A Normalized Light CNN for Face Recognition

Hong Hui ZHENG, Yun Xiao ZU[*]

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

zuyx@bupt.edu.cn

Abstract. In recent years, achievements of deep learning in face recognition have exceeded those of traditional machine learning methods do. To achieve better recognition performance, the architectures of the neural network tend to be deeper and more complex, which wastes the time and space. Aiming at this issue, a series of light convolutional neural network (Light CNN) has been proposed, which consume less hardware resources while has excellent performance. In this paper, we propose a Normalized Light CNN model modified from a Light CNN model. We introduced a normalized layer to this model in both training and test phase. The normalized layer normalizes the output features, making it represent images better. We evaluate our model on LFW dataset. The accuracy of face verification reaches 98.46%, which is better than the original model.

1. Introduction

The research of face recognition started in 1950s. As an important biometric identification technology, it has the advantages of direct, friendly, convenient, interactive and so on. It has been paid close attention to by researchers. Machine vision based face recognition methods have achieved fruitful results in last decade. At first, people extract texture feature like LBP[11], sift[12] to represent face data. But such traditional method of face recognition based on shallow learning cannot obtain satisfying performance in dealing with the complex and nonlinear face data. In opposite, deep learning simulates human visual perception of the nervous system, which can learn more deep and representative feature.

Face recognition generally try to figure out two problems — face identification and verification. The current face identification known as the 1:1 problem[2], focusing on the issue that whether the two images belong to the same identity. While face identification, also known as the 1:N problem[2], focusing on the issue that finding the given identity in a dataset.

We train our network on CASIA-WebFace dataset and a private dataset. CASIA-WebFace dataset includes 494414 images from 10575 identities. All the images in CASIA-WebFace dataset are collected from the internet. Our private dataset includes 365866 images from 15370 identities. We evaluate our network in LFW dataset. LFW dataset mainly test the accuracy of face recognition. LFW data set consists of more than 13000 world famous people’s face images from more than 5000 identities. All images collected in natural scenes with different directions, expressions and lighting environment.

2. Related Work

Before 2012, people usually represent a image by extracting its the texture features, such as LBP. But such method has poor performance in large scale dataset. In 2012, AlexNet was proposed by A.

Krizhevsky[22], which got the first of the ILSVRC 2012 on classification task. After that, deep learning is widely used in various fields of artificial intelligence gradually, including face recognition.

Deepface[3] is the first deep model that is trained on 4.4M face images. It has a complex preprocessing method that employs a 3D alignment for each face images. Besides, it outputs 4096-d feature, which increase the amount of computation. Deepface achieves 97.35% on the LFW, breaking the previous record.

Google proposed FaceNet[4] in 2015, introducing triple loss into CNN. Tradition methods usually take face recognition as a classification issue, which use softmax method to train model then select one layer to extract feature. Different from that, FaceNet learn an end to end encoding method of transform images to an Euclidean space, then use this code to do face recognition tasks. FaceNet is trained on a private dataset that include 200M face images from 8M identities. It achieves 99.63 % on the LFW.

Center loss[5] was proposed in 2016. It gets a center point from each classifier and sets the penalty function based on the distance between the feature and its center, which raise the intra-class compactness. Center loss achieve 99.28% on LFW.

Besides above, DeepID series[6][7][8] and other algorithms have achieved excellent performance in face recognition. However, these models are often trained on large-scale datasets, and the network architecture is relatively complex. To alleviate this issue, Xiang Wu proposed two light neural network called Light CNN[9] in 2015. Light CNN model A has only 4 convolution layers while it obtains 97.77% on LFW. Light CNN model B has 5 convolution layers while it obtains 98.13% on LFW.

3. Architecture

In this section, we first introduce Light CNN and its activation function called Max-Feature-Map (MFM)[9]. Then, we introduce our improved model.

3.1. Light CNN & MFM

Since the traditional activation functions like Sigmoid or Tanh may lead to gradient vanishing in back propagation process, some scholars have proposed the ReLU[13] activation function that have been widely used. ReLU can solve the gradient vanishing and overfitting problems for it can make feature sparse, but it may lead to the loss of some information because the value is 0 if the unit is not activated[1]. After that, some scholars have proposed activation functions such as Leaky ReLU[14], PReLU[15] and RReLU[16] to improve the ReLU's disadvantage. However, all these activation functions are combinations of simple linear functions, which cannot well represent the features in some cases[1].

To solve this problem, Xiang Wu proposed a new activation function called Max-Feature-Map (MFM). Contrary to ReLU, MFM can help to represent feature compact. As is shown is fig1, given an input convolution layer $x^n \in R^{H\times W}$, where $n = \{1,2,\dots,2N\}$, W and H denote the width and height of feature map, MFM output the element-wise maximum feature map, which can be written as

$$\hat{x}_{ij} = max(x_{ij}^{k}, x_{ij}^{k+N}) \tag{1}$$

where the channel of input convolution layer is $2N$, $1 \le k \le N$, $1 \le i \le H$, $1 \le j \le W$. we can get gradient of Eq(1)

$$\frac{\partial \hat{x}_{ij}^{k}}{\partial x^{k'}} = \begin{cases} 1, & x_{ij}^{k} \ge x_{ij}^{k+N} \\ 0, & otherwise \end{cases} \tag{2}$$

where $1 \le k' \le 2N$ and

$$k = \begin{cases} k', & 1 \le k' \le N \\ k' - N, & N+1 \le k' \le 2N \end{cases} \tag{3}$$

According to equation above, we obtain half of gradients information.

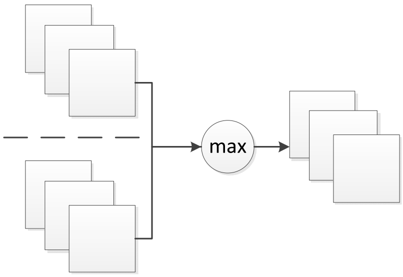


Figure1. Operation performed by MFM

3.2. A Normalized Light CNN

Normalized Light CNN consists of 5 convolution layers, 4 max pooling layers, 2 fully connected layers and a normalized layer. Especially, conv2 to conv5 layer draw lessons from Network in Network (NIN)[10].

The architecture of Light CNN model B and Normalized Light CNN are presented in Table1. Light CNN model B input gray images, which may loss some information compared with RGB images. Therefore, we choose RGB images as input.

Since the model uses the cosine similarity to measure the image relevance, in order to represent the feature more precisely, we add a normalized layer after MFM_fc1 layer. Given two feature vector $f1$， $f2$ extracted from two different images, where $f1 = (x_1, x_2, x_{3,} \dots, x_{256})$， $f2 = (y_1, y_2, y_3, \dots, y_{256})$, we can calculate their cosine similarity according to the Eq(4)：

$$cos < f1, f2 > = \frac{f1 \cdot f2}{\|f1\| \cdot \|f2\|} = \frac{\sum_{i=1}^{256} x_i y_i}{\sqrt{\sum_{i=1}^{256} x_i{}^2}\sqrt{\sum_{i=1}^{256} y_i{}^2}} \tag{4}$$

we normalized feature to between -1 and 1

$$f1_{norm} = \frac{f1}{\|f1\|} = \frac{f1}{\sqrt{\sum_{i=1}^{256} x_i{}^2}} \tag{5}$$

therefore, we can use Eq(6) to calculate the cosine similarity between two image

$$cos < f1_{norm}, f2_{norm} > = f1_{norm} \cdot f2_{norm} \tag{6}$$

The normalized feature can represent the image better.

Table 1. the architecture of Light CNN Model B & Normalized Light CNN

Light CNN Model B			Normalized Light CNN		
Type	Filter Size /Stride, Pad	Output Size	Type	Filter Size /Stride, Pad	Output Size
data	-	128×128×1	data	-	128×128×3
Conv1	5×5/1,2	128×128×96	Conv1	7×7/3,2	126×126×96
MFM1	-	128×128×48	MFM1	-	126×126×48
Pool1	2×2/2	64×64×48	Pool1	2×2/2	63×63×48
Conv2a	1×1/1	64×64×96	Conv2a	1×1/1	63×63×120
MFM2a	-	64×64×48	MFM2a	-	63×63×60

Conv2	3×3/1,1	64×64×192	Conv2	3×3/1,1	63×63×180
MFM2	-	64×64×96	MFM2	-	63×63×90
Pool2	2×2/2	32×32×96	Pool2	2×2/2	32×32×90
Conv3a	1×1/1	32×32×192	Conv3a	1×1/1	32×32×200
MFM3a	-	32×32×96	MFM3a	-	32×32×100
Conv3	3×3/1,1	32×32×384	Conv3	3×3/1,1	32×32×256
MFM3	-	32×32×192	MFM3	-	32×32×128
Pool3	2×2/2	16×16×192	Pool3	2×2/2	16×16×128
Conv4a	1×1/1	16×16×384	Conv4a	1×1/1	16×16×256
MFM4a	-	16×16×192	MFM4a	-	16×16×128
Conv4	3×3/1,1	16×16×256	Conv4	3×3/1,1	16×16×300
MFM4	-	16×16×128	MFM4	-	16×16×150
Conv5a	1×1/1	16×16×256	Conv5a	1×1/1	16×16×256
MFM5a	-	16×16×128	MFM5a	-	16×16×128
Conv5	3×3/1,1	16×16×256	Conv5	3×3/1,1	16×16×200
MFM5	-	16×16×128	MFM5	-	16×16×100
Pool4	2×2/2	8×8×128	Pool4	2×2/2	8×8×100
fc1	-	512	fc1	-	512
MFM_fc1	-	256	MFM_fc1	-	256
			Normalize	-	256
Drop1	-	-	Drop1	-	-
fc2	-	10575	fc2	-	25945

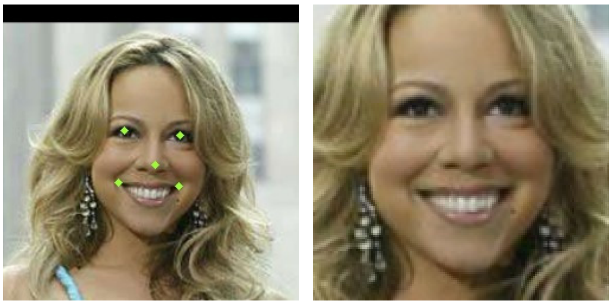
loss	-	10575	loss	-	25945

4. Experiment

In this section, we introduce data preprocessing and training methods at first. Then we evaluate our model on LFW.

4.1. Data Preprocessing

Before training or test our model, all the images data need to be processed in the same preprocessing way. First, we use Faster R-CNN[20] to obtain a face detected box. Second, we detect 5 points facial landmarks by a pre-trained CNN model[19]. Third, we align images according to the 5 points facial landmarks detected in phase2 then crop the region of face. The way we align image can overcome the pose variations in roll angle. All the output images are resized into 144×144. As is shown in Fig.2, Fig.2(a) show the 5 points facial landmarks that we detect. Fig.2(b) show the final image after being aligned and cropped, where we can see that both eyes and corners of the mouth are rotated to the same horizontal line.



(a) **(b)**

Figure2. (a) is the 5 points facial landmarks detection results and
(b) is the final image after being preprocessed

4.2. Training Methodology

We use open source deep learning framework Caffe[17] to train our model. We divide images of each identity into train set and validation set at ratio of 39:1 randomly and make sure each validation set has at least one image.

The input of our model is the 144×144 RGB images. In training phase, all the images will be cropped into 128×128 randomly. Besides, we use mirror and shuffle method to augment the dataset. In case of overfitting, we set dropout layer after fully connected layer and the ratio is set to 0.5. Moreover, we use Adam[21] method to train the model. The momemtum and momemtum2 are set to 0.9 and 0.999. The base learning rate is set to 10e-6. The parameter initialization for each convolution is Xavier.

4.3. Experiment Results

We evaluate our model on LFW verification. In LFW verification task, 10 given group of face pairs are verified to tell if they are from the same person. Each group contains 300 pairs images from the same identity while the other 300 pairs images from the different[18].

As is shown in Table 2, our model achieve 98.46%. It can be seen that both FaceNet and DeepID series have better performance than our model, but all these models are trained on a large scale dataset and have a complex and deep architecture. By contrast, Light CNN series models only have 5 convolution layers at most, while it can achieve the similar accuracy as many other large networks. The Light CNN Model B even exceed DeepFace, which is known as the first deep model trained on a large-scale dataset. Our model improve the feature representation of Light CNN, which raise the accuracy on LFW verification.

Table 2. Comparision with other methods on the LFW verification

Method	Accuracy	TPR@FAR=0.1%	Protocol
High-dim LBP	95.17%	-	unrestricted
TL Joint Bayesian	96.33%		unrestricted
DeepFace	97.35%	-	unrestricted
Web-Scale	98.37%	-	unrestricted
WebFace + Joint Bayes	97.73%	80.26%	unrestricted
VGG	97.27%	81.90%	unsupervised
DeepID	97.45%	-	unsupervised
DeepID2	99.15%	-	unsupervised
DeepID3	99.53%	-	unsupervised
FaceNet	99.63%	-	unrestricted
Light CNN Model A	97.77%	84.37%	unsupervised
Light CNN Model B	98.13%	87.13%	unsupervised
Normalized Light CNN	98.46%	90.05%	unsupervised

5. Conclusion

In this paper, we propose a normalized Light CNN, which achieve the accuracy 98.46% on LFW. Although don’t have a deep architecture, Light CNN can achieve comparable performance to other large networks. Our model cost 110ms on extracting feature from a image on a single core i7-6700K, which is faster than most other network.

Acknowledgment

This paper is supported by Beijing Key Laboratory of Work Safety Intelligent Monitoring (Beijing University of Posts and Telecommunications).

References

[1] Wu X, He R, Sun Z, et al. A Light CNN for Deep Face Representation with Noisy Labels[J]. Computer Science, 2017

[2] Klare B F, Klein B, Taborsky E, et al. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2015:1931-1939.

[3] Taigman Y, Yang M, Ranzato M, et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2014:1701-1708.

[4] Schroff F, Kalenichenko D, Philbin J. FaceNet: A unified embedding for face recognition and clustering[J]. 2015:815-823.

[5] Wen Y, Zhang K, Li Z, et al. A Discriminative Feature Learning Approach for Deep Face Recognition[M]// Computer Vision – ECCV 2016. Springer International Publishing, 2016:499-515.

[6] Ouyang W, Luo P, Zeng X, et al. DeepID-Net: multi-stage and deformable deep convolutional neural networks for object detection[J]. Eprint Arxiv, 2014.

[7] Ouyang W, Wang X, Zeng X, et al. DeepID-Net: Deformable deep convolutional neural networks for object detection[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2015:2403-2412.

[8] Ouyang W, Zeng X, Wang X, et al. DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2016, PP(99):1-1.
[9] Wu X, He R, Sun Z, et al. A Light CNN for Deep Face Representation with Noisy Labels[J]. Computer Science, 2015.
[10] Lin M, Chen Q, Yan S. Network In Network[J]. Computer Science, 2013.
[11] Wang X, Han T X, Yan S. An HOG-LBP human detector with partial occlusion handling[C]// IEEE, International Conference on Computer Vision. IEEE, 2010:32-39.
[12] Luo J, Ma Y, Takikawa E, et al. Person-Specific SIFT Features for Face Recognition[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2007:II-593 - II-596.
[13] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]// International Conference on International Conference on Machine Learning. Omnipress, 2010:807-814.
[14] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In Proceedings of the 30th International Conference on Machine Learning, volume 30, 2013.
[15] He K, Zhang X, Ren S, et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification[J]. 2015:1026-1034.
[16] Xu B, Wang N, Chen T, et al. Empirical Evaluation of Rectified Activations in Convolutional Network[J]. Computer Science, 2015.
[17] Jia, Yangqing, Shelhamer, et al. Caffe: Convolutional Architecture for Fast Feature Embedding[J]. 2014:675-678.
[18] Huang G B, Mattar M, Berg T, et al. Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments[J]. Month, 2008.
[19] Sun Y, Wang X, Tang X. Deep Convolutional Network Cascade for Facial Point Detection[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2013:3476-3483.
[20] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]// International Conference on Neural Information Processing Systems. MIT Press, 2015:91-99.
[21] Kingma D, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.
[22] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012:1097-1105.