A Lightened CNN for Deep Face Representation

Xiang Wu
School of Computer and Communication Engineering
University of Science and Technology Beijing, Beijing, China
aflredxiangwu@gmail.com

Ran He, Zhenan Sun
National Laboratory of Pattern Recognition
Institute of Automation Chinese Academy of Sciences, Beijing, China
{rhe, znsun}@nlpr.ia.ac.cn

Abstract

Convolution neural network (CNN) has significantly pushed forward the development of face recognition techniques. To achieve ultimate accuracy, CNN models tend to be deeper or multiple local facial patch ensemble, which result in a waste of time and space. To alleviate this issue, this paper studies a lightened CNN framework to learn a compact embedding for face representation. First, we introduce the concept of maxout in the fully connected layer to the convolution layer, which leads to a new activation function, named Max-Feature-Map (MFM). Compared with widely used ReLU, MFM can simultaneously capture compact representation and competitive information. Then, one shallow CNN model is constructed by 4 convolution layers and totally contains about 4M parameters; and the other is constructed by reducing the kernel size of convolution layers and adding Network in Network (NIN) layers between convolution layers based on the previous one. These models are trained on the CASIA-WebFace dataset and evaluated on the LFW and YTF datasets. Experimental results show that the proposed models achieve state-of-the-art results. At the same time, a reduction of computational cost is reached by over 9 times in comparison with the released VGG model.

1. Introduction

In the last decade, convolution neural network (CNN) has become one of the most popular techniques for computer vision. Numerous vision tasks, such as image classification [5], object detection [23], face recognition [19, 24, 27], have benefited from the robust and discriminative representation learnt via CNN models. Their performances have obtained great progress and some algorithms have been applied to commercial systems. For example, the accuracy on the challenging LFW benchmark has been improved from 97% [24] to 99% [12, 16, 17, 19] by using CNN based model. Face verification is one of the most successful applications of CNN. It identifies whether two facial images are from the same person (one-to-one matching).

CNN methods for face verification can be generally categorized into three groups. The first group resorts to multi-class classification [21, 24] to extract face feature vectors and then processes these vectors by classifiers or multi-patch ensemble models. Some methods in this group are often based on strong assumptions of data distributions that may not make effect on various situations, such as Joint Bayesian[2] and Gaussian Processing [13]. Although multi-patch ensemble enhances robustness of features from multi-class classification, it is time consuming. The second group aims to directly optimize the verification loss for matching and non-matching pairs [3, 7, 17]. These methods potentially overcome the bottleneck of multi-class classification based networks, which may not be generalized for a new identity that does not exist in the training set. However, one limitation of these methods is that it is difficult to select training dataset for negative pairs and the threshold in the verification loss is manually determined. The last group employs a joint identification and verification constraint to optimize deep face models [12, 16, 19, 27]. Multi-task learning provides an efficient way to enhance the generalization ability of face representation. However, convergence is still challenge for multi-task based CNNs. The trade-off between identification and verification is manually determined and depends on the training set.

Although previous methods [12, 16, 17, 22, 25] have achieved ultimate accuracy on the LFW dataset, computational costs are still an ongoing issue, which has been a bottleneck for face recognition systems on embedding de-

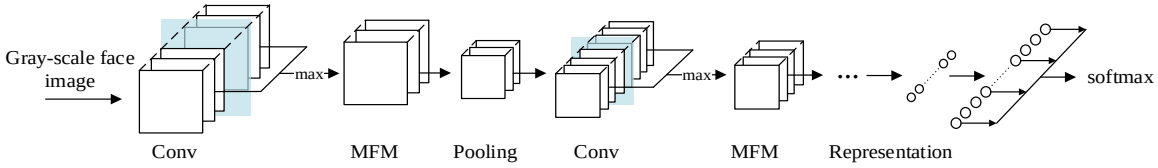


Figure 1. An illustration of the architecture of our lightened convolution networks model. The Max-Feature-Map (MFM) is the maximum between two convolution feature map candidate nodes.

vices or smart phones. First, the very deep CNN leads to a large model size and long computation time for feature extraction on CPU or GPU. Particularly, the usage of multiple facial patches requires much time and introduces uncertainty incurred by the automatic location of facial landmarks. Second, most of the state-of-the-art CNN models are based on the ReLU activation function that makes learnt features usually high dimensional and sparse. To obtain a low-dimensional and compact representation, one often utilizes Joint Bayesian [19] or metric learning [12] to reduce the learnt high-dimensional sparse features in another independent stage. Hence, it is important to directly seek a CNN model with a small size, fast speed of feature extraction and low-dimensional representation.

This paper studies a lightened CNN framework (as shown in Fig. 1) to learn a deep face representation as feature extractors. We define a Max-Feature-Map (MFM) activation function for compact representation and feature selection as an alternative of ReLU. MFM in convolution layers is a variation of the maxout operator [4] in fully connected layers. The CNN model is trained on CASIA-WebFace dataset[1] and evaluated on LFW and YTF datasets. Our proposed single net model achieves **98.13%** and **91.6%** on LFW and YTF respectively. Moreover, the CPU time is improved from 581ms[2] nearly **67ms** to extract one face image representation on a single core i7-4790. The contributions are summarized as follows:

(1) A new Max-Feature-Map (MFM) activation function is introduced to the convolution layers of CNN. Its learnt features are compact whereas ReLU learnt features are sparse and often high-dimensional. It is an approach of aggregate statistics to obtain more notable and discriminative nodes in both convolution and fully connected layers.

(2) Two lightened convolution neural networks are designed for extracting face representation of face images. One contains 4 convolution layers, 4 max-pooling layers and 2 fully connected layers and totally contains about 4M parameters, the other reduces the kernel size of convolution layers and employs Network in Network (NIN) [11] between convolution layers. These configurations not only ensure the generalization but also make great improvements on speed and storage space.

(3) The proposed lightened CNN models obtain comparable performance on the LFW and YTF datasets. The size of model file is about **20-30MB** and the CPU time of extracting face feature vector based on CNN is nearly **67ms**, which has significant potential to be deployed on real-time applications.

The paper is organized as follows. In Section 2, we briefly review the CNN based face representation methods. Section 3 describes the proposed lighten CNN framework. Finally, we present our experimental results in Section 4 and conclude in Section 5.

2. Related Work

Current methods for face recognition are often designed by CNN to obtain a robust face feature extractor. To the best of our knowledge, DeepFace [24] is the first one to train CNN by 4.4M face images as a feature extractor for face verification tasks. It employs a 3D alignment for data pre-processing and its network contains convolution and local connected layers. It achieves 97.35% accuracy on LFW with 4096-d feature vectors. As the extension of DeepFace, Web-Scale [25] applied a semantic bootstrapping method to select an efficient training set from a large dataset. It certificates that high dimensional feature vectors are not necessary for face recognition problems because the low dimensional features of Web-Scale can outperform DeepFace. And [25] also discuss stabler protocol [1] of the LFW benchmark, which can indicate the robustness of face features more representatively.

[1] http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html

[2] The CPU time is tested by VGG model which is released on http://www.robots.ox.ac.uk/ vgg/software/vgg_face/

To further improve accuracy, Sun *et al.* [19, 21] resorts to a multi-patch ensemble model. An ensemble of 25 CNN models is trained on different local patches and Joint Bayesian is applied to obtain a robust embedding space. Verification loss and classification loss are further combined to increase the interclass distance and decrease the intraclass distance. Compared with DeepFace, they did not use 3D face alignment and trained multiple CNN models on 0.2M face images. Their final performance obtains 99.47% [22] on LFW.

Recently, triplet loss is introduced into CNN, which leads to a new method named FaceNet [17]. FaceNet is trained on totally about 100-200M face images with 8M face identities. Since the selection of triplet pairs is important for accuracy, FaceNet presents a novel online triplet mining method for training triplet-based CNN and achieves good performance (99.63%). Then Parkhi *et al.* [16] combined the very deep convolution neural network [18] and the triplet embedding. The input of their model is 224×224 as imagenet task. They trained the CNN model on 2622 identities of 2.6M images collected from Internet and then finetuned the model via triplet-based metric learning method like FaceNet. The classification-based net obtains 97.27% and the deep embedding model achieves 98.95% on LFW.

Besides, by taking advantages of multiple CNNs and triple-based method, Liu *et al.* [12] proposed a two-stage algorithm that combined multi-patch deep CNN and deep metric learning. They cropped 7 local patches from face images according to the facial landmarks and trained CNN models, contained 7 convolution layers, separately. Then a deep metric learning algorithm on triplets is used to reduce the dimension of features and enhance discrimination. The ensemble model is targeting the accuracy of 99.77%.

Although these CNN based methods have achieved ultimate accuracy on the LFW dataset, their computational costs are still high due to deep architectures and multiple local facial patches. Hence, it is important to seek a CNN model with a small size, fast speed of feature extraction and low dimensional representation for real-time applications.

3. Architecture

In this section, we first define the compact Max-Feature-Map activation function for CNN and then introduce the framework of our deep face representation models.

3.1. Max-Feature-Map Activation Function

Sigmoid or Tanh activation function is a nonlinear activation for neural networks and often leads to robust optimization during DNN training [6]. But it may suffer from a vanishing gradient when lower layers have gradients of nearly 0 because higher layer units are nearly saturate at -1 or 1. The vanishing gradient may lead to a slow convergence or a poor local optima for convolution neural networks.

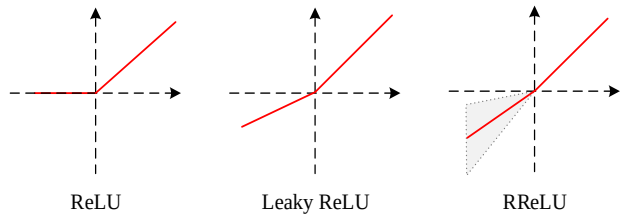


Figure 2. Comparison of ReLU, Leaky ReLU, PReLU and RReLU. For Leaky ReLU, the slope for $x \leq 0$ is fixed while PReLU is learnt from data. And the slope for RReLU is a random variable sampled from uniform distribution.

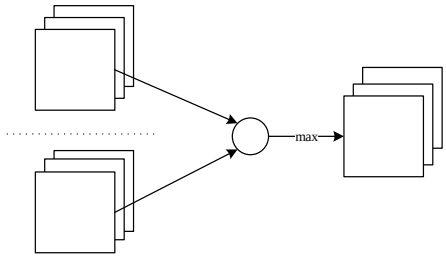


Figure 3. Operation performed by Max-Feature-Map activation function.

To overcome vanishing gradient, the Rectified linear unit(ReLU) [15] offers a sparse representation. However, ReLU is at a potential disadvantage during optimization because the value is 0 if the unit is not active. It might lead to the loss of some information especially for the first several convolution layers because these layers are similar to Gabor filter which both positive and negative responses are respected.

To alleviate this problem, the leaky rectified linear (Leaky ReLU) [14], parametric rectified linear (PReLU) [5] and randomized rectified linear (RReLU) [26] are proposed. Fig. 2 shows the difference among them. However, these activation functions are simple piece-wise linear activation functions, therefore, it can not represent the features efficiently in some cases.

In order to make the representation compact instead of sparsity in ReLU, we propose the Max-Feature-Map(MFM) activation function which is inspired by maxout networks [4]. Given an input convolution layer $C \in \mathbb{R}^{h \times w \times 2n}$, as is shown in Fig. 3, the Max-Feature-Map activation function can be written as

$$f_{ij}^{k} = \max_{1 \leq k \leq n} (C_{ij}^{k}, C_{ij}^{k+n}) \tag{1}$$

where the channel of the input convolution layer is $2n$, $1 \leq i \leq h$, $1 \leq j \leq w$. As is shown in Eq.(1), the output f via MFM activation function belongs to $\mathbb{R}^{h \times w \times n}$.

According to the Eq.(1), the gradient of this activation

Table 1. The architectures of the lightened CNN model A and B.

A				B			
Name	Filter Size /Stride	Output Size	#param	Name	Filter Size /Stride, Pad	Output Size	#param
input	-	$144 \times 144 \times 1$	-	input	-	$144 \times 144 \times 1$	-
crop	-	$128 \times 128 \times 1$	-	crop	-	$128 \times 128 \times 1$	-
conv1_1	$9 \times 9/1$	$120 \times 120 \times 48$	3.8K	conv1_1	$5 \times 5/1, 2$	$128 \times 128 \times 48$	1.2K
conv1_2	$9 \times 9/1$	$120 \times 120 \times 48$	3.8K	conv1_2	$5 \times 5/1, 2$	$128 \times 128 \times 48$	1.2K
mfm1	-	$120 \times 120 \times 48$	-	mfm1	-	$128 \times 128 \times 48$	-
pool1	$2 \times 2/2$	$60 \times 60 \times 48$	-	pool1	$2 \times 2/2$	$64 \times 64 \times 48$	-
				conv2_a	$1 \times 1/1$	$64 \times 64 \times 48$	0.04K
conv2_1	$5 \times 5/1$	$56 \times 56 \times 96$	2.4K	conv2_1	$3 \times 3/1, 1$	$64 \times 64 \times 96$	0.8K
conv2_2	$5 \times 5/1$	$56 \times 56 \times 96$	2.4K	conv2_2	$3 \times 3/1, 1$	$64 \times 64 \times 96$	0.8K
mfm2	-	$56 \times 56 \times 96$	-	mfm2	-	$64 \times 64 \times 96$	-
pool2	$2 \times 2/2$	$28 \times 28 \times 96$	-	pool2	$2 \times 2/2$	$32 \times 32 \times 96$	-
				conv3_a	$1 \times 1/1$	$32 \times 32 \times 96$	0.09K
conv3_1	$5 \times 5/1$	$24 \times 24 \times 128$	3.2K	conv3_1	$3 \times 3/1, 1$	$32 \times 32 \times 192$	1.7K
conv3_2	$5 \times 5/1$	$24 \times 24 \times 128$	3.2K	conv3_2	$3 \times 3/1, 1$	$32 \times 32 \times 192$	1.7K
mfm3	-	$24 \times 24 \times 128$	-	mfm3	-	$32 \times 32 \times 192$	-
pool3	$2 \times 2/2$	$12 \times 12 \times 128$	-	pool3	$2 \times 2/2$	$16 \times 16 \times 192$	-
				conv4_a	$1 \times 1/1$	$16 \times 16 \times 192$	0.19K
conv4_1	$4 \times 4/1$	$9 \times 9 \times 192$	3K	conv4_1	$3 \times 3/1, 1$	$16 \times 16 \times 128$	1.1K
conv4_2	$4 \times 4/1$	$9 \times 9 \times 192$	3K	conv4_2	$3 \times 3/1, 1$	$16 \times 16 \times 128$	1.1K
mfm4	-	$9 \times 9 \times 192$	-	mfm4	-	$16 \times 16 \times 128$	-
pool4	$2 \times 2/2$	$5 \times 5 \times 192$	-	pool4	$2 \times 2/2$	$8 \times 8 \times 128$	-
				conv5_a	$1 \times 1/1$	$8 \times 8 \times 128$	0.12K
				conv5_1	$3 \times 3/1, 1$	$8 \times 8 \times 128$	1.1K
				conv5_2	$3 \times 3/1, 1$	$8 \times 8 \times 128$	1.1K
				mfm5	-	$8 \times 8 \times 128$	-
				pool5	$2 \times 2/2$	$4 \times 4 \times 128$	-
fc1	-	256	1,228K	fc1	-	256	524K
fc2	-	10,575	2,707K	fc2	-	10,575	2,707K
loss	-	10,575	-	loss	-	10,575	-
total			3,961K				3,244K

function can be shown as

$$\frac{\partial f}{\partial C^{k'}} = \begin{cases} 1, \text{ if } C_{ij}^{k} \geq C_{ij}^{k+n} \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

where $1 \leq k^{'} \leq 2n$ and

$$k = \begin{cases} k^{'} & 1 \leq k^{'} \leq n \\ k^{'} - n & n+1 \leq k^{'} \leq 2n \end{cases} \tag{3}$$

As is shown in Eq.(3), the 50% gradients of the activation layers are 0. Therefore, the MFM activation function can get sparse gradients which reflect the variance of the data conditioned on response variables.

The MFM activation function is not a normal single-input-single-output function such as sigmoid or ReLU, while it is the maximum between two convolution feature map candidate nodes. The MFM activation function utilizes an approach of aggregate statistics and can not only obtain a compact representation, but also obtain sparse gradients which achieve variable selection and dimension reduction. Moreover, the MFM activation function can also be treated as the sparse connection between two convolution layers, which encodes the information sparsely into a feature space.

3.2. The Lightened CNN Framework

For our proposed lighten CNN-based framework shown in Fig. 1, we discuss two different architectures. The deep model A is constructed by 4 convolution layers, Max-Feature-Map activation functions, 4 max-pooling layers and 2 fully connected layers which is inspired from AlexNet [10]. The detail parameter setting is presented in Table 1.

The deep model B contains 5 convolution layers, 4 Network in Network (NIN) layers [11], Max-Feature-Map activation functions, 4 max-pooling layers and 2 fully con-

(a) (b)

Figure 4. Face image alignment for WebFace dataset. (a) is the facial points detection results and (b) is the normalized face image.

Table 2. The setting of face normalization for training and testing datasets. The ec_mc_y stands for the distance between the midpoint of eyes and the midpoint of mouth and the y axis of midpoint of eyes is denoted as ec_y.

Dataset	size	ec_mc_y	ec_y
CASIA-WebFace	144×144	48	48
LFW	128×128	48	40
YTF	128×128	48	40

nected layers which the detail is presented in Table 1. The model B with NIN and small convolution kernel size are from [18] because NIN can do feature selection between convolution layers and small convolution kernel can reduce the number of parameters for the model.

The input image is 144×144 gray-scale face image from CASIA-WebFace dataset. We crop each input image randomly into 128×128 patch as the input of the first convolution layer for training. Each convolution layer of network A and B is combined with two independent convolution parts calculated from the input. The Max-Feature-Map activation function and max pooling layer are used later. The fc1 layer is a 256-dimensional face representation. And the fc2 layer is used as the input of the softmax cost function and is set to the number of CASIA-WebFace dataset identities(10,575). Besides, the proposed network A has 3,961K parameters and network B has 3,244K parameters which are both smaller than DeepFace, WebFace and VGG models.

4. Experiments

In this section, we evaluate our lightened CNN models on the LFW and YTF datasets. Besides, we analyze the performance of MFM activation and the speed of our proposed models compared with other CNN methods. The network A model is released on my github[3] and the network B model will be released soon.

4.1. Data Pre-processing

The CASIA-WebFace dataset is used to train our lightened convolution neural network. It contains 493,456 face images of 10,575 identities and all the face images are converted to gray-scale and normalized to 144×144 via landmarks as shown in Fig. 4(a). The normalized face image is shown in Fig. 4(b). According to the 5 facial points extracted by [20] and manually adjusted, we rotate two eye points horizontally which can overcome the pose variations in roll angle. The distance between the midpoint of eyes and the midpoint of mouth, as well as the y axis of midpoint of eyes, will be used for facial image normalization, because the distance between the midpoint of eyes and the midpoint of mouth is relative invariant to pose variations in yaw angle.

The evaluation is performed on the LFW dataset[4] and the YTF dataset[5] in detail. LFW contains 13,233 images of 5,749 people for face verification and YTF contains 3,425 videos of 1,595 different people. And all the images in the LFW and YTF datasets are processed by the same pipeline as the training dataset and normalized to 128×128. The details of face image normalization setting is shown in Table 2

4.2. Training Methodology

To train the convolution network A and B, we randomly select one face image from each identity as the validation set and the other images as the training set. The open source deep learning framework *Caffe* [9] is used for training the model.

The input for the lightened CNN models is the 144×144 gray-scale face image and we crop the input image into 128×128 randomly and mirror it. These data augmentation methods can improve the generalization of the convolution neural network and overcome the overfitting [10]. Dropout is also used for fully connected layers and the ratio is set to 0.7.

Moreover, the momentum is set to 0.9, and the weight decay is set to 5e-4 for convolution layer and fully connected layer except the fc2 layer. It is obvious that the fc1 fully connected layer is the face representation used for face verification tasks. However, the parameters from fc1 layer to fc2 layer is very large and is not used for feature extractor. Therefore, it might lead to overfitting for learning the large fully-connected layer parameters. To overcome it, we set the weight decay of fc2 layer to 5e-3.

The learning rate is set to 1e-3 initially and reduced to 5e-5 gradually. The parameter initialization for convolution is Xavier and Gaussian is used for fully-connected lay-

[3] https://github.com/AlfredXiangWu/face_verification_experiment
[4] http://vis-www.cs.umass.edu/lfw/
[5] http://www.cs.tau.ac.il/ wolf/ytfaces/

Table 3. Comparison with other state-of-the-art methods on the LFW verification and identification protocol.

Method	#Net	Accuracy	TPR@FAR=0.1%	Protocol	Rank-1	DIR@FAR=1%
DeepFace [24]	1	95.92%	-	unsupervised	-	-
DeepFace [24]	7	97.35%	-	unrestricted	-	-
Web-Scale [25]	1	98.00%	-	unrestricted	82.1%	59.2%
Web-Scale [25]	4	98.37%	-	unrestricted	82.5%	61.9%
DeepID2 [19]	1	95.43%	-	unsupervised	-	-
DeepID2 [19]	4	97.75%	-	unsupervised	-	-
DeepID2 [19]	25	98.97%	-	unsupervised	-	-
WebFace [27]	1	96.13%	-	unsupervised	-	-
WebFace+PCA [27]	1	96.30%	-	unsupervised	-	-
WebFace+Joint Bayes [27]	1	97.30%	-	unsupervised	-	-
WebFace+Joint Bayes [27]	1	97.73%	80.26%	unrestricted	-	-
FaceNet [17]	-	99.63%	-	unrestricted	-	-
VGG [16]	1	97.27%	81.90%	unsupervised	74.10%	52.01%
Our model A	1	97.77%	84.37%	unsupervised	84.79%	63.09%
Our model B	1	**98.13%**	**87.13%**	unsupervised	**89.21%**	**69.46%**

ers. Moreover, the deep model is trained on GTX980 for 2 weeks.

4.3. Results on the LFW Benchmark

We evaluate our lightened CNN model A and B on LFW with the verification protocol(1:1) [8] and the probe-gallery identification protocol(1:N) [1].

For the verification protocol, face images are divided in 10 folds which contain different identities and 600 face pairs. There are two evaluation settings about LFW training and testing: restricted and unrestricted. In restricted setting, the pre-defined image pairs are fixed by the author [8] (each fold contains the 5400 pairs for training and the 600 pairs for testing). And in unrestricted setting, the identities within each fold for training is allowed to be much larger.

For the probe-gallery identification testing, there are two new protocols called the close set task and the open set task on LFW. (i) For the close set identification task, the gallery contains 4,249 identities, each with only a single face image, and the probe set contains 3,143 face images belonging to the same set of identities. The performance is measured by Rank-1 identification accuracy. (ii) For the open set identification task, the gallery set includes 3,143 images of 596 identities. The probe set includes 10,090 images which is constructed by 596 genuine probes and 9,494 impostor ones. The accuracy is evaluated by the Rank-1 Detection and Identification Rate (DIR), which is genuine probes matched in Rank-1 at a 1% False Alarm Rate (FAR) of impostor ones that are not rejected.

We test the performance of our model A and B with cosine similarity. As shown in Table 3, the model A achieves **97.77%** and model B obtains **98.13%** on the LFW dataset with unsupervised setting[6]. The results of our model A and B on LFW verification protocol outperform those of DeepFace [24], DeepID2 [19], WebFace [27] and VGG [16] for **single net**. For the probe-gallery identification protocol, our model performance also outperforms the results of DeepFace and VGG[7] with **89.21%** and **69.46%**, respectively.

Besides, our dataset is inferior to DeepFace, Google and VGG. Their training sets contain millions of images while CASIA-WebFace only includes 0.5M images. Considering the limitation of the GPU computation resources and the speed on practical systems, we don't train several networks and perform model ensemble to improve face verification performance.

4.4. Results on the YTF Benchmark

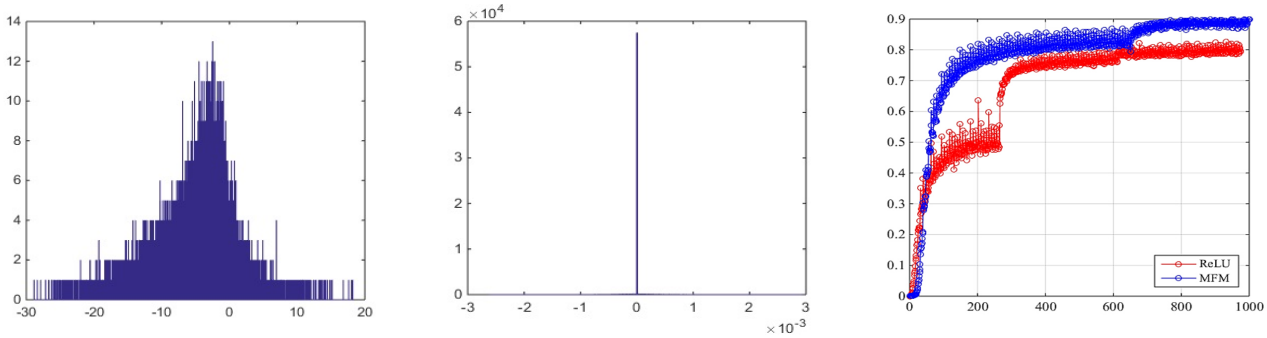
To evaluate the generalization of our lightened CNN models, we also test them on the Youtube Face dataset. Due to low resolution and motion blur, the quality of images in the YTF dataset is worse than LFW. For the evaluation protocol, YTF dataset is divided into 10 splits. Each split includes 250 positive pairs and 250 negative ones. We randomly select 100 samples from each video and compute the average similarities. As shown in Table 4, we obtain **91.6%** on YTF dataset with unsupervised setting for a single model which outperform the results of DeepFace and WebFace.

4.5. MFM Activation Function Analysis

The Max-Feature-Map (MFM) activation function is proposed to alleviate the disadvantage of the simple piecewise functions such as ReLU and its variations. Fig. 5 presents the example of MFM activation functions over our lightened CNN models. It is obvious that the network can

[6]The unsupervised setting means that the model is not trained on LFW in supervised way.

[7]The VGG performance on the probe-gallery identification protocol are evaluated based on VGG released model.



(a) (b) (c)

Figure 5. The performance of the MFM activation function. (a) The histogram of MFM values; (b) The histogram of MFM gradient values; (c)Comparison with ReLU and Max-Feature-Map activation functions in terms of validation accuracy for CNN training.

Table 4. Comparison with other state-of-the-art methods on YTF.

Method	#Net	Accuracy	Protocol
DeepFace [24]	1	91.40%	supervised
WebFace [27]	1	88.00%	unsupervised
WebFace+PCA [27]	1	90.60%	unsupervised
VGG [16]	1	92.80%	unsupervised
Our model A	1	90.72%	unsupervised
Our model B	1	91.60%	unsupervised

obtain a compact representation from the MFM activation function as shown in Fig. 5, while the gradient of MFM is sparse.

Due to the sparse gradient, on the one hand, when doing backpropagation for training CNN, the processing of stochastic gradient descent (SGD) can only make effects on the neuron of response variables. On the other hand, when extracting features for testing, the MFM can obtain more competitive nodes from previous convolution layers by activating the maximum of two feature maps. These appearances result in the properties that MFM can realize feature selection and sparse connection in our lightened CNN models.

Compared with ReLU and Max-Feature-Map, we observe in Fig. 5(c) that the speed of convergence for Max-Feature-Map network is slower than that for ReLU due to the complexity of the activation and the randomness of initial parameters. However, with the progress of training, the validation accuracy for Max-Feature-Map outperforms ReLU finally. Table 5 presents the performance of ReLU and MFM on LFW. Although there are little difference between ReLU and MFM on verification accuracy, the identification performance of MFM has been improved about 4%-5%, compared with ReLU.

Table 5. Comparison with other state-of-the-art method on LFW verification and identification protocol.

Method	Accuracy	Rank-1	DIR@FAR=1%
A(ReLU)	97.45%	78.79%	59.09%
B(ReLU)	97.73%	84.19%	62.92%
A(MFM)	97.77%	84.79%	63.09%
B(MFM)	**98.13%**	**89.21%**	**69.46%**

Table 6. Comparison between our model and VGG released model. The speed is tested on a single core i-7 4790. The storage space is measured by the size of model generated by Caffe.

Model	#Parameters	Times	Storage Space
VGG	27749K	581ms	553MB
A	3,961K	71ms	26.0MB
B	3,244K	67ms	31.2MB

4.6. Performance on Speed and Storage Space

Since most of CNN based face representation extractors are based on either the very deep convolution neural networks or multi-patch ensemble, they may be time-consuming for practical systems. In this section, we compare our lightened CNN models with the VGG released model.

As shown in Table 6, the size of our lightened CNN model is 20 times smaller than that of the VGG model, while the CPU time is about 9 times faster. The results indicate that our lightened models are potentially suitable and practical on embedding devices and smart phones for real-time applications than other complex models.

5. Conclusions

In this paper, we have developed a lightened convolution neural network framework to learn a robust face representation. We have proposed a Max-Feature-Map activation function to obtain a compact low-dimensional face representation and achieve the accuracy **98.13%** on LFW by only using a single model. One advantage of our frame-

work is that it is faster and smaller than other CNN methods. Its time to extract one face image representation is about **67ms** on a single core i7-4790, and its storage only occupies **31.2MB** on hard disk. Experimental results on LFW and YTF datasets show that MFM is effective for learning compact features and the proposed lightened CNN framework has practical value for real-time face recognition systems. Future work will focus on better performance of lightened CNN models.

References

[1] L. Best-Rowden, H. Han, C. Otto, B. Klare, and A. K. Jain. Unconstrained face recognition: Identifying a person of interest from a media collection. *IEEE Transactions on Information Forensics and Security*, 9(12):2144–2157, 2014. 2, 6

[2] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Proceedings of the 12th European conference on Computer Vision-Volume Part III*, pages 566–579, 2012. 1

[3] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. Learning deep face representation. *CoRR*, abs/1403.2802, 2014. 1

[4] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1319–1327, 2013. 2, 3

[5] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. 1, 3

[6] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 3

[7] J. Hu, J. Lu, and Y. Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1875–1882, 2014. 1

[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007. 6

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678, 2014. 5

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012. 4, 5

[11] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 2, 4

[12] J. Liu, Y. Deng, T. Bai, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *CoRR*, abs/1506.07310, 2015. 1, 2, 3

[13] C. Lu and X. Tang. Surpassing human-level face verification performance on LFW with gaussianface. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3811–3819, 2015. 1

[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. 3

[15] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010. 3

[16] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. *Proceedings of the British Machine Vision*, 2015. 1, 3, 6, 7

[17] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. 1, 3, 6

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3, 5

[19] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Proceedings of Advances in Neural Information Processing Systems 27*, pages 1988–1996, 2014. 1, 2, 3, 6

[20] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3476–3483, 2013. 5

[21] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10, 000 classes. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014. 1, 3

[22] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015. 1, 3

[23] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *CoRR*, abs/1412.1441, 2014. 1

[24] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 1, 2, 6, 7

[25] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2746–2754, 2015. 1, 2, 6

[26] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *CoRR*, abs/1505.00853, 2015. 3

[27] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014. 1, 6, 7