, Q. Du, Nuclear norm-based matrix regression preserving embedding for face recognition, Neurocomputing 311 (2018) 279–290, doi:10.1016/j.neucom.2018.05.078.
[43] M. Wozniak, D. Polap, Object detection and recognition via clustered features, Neurocomputing 320 (2018) 76–84, doi:10.1016/j.neucom.2018.09.003.
[44] L. Huang, A. Shimizu, Y. Hagihara, H. Kobatake, Gradient feature extraction for classification-based face detection, PR 36 (11) (2003) 2501–2511, doi:10.1016/S0031-3203(03)00130-4.
[45] D.G. Lowe, Distinctive image features from scale-invariant keypoints, IJCV 60 (2) (2004) 91–110, doi:10.1023/B:VISI.0000029664.99615.94.
[46] J. Meynet, V. Popovici, J. Thiran, Face detection with boosted gaussian features, PR 40 (8) (2007) 2283–2291, doi:10.1016/j.patcog.2007.02.001.
[47] B. Jun, D. Kim, Robust face detection using local gradient patterns and evidence accumulation, PR 45 (9) (2012) 3304–3316, doi:10.1016/j.patcog.2012.02.031.
[48] J. Shen, X. Zuo, J. Li, W. Yang, H. Ling, A novel pixel neighborhood differential statistic feature for pedestrian and face detection, PR 63 (2017) 127–138, doi:10.1016/j.patcog.2016.09.010.
[49] S.C. Brubaker, J. Wu, J. Sun, M.D. Mullin, J.M. Rehg, On the design of cascades of boosted ensembles for face detection, IJCV 77 (1–3) (2008) 65–86, doi:10.1007/s11263-007-0060-1.
[50] M.-T. Pham, T.-J. Cham, Fast training and selection of haar features using statistics in boosting-based face detection, in: ICCV, 2007, pp. 1–7, doi:10.1109/ICCV.2007.4409038.
[51] J. Chen, X. Chen, J. Yang, S. Shan, R. Wang, W. Gao, Optimization of a training set for more robust face detection, PR 42 (11) (2009) 2828–2840, doi:10.1016/j.patcog.2009.02.006.
[52] Y. Ban, S. Kim, S. Kim, K. Toh, S. Lee, Face detection based on skin color likelihood, PR 47 (4) (2014) 1573–1585, doi:10.1016/j.patcog.2013.11.005.
[53] B. Heisele, T. Serre, S. Prentice, T.A. Poggio, Hierarchical classification and feature reduction for fast face detection with support vector machines, PR 36 (9) (2003) 2007–2017, doi:10.1016/S0031-3203(03)00062-1.
[54] L. Bourdev, J. Brandt, Robust object detection via soft cascade, in: CVPR, Vol. 2, 2005, pp. 236–243, doi:10.1109/CVPR.2005.310.
[55] L. Huang, A. Shimizu, A multi-expert approach for robust face detection, PR 39 (9) (2006) 1695–1703, doi:10.1016/j.patcog.2005.11.020.
[56] S.Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, in: ECCV, 2002, pp. 67–81, doi:10.1007/3-540-47979-1_5.
[57] S. Wu, M. Kan, Z. He, S. Shan, X. Chen, Funnel-structured cascade for multi-view face detection with alignment-awareness, Neurocomputing 221 (2017) 138–145, doi:10.1016/j.neucom.2016.09.072.
[58] J. Yan, Z. Lei, L. Wen, S.Z. Li, The fastest deformable part model for object detection, in: CVPR, 2014, pp. 2497–2504, doi:10.1109/CVPR.2014.320.
[59] J. Yan, X. Zhang, Z. Lei, S.Z. Li, Face detection by structural models, in: Image and Vision Computing, doi:10.1016/j.imavis.2013.12.004.
[60] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: CVPR, 2012, pp. 2879–2886, doi:10.1109/CVPR.2012.6248014.
[61] M. Mathias, R. Benenson, M. Pedersoli, L.V. Gool, Face detection without bells and whistles, in: ECCV, 2014, pp. 720–735, doi:10.1007/978-3-319-10593-2_47.
[62] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: CVPR, 2015, pp. 5325–5334, doi:10.1007/978-3-319-64698-5_32.
[63] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, SPL 23 (10) (2016) 1499–1503, doi:10.1109/LSP.2016.2603342.
[64] J. Yu, Y. Jiang, Z. Wang, Z. Cao, T. Huang, Unitbox: an advanced object detection network, in: ACMMM, 2016, pp. 516–520, doi:10.1145/2964284.2967274.
[65] Z. Hao, Y. Liu, H. Qin, J. Yan, X. Li, X. Hu, Scale-aware face detection, in: CVPR, 2017, pp. 1913–1922, doi:10.1109/CVPR.2017.207.
[66] W. Chu, W. Li, Manga face detection based on deep neural networks fusing global and local information, PR 86 (2019) 62–72, doi:10.1016/j.patcog.2018.08.008.
[67] G. Song, Y. Liu, M. Jiang, Y. Wang, J. Yan, B. Leng, Beyond trade-off: accelerate fcn-based face detector with higher accuracy, CVPR, 2018.
[68] X. Shi, S. Shan, M. Kan, S. Wu, X. Chen, Real-time rotation-invariant face detection with progressive calibration networks, CVPR, 2018.
[69] Y. Bai, Y. Zhang, M. Ding, B. Ghanem, Finding tiny faces in the wild with generative adversarial network, CVPR, 2018.
[70] H. Jiang, E. Learned-Miller, Face detection with the faster r-cnn, in: FG, 2017, pp. 650–657, doi:10.1109/FG.2017.82.
[71] H. Wang, Z. Li, X. Ji, Y. Wang, Face r-cnn, arXiv:1706.01061.
[72] Y. Wang, X. Ji, Z. Zhou, H. Wang, Z. Li, Detecting faces using region-based fully convolutional networks, arXiv:1709.05256.
[73] X. Sun, P. Wu, S.C. Hoi, Face detection using deep learning: an improved faster rcnn approach, Neurocomputing (2016), doi:10.1016/j.neucom.2018.03.030.
[74] C. Zhang, X. Xu, D. Tu, Face detection using improved faster rcnn, arXiv:1802.02142.
[75] Y. Li, B. Sun, T. Wu, Y. Wang, Face detection with end-to-end integration of a convnet and a 3d model, in: ECCV, 2016, pp. 420–436, doi:10.1007/978-3-319-46487-9_26.
[76] D. Triantafyllidou, P. Nousi, A. Tefas, Fast deep convolutional face detection in the wild exploiting hard sample mining, in: Big Data Research, doi:10.1016/j.bdr.2017.06.002.
[77] P. Hu, D. Ramanan, Finding tiny faces, in: CVPR, 2017, pp. 1522–1530, doi:10.1109/CVPR.2017.166.
[78] M. Najibi, P. Samangouei, R. Chellappa, L. Davis, Ssh: single stage headless face detector, in: ICCV, 2017, pp. 4885–4894, doi:10.1109/ICCV.2017.522.
[79] J. Zhang, X. Wu, J. Zhu, S.C. Hoi, Feature agglomeration networks for single stage face detection, arXiv:1712.00721.
[80] J. Wang, Y. Yuan, G. Yu, Face attention network: an effective face detector for the occluded faces, arXiv:1711.07246.
[81] C. Zhu, R. Tao, K. Luu, M. Savvides, Seeing small faces from robust anchor's perspective, CVPR, 2018.
[82] X. Tang, D.K. Du, Z. He, J. Liu, Pyramidbox: a context-assisted single shot face detector, in: ECCV, 2018, pp. 812–828, doi:10.1007/978-3-030-01240-3_49.
[83] C. Chi, S. Zhang, J. Xing, Z. Lei, S.Z. Li, X. Zou, Selective refinement network for high performance face detection, AAAI, 2019.
[84] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, F. Huang, Dsfd: dual shot face detector, arXiv:1810.10220.
[85] W. Shang, K. Sohn, D. Almeida, H. Lee, Understanding and improving convolutional neural networks via concatenated rectified linear units, in: CML, 2016, pp. 2217–2225.

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064

[86] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: CVPR, 2015, pp. 1–9, doi:10.1109/CVPR.2015.7298594.
[87] A.G. Howard, Some improvements on deep convolutional neural network based image classification, arXiv:1312.5402.
[88] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, in: CVPR, 2014, pp. 2147–2154, doi:10.1109/CVPR.2014.276.
[89] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: ACMMM, 2014, pp. 675–678, doi:10.1145/2647868.2654889.
[90] D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun, Joint cascade face detection and alignment, in: ECCV, 2014, pp. 109–122, doi:10.1007/978-3-319-10599-4_8.
[91] B. Yang, J. Yan, Z. Lei, S.Z. Li, Aggregate channel features for multi-view face detection, in: IJCB, 2014, pp. 1–8, doi:10.1109/BTAS.2014.6996284.
[92] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, W. Liu, Detecting faces using inside cascaded contextual cnn, in: ICCV, 2017, pp. 3190–3198, doi:10.1109/ICCV.2017.344.
[93] V. Jain, E. Learned-Miller, Fddb: a benchmark for face detection in unconstrained settings, university of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009 2 (7) (2010) 8.
[94] S. Yang, P. Luo, C.C. Loy, X. Tang, Wider face: a face detection benchmark, in: CVPR, 2016, pp. 5525–5533, doi:10.1109/CVPR.2016.596.
[95] S. Liao, A.K. Jain, S.Z. Li, A fast and accurate unconstrained face detector, TPAMI 38 (2) (2016) 211–223, doi:10.1109/TPAMI.2015.2448075.
[96] X. Shen, Z. Lin, J. Brandt, Y. Wu, Detecting and aligning faces by image retrieval, in: CVPR, 2013, pp. 3460–3467, doi:10.1109/CVPR.2013.444.
[97] Z. Kalal, J. Matas, K. Mikolajczyk, Weighted sampling for large-scale boosting, in: BMVC, 2008, pp. 1–10, doi:10.5244/C.22.42.
[98] S.S. Farfade, M.J. Saberian, L.J. Li, Multi-view face detection using deep convolutional neural networks, in: ICMR, 2015, pp. 643–650, doi:10.1145/2671188.2749408.
[99] V. Kumar, A. Namboodiri, C. Jawahar, Visual phrases for exemplar face detection, in: ICCV, 2015, pp. 1994–2002, doi:10.1109/ICCV.2015.231.
[100] H. Li, G. Hua, Z. Lin, J. Brandt, J. Yang, Probabilistic elastic part model for unsupervised face detector adaptation, in: ICCV, 2013, pp. 793–800, doi:10.1109/ICCV.2013.103.
[101] H. Li, Z. Lin, J. Brandt, X. Shen, G. Hua, Efficient boosted exemplar-based face detection, in: CVPR, 2014, pp. 1843–1850, doi:10.1109/ICCV.2013.103.
[102] J. Li, Y. Zhang, Learning surf cascade for fast and accurate object detection, in: CVPR, 2013, pp. 3468–3475, doi:10.1109/CVPR.2013.445.
[103] R. Ranjan, V.M. Patel, R. Chellappa, A deep pyramid deformable part model for face detection, in: BTAS, 2015, pp. 1–8, doi:10.1109/BTAS.2015.7358755.
[104] D. Triantafyllidou, A. Tefas, A fast deep convolutional neural network for face detection in big visual data, in: INNS Conference on Big Data, 2016, pp. 61–70, doi:10.1007/978-3-319-47898-2_7.



Shifeng Zhang received the B.S. degree from the University of Electronic Science and Technology of China (UESTC), in 2015. Since September 2015, he has been a Ph.D. candidate at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science (CASIA). His research interests include computer vision, pattern recognition, especially with a focus on object detection, face detection, pedestrian detection, video detection.

Xiaobo Wang received the B.S. and M.E. degrees from the School of Science, Tianjin University, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include machine learning, deep learning, data mining, and computer vision.

Zhen Lei received the B.S. degree in automation from the University of Science and Technology of China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2010, where he is currently a professor. He has published more than 130 papers in international journals and conferences. His research interests are in computer vision, pattern recognition, image processing, and face recognition in particular. He served as an area chair of the International Joint Conference on Biometrics in 2014, the IAPR/IEEE International Conference on Biometric in 2015, 2016, 2018, and the IEEE International Conference on Automatic Face and Gesture Recognition in 2015. He is the winner of 2019 IAPR Young Biometrics Investigator Award. He is a senior member of the IEEE.

Stan Z. Li received the B.Eng degree from Hunan University, China, the M.Eng degree from National University of Defense Technology, China, and the Ph.D. degree from Surrey University, United Kingdom. He is currently a professor and the director of Center for Biometrics and Security Research (CBSR), Institute of Automation, Chinese Academy of Sciences (CASIA). He was with Microsoft Research Asia as a researcher from 2000 to 2004. Prior to that, he was an associate professor in the Nanyang Technological University, Singapore. His research interests include pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance. He has published more than 300 papers in international journals and conferences, and authored and edited eight books. He was an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and is acting as the editor-in-chief for the Encyclopedia of Biometrics. He served as a program co-chair for the International Conference on Biometrics 2007, 2009, 2013, 2014, 2015, 2016 and 2018, and has been involved in organizing other international conferences and workshops in the fields of his research interest. He was elevated to IEEE fellow for his contributions to the fields of face recognition, pattern recognition and computer vision and he is a member of the IEEE Computer Society.

Please cite this article as: S. Zhang, X. Wang and Z. Lei et al., Faceboxes: A CPU real-time and accurate unconstrained face detector, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.07.064