ACCELERATING PROPOSAL GENERATION NETWORK FOR FAST FACE DETECTION ON MOBILE DEVICES

Heming Zhang[1], *Xiaolong Wang*[2*], *Jingwen Zhu*[2], *C.-C. Jay Kuo*[1]

[1] University of Southern California
Los Angeles, CA, USA

[2] Samsung Research America
Mountain View, CA, USA

ABSTRACT

Face detection is a widely studied problem over the past few decades. Recently, significant improvements have been achieved via the deep neural network, however, it is still challenging to directly apply these techniques to mobile devices for its limited computational power and memory. In this work, we present a proposal generation acceleration framework for real-time face detection. More specifically, we adopt a popular cascaded convolutional neural network (CNN) as the basis, then apply our acceleration approach on the basic framework to speed up the model inference time. We are motivated by the observation that the computation bottleneck of this framework arises from the proposal generation stage, where each level of the dense image pyramid has to go through the network. In this work, we reduce the number of image pyramid levels by utilizing both global and local facial characteristics (i.e., global face and facial parts). Experimental results on public benchmarks WIDER-face and FDDB demonstrate the satisfactory performance and faster speed compared to the state-of-the-arts.

Index Terms— Face detection, mobile devices

1. INTRODUCTION

Face detection has been studied for a long time for its important prerequisite of these face related applications, e.g., face recognition [1], face alignment [2], face editing [3], face manipulation [4] and tracking [5]. Early works on face detection mainly rely on hand-crafted features with classifiers. Viola-Jones detector [6] is one typical approach which combines the Haar features with AdaBoost classifier. It is still a popular method nowadays due to its small model size and fast speed. Then, deformable part models (DPM) based techniques [7, 8] are becoming popular where latent support vector machine (SVM) is applied to find the parts and their geometric relationship. Although DPM-based methods have achieved remarkable performance, they are computationally expensive and sensitive to handcraft features. Recently, a boosted-decision-tree-based face detector [9] outperforms all other non-CNN techniques while operating at the fast speed. However, as discussed in [9], the detection performance of these boosted trees model is still limited. One major drawback for these approaches is the feature is not learned from the data. This limits the improvement space with additional data and modeling capacity.

With the powerful discriminative capability of the deep neural network, many CNN-based methods have been proposed to solve the face detection problem. One earlier work is proposed by Farfade et al. [10] where a pre-trained AlexNet [11] is used as the basic network structure and converted to a fully-convolutional structure to fit different input face sizes. The feature map is directly used as the heatmap to localize faces.

Recently, many CNN-based works aim to improve the detection accuracy while compromising model size and speed which can facilitate the mobile device applications. Bai et al. [12] add multi-scale branches to the end of the network and reduces the image pyramid to octave-space scales. Yang et al. [13] utilize multiple proposal networks to avoid image pyramid. Recent work [14] has achieved significant improvements in tiny face detection by training separate detectors and defining multiple templates for different scales. In [15], facial parts heatmaps are obtained from five different networks and combined into a single heatmap. The faceness measure of a candidate bounding box is calculated based on the geometry of each part. The face proposals are then refined by fine-tuned AlexNet [11]. Hao et al. [16] come with a scale-aware framework where possible sizes of faces are estimated by the scale proposal network. However, the model sizes and computation complexities of these works are still not suitable for mobile devices.

Cascaded CNN face detections [2, 17, 18, 19] are favored for its small model size and fast speed compared to other frameworks introduced previously. The cascaded CNN [17] is combined with several shallow networks at different resolutions. However, the network grows larger along with the added cascade. Instead of training each network in the cascade separately, a joint training framework is proposed in [18]. To further improve the accuracy of cascaded CNN, face detection and alignment are jointly learned in [2]. Although these approaches discussed above have relatively small model sizes, the images need to be pre-processed to form the pyramid in order to tolerate various input face sizes. During the inference stage, the input image has to be repeated for each level of the image pyramid, which significantly increases the inference time.

To adapt the CNN-based approaches from high-performance platforms to mobile devices, it is not possible to apply a large and deep structure as discussed above. In this case, we follow the general pipeline as discussed in recent popular cascaded CNN frameworks [17], which consists of several light-weighted networks. As designed, the first stage of the cascade network serves as a proposal detection stage which quickly scans through the whole image to obtain face candidates. However, the proposal networks can only detect faces within a small range of sizes. For these faces whose sizes exceed the receptive field, it will fail to capture the global facial characteristics. Existing works solve this issue by rescaling the given image to different sizes. The generated images at different scales have to go through the given network which is highly computation inefficient. To speed up the inference procedure, we propose a proposal generation acceleration framework which utilizes both global and local facial cues and enables the multi-scale capability

* indicates corresponding author

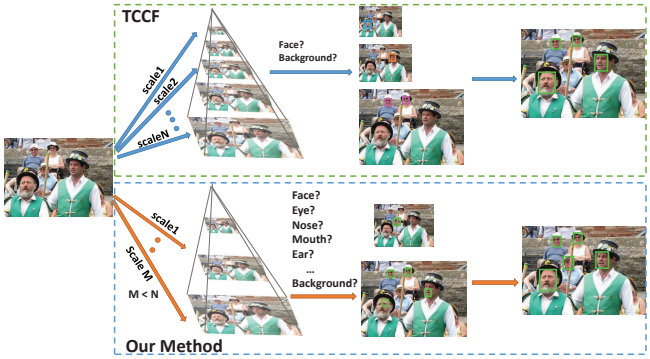

Fig. 1. Comparison between our method and typical cascaded CNN frameworks. TCFF indicates the general structure of these typical cascaded CNN frameworks. As we can see, since we also capture local characteristics like eyes, nose, mouth, etc., a single level of the pyramid encodes multiple scales of faces, thus we can reduce the number of pyramid levels and speed up the proposal generation process.

of the proposal stage. For these faces which exceed the processing size, we utilize local captured facial characteristics as cues to infer the face locations. Consequently, face regions with multiple sizes can be found in a single forward pass. In this way, locating faces of different sizes requires fewer pyramid levels.

The main contributions of this paper can be summarized as:

- We introduce a new pipeline to accelerate face proposals generation by capturing both global and local facial characteristics. This greatly reduces the number of pyramid levels for the given input image.
- Our proposed face detector has satisfactory performance and yet meets the crucial memory and speed requirements of mobile devices.
- Our approach can quickly infer the location of face regions using local facial characteristics instead of relying on the global face.

2. PROPOSED METHOD

2.1. Observation and Motivation

Although the cascaded CNN framework is faster compared to other deep learning structures, it is still not feasible for real-time face detection on mobile devices. This is due to the observation that most of the time is spent on the first stage where it serves as a proposal network and takes each level of the image pyramid as input. Motivated by this observation, we focus on designing a new proposal network to reduce the total amount of image pyramid levels.

In previous cascaded CNN frameworks, the image needs to be resampled to the right size to make sure the face region matches to the receptive field of the proposal network (12×12 is used in [2, 17, 18, 19]). Each pyramid level corresponds to a specific scale of the face. As a result, we need to form a dense image pyramid in order to achieve high detection accuracy. One way to reduce the computation time is to directly reduce the number of pyramid levels. However, the accuracy will drop rapidly. If we can encode more scales per pyramid level, less sparse pyramid levels will be needed, and then the proposal generation process will be accelerated.

Table 1. Proposal network architecture

Layer	Kernel size	Output size
Input		$12\times12\times3$
Conv1	3×3	$12\times12\times16$
Pool1	3×3	$6\times6\times16$
Conv2	3×3	$4\times4\times32$
Conv3	3×3	$2\times2\times32$
Conv4	2×2	$1\times1\times64$
Conv5	1×1	$1\times1\times8$

Based on this motivation, we propose a novel proposal module that not only focuses on the global characteristics of the face, but also captures some local cues. If the global characteristic is captured, the input patch will be directly passed to the next stage as a face proposal. On the other hand, when local cues are captured, the location of the face is inferred and the corresponding region is used as the face proposal. A comparison between our method and previous cascaded CNN frameworks is demonstrated in Fig.1. Our proposed proposal acceleration pipeline is illustrated in Fig. 2.

2.2. Proposal Network Design

Our proposal network needs to be able to capture both global and local characteristics of faces. Therefore we design the network as a multi-label classifier which can classify an input patch to the background, global face or other facial parts. Since we do not want to confuse the classifier with similar regions such as cheek and forehead, we only choose the most distinctive facial parts such as eye, nose and mouth.

Table. 1 gives the details of our proposal network for training. The input resolution of the proposal network is 12×12, the same as previous work [2, 17, 18, 19]. Inspired by [2] and [10], we design the network to be fully-convolutional. Therefore we can directly apply the network to images with arbitrary dimension and avoid cropping out patches using sliding window. The stride of the whole network is set to 2. When it scans through an input image during testing, it is equivalent to using sliding window of stride 2.

2.3. Proposal Generation

During the detection, given heatmaps of the global face and facial parts, we need to generate proposals in terms of bounding boxes. The bounding boxes generation from face heatmap and facial part heatmaps are processed separately. For face heatmap, similar to [10], a threshold τ_f is applied to the heatmap and local maximums on the heatmap are extracted to generate bounding boxes.

Fast proposal generation from parts is a non-trivial problem. Previous approaches relying on facial parts [7, 15, 20] are computational intensive when combining facial part region with the global face via sliding windows. Unlike these works, our proposed method aims at utilizing facial parts to reduce the number of image pyramid levels and speed up the detection process. Since the number of faces is much less than the number of sliding windows, it is time-consuming to evaluate each sliding window. Furthermore, we do not want to spend extra time on generating generic object proposals. Therefore instead of using candidate window approach and scoring each window, we propose to directly generate candidate bounding boxes from our facial part heatmaps. These bounding boxes are then combined and evaluated simultaneously. Our pipeline consists of three steps:

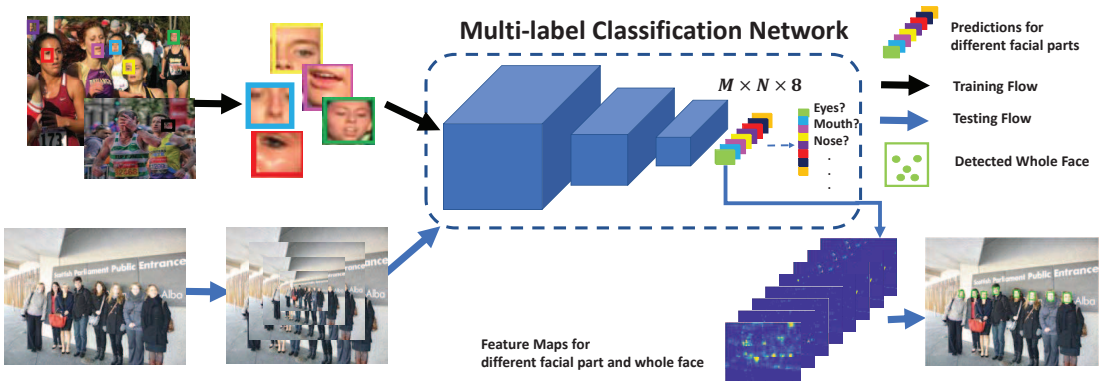

Fig. 2. An illustration of the proposal module. In the training stage, face and facial part patches are randomly cropped from training images, and used to train a multi-label classification network. In the testing stage, a test image is resized to form a sparse pyramid, and fed into the multi-label classification network to generate heatmaps of face and facial parts. Based on the heatmaps and bounding box templates of each facial parts, we can generate face proposals. These face proposals will be sent to the next stage of the cascaded framework.

1. **Finding local maxima**
   For each facial part heatmap, we first apply threshold τ_p to find the strong response, where p denotes a certain facial part. Non-maximum suppression (NMS) is then applied to obtain the strongest response points in local regions of heatmap.
2. **Bounding box generating using templates**
   We define bounding box template(s) of the face for each facial part. For eyes, we define two templates since we do not identify left eyes and right eyes. Each bounding box is determined by coordinates of its upper-left vertex (x_1, y_1) and bottom-right vertex (x_2, y_2). We denote a bounding box i's location as $\mathbf{b}_i = (x_{i1}, y_{i1}, x_{i2}, y_{i2})$. For bounding box i, we define its score p_i as the corresponding value on the heatmap. In this way, we can roughly sketch the bounding boxes of the face based on detected local maximums from the previous step.
3. **Part box combination**
   For the bounding boxes generated from different facial parts, we employ a similar way as NMS to combine them, which is described as follows.

   Given a set of bounding boxes, we start from the bounding box with highest score and find all bounding boxes that have intersection over union (IoU) with it higher than threshold τ_{IoU}. By taking the average of their coordinates, those bounding boxes are merged together:

$$\mathbf{b}_{m,i} = \frac{1}{|\mathcal{C}_i|} \sum_{j \in \mathcal{C}_i} \mathbf{b}_j, \quad \text{where } \mathcal{C}_i = \{\mathbf{b}_i\} \bigcup \{\mathbf{b}_j : IoU(\mathbf{b}_i, \mathbf{b}_j) > \tau_{IoU}\}. \tag{1}$$

   The score of the merged bounding box is defined as

$$p_{m,i} = 1 - \prod_{j \in \mathcal{C}_i} (1 - p_j), \tag{2}$$

   which resembles the statistic rule of combining two independent events. The merged bounding box is assigned to the proposal set and the bounding boxes used for merging are eliminated from the original set. Then we repeat the searching and merging process for the remaining bounding boxes in the original set. This process is repeated until there is no remaining bounding boxes left.

3. EXPERIMENTS

3.1. Experimental Setup

As stated before, our accelerating proposal module can be combined with any face classifiers. To train a small model with satisfactory performance, we cascade it with a CNN to construct the whole face detection pipeline. Specifically, after the proposal module, we adopt two successive sub-networks that follow the same structure as the RNet (second stage) and the ONet (last stage) used in the MTCNN [2]. As a result, we form a three-stage cascaded lightweight deep face detector.

We evaluate the proposed face detector on two popular benchmarks: the WIDER-face [13] and the FDDB [21]. The WIDER-face dataset has 393,703 labeled face bounding boxes from 32,203 images while the FDDB dataset contains 5,171 annotated faces. To build our training dataset, we use the WIDER-face [13] training set to extract background and face patches. The WIDER-face dataset consists of 32,000 images, where 50% of them are used for testing, 40% for training and the remaining ones are for validation. Furthermore, the eye, nose and mouth patches for training are extracted from the CelebA [22], which has around 200,000 images and most of the images contain a single face with landmark locations provided.

3.2. Evaluation of Model Size

We compare the size of our model with other works. The results are listed in Table. 2, where * denotes our calculation based on the information from the literature, and the rest is directly measured. From the comparison, our model is much smaller compared to those complicated CNN frameworks such as DDFD [10], HR [14], CEDN [20]. It is even smaller than LCDF+ [9], which is the state-of-the-art method that uses manually-crafted feature framework. Our framework also has comparable size to other cascaded CNN-based models (MTCNN [2], nested CNN detector[19]).

3.3. Evaluation of Face Detection

Multi-scale capability

We compare the multi-scale capabilities of our face detector and the MTCNN [2], which is the state-of-the-art cascaded CNN face detection engine. The WIDER-face validation set [13] with large face scale variety is used for evaluation. . It consists of three subset, namely the easy, medium and hard sets. Since this experiment

Table 2. Comparisons of model size with state-of-the-art networks.

Work	Model size
CEDN [20]	1.1GB*
DDFD [10]	233MB*
HR [14]	98.9MB
LCDF+ [9]	2.33MB
MTCNN [2]	1.9MB
Nested [19]	1.6MB*
Ours	1.96MB

Table 3. Comparisons of detection performance with MTCNN[2] on WIDER-face validation set [13] with different scale factors.

Scale factor		0.79	0.50	0.25
Easy	MTCNN [2]	0.836	0.817	0.755
	Ours	0.844	0.842	0.826
Medium	MTCNN [2]	0.809	0.798	0.744
	Ours	0.809	0.805	0.794
Hard	MTCNN [2]	0.622	0.600	0.529
	Ours	0.603	0.568	0.519

targets at evaluating the performance in multi-scale detectability, we set different levels of scaling factor for the image pyramid. For fair comparison, we use the model provided and follow the same parameter setting in [2]. The results are listed in Table 3.

From the results, we can find that both face detectors achieve satisfactory accuracy with the dense image pyramid at the scale factor of 0.79. This scale is also chosen by MTCNN. Our detector outperforms the MTCNN on the Easy set, while the MTCNN performs better on the Hard set. It is worth noting that the MTCNN utilizes joint training for face detection and facial landmark localization. The latter is not used in our detector training.

As the image pyramid becomes sparse, MTCNN's accuracy drops rapidly. When the scale factor decreases from 0.79 to 0.25, its accuracy degrades by 8.1%, 6.5% and 9.3% on the easy, medium and hard sets, respectively. In contrast, the accuracy of our method without model acceleration drops by 1.8%, 1.5%, 8.4%, respectively.

Accuracy benchmarks

We conduct face detection experiments on the FDDB [21]. We use 0.25 as the pyramid scaling factor and add an extra layer to the image pyramid with half of the size of the largest scale. As illustrated in Fig. 3, our method outperforms many others such as the CEDN [20] and the nested CNN detector [19]. In terms of detection accuracy, our model can achieve 94.35%. As compared to 83.29% obtained by the nested CNN detector, we have 9.2% improvement. It also achieves comparable accuracy as compared to the MTCNN [2] that uses more pyramid levels. The HR [14] outperforms our model by a small margin, yet its model size is too large to be deployed on mobile devices.

3.4. Evaluation of Runtime Efficiency

Runtime comparison with MTCNN [2]

We compare the detection speed with the MTCNN method using their provided Matlab codes. For fair comparison, our method is also implemented in Matlab codes. The experiment was conducted on the WIDER-face validation set [13] using the GeForce GTX TITAN. We use original images without re-sampling to a fixed resolution. The

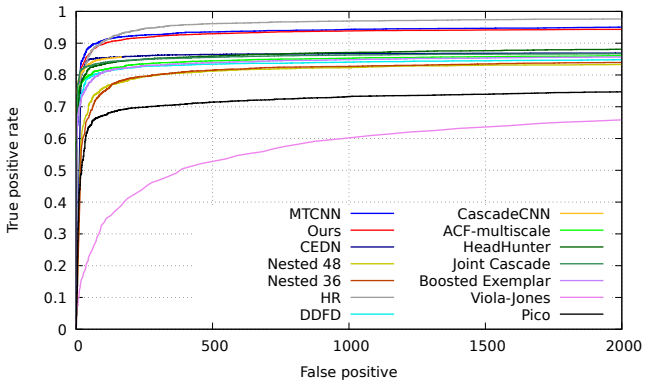

Fig. 3. Evaluation results on FDDB.

runtime is calculated by averaging the time over the entire validation set. For both detectors, the minimum face size to detect is set to 10 as used by the MTCNN. The scaling factor of the MTCNN is 0.79 as given in its original setting, while scaling factor of 0.25 and an extra pyramid layer is the setting for our detector with comparable accuracy listed in Sec. 3.3. For some images in the WIDER-face validation set, the number of face proposals generated by the MTCNN is more than that our GPU memory (12GiB) can take. Therefore, we take at most 20000 proposals per image for the MTCNN, which in fact reduces the average runtime of the MTCNN. The average runtime of the MTCNN in this case is 0.595s while that of our detector is 0.499s. We reach more than 16% acceleration. Clearly, our detector achieves comparable accuracy with a faster speed.

Runtime comparison with Nested CNN Detector [19]

The running time claimed by the nested CNN detector [19] is 40.1ms using the CPU only, where 640×480 VGA image with 80×80 as the minimum size. For comparison, we follow the same setting of the resolution and the minimum face size. The data used for runtime evaluation was not mentioned in [19]. Here, we evaluate detection accuracy and running time on FDDB [21] with the same dataset in [19] for performance benchmarking. With model acceleration, our model can get 39.1ms compared to 40.1ms achieved by the nested CNN detector. It shows that we can still get a faster s peed with significant accuracy improvement as indicated in Sec. 3.3.

Speed on mobile devices

We implemented our method on Samsung Galaxy S8 using Caffe. The face detector received images of high resolution (1280×720) from the back camera continuously. By setting the minimum face size to 100 and scaling factor to 0.25, even with the extra pyramid layer mentioned in previous experiments, the detection speed still achieves 8 to 10 FPS in different scenarios on mobile CPU.

4. CONCLUSION

In this paper, we presented an efficient face detector. Particularly, we proposed a new framework to quickly generate face proposals by capturing both global and local facial cues to reduce image pyramid levels and introduced a method to infer face locations from local facial characteristics. We validated the proposed methods on two popular benchmarks. The promising performance over the state-of-the-art in terms of accuracy, model size and detection speed demonstrates the potential of our approach towards the real deployment on mobile devices.

5. REFERENCES

[1] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[2] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[3] Brandon M Smith, Li Zhang, Jonathan Brandt, Zhe Lin, and Jianchao Yang, "Exemplar-based face parsing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3484–3491.

[4] Zhifei Zhang, Yang Song, and Hairong Qi, "Age progression/regression by conditional adversarial autoencoder," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry Rowley, "Face tracking and recognition with visual constraints in real-world videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[6] Paul Viola and Michael J Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[7] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool, "Face detection without bells and whistles," in *European Conference on Computer Vision*. Springer, 2014, pp. 720–735.

[8] Xiangxin Zhu and Deva Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2879–2886.

[9] Eshed Ohn-Bar and Mohan M Trivedi, "To boost or not to boost? on the limits of boosted trees for object detection," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 3350–3355.

[10] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li, "Multi-view face detection using deep convolutional neural networks," in *5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 643–650.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[12] Yancheng Bai, Wenjing Ma, Yucheng Li, Liangliang Cao, Wen Guo, and Luwei Yang, "Multi-scale fully convolutional network for fast face detection.," in *BMVC*, 2016.

[13] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5525–5533.

[14] Peiyun Hu and Deva Ramanan, "Finding tiny faces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[15] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang, "From facial parts responses to face detection: A deep learning approach," in *IEEE International Conference on Computer Vision*, 2015, pp. 3676–3684.

[16] Zekun Hao, Yu Liu, Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu, "Scale-aware face detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[17] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua, "A convolutional neural network cascade for face detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5325–5334.

[18] Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu, "Joint training of cascaded cnn for face detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3456–3465.

[19] Jingjing Deng and Xianghua Xie, "Nested shallow cnn-cascade for face detection in the wild," in *IEEE International Conference on Automatic Face & Gesture Recognition(FG)*. IEEE, 2017, pp. 165–172.

[20] Lezi Wang, Xiang Yu, and Dimitris N Metaxas, "A coupled encoder-decoder network for joint face detection and landmark localization," in *IEEE International Conference on Automatic Face & Gesture Recognition(FG)*, 2017, pp. 251–257.

[21] Vidit Jain and Erik Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," Tech. Rep., Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *IEEE International Conference on Computer Vision*, 2015.