A Tech Report: FCN-based Face Detector

Gehua Ma[1], Yi Huang[1], Ruoyu Chen[2], You Jiang[2], Shanting Su[2], Jing Chen[2], Bei Chen[2], Yuan Qiu[2], Jing Zhang[2]
∗ In no particular order

[1]College of Automation, NUAA
[2]College of Computer Science, NUAA

Abstract. We presented a fully convolutional network based face detector. Experimental results showed the outstanding performance of our method.

1 Introduction

Face detect techniques are widely used nowadays, there are face-recognition-based systems everywhere. But in fact, face detection technology is one of the most difficult problems of pattern recognition field. Before deep-learning-based method become a mainstream, the most puzzling issues to be considered are the extraction of feature and the selection of detector. As the feature design almost entirely depends on prior knowledge, it is really hard for researchers to make improvements. However, deep-learning-based methods address that limitation, multi-layer-perceptrons can learn from labeled data, complete the feature extraction automatically. Benefiting from the rapid growth of data on the internet and the capacity of processing units, deep-learning methods' application scenarios were broadened. Most brought-in face detection methods are based on deep-learning or partly based on it. To carry through practice, we proposed a fully convolutional network based face detection technique, which enables high-quality face detection in a complex and changeable environment. Our method showed good performance and robustness during the test.

2 Related Work

CNNs CNNs have been successfully used in audio, image and text classification, analysis and generation. In 1998, Yann et al. proposed a convolutional-neural-network-based method to recognize the handwritten postal codes[1, 2]. Step into the twenty-first century, AlexNet achieved great success on the large-scale image classification challenge[3]. AlexNet also faced several challenges to its dominant position in the next couple of years[4–6]. Main architectures of those nets were proved to be sufficient and were widely used as backbone parts in other computer vision tasks.

FCNS As mentioned above, convolutional nets demonstrate excellent performance on the classification task, seemingly it's possible to implement pixel-level classification. The work of Long et al.[7] offered an effective way to solve this problem. They define and detail the space of fully convolutional networks, explain their application to spatially dense prediction tasks, and draw connections to prior models. They adapt contemporary classification networks (AlexNet, the VGG net, and GoogLeNet[3–5]) into fully convolutional networks and transfer their learned representations by fine-tuning to the segmentation task. A skip architecture was defined, which combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentation.

Batch Normalization During the training of deep convolutional nets, the distribution of each layers inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization and makes it notoriously hard to train models with saturating nonlinearities. Ioffe et al.[8] refer to this phenomenon as internal covariate shift and address the problem by normalizing layer inputs. Batch Normalization[8] method draws its strength from making normalization a part of the model architecture and performing the normalization for each training mini-batch. Batch Normalization allows us to use much higher learning rates and be less careful about initialization, it also eliminates the need for Dropout.

Residual Block Network depth is of crucial importance in neural network architectures, however, deeper networks are much more difficult to train. The residual learning framework eases the training of these networks and enables them to be substantially deeper. That leading to improved performance in both visual and non-visual tasks. These residual networks are much deeper than counterparts without residual structure, yet they require a similar number of parameters. Researches around residual blocks have effectively improved the performance on many tasks[6, 9].

SGD Solver Stochastic gradient descent (SGD) in contrast performs a parameter update for each training example x_i and label y_i,

$$\theta = \theta - \eta \bigtriangledown_\theta J(\theta, x_i, y_i) \tag{1}$$

Gradient descent performs redundant computations for large datasets, as it recomputes gradients for similar examples before each parameter update. So we choose mini-batch SGD in practice. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster and effective.

Darknet Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation[10–12].

MSCOCO Microsoft COCO[13] is a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. That dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation.

3 Method

Network Architecture Improvements to CNNs such as residual block[6, 9], Xception module[14] achieved state-of-art results dealing with computer vision tasks, we applied some of them to promote our network. Our network uses $3 \times 640 \times 640$ raw image as input, but as the network is fully-convolutional, it can be applied to input images of any size at employ-time. Larger input within limits can produce better result theoretically. The whole structure of the detector network is shown in Fig. 1. The backbone of the net is a combination of ResNet and Darknet53, residual blocks were applied to increase the receptive field and get full use of semantic information. Each convolutional layer is equipped with batch normalization operation to help the net better converge to the optimal. According to the spatial-pyramid theory, we use feature maps at different scales to produce detections, that avoid the information loss due to the downsample operation. Multi-scale detection increased the performance dealing with small objects.

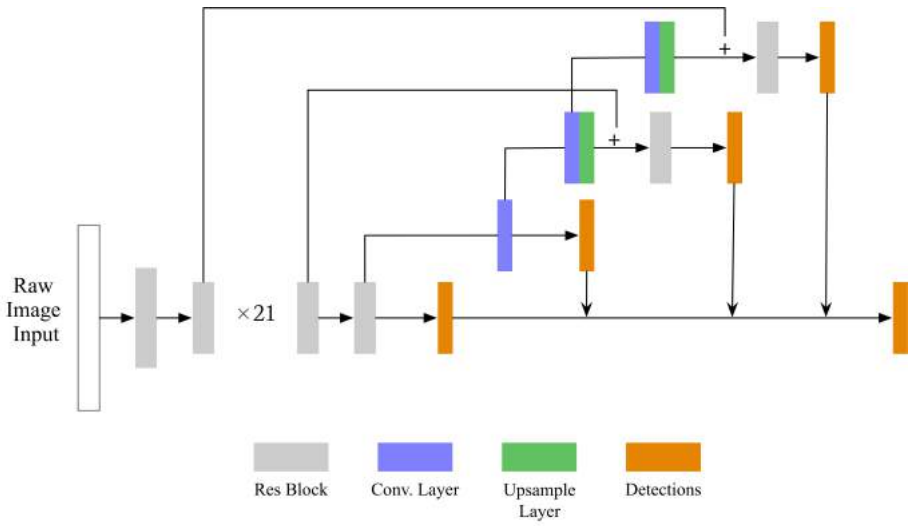

Fig. 1. Network architecture of ours

Dataset Deep-learning suffered from the manually-labeling, of course we will not do that job happily. To prevent that tragedy, we develop the auto labeled data generator. According to the prior knowledge and statistical analysis, the face part is usually the top half of the body. So the annotations of class 'person' in MSCOCO can be translated to face type using that principle. After all, we get more than $50,000$ labeled samples, most of them were annotated correctly. Some samples in our training set were shown in Fig. 2, to be honest, they were selected randomly.

Fig. 2. Examples in our set. Actually, there are some none-face bounding box in our set, but that kind of noise was at a reasonable level.

Training We used SGD solver with momentum. Every batch consists of 64 samples during the training, the learning rate obeys a step-decay policy. To make full use of existing resources, we fine-tuned our net on the basis of a network trained on Microsoft COCO dataset. That ensured the high-quality low-level features extracted using the large dataset can be reserved and help accelerating the training procedure. Small tricks like exposure, saturation, hue variation and jitter were also applied to improve the generalization.

4 Results and conclusions

Several real examples were shown in Fig 3∼Fig 6. We displayed outputs of the final-version detector. Although the shot environments, light conditions and postures are varying in our test photos, our detector showed fairly good robustness and produces quite accurate bounding boxes. Due to the limited time, we did not carry a quantitative evaluation, which includes the Recall-Precision curve and the mean-IoU index.

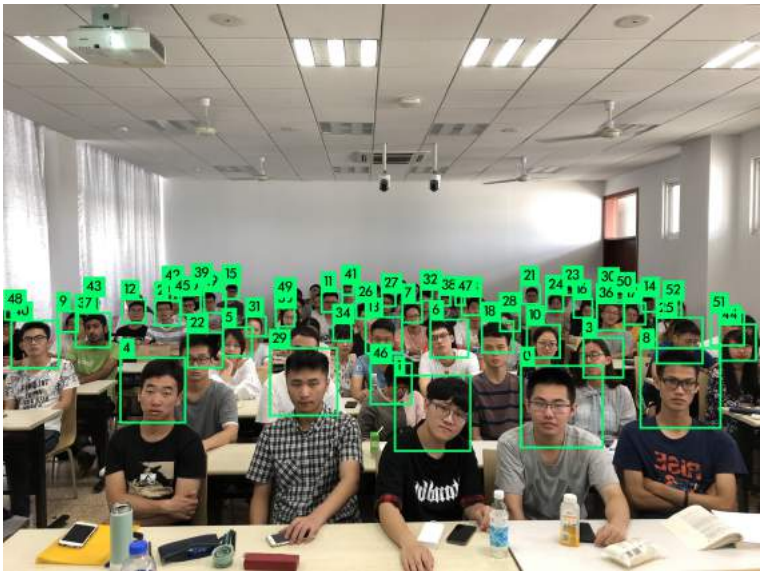

Fig. 3. One real example.

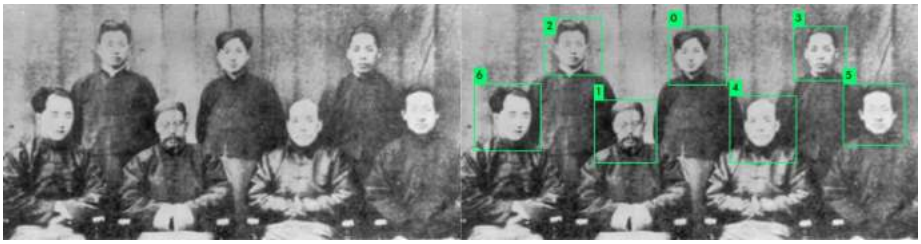

Fig. 4. One real example: An old photo of Ji Li, Guowei Wang, Qichao Liang, Yuanren Zhao, Zhaohuang Zhang, Weizhao Lu and Tingcan Liang.

Fig. 5. One real example: There are 101 faces in the image, although the resolution is low, most of them were detected precisely during our test.

Fig. 6. One real example.

5 Intra-group Assessment

We show our intra-group assessment using a heatmap Fig 7.

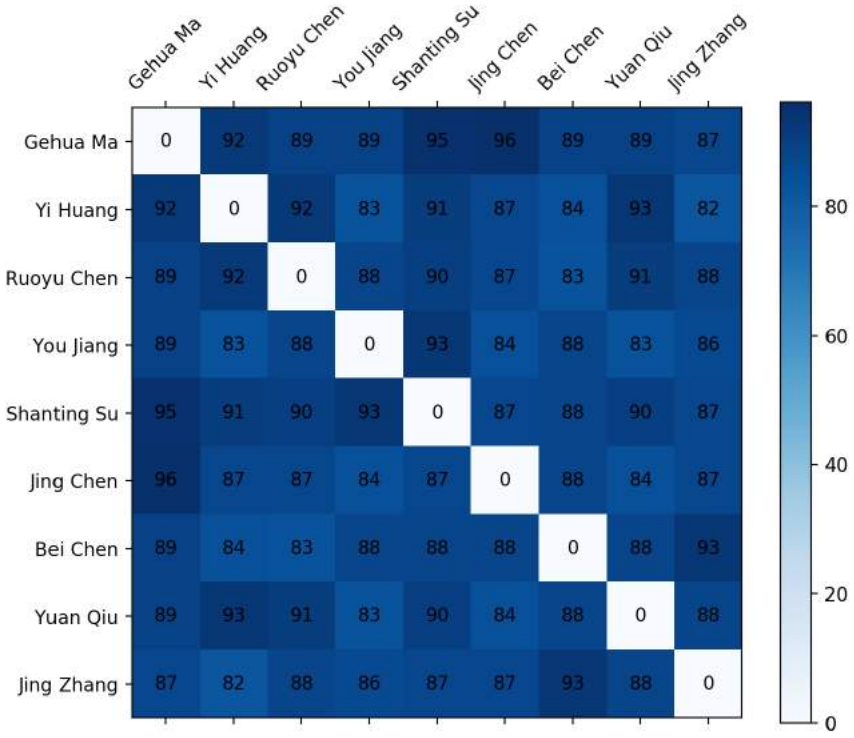

Fig. 7. Peer review confusion matrix

Table 1. Average score.

ID	Name	Score (Average)
SX1716035	Jing Chen	87.5
SX1716028	Bei Chen	86.4
SZ1716046	Jing Zhang	86.3
SZ1703117	Yi Huang	88
SF1703009	Gehua Ma	90.7
SX1716085	You Jiang	86.7
SX1716044	Ruoyu Chen	88.5
SX1716081	Yuan Qiu	88.3
SZ1716040	Shanting Su	90.1

References

1. Yann LeCunn, L.N.: Gradient-based learning applied to document recognition. Proceedings of IEEE. (1998)
2. LeCun Y, Boser B, D.J.S.: Backpropagation applied to handwritten zip code recognition. (Neural Computation)
3. Krizhevsky A, Sutskever I, H.G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012)
4. Karen Simonyan, A.Z.: Very deep convolutional networks for large-scale image recognition. arxiv.org (2014)
5. Szegedy C, Liu W, J.Y.: Going deeper with convolutions. In: CVPR. (2015)
6. He K, Zhang X, R.S.: Deep residual learning for image recognition. In: CVPR. (2016)
7. Jonathan Long, Evan Shelhamer, T.D.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
8. Sergey Ioffe, C.S.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. (2015)
9. Xie S, Girshick R, D.P.: Aggregated residual transformations for deep neural networks. In: CVPR. (2017)
10. Joseph Redmon, Divvala S, G.R.: You only look once: Unified, real-time object detection. In: CVPR. (2016)
11. Joseph Redmon, A.F.: Yolo9000: Better, faster, stronger. In: CVPR. (2017)
12. Redmon, J.: Darknet: Open source neural networks in c. (2014)
13. Tsung-Yi Lin, Michael Maire, S.B.L.B.R.G.J.H.P.P.D.R.C.L.Z.P.D.: Microsoft coco: Common objects in context. arxiv.org (2014)
14. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. arxiv.org (2016)