Neurocomputing 284 (2018) 119–127

Contents lists available at ScienceDirect

Neurocomputing

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

Detecting Face with Densely Connected Face Proposal Network

Shifeng Zhang[a,b], Xiangyu Zhu[a,b], Zhen Lei[a,b,*], Xiaobo Wang[a,b], Hailin Shi[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 6 July 2017
Revised 3 January 2018
Accepted 10 January 2018

Communicated by Dr Xiaoming Liu

Keywords:
Face detection
Small face
Region proposal network

ABSTRACT

Accuracy and efficiency are two conflicting challenges for face detection, since effective models tend to be computationally prohibitive. To address these two conflicting challenges, our core idea is to shrink the input image and focus on detecting small faces. Reducing the image resolution can significantly improve the detection speed, but it also results in smaller faces that need to pay more attention. Specifically, we propose a novel face detector, dubbed the name Densely Connected Face Proposal Network (DCFPN), with high accuracy as well as CPU real-time speed. Firstly, we subtly design a lightweight-but-powerful fully convolution network with the consideration of efficiency and accuracy. Secondly, we present a dense anchor strategy and a scale-aware anchor matching scheme to improve the recall rate of small faces. Finally, a fair L1 loss is introduced to locate small faces well. As a consequence, our proposed method can detect faces at 30 FPS on a single 2.60 GHz CPU core and 250 FPS using a GPU for the VGA-resolution images. We achieve state-of-the-art performance on the common face detection benchmark datasets.

1. Introduction

Face detection is one of the fundamental problems in computer vision and pattern recognition. It plays an important role in face based applications, since accurate and efficient face detection usually needs to be done first. With the great progress, face detection has been successfully applied in our daily life. However, there are still some tough challenges in the uncontrolled face detection problem. The challenges mainly come from two requirements for face detectors: (1) The large variation of facial changes requires face detectors to accurately address a complicated face and non-face classification problem; (2) The large search space of arbitrary 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.

To meet these challenges, face detection has been studied mainly in two different ways. One way is the cascade based methods and it starts from the pioneering work of Viola-Jones face detector [1]. Since then, the boosted cascade with simple features becomes the most popular and effective design for practical face detection. A number of improvements to the Viola-Jones face detector have been proposed in the past decade [2], which can be seen as a history of more efficiently sampling the output space to a solvable scale and more effectively evaluation per configuration. The other way is Convolutional Neural Network (CNN) [3] based methods and with the development of deep learning techniques, the CNN has been successfully applied in face detection tasks. Recently, some works based on R-CNN [4] demonstrate state-of-the-art performance on face detection tasks.

However, these two ways focus on different aspects. The former pays more attention to efficiency while the latter cares more about accuracy. To make face detector perform well on both speed and accuracy, one natural idea is to combine the advantages of them. Therefore, cascade CNN based methods [5] are proposed that put features learned by CNN into cascade framework so as to boost the performance and keep efficient. However, there are three problems in cascaded CNN based methods: (1) Their speed is negatively related to the number of faces on the image. The speed would dramatically degrade as the number of faces increases; (2) The cascade based detectors optimize each component separately, making the training process extremely complicated and the final model sub-optimal; (3) For the VGA-resolution images, their runtime efficiency on the CPU is about 14 FPS, which is not fast enough to reach the real-time speed (25 FPS).

Therefore, it is still one of the remaining open issues for practical face detectors to achieve CPU real-time speed as well as maintain high performance. In this work, we develop a state-of-the-art face detector with CPU real-time speed. The core idea is to shrink the input image and focus on detecting small faces.

* Corresponding author at: CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
E-mail addresses: shifeng.zhang@nlpr.ia.ac.cn (S. Zhang), xiangyu.zhu@nlpr.ia.ac.cn (X. Zhu), zlei@nlpr.ia.ac.cn (Z. Lei), xiaobo.wang@nlpr.ia.ac.cn (X. Wang), hailin.shi@nlpr.ia.ac.cn (H. Shi), szli@nlpr.ia.ac.cn (S.Z. Li).

https://doi.org/10.1016/j.neucom.2018.01.012
0925-2312/

Reducing the high-resolution input image into the low-resolution image can significantly improve the detection speed, but it also results in smaller faces that need to pay more attention in order to maintain high performance. Specifically, our DCFPN has a lightweight-but-powerful network with the consideration of efficiency and accuracy. To improve the recall rate of small faces, a dense anchor strategy and a scale-aware anchor matching scheme are introduced. Besides, we present a fair L1 loss to locate small faces well. Consequently, for VGA images to detect faces bigger than 40 pixels, our face detector can run at 30 FPS on a single CPU core and 250 FPS on a GPU card. More importantly, the speed of DCFPN is invariant to the number of faces on the image.

A preliminary version of this work has been published on Chinese Conference on Biometric Recognition (CCBR) 2017.[1] Comparing with the preliminary version, this paper proposes a new scale-aware anchor matching scheme and further improves the state-of-the-art performance. For clarity, the main contributions of this work can be summarized as four-fold:

- We design a lightweight-yet-powerful fully convolution network with the consideration of efficiency and accuracy for the face detection task.
- We present a dense anchor strategy and a scale-aware anchor matching scheme to improve the recall rate of small faces.
- We introduce a fair L1 loss function that directly regresses box's relative center and size in order to locate small faces well.
- We achieve state-of-the-art performance on the common face detection benchmark datasets with CPU real-time speed.

2. Related work

Face detection approaches can be roughly divided into two different categories. One is based on hand-craft features, and the other one is built on CNN. This section briefly reviews them and refer more detailed survey to [2,6,7].

Hand-craft based methods. Previous face detection systems are mostly based on the hand-crafted features. The milestone work of Viola-Jones [1] proposes to use Haar feature, Adaboost learning and cascade inference for face detection. After that, many subsequent works focus on new local features [8,9], new boosting algorithms [10–12] and new cascade structures [13–15]. Besides the cascade framework, the seminal work deformable part model (DPM) [16] is introduced into the face detection task by [17–21] , which use supervised parts, more pose partition, better training or more efficient inference to achieve better performance.

CNN based methods. Recently, CNN based methods have showed advantages in face detection. CCF [22] uses boosting on top of CNN features for face detection. Farfade et al. [23] fine-tune CNN model trained on 1 k ImageNet classification task for face and background classification task. Faceness [24] trains a series of CNNs for facial attribute recognition to detect partially occluded faces. CascadeCNN [5] uses six cascaded CNNs to efficiently reject backgrounds in three stages. STN [25] proposes a new Supervised Transformer Network and a ROI convolution for face detection. Similar to Chen et al. [26], MTCNN [27] presents a multi-task cascaded CNNs based framework for joint face detection and alignment. UnitBox [28] introduces a new intersection-over-union loss function. CMS-RCNN [29] uses Faster R-CNN in face detection with body contextual information. Convnet [30] integrates CNN with 3D face model in an end-to-end multi-task learning framework.

Generally, hand-craft based methods are able to achieve CPU real-time speed, but they are not accurate enough for the uncontrolled face detection problem. With learned feature and classifier directly from the image, CNN based methods can differentiate faces from highly cluttered backgrounds, while they are too time-consuming to reach real-time speed. Notably, our proposed DCFPN is able to achieve real-time speed on the CPU devices as well as maintain state-of-the-art detection performance.

Table 1
The receptive field of the last convolutional layer and the default anchor of our DCFPN.

Receptive field	75×75, 107×107, 139×139, 171×171, 203×203, 235×235
Default anchor	16×16, 32×32, 64×64, 128×128, 256×256

3. Densely connected face proposal network

This section presents detail of DCFPN. It includes four key contributions that make it accurate and efficient for face detection: lightweight-but-powerful architecture, dense anchor strategy, scale-aware anchor matching scheme and fair L1 loss.

3.1. Lightweight-but-powerful architecture

The architecture of DCFPN encourages feature reuse and leads to a substantial reduction of parameters. As illustrated in Fig. 1 , it consists of two parts.

Rapidly Digested Convolutional Layers. It is designed for high efficiency via quickly reducing the input image spatial size by 16 times with narrow but large kernels. On one side, face detection is a two classification problem and does not require very wide network, hence the narrow kernels is powerful enough and can result in faster running speed, especially for CPU devices. On the other side, the large kernels are to alleviate the information loss brought by spatial size reducing.

Densely Connected Convolutional Layers. Inspired by Huang et al. [31], each layer in DCCL is directly connected to every other layer in a feed-forward fashion. It ends with two micro inception layers. There are two motivations behind the design of DCCL. Firstly, the DCCL is designed to enrich the receptive field of the last convolutional layer that is used to predict the detection results. As listed in Table 1, the last convolutional layer of DCFPN has a large scope of receptive field from 75 to 235 pixels, which is consistent with our default anchors and is important for the network to learn visual patterns for different scales of faces. Secondly, the DCCL aims at combining coarse-to-fine information across deep CNN models to improve the recall rate and precision of detection. Deep and shallow CNN features are really complementary for detection task, since the information of the interest region is distributed over all levels of the convolution network with multiple level abstraction, and they should be well organized.

To sum up, our lightweight-but-powerful architecture consists of RDCL and CCL. The former is designed to achieve CPU real-time speed. The latter aims at enriching the receptive fields and combining coarse-to-fine information across different layers to handle faces of various scales.

3.2. Dense anchor strategy

As listed in Table 1, we use 5 default anchors that are associated with the last convolutional layer. Hence, these 5 default anchors have the same tiling interval on the image (i.e., 16 pixels). It is obviously that there is a tiling density imbalance problem. Comparing with large anchors (i.e., 64×64, 128×128 and 256×256), small anchors (i.e., 16×16 and 32×32) are too sparse, which results in low recall rate of small faces.

To improve the recall rate of small faces, we proposed the dense anchor strategy for small anchor. Specifically, without our dense anchor strategy, there are 5 anchors for every receptive filed center

[1] http://ccbr2017.org/