Joint Training of Cascaded CNN for Face Detection

Liangjie Cao

Jun 18, 2018

Abstract

Cascade has been widely used in face detection, where classifier with low computation cost can be firstly used to shrink most of the background while keeping the recall. The cascade in detection is popularized by seminal Viola-Jones framework and then widely used in other pipelines, such as DPM and CNN. However, to the authors' best knowledge, most of the previous detection methods use cascade in a greedy manner, where previous stages in cascade are fixed when training a new stage. So optimizations of different CNNs are isolated. In this paper, they propose joint training to achieve end-to-end optimization for CNN cascade. They show that the back propagation algorithm used in training CNN can be naturally used in training CNN cascade. They present how jointly training can be conducted on naive CNN cascade and more sophisticated region proposal network (RPN) and fast R-CNN. Experiments on face detection benchmarks verify the advantages of the joint training.

1. Introduction

Face detection plays an important role in face based image analysis and is one of the fundamental problems in computer vision. The performances of various face based applications, from face identification and veri?cation to face clustering, tagging and retrieval, rely on accurate and e?cient face detection. Recent works in face detection focus on faces in uncontrolled setting, which is challenging due to the variations in subject level (e.g., *a face can have many different poses), category level* (e.g., *adult and baby) and image level* (e.g., *illumination and cluttered background).*

Given a novel image I, the face detector is expected to return a bounding box configuration $B = (b_i, c_i)_N$, *where the* b_i *and* c_i *specify the localization and confidence of a face. The number of detected faces N always vary in different images. Considering that the* b_i *can possibly appear in any scale and position, the face detection problem has a output space of size* $\frac{(w \times h)^2}{2}$ *, where w and h denote width and height respectively. Considering that it can be* $\frac{(500 \times 350)_2}{2} \approx 1010$ *for a typical 500 350 image, it is actually impossible to evaluate them all at a acceptable cost. Actually, only a few of them correspond to faces and most of the configurations in the output space belongs to the background.*

The previous face detection research can be seen as a history of more efficiently sampling the output space to a solvable scale and more effectively evaluating per configuration. One natural idea to achieve this is using cascade, where classifier with low computation cost can be firstly used to shrink background while keeping the faces. The pioneering work [5] popularized this, which combined classifiers in different stages, to allow background regions quickly discarded while spending more computation on promising face-like regions. The cascade made efficient detection possible and was widely used in subsequent works. For example, two other detection pipelines DPM [1] and CNN [4] can both use cascade for acceleration.

In this paper, they show that in CNN based cascade detection, other than enjoying the advantages in efficiency as traditional cascade, different stages in the cascade can be jointly trained to achieve better performance. They show that the back propagation algorithm used in training CNN can be naturally used in training CNN cascade. Joint training can be conducted on naive CNN cascade and more sophisticated cascade such as region proposal network (RPN) and fast RCNN. The authors show that the jointly trained cascade CNN as well as the jointly trained RPN and fast R-CNN can achieve leading performance on face decttion.

2. Related Work

Numerous works have been proposed for face detection and some of them have been delivered to real applications. Similar to many other computer vision tasks, leading algorithms in face detection are based on convolutional neural network in the 1990s, then based on hand-craft feature and model, and recently based on convolutional neural network again. In this part, they briefly review the three kinds of methods and refer more detailed survey to [6, 3].

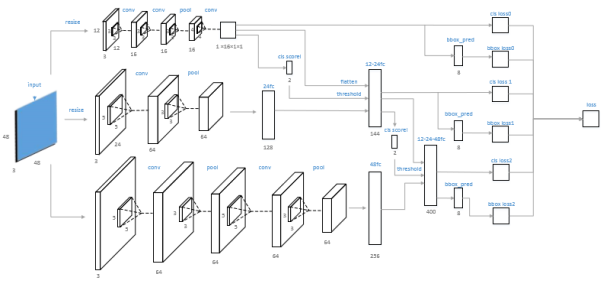

Figure 1. **Joint training architecture. During training, the network takes an image of size 48 × 48 as input, and outputs one joint loss of three branches. The network is optimized through back-propagation. Compared to separate networks, the joint network also use threshold control layers, to decide which proposals from up branches contribute to the loss of the down branches**

3. Cascaded CNNs

The cascaded CNN for face detection in [4] contains three stages. In each stage, they use one detection network and one calibration network. There are totally six CNNs. In practice, this makes the training process quite complicated. They have to carefully prepare the training samples for all the stages and optimize the networks one by one. One natural question is how about they jointly train all the stages in one network?

Firstly, detection network and calibration network can share a multi-loss network used for both detection and bounding-box regression. Multi-loss optimization has been proved effective in general objection detection.

Secondly, if multi-resolution is used during training the later stages, as the authors did in [4], the network of the later stage contains the network of the previous one. So theoretically, the convolution layers can be shared by three stages. Meanwhile, shared convolutional layers results in smaller model size. In the joint training network, the model size is approximately the same as the final stage in separate cascaded CNNs.

Thirdly, in cascaded CNNs, the separate first stage used for generating proposals is only optimized by itself. In the joint network, it is jointly optimized by larger scale branches. In this way, each branch benefits from other branches. Together, the joint network is expected to achieve end-to-end optimization.

4. Joint Training of Cascaded CNN

They design a joint training architecture to train the network once for all. They call this architecture FaceCraft. Fig. 1 demonstrates this joint training architecture. During training, the network takes an image of size 48 × 48 as input, and outputs one joint loss of three branches. The three branches are called x12, x24, x48 respectively, corresponding to the input size of each network. They use ReLU for non-linear layers and drop-out before classification or regression layer. It is optimized through back-propagation. Compared to separate networks, the joint network also use threshold control layers to decide which proposals from up branches contribute to the loss of the down branches.

They use a multi-task loss of classification and bounding-box regression to jointly optimize this branch. They use softmax loss for classification and smooth L1 loss for bounding-box regression:

$$L(p, u, t^u, v) = L_cls(p, u) + \lambda \mid u \geq 1 \mid L_loc(t^u, v) \quad (1)$$

where $L_cls(p, u)$= -log p_u is log loss for true class u.

They set λ = 1 in their experiment, this is appropriate for all three separate CNN networks.

For the regression offsets, they set 4 coordinates defined in [2]:

$$t_x^* = (x^* - x_p^*)w_p \quad (2)$$

$$t_y^* = (y^* - y_p)h_p \quad (3)$$

$$t_w^* = log(w^* w_p) \quad (4)$$

$$t_h^* = log(h^* h_p) \quad (5)$$

where x and y denote the two coordinates of the box center, w and h denote box width and height respectively . The variables x_p, and x^* are for the proposal box and groundtruth box respectively. In this way, they can optimize the regression targets and regress the bounding-box from a proposal box to a nearby ground-truth box.

References

[1] *P. F. Felzenszwalb, R. B. Girshick, and D. A. Mcallester. Cascade object detection with deformable part models. In* CVPR, *2013.*

[2] *R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In* CVPR, *2014.*

[3] *X. Jin and X. Tan. Face alignment in-the-wild: A survey.* Computer Vision and Image Understanding, *2017.*

[4] *H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In* CVPR*, 2015.*

[5] *P. Viola and M. J. Jones. Robust real-time face detection.* IJCV*, 2004.*

[6] *M. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: a survey.* IEEE TPAMI*, 2002.*