A Light CNN for Deep Face Representation with Noisy Labels

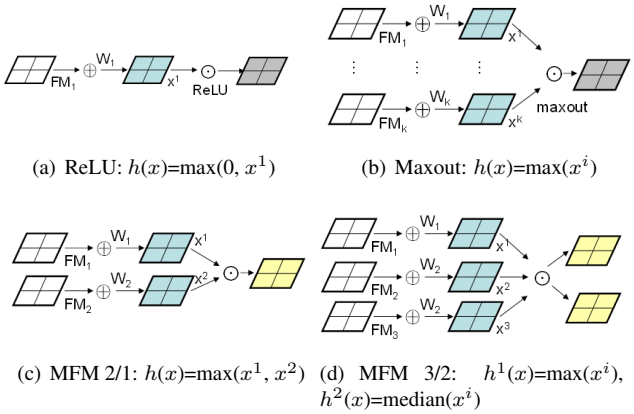
Xiang Wu, Ran He, Zhenan Sun, Tieniu Tan
Center for Research on Intelligent Perception and Computing (CRIPAC),
National Laboratory of Pattern Recognition (NLPR),
Institute of Automation, Chinese Academy of Sciences, Beijing, P. R. China, 100190
alfredxiangwu@gmail.com, {rhe, znsun, tnt}@nlpr.ia.ac.cn

Abstract

Current CNN models for face recognition tend to be deeper and larger to better fit large amounts of training data. Besides, when training data are from internet, their labels are often ambiguous and inaccurate. This paper presents a light CNN framework to learn a compact embedding on the large-scale face data with massive noisy labels. First, we introduce the concept of maxout activation into each convolutional layer of CNN, which results in a Max-Feature-Map (MFM). Different from Rectified Linear Unit that suppresses a neuron by a threshold (or bias), MFM suppresses a neuron by a competitive relationship. MFM can not only separate noisy signals and informative signals but also plays a role of feature selection. Second, three networks are carefully designed to not only obtain better performance but also reduce parameters and time-consuming. Lastly, a semantic bootstrapping method is accordingly designed to make the prediction of the models be better consistent with noisy labels. Experimental results show that the proposed framework can utilize large-scale noisy data to learn a light model in terms of both computational cost and storage space. The learnt single model with a 256-D representation achieves state-of-the-art results on five face benchmarks without fine-tuning.

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 [13], object detection [39], face recognition [44, 48, 58], have benefited from the robust and discriminative representation learnt via CNN models. Their performances have obtained great improvement, for example, the accuracy on the challenging LFW benchmark has been improved from 97% [48] to 99% [38, 41, 44]. This improvement is mainly due to the fact that CNN can learn a complex data distribution form the large-scale training datasets consisting of many identities. To achieve ultimate accuracy, the training dataset for CNN is becoming larger. Several face datasets have been published such as CASIA-WebFace [58], CelebFaces+ [44], VGG face dataset [38] and MS-Celeb-1M [11]. However, large-scale datasets often contain massive noisy labels especially when they are automatically collected from image search engines or movies.



(a) ReLU: $h(x)$=max(0, x^1) (b) Maxout: $h(x)$=max(x^i)

(c) MFM 2/1: $h(x)$=max(x^1, x^2) (d) MFM 3/2: $h^1(x)$=max(x^i), $h^2(x)$=median(x^i)

Figure 1. A comparison of different neural inhabitation. (a) ReLU suppresses a neuron by thresholding magnitude responses. (b) Maxout with enough hidden units makes a piecewise linear approximation to an arbitrary convex function. (c) MFM 2/1 suppresses a neuron by a competitve relationship. It is the simplest case of maxout activations. (d) MFM 3/2 activates two neurons and suppresses one neuron.

This paper studies a light CNN framework to learn a deep face representation from the large-scale data with massive noisy labels. As shown in Fig. 1, we define a Max-Feature-Map (MFM) operation for a compact representation and feature filter selection. MFM is an alternative of ReLU to suppress low-activation neurons in each layer. It can be considered as a special implementation of maxout activation [9] to separate noisy signals and informative signals. We implement the light CNN including MFM, small convolution filters and Network in Network, which is

trained on the MS-Celeb-1M dataset. To handle noisy labeled images, we propose a semantic bootstrapping method to automatically re-label training data by the pre-trained deep networks. We assume that the consistency of the same predictions can be made by given similar percepts. Of course, too much skepticism of the original training label may lead to a wrong relabeling. Hence, it is important to balance the trade-off between the prediction and original label. Extensive experimental evaluations demonstrate that the proposed light CNN is effective and achieves state-of-the-art results on five face benchmarks without supervised fine-tuning. The contributions are summarized as follows:

1) This paper introduces MFM operation, a special case of maxout to learn a light CNN, which has a small number of parameters. Compared to ReLU whose threshold is learned from training data, MFM adopts a competitive relationship so that it has better generalization ability and is applicable for different data.
2) The light CNNs based on MFM are designed to learn a universal face representation. We propose three light CNN architectures which are followed the ideas of AlexNet, VGG and ResNet, respectively. The proposed models lead to better performance in terms of speed and storage space.
3) A semantic bootstrapping method via a pretrained deep network is proposed to handle noisy labeled images in a large-scale dataset. Inconsistent labels can be effectively detected by the probabilities of predictions, and then are relabeled or removed for training.
4) The proposed single model with a 256-D representation obtains state-of-the-art results on various different face benchmarks, i.e., large-scale, video-based, cross-age face recognition, heterogenous and cross-view face recognition datasets. The models contain fewer parameters and extract a face representation faster than other open source face models on CPU and embedding systems.

The paper is organized as follows. In Section 2, we briefly review some related work on face recognition and noisy label problems. Section 3 describes the proposed lighten CNN framework and the semantic bootstrapping method. Finally, we present experimental results in Section 4 and conclude this paper in Section 5.

2. Related Work

2.1. Face Recognition

Current face recognition methods are often based on CNN to obtain a robust feature extractor. Earlier DeepFace [48] trains CNN on 4.4M face images and uses CNN as a feature extractor for face verification. It achieves 97.35% accuracy on LFW with a 4096-D feature vector. As an extension of DeepFace, Web-Scale [49] applies 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 it also discusses more stable protocol [2] of LFW, which can indicate the robustness of face features more representatively. To further improve accuracy, Sun *et al.* [44] 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. In [46], verification loss and classification loss are further combined to increase inter-class distance and decrease intra-class distance. The ensemble model obtains 99.47% on LFW.

Recently, triplet loss is introduced into CNN, resulting in a new method named FaceNet [41]. 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 an online triplet mining method for training triplet-based CNN and achieves good accuracy (99.63%). Then Parkhi *et al.* [38] combine the very deep convolution neural network [42] and the triplet embedding. They train the CNN model on 2622 identities of 2.6M images collected from Internet and then fine-tune 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.

The performance improvement of face recognition benefits from CNN and large-scale face datasets. However, large-scale datasets often contain massive noisy labels especially when they are automatically collected from internet. Hence, how to learn a light CNN model from the large-scale face data with massive noisy labels becomes an important issue.

2.2. Noisy Label Problems

Noisy label is an important issue in machine learning when datasets tend to be large-scale. Many methods [7] are devoted to deal with noisy label problems. These methods can generally be classified into three categories. In the first category, robust loss [1] is designed for classification tasks, so that the learnt classification models are robust to the presence of label noise. The second category [54] aims to improve the quality of training data by identifying mislabeled instances. The third category [22] directly models the distribution of noisy label during learning. The advantage of this approach allows using information about noisy labels during learning.

Recently, learning with noisy label data also draws much attention in deep learning, because deep learning is a data-driven approach and large-scale label annotation is quite expensive. Mnih and Hinton [34] introduce two robust loss functions for noisy label aerial images. However,

their method is only applicable for binary classification. Sukhbaatar *et al.* [43] consider multi-class classification for modeling class dependent noise distribution. They propose a bottom-up noise model to change the label probabilities output for back-propagation and a top-down model to change given noisy labels before feeding data. Moreover, with the notion of perceptual consistency, the work of [40] extends the softmax loss function by weakly supervised training. The idea is to dynamically update the targets of the prediction objective function based on the current model. They use a simple convex combination of training labels and the predictions of a current model to generate training targets. Although some strategies have been studied for noisy label problem, massive noisy labels are still an ongoing issue for deep learning methods.

3. Architecture

In this section, we first propose Max-Feature-Map operation for CNN to simulate neural inhibition, resulting in a new light CNN framework for face analysis and recognition. Then, the semantic bootstrapping method for noisy labeled training dataset is addressed in detail.

3.1. Max-Feature-Map Operation

Large-scale face training dataset often contains various types of noise and massive noisy labels. If the errors incurred by these noisy signals are not well treated, CNN will learn a bias result. Rectified Linear Unit (ReLU) [35] activation offers a way to separate noisy signals and informative signals. It makes use of a threshold (or bias) to determine the activation of one neuron. If the neuron is not active, its output value will be 0. However, this thresholding might lead to the loss of some information especially for the first several convolution layers because these layers are similar to Gabor filters (i.e., both positive and negative responses are respected). To alleviate this problem, the Leaky Rectified Linear Units (LReLU) [33], Parametric Rectified Linear Units(PReLU) [12] and Exponential Linear Units (ELU) [6] are proposed.

In neural science, lateral inhibition (LI) increases the contrast and sharpness in visual or audio response and aids the mammalian brain in perceiving contrast within an image. To take visual LI as an example, if an excitatory neural signal is released to horizontal cells, the horizontal cells will send an inhibitory signal to its neighboring or related cells. This inhibition produced by horizontal cells creates a more concentrated and balanced signal to cerebral cortex. To take auditory LI as an example, if certain sound frequencies create a greater contribute to inhibition than excitation, tinnitus can be suppressed. Considering LI and noisy signals, we expect the activation function in one convolution layer to have the following characters:

1) Since large-scale dataset often contains various types of noise, we expect that noisy signals and informative signals can be separated.
2) When there is a horizontal edge or line in an image, the neuron corresponding to horizontal information is excited whereas the neuron corresponding to vertical information is inhibited.
3) The inhabitation of one neuron is parameter free so that it does not depend on training data very much.

To achieve the above characters, we propose the Max-Feature-Map (MFM) operation, which is an extension of Maxout activation [9]. Different from Maxout activation that uses enough hidden neurons to approximate an arbitrary convex function, MFM suppresses only a small number of neurons to make CNN models light and robust. We define two MFM operations to obtain competitive feature maps.

Given an input convolution layer $x^n \in \mathbb{R}^{H\times W}$, where $n = \{1, ..., 2N\}$, W and H denote the spatial width and height of the feature map. The MFM 2/1 operation which combines two feature maps and outputs element-wise maximum one as shown in Fig. 1(c) can be written as

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

where the channel of the input convolution layer is $2N$, $1 \leq k \leq N, 1 \leq i \leq H, 1 \leq j \leq W$. As is shown in Eq. (1), the output $\hat{x}$ via MFM operation belongs to $\mathbb{R}^{H\times W\times N}$.

The gradient of Eq.(1) takes the following form,

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

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

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

Considering MFM 2/1, we mainly obtain 50% informative neurons from input feature maps via the element-wise maximum operation across feature channels.

Furthermore, as shown in Fig. 1(d), to obtain more comparable feature maps, the MFM 3/2 operation, which inputs three feature maps and removes element-wise minimum one, can be defined:

$$\begin{cases} \hat{x}_{ij}^{k_1} = \max(x_{ij}^k, x_{ij}^{k+N}, x_{ij}^{k+2N}) \\ \hat{x}_{ij}^{k_2} = \text{median}(x_{ij}^k, x_{ij}^{k+N}, x_{ij}^{k+2N}) \end{cases} \tag{4}$$

where $x^n \in \mathbb{R}^{H\times W}$, $1 \leq n \leq 3N, 1 \leq k \leq N$ and median() is the middle value of input feature maps. The gradient of MFM 3/2 is similar to Eq. (3), in which the value of gradient is 1 when the feature map x_{ij}^k is activated, and it tends to be 0 otherwise. In this way, we select and reserve 2/3 information from input feature maps.

3.2. The Light CNN Framework

When MFM operation is introduced to CNN, it performs a similar role of the feature selection of local features in biometrics. It selects the best feature at the same location learnt by different filters. It results in binary gradient (0 and 1) to excite or suppress one neuron during back propagation. The binary gradient plays a similar role of ordinal measure [47] that is famous local feature and widely used in biometrics.

The CNN with MFM can obtain a compact representation while the gradient of MFM layer is sparse. Due to the sparse gradient, on the one hand, when doing back propagation 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, 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 perform feature selection and sparse connection in our light CNN models.

In this section, we discuss three architectures for Light CNN. The first one is constructed by 4 convolution layers with Max-Feature-Map operations and 4 max-pooling layers like Alexnet [21] (as shown in Table 1). It contains about 4,095K parameters and about 1.5G FLOPS.

Table 1. The architectures of the Light CNN-4 model.

Type	Filter Size /Stride	Output Size	#Params
Conv1	$9 \times 9/1$	$120 \times 120 \times 96$	7.7K
MFM1	-	$120 \times 120 \times 48$	-
Pool1	$2 \times 2/2$	$60 \times 60 \times 48$	-
Conv2	$5 \times 5/1$	$56 \times 56 \times 192$	230.4K
MFM2	-	$56 \times 56 \times 96$	-
Pool2	$2 \times 2/2$	$28 \times 28 \times 96$	-
Conv3	$5 \times 5/1$	$24 \times 24 \times 256$	614K
MFM3	-	$24 \times 24 \times 128$	-
Pool3	$2 \times 2/2$	$12 \times 12 \times 128$	-
Conv4	$4 \times 4/1$	$9 \times 9 \times 384$	786K
MFM4	-	$9 \times 9 \times 192$	-
Pool4	$2 \times 2/2$	$5 \times 5 \times 192$	-
fc1	-	512	2,457K
MFM_fc1	-	256	-
Total	-	-	4,095K

Since Network in Network (NIN) [27] can potentially do feature selection between convolution layers and small convolution kernel can reduce the number of parameters like VGG [42], we integrate NIN and small convolution kernel size into the network with MFM. Finally, the 9-layer light CNN is designed, which contains 5 convolution layers, 4 Network in Network (NIN) layers, Max-Feature-Map layers and 4 max-pooling layers as shown in Table 2. The Light CNN-9 contains about 5,556K parameters and totally about 1G FLOPS which is deeper and faster than the Light CNN-4 model.

Table 2. The architectures of the Light CNN-9 model.

Type	Filter Size /Stride, Pad	Output Size	#Params
Conv1	$5 \times 5/1, 2$	$128 \times 128 \times 96$	2.4K
MFM1	-	$128 \times 128 \times 48$	-
Pool1	$2 \times 2/2$	$64 \times 64 \times 48$	-
Conv2a	$1 \times 1/1$	$64 \times 64 \times 96$	4.6K
MFM2a	-	$64 \times 64 \times 48$	-
Conv2	$3 \times 3/1, 1$	$64 \times 64 \times 192$	165K
MFM2	-	$64 \times 64 \times 96$	-
Pool2	$2 \times 2/2$	$32 \times 32 \times 96$	-
Conv3a	$1 \times 1/1$	$32 \times 32 \times 192$	18K
MFM3a	-	$32 \times 32 \times 96$	-
Conv3	$3 \times 3/1, 1$	$32 \times 32 \times 384$	331K
MFM3	-	$32 \times 32 \times 192$	-
Pool3	$2 \times 2/2$	$16 \times 16 \times 192$	-
Conv4a	$1 \times 1/1$	$16 \times 16 \times 384$	73K
MFM4a	-	$16 \times 16 \times 192$	-
Conv4	$3 \times 3/1, 1$	$16 \times 16 \times 256$	442K
MFM4	-	$16 \times 16 \times 128$	-
Conv5a	$1 \times 1/1$	$16 \times 16 \times 256$	32K
MFM5a	-	$16 \times 16 \times 128$	-
Conv5	$3 \times 3/1, 1$	$16 \times 16 \times 256$	294K
MFM5	-	$16 \times 16 \times 128$	-
Pool4	$2 \times 2/2$	$8 \times 8 \times 128$	-
fc1	-	512	4,194K
MFM_fc1	-	256	-
Total	-	-	5,556K

With the development of residual networks [13], the ultimate deep convolution neural networks are widely used and obtain high performance in various computer vision tasks. We also introduce the idea of residual blocks to light CNN and design a 29-layer convolution network for face recognition. The residual blocks contain two 3×3 convolution layers and two MFM operations without batch normalization. There are 12,637K parameters and about 3.9G FLOPS in Light CNN-29. The details of Light CNN-29 are presented in Table 3.

Note that there are some differences between our MFM operation residual blocks and the original residual blocks [13]. On the one hand, we remove batch normalization from original residual block. Although batch normalization is efficient to accelerate the convergence when training and avoid overfitting, the means and variances depend on a specific training dataset. Besides, in practice the model usually employ moving averages of minibatch means and variances, which the batch statistics may diminish when training mini-batches are small or the testing samples are independent from the training ones.

On the other hand, we employ fully connected layer instead of global average pooling layer in the end. Although global average pooling can efficiently reduce the number of parameters in the network to avoid overfitting. In our training scheme, the input images are all aligned, therefore each nodes for high-level feature maps contain not only semantic information but also spatial one which may be damaged by the global average pooling.

An input image is 144×144 gray-scale face image from the training dataset. We crop each input image randomly into 128×128 patch as the input of the first convolution layer for training. Each convolution layer is combined with two independent convolution parts calculated from its input. Max-Feature-Map layer and max pooling layer are used later. The fc1 layer is a 256-dimensional face representation. The fc2 layer is used as the input of the softmax cost function and is simply set to the number of training set identities.

3.3. Semantic Bootstrapping for Noisy Label

Bootstrapping, also called "self-training", provides a simple means for sample estimation. It is widely used to estimate a sampling distribution due to its simplicity and effectiveness. Its basic idea is that the inference about a training sample can be modeled by re-sampling and performing inference from original labeled samples to re-labeled them. It can estimate standard errors and confidence intervals for a complex data distribution and it is also appropriate to control the stability of the estimation.

Let $x \in X$ and t be the data and their labels, respectively. The CNN based on softmax loss function regresses x onto t, which its prediction can be represented as a conditional probability $p(t|f(x)), \sum_i p(t_i|f(x)) = 1$. Obviously, the maximum probability $p(t_i|f(x))$ determines the prediction label and is more confident, especially for a large number of subjects.

Due to the observations, we propose a semantic bootstrapping method to sample the training data from the large dataset with massive noisy label. The MFM operation potentially leads to robustness for the property of perceptual consistency for training because the gradient of MFM is sparse. Therefore the light CNN model can converge stably even if there are lots of the noisy labeled data in the training dataset. As stated above, firstly, we train a light CNN model on the original noisy labeled dataset. Secondly, we employ the trained model to predict the labels of noisy labeled training dataset. And then we set a threshold to decide whether accept or reject the prediction according to the conditional probabilities $p(t_i|f(x))$. Finally, we retrain the light CNN model on the re-labeled training dataset. The details for bootstrapping the MS-Celeb-1M dataset are shown in Section 4.5.

Table 3. The architectures of the Light CNN-29 model.

Type	Filter Size /Stride, Pad	Output Size	#Params
Conv1	$5 \times 5/1, 2$	$128 \times 128 \times 96$	2.4K
MFM1	-	$128 \times 128 \times 48$	-
Pool1	$2 \times 2/2$	$64 \times 64 \times 48$	-
Conv2_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 1$	$64 \times 64 \times 48$	82K
Conv2a	$1 \times 1/1$	$64 \times 64 \times 96$	4.6K
MFM2a	-	$64 \times 64 \times 48$	-
Conv2	$3 \times 3/1, 1$	$64 \times 64 \times 192$	165K
MFM2	-	$64 \times 64 \times 96$	-
Pool2	$2 \times 2/2$	$32 \times 32 \times 96$	-
Conv3_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 2$	$32 \times 32 \times 96$	662K
Conv3a	$1 \times 1/1$	$32 \times 32 \times 192$	18K
MFM3a	-	$32 \times 32 \times 96$	-
Conv3	$3 \times 3/1, 1$	$32 \times 32 \times 384$	331K
MFM3	-	$32 \times 32 \times 192$	-
Pool3	$2 \times 2/2$	$16 \times 16 \times 192$	-
Conv4_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 3$	$16 \times 16 \times 192$	3981K
Conv4a	$1 \times 1/1$	$16 \times 16 \times 384$	73K
MFM4a	-	$16 \times 16 \times 192$	-
Conv4	$3 \times 3/1, 1$	$16 \times 16 \times 256$	442K
MFM4	-	$16 \times 16 \times 128$	-
Conv5_x	$\begin{bmatrix} 3 \times 3/1, 1 \\ 3 \times 3/1, 1 \end{bmatrix} \times 4$	$16 \times 16 \times 128$	2356K
Conv5a	$1 \times 1/1$	$16 \times 16 \times 256$	32K
MFM5a	-	$16 \times 16 \times 128$	-
Conv5	$3 \times 3/1, 1$	$16 \times 16 \times 256$	294K
MFM5	-	$16 \times 16 \times 128$	-
Pool4	$2 \times 2/2$	$8 \times 8 \times 128$	-
fc1	-	512	4,194K
MFM_fc1	-	256	-
Total	-	-	12,637K

4. Experiments

In this section, we evaluate our light CNN models on various face recognition tasks. We first introduce the training methodology and databases, and then present the comparison with state-of-the-art face recognition methods, as well as algorithmic analysis and detailed evaluation. Finally, we discuss the effectiveness of the semantic bootstrapping method for selecting training dataset.

4.1. Training Methodology and Preprocessing

To train the light CNN, we randomly select one face image from each identity as the validation set and the remaining images as the training set. The open source deep learning framework *Caffe* [16] is employed to train the CNN model. Dropout is used for fully connected layers and the

(a) (b)

Figure 2. Face image alignment for training dataset. (a) is the facial points detection results and (b) is the normalized face image.

ratio is set to 0.7. The momentum is set to 0.9, and the weight decay is set to 5e-4 for convolution layers and a fully-connected layer except the fc2 layer. Obviously, the fc1 fully-connected layer contains the face representation that can be used for face verification. Note that, the number of the parameters from fc1 layer to fc2 layer is very large. But these parameters are not used for feature extractor. Therefore, they may lead to over-fit if the large fully-connected layer parameters are learnt. To overcome this over-fitting, 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 layers.

The CASIA-WebFace and MS-Celeb-1M datasets are used to train our light convolution neural networks. All face images are converted to gray-scale and normalized to 144×144 via landmarks as shown in Fig. 2(a). The normalized face image is shown in Fig. 2(b). According to the 5 facial points extracted by [45] and manual adjustment, we rotate two eye points to be horizontal, 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, are set to 48 pixels for training set, for the distance between the midpoint of eyes and the midpoint of mouth is relatively invariant to pose variations in yaw angle. Since the input 144×144 image is randomly cropped into 128×128 for the first convolution input, the y axis of midpoint of eyes is set to 48 for training and 40 for testing, respectively.

4.2. The Testing Protocols

Four face databases are used to systematically evaluate the performance of the proposed light CNN. These databases corresponds to large-scale, low-resolution and heterogeneous face recognition (or verification) tasks respectively. **Note that we do not re-train or fine-tune the light CNN model on any testing database.** That is all the training sets in the five databases are not used for training or fine-tuning. We directly extract the features of the light CNN leant on the MS-Celeb-1M dataset and compute the similarity of these features in cosine similarity.

The first testing database is the commonly used LFW dataset [15] that contains 13,233 images of 5,749 people. For the verification protocol [15], face images are divided in 10 folds which contain different identities and 600 face pairs. In unrestricted setting, the identities within each fold for training are allowed to be much larger. For the probe-gallery identification testing [2], there are two new protocols called the close set and open set identification tasks. 1) For the close set 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. 2) For the open set task, the gallery set includes 3,143 images of 596 identities. The probe set includes 10,090 images which are 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.

The Benchmark of Large-scale Unconstrained Face Recognition (BLUFR) [26] is a new benchmark for LFW evaluations, which contains both verification and open-set identification. There are 10-fold experiments, with each fold containing about 156,915 genuine matching and 46,960,863 impostor matching on average for performance evaluation. It is more challenging and generalized for LFW.

The second testing databases for video-based face recognition contain the YouTube Face (YTF) databse [55], YouTube Celebrities (YTC) [20] and Celebrity-1000 [28] that are widely used to evaluate the performance of video-based face recognition methods.

The YTF dataset contains 3,425 videos of 1,595 different people. Due to low resolution and motion blur, the quality of images in the YTF dataset is worse than LFW. For the evaluation protocol, It is divided into 10 splits. Each split includes 250 positive pairs and 250 negative ones. As in [38, 41], we randomly select 100 samples from each video and compute the average similarities.

The YTC database is composed of 1,910 videos from 47 subjects with high compression rate and large appearance variations. Following the standard evaluation protocols, the YTC testing set is divided into five-fold cross validation. Each fold contains 423 videos, where the gallery set contains 141 videos and the other is probe set.

Celebrity-1000 contains 159,726 video sequences from 1000 subjects covering various resolutions, illuminations and poses. There are two types of protocols: close-set and open-set. For the close-set protocol, the training and testing subsets contain the same identities and they are divided into four scales: 100, 200, 500 and 1000 for probe-gallery identification. For the open-set protocol, the generic training set contains 200 subjects and the other exclusive 800 subjects are used in testing stage. The probe and gallery set are used in testing stage and they are further divided into four scale:

Table 4. Comparison with other state-of-the-art methods on the LFW and YTF datasets. The unrestricted protocol follows the LFW unrestricted setting and the unsupervised protocol means the model is not trained on LFW in supervised way.

Method	#Net	Acc on LFW	VR@FAR=0	Protocol	Rank-1	DIR@FAR=1%	Acc on YTF
DeepFace [48]	7	97.35%	46.33%	unrestricted	64.90%	44.50%	91.40%
Web-Scale [49]	4	98.37%	-	unrestricted	82.50%	61.90%	-
DeepID2+ [46]	25	99.47%	**69.36%**	unrestricted	**95.00%**	**80.70%**	93.20%
WebFace [58]	1	97.73%	-	unrestricted	-	-	90.60%
FaceNet [41]	1	**99.63%**	-	unrestricted	-	-	95.10%
SeetaFace [29]	1	98.62%	-	unrestricted	92.79%	68.13%	-
VGG [38]	1	97.27%	52.40%	unsupervised	74.10%	52.01%	92.80%
CenterLoss [53]	1	98.70%	61.40%	unsupervised	94.05%	69.97%	94.90%
Light CNN-4	1	97.97%	79.20%	unsupervised	88.79%	68.03%	90.72%
Light CNN-9	1	98.80%	94.97%	unsupervised	93.80%	84.40%	93.40%
Light CNN-29	1	**99.33%**	**97.50%**	unsupervised	**97.33%**	**93.62%**	**95.54%**

100, 200, 400 and 800.

The third testing database is the very challenging MegaFace database [19] that aims at the evaluation of face recognition algorithms at million-scale. It includes probe and gallery set. The probe set is FaceScrub [36], which contains 100K images of 530 identities, and the gallery set consists of about 1 million images from 690K different subjects. As in [53], we evaluate the proposed light CNN by the provided code[1], which only tests on one of the three gallery set (set 1) for both face identification and face verification protocols.

The fourth testing databse is the CACD-VS dataset [3] that contains large variations in aging. It includes 4,000 image pairs (2000 positive pairs and 2000 negative pairs) by collecting celebrity images on Internet.

The fifth testing database is Multi-PIE [10], the largest dataset for evaluating face recognition under pose, illumination and expression variations in controlled environment. Multi-PIE contains 754,204 images of 337 identities and following the setting in [64], they use 337 subjects with neutral expression, nine pose within $\pm 60^\circ$ and 20 illuminations and the first 200 subjects are used for training and the rest 137 for testing. We don't re-train or finetune Light CNN models on Multi-PIE. Therefore, for a fair comparison, we only evaluate the Light CNN models on the rest 137 identities.

The final testing database is the largest public and challenging CASIA NIR-VIS 2.0 database [24] for heterogeneous face recognition. It consists of different modality face images. We follow the standard protocol in View 2. There are 10 fold experiments and each fold contains 358 subjects in the testing set. For testing, the gallery set contains only 358 VIS images for each subjects and the probe set consist of 6,000 NIR images from the same 358 subjects.

[1] http://megaface.cs.washington.edu/participate/challenge.html

Table 5. The performance on LFW BLUFR protocols.

Method	FAR=0.1%	DIR@FAR=1%
HighDimLBP [26]	41.66%	18.07%
WebFace [58]	80.26%	28.90%
CenterLoss [53]	93.35%	67.86%
Light CNN-4	87.21%	60.24%
Light CNN-9	97.45%	84.89%
Light CNN-29	**98.71%**	**90.42%**

4.3. Method Comparison

In this subsection, we compare the proposed light CNNs with state-of-the-art methods according to the testing protocols in Section 4.2. For one private business method, we directly report its results from its published paper. For the published method whose source code of feature extraction is available, we extract its deep features and then report its results based on these features. Since the computational cost of the feature extraction of SeetaFace [29] is high, we do not report its results on some databases. Tables 4-11 show the results of different methods on the five face databases. We have the following observations.

The proposed three light CNNs achieve better and comparable results than the competitors. Although it performs lower than several private business methods on the LFW and MegaFace, it outperforms the published methods, including VGG [38], CenterLoss [53] and SeetaFace [29]. It is one of the best published CNN methods for face recognition.

On the LFW and YTF database (as shown in Table 4), we evaluate our 4-layer, 9-layer and 29-layer light CNN models with unsupervised setting, which means our model is not trained or fine-tuned on the LFW and YTF training dataset in a supervised way. The results of our models on the LFW verification protocol are better than those of DeepFace[48], DeepID2+ [46], WebFace [58], VGG [38], CenterLoss [53] and SeetaFace [29] for **a single net**. Al-

Table 6. Comparison of Rank-1 accuracy (%) with other state-of-the-art methods on the Celebrity-1000 dataset.

Method	Close-Set				Open-Set			
	100	200	500	1000	100	200	400	800
MTJSR [61]	50.60	40.80	35.46	30.04	46.12	39.84	37.51	33.50
DELM [51]	49.80	45.21	38.88	28.83	-	-	-	-
Eigen-RER [23]	50.60	45.02	39.97	31.94	51.55	46.15	42.33	35.90
GoogleNet+AvePool [56]	84.46	78.93	77.68	73.41	84.11	79.09	78.40	75.12
GoogleNet+NAN [56]	90.44	83.33	82.27	77.17	88.76	85.21	82.74	79.87
Light CNN-4	79.68	71.48	67.95	63.19	77.04	70.50	70.38	64.61
Light CNN-9	81.27	74.37	72.96	67.71	77.82	75.84	74.54	68.90
Light CNN-29	**88.54**	**81.70**	**79.62**	**76.31**	**85.99**	**82.38**	**81.32**	**77.31**

Table 7. Average Rank-1 Accuray on YouTube Celebrities (YTC) dataset.

Method	Rank-1 Accuracy (%)
LMKML [32]	78.20
MMDML [31]	78.50
MSSRC [37]	80.75
SFSR [62]	85.74
RRNN [25]	86.60
CRG [5]	86.70
VGG [38]	93.62
Light CNN-4	88.09
Light CNN-9	91.56
Light CNN-29	**94.18**

Table 8. MegaFace performance comparison with other methods on rank-1 identification accuracy with 1 million distractors and verification TAR for 10^{-6} FAR.

Method	Rank-1	VR@FAR=10^{-6}
NTechLAB	73.300%	85.081%
FaceNet v8 [41]	70.496%	86.473%
Beijing Faceall Co.	64.803%	67.118%
3DiVi Company	33.705%	36.927%
Barebones_FR	59.363%	59.036%
CenterLoss [53]	65.234%	76.510%
Light CNN-4	60.236%	62.341%
Light CNN-9	67.109%	77.456%
Light CNN-29	**73.494%**	**84.731%**

though several business methods have achieved ultimate accuracy on 6000-pairs face verification task, the more practical criterion may be the verification rate at the extremely low false acceptance rate (eg., VR@FAR=0). We achieve **97.50%** at VR@FAR=0 for 29 layer light CNN, while other methods' results are lower than 70%. Moreover, open-set identification rate at low false acceptance rate is even more challenging but applicable. As shown in Table 3, we obtain **93.62%** which outperforms DeepFace, DeepID2+, SeetaFace, CenterLoss and VGG. These results suggest that the MFM operations are effective for different general CNN architectures and our light CNNs learn more discriminative embedding than other CNN methods.

On the BLUFR protocols (as shown in Fig. 5), the Light CNN obtains **98.71%** on TPR@FAR=0.1% for face verification and **90.42%** on DIR@FAR=1% for open-set identification, which also outperforms other state-of-the-art methods.

Due to low resolution and motion blur, the quality of images, the quality of images in YTF is worse than LFW. The light CNN-29 obtains **95.54%** without fine-tuning on YTF by using a single model, which outperforms the results other state-of-the-art methods, such as DeepFace, DeepID2+, WebFace, FaceNet, SeetaFace, VGG and CenterLoss.

As is shown in Table 7, we compare the Light CNN with other video-based face recognition methods such as Localized Multi-Kernel Metric Learning (LMKML) [32], Multi-Manifold Deep Metric Learning (MMDML) [31], Mean Sequence Sparse Representation-based Classification (MSSRC) [37], Simultaneous Feature and Sample Reduction (SFSR) [62], Recurrent Regression Neural Network (RRNN) [25], Covariate-Relation Graph (CRG) [5] and VGG [38]. Obviously, the Light CNN-29 obtains **94.18%** rank-1 accuracy which outperforms other state-of-the-art methods.

We evaluate the performance of the Light CNN models on Celebrity-100, compared with the state-of-the-art methods including Multi-task Joint Sparse Representation (MTJSR) [61], Eigen Probabilistic Elastic Part (Eigen-PEP) [23], Deep Extreme Learning Machines (DELM) [51] and Neural Aggregation Network (NAN) [56]. In Table 6, the Light CNN-29 outperforms other state-of-the-art methods such as MTJSR [61], DELM [51], Eigen-RER [23] and GoogleNet+AvePool [56] on both close-set and open-set protocols. Although, the performance of Light CNN-29 is lower than GoogleNet+NAN [56], this is because the Light CNN models are not trained on Celebrity-1000 and we only employ average pooling along each feature dimension for aggregation as described in [56].

On the challenge MegaFace database (as shown in Ta-

Table 9. Accuracy of different methods on CACD-VS.

Method	Accuracy
HD-LBP [4]	81.60%
HFA [8]	84.40%
CARC [3]	87.60%
VGG [38]	96.00%
SeetaFace [29]	95.50%
Light CNN-4	95.50%
Light CNN-9	97.95%
Light CNN-29	**98.55%**
Human, Average	85.70%
Human, Voting	94.20%

Table 10. Comparison of Rank-1 accuracy (%) with other state-of-the-art methods on the Multi-PIE dataset.

Method	±15°	±30°	±45°	±60°
Zhu et al. [63]	90.7	80.7	64.1	45.9
Zhu et al. [64]	92.8	83.7	72.9	60.1
Kan et al. [18]	100	**100**	90.6	85.9
Yin et al. [60]	99.2	98.0	90.3	92.1
Yim et al. [59]	95.0	88.5	79.9	61.9
Tran et al. [50]	94.0	90.1	86.2	83.2
Light CNN-4	90.1	78.1	59.9	25.4
Light CNN-9	92.1	78.2	63.6	35.7
Light CNN-29	**100**	99.9	**99.6**	**95.0**

ble 8), we compare the light CNNs against FaceNet [41], NTechLAB, CenterLoss [53], Beijing Faceall Co., Barebones_FR and 3DiVi Company. To improve the robustness of our algorithm, we extract the features for each image and its horizontal mirror one, and then concatenate them as the representation followed by [53]. The 29-layer light CNN achieves **73.494%** on rank-1 accuracy and **84.731%** on VR@FAR=10^{-6} which outperforms Barebones_FR, CenterLoss, Beijing Faceall Co. and 3DiVi Company. Besides, the light CNNs obtains equivalent results compared with some commercial face recognition systems such as Google FaceNet and NTechLAB. Note that they achieve better performance than our model due to the large-scale private training dataset (500M for Google and 18M for NTechLAB) and unknown preprocessing techniques.

Table 9 shows the results on the CACD-VS dataset. The results of our models on CACD-VS is **98.55%** and outperform other age-invariant face recognition algorithms [3, 4, 8] and two open source models [38, 29]. This indicates that our light CNN is potentially robust for age-variant problems.

We also compare the Light CNNs with multi-view face recognition methods [18, 63, 64, 60] and pose-aware face image synthesis [50, 59] in Table 10. It is obvious that the Light CNN-29 obtains great performance on Multi-PIE, which the accuracy on ±60° is about **95.0%**. Note that all the comparison methods are trained on Multi-PIE, while the

Table 11. Rank-1 accuracy and VR@FAR=0.1% of different methods on CASIA 2.0 NIR-VIS Face Database.

Method	Rank-1	VR@FAR=0.1%
Gabor+RBM [57]	86.16±0.98%	81.29±1.82%
DLBP [17]	78.46±1.67%	85.80%
TRIVET [30]	95.74±0.52%	91.03±1.26%
IDR [14]	95.82±0.76%	94.03±1.06%
VGG [38]	62.09±1.88%	39.72±2.85%
SeetaFace [29]	68.03±1.66%	58.75±2.26%
CenterLoss [53]	87.69±1.45%	69.72±2.07%
Light CNN-4	85.45±1.65%	70.91±1.95%
Light CNN-9	91.88±0.58%	85.31±0.95%
Light CNN-29	**96.72±0.23%**	**94.77±0.43%**

Light CNN models are trained on MS-Celeb-1M which the imaging condition is quite different from Multi-PIE. The results indicate that the Light CNN framework can efficiently capture the characteristics of different identities and obtain the pose and illumination invariant features for face recognition.

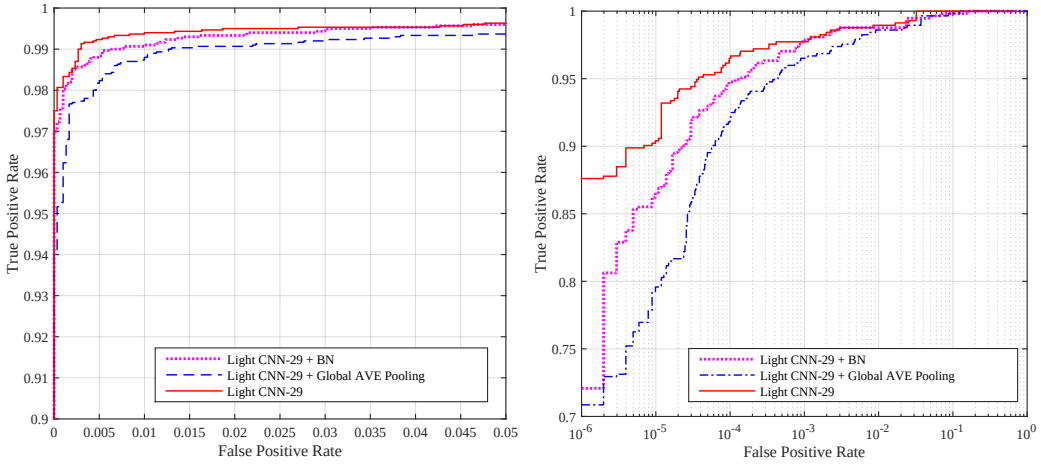
It is interesting to observe that our light CNN also performs very well on the NIR-VIS dataset. It not only outperforms the three CNN methods but also significantly improves state-of-the-art results such as TRIVET [30] and IDR [14] which is trained on the CASIA NIR-VIS 2.0 dataset in supervised way. We improve the best rank-1 accuracy from 95.82%±0.76% to **96.72%±0.23%** and the VR@FAR=0.1% is further improved from 94.03%±1.06% to **94.77%±0.43%**. Note that all Light CNN methods are not fine-tuned on the CASIA NIR-VIS 2.0 dataset. The improvement of our light CNN may benefit from its character of parameter free in an activation function. Obviously, compared with other CNN methods which are not trained on the cross-modal NIR-VIS dataset, our light CNNs based on MFM operations depend on a competitive relationship rather than a threshold of ReLU so that it is naturally adaptive to different appearances from different modalities.

All of the experiments suggest that the proposed light CNNs obtain discriminative face representations and have good generalization for various face recognition tasks.

4.4. Network Analysis

MFM operation plays an important role in our light CNN. Hence we give a detail analysis of MFM on the light CNN-9 model in this subsection.

First, we compare the performance of MFM 2/1 and MFM 3/2 with ReLU, PReLU and ELU on the LFW database. To simplify the computation of MFM 3/2, we employ $h^1(c) = \max(c^1, c^2), h^2(c) = \max(c^2, c^3)$ to approximate Eq. (4). Although this approximation will induce some redundancy, experimental results show that this approximation can further improve verification rates. The experimental results of different activation functions are



(a) The ROC curves on LFW. (b) The ROC curves on VAL1.

Figure 3. Comparisons on different configurations of Light CNN-29. (a) shows the ROC curves of LFW. (b) shows the ROC curves of VAL1.

shown in Table 12.

We observe that the CNN with MFM 3/2 achieves the highest performance in terms of accuracy, rank-1 and DIR-FAR=1%. For our lighted CNN, MFM 3/2 and MFM 2/1 are superior to the three other activation functions. This may be due to that our MFM uses a competitive relationship rather than a threshold (or bias) to active a neuron. Since the training and testing sets are from different data sources, MFM has better generalization ability to different sources. Compared with MFM 2/1, MFM 3/2 can further improve performance. This indicates that when using MFM, it would be better to keep only a small number of neurons to be inhibited so that more information can be preserved to the next convolution layer. That is, the ratio between input neurons and output neurons is set to between 1 and 2.

Second, as shown in Table 2-5 in Section 4.3, it is obvious that all the proposed Light CNN-4, Light CNN-9 and Light CNN-29 models obtain good performance on various face recognition datasets. It is shown that the MFM operation are suitable for the different general CNN architectures such as AlexNet, VGG and ResNet.

Third, we analyze the MFM operation residual blocks on two validation datasets. The one is LFW which contains 6,000 pairs for evaluations and the other, denoted as VAL1, contains 1,424 face images of 866 identities, which contains 573 positive pairs and 1,012,603 negative pairs. The images in VAL1 are strictly independent from MS-Celeb-1M, because they are not collected from Internet.

In Fig.3, we present the performance of different configurations on Light CNN-29. Obviously, the performance of Light CNN-29 with global average pooling on both LFW and VAL1 is lower than the performance of Light CNN-29, since the global average pooling don't consider the spatial information of each nodes in high-level feature map.

Table 12. Comparison with different activation functions on LFW verification and identification protocol by the Light CNN-9 model.

Method	Accuracy	Rank-1	DIR@FAR=1%
ReLU [35]	98.30%	88.58%	67.56%
PReLU [12]	98.17%	88.30%	66.30%
ELU [6]	97.70%	84.70%	62.09%
MFM 2/1	98.80%	93.80%	84.40%
MFM 3/2	**98.83%**	**94.97%**	**88.59%**

In terms of Batch Normalization, as shown in Fig.3(a), the model with BN is closed to the one without BN, however, the Light CNN-29 substantially outperforms the one with BN shown in Fig.3(b). This is mainly because the images in LFW are also collected from Internet, which the identities are celebrities and the imaging condition is also similar with MS-Celeb-1M, while the images in VAL1 are not from Internet, which is strictly independent from the training dataset. Obviously, BN diminishes when testing samples are quite different from the training dataset, because the means and variances in BN depend on the statistics of the training samples.

As stated above, in the Light CNN-29 model, we remove batch normalization and use fully-connected layer instead of global average pooling, which leads to more generalized face recognition models.

And then very deep convolution neural networks or multi-patch ensemble are common ways to improve recognition accuracy. But they are often time consuming for practical systems. Computational efficiency is also an important issue for CNN models. To verify the computational efficiency of our light CNNs, we compare our CNN with three public CNNs, i.e., the VGG released model [38], open source SDK SeetaFace [29] and CenterLoss [53].

Table 13. The time cost and the number of parameters of our model compared with VGG, CenterLoss released model and SeetaFace. The speed is tested on a single core i7-4790.

Model	#Param	#Dim	Times
VGG [38]	134,251K	4096	581ms
SeetaFace [29]	50,021K	2048	245ms
CenterLoss [53]	19,596K	1024	160ms
Light CNN-4	4,095K	256	75ms
Light CNN-9	5,556K	256	67ms
Light CNN-29	12,637K	256	121ms

As shown in Table 13, the size of our biggest light CNN model (Light CNN-29) is 10 times smaller than that of the well-known VGG model, while the CPU time is about 5 times faster. Compared with the open source face SDK SeetaFace and CenterLoss, our light CNN also performs well in terms of time cost, the number of parameters and feature dimension. The results indicate that our light CNN is potentially suitable and practical on embedding devices and smart phones for real-time applications than its competitors. Particularly, these results also suggest that MFM (as a special case and extension of Maxout activation) can result in a convolution neural network with a small parameter space and a small feature dimension. If MFM is well treated and carefully designed, the learnt CNN can use smaller parameter space to achieve better recognition accuracy.

Besides, we employ our light CNNs on MaPU[52], a novel architecture which is suitable for data-intensive computing with great power efficiency and sustained computation throughput. The Light CNN-9 for feature extractions are only about 40ms on MaPU, which is implemented by floating-point calculation. It is shown that our light CNNs can be deployed on embedded system without any precision losing.

4.5. Noisy Label Data Bootstrapping

In this subsection, we verify the efficiency of the proposed semantic bootstrapping method on the MS-Celeb-1M database. We select Light CNN-9 model to do semantic bootstrapping, because the Light CNN-4 is not able to deal with the original massive data, while the Light CNN-29 is too powerful to overfit on the noisy label data. The tests are performed on two datasets in which face images are subject variations in viewpoint, resolution and illumination. The first dataset, denoted as VAL1, contains 1,424 face images of 866 identities. The second dataset contains 675 identities and totally 3,277 images, which is denoted as VAL2. There are 573 positive pairs and 1,012,603 negative pairs (i.e., totally 1,013,176 pairs) in VAL1. VAL2 contains 2,632,926 pairs that are composed of 4,015 positive and 2,628,911 negative pairs. All the face images in VAL1 and VAL2 are independent from the CASIA-WebFace and MS-Celeb-1M database. Considering highly imbalance dataset evaluations, we employ Receiver Operator Characteristic (ROC) curves and Precision-Recall (PR) curves to evaluate the performance of the retrain models via bootstrapping. Both on VR@FAR for ROC curves and AUC for PR curves are reported.

First, we train a light CNN model on the CASIA-WebFace database that contains 10,575 identities totally about 50K images. Then, we fine-tune our light CNN on MS-Celeb-1M, initialized by the pre-trained model on CASIA-WebFace. Since MS-Celeb-1M contains 99,891 identities, the fully-connected layer from feature representation (256-d) to identity label, which is treated as the classifier, has a large number of parameters (256*99,891=25,572,096). To alleviate the difficulty of CNN convergence, we firstly set the learning rate of all the convolution layers to 0. The softmax loss only contributes to the last fully-connected layer to train the classifier. When it converges coarsely, the learning rate of all the convolution layers is set to the same. And then the learning rate is gradually decreased from 1e-3 to 1e-5.

Second, we employ the trained model in the first step to predict the MS-Celeb-1M dataset and obtain the probability $\hat{p}_i$ and $\hat{t}_i$ for each sample $x_i \in X$. Since the abilities of perceptual consistency can be influenced by the noisy labeled data in the training set, the strict bootstrapping rules are employed to select samples. We accept the re-labeling sample whose prediction label $\hat{t}$ is the same as the ground truth label t and whose probability $\hat{p}_i$ is upper than the threshold p_0 which is set to 0.7. In this way, the MS-Celeb-1M re-labeling dataset, defined as MS-1M-1R, contains 79,077 identities totally 4,086,798 images.

Third, MS-1M-1R is used to retrain the light CNN model following the training methodology in Section 4.1. Furthermore, the original noisy labeled MS-Celeb-1M database is re-sampled by the model trained on MS-1M-1R. Assuming that there are few noisy labeled data in MS-1M-1R, we accept the following samples: 1) The prediction $\hat{t}$ is the same as ground truth label t; 2) The prediction $\hat{t}$ is different from the ground truth label t, but the probability p_i is upper than the threshold p_1 which is set to 0.7. The dataset after bootstrapping, which contains 5,049,824 images for 79,077 identities, denoted as MS-1M-2R.

Finally, we retrain the light CNN on MS-1M-2R. Table 14 shows experimental results of the CNN models learnt on different subsets. We have the following observations:1) The MS-Celeb-1M database contains massive noisy labels. If the noisy labels are well treated, the performances on the two testing datasets can be improved. Our proposed semantic bootstrapping method provides a practical way to deal with the noisy labels on the MS-Celeb-1M database. 2) Verification performance benefits from the larger dataset. The model trained on the original MS-Celeb-1M database with

Table 14. The performance on **VAL1** and **VAL2** for different database trained Light CNN-9 model. It compares the performance of light CNN model trained on CASIA-WebFace, MS-Celeb-1M, MS-Celeb-1M after 1 times bootstrapping(MS-1M-1R) and MS-Celeb-1M after 2 times bootstrapping(MS-1M-2R). The area under Precision-Recall curve(AUC) and verification rate(VR)@fasle acceptance rate(FAR) for different models are shown.

VAL1	AUC	FAR=0.1%	FAR=0.01%
CASIA	89.72%	92.50%	84.82%
MS-Celeb-1M	92.03%	94.42%	88.48%
MS-1M-1R	94.82%	96.86%	92.15%
MS-1M-2R	95.34%	97.03%	93.54%
VAL2	AUC	FAR=0.1%	FAR=0.01%
CASIA	62.82%	62.84%	44.46%
MS-Celeb-1M	75.79%	77.93%	61.38%
MS-1M-1R	81.04%	82.66%	68.91%
MS-1M-2R	82.94%	84.55%	71.39%

noisy labels outperforms the model trained on the CASIA-WebFace database in terms of both ROC and AUC. 3) After two bootstrapping steps, the number of identities is from 99,891 to 79,077 and performance improvement tends to be small. These indicate that our semantic bootstrapping method can obtain a better dataset to train our light CNN.

5. Conclusions

In this paper, we have developed a light convolution neural network framework to learn a robust face representation on noisy labeled dataset. Inspired by neural inhabitation and maxout activation, we proposed a Max-Feature-Map operation to obtain a compact and low dimensional face representation. Small kernel sizes of convolution layers, Network in Network layers and Residual Blocks have been implemented to reduce parameter space and improve performance. One advantage of our framework is that it is faster and smaller than other published CNN methods. It extracts one face representation by using about 121ms on a single core i7-4790, and it only occupies 12,637K parameters for a Light CNN-29 model. Besides, an effective semantic bootstrapping has been proposed to handle the noisy label problem. Experimental results on various face recognition datasets show that the proposed light CNN framework has potential value for some real-time face recognition systems.

References

[1] E. Beigman and B. B. Klebanov. Learning with annotation noise. In *ACL*, 2009.

[2] 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. *TIFS*, 9(12):2144–2157, 2014.

[3] B. Chen, C. Chen, and W. H. Hsu. Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. *IEEE Transactions on Multimedia*, 17(6):804–815, 2015.

[4] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *CVPR*, 2013.

[5] Z. Chen, B. Jiang, J. Tang, and B. Luo. Image set representation and classification with covariate-relation graph. In *ACPR*, 2015.

[6] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *ICLR*, 2016.

[7] B. Frénay and M. Verleysen. Classification in the presence of label noise: A survey. *TNNLS*, 25(5):845–869, 2014.

[8] D. Gong, Z. Li, D. Lin, J. Liu, and X. Tang. Hidden factor analysis for age invariant face recognition. In *ICCV*, 2013.

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. In *ICML*, 2013.

[10] R. Gross, I. A. Matthews, J. F. Cohn, T. Kanade, and S. Baker. Multi-pie. *IVC*, 28(5):807–813, 2010.

[11] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] R. He, X. Wu, Z. Sun, and T. Tan. Learning invariant deep representation for nir-vis face recognition. In *AAAI*, 2017.

[15] 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.

[16] 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 *ACM MM*, 2014.

[17] F. Juefei-Xu, D. K. Pal, and M. Savvides. NIR-VIS heterogeneous face recognition via cross-spectral joint dictionary learning and reconstruction. In *CVPRW*, 2015.

[18] M. Kan, S. Shan, and X. Chen. Multi-view deep network for cross-view classification. In *CVPR*, 2016.

[19] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016.

[20] M. Kim, S. Kumar, V. Pavlovic, and H. A. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, 2008.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[22] N. D. Lawrence and B. Schölkopf. Estimating a kernel fisher discriminant in the presence of label noise. In *ICML*, 2001.

[23] H. Li, G. Hua, X. Shen, Z. L. Lin, and J. Brandt. Eigen-pep for video face recognition. In *ACCV*, pages 17–33, 2014.

[24] S. Z. Li, D. Yi, Z. Lei, and S. Liao. The CASIA NIR-VIS 2.0 face database. In *CVPRW*, 2013.

[25] Y. Li, W. Zheng, and Z. Cui. Recurrent regression for face recognition. *CoRR*, abs/1607.06999, 2016.

[26] S. Liao, Z. Lei, D. Yi, and S. Z. Li. A benchmark study of large-scale unconstrained face recognition. In *IJCB*, 2014.

[27] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.

[28] L. Liu, L. Zhang, H. Liu, and S. Yan. Toward large-population face identification in unconstrained videos. *IEEE TCSVT*, 24(11):1874–1884, 2014.

[29] X. Liu, M. Kan, W. Wu, S. Shan, and X. Chen. VIPLFaceNet: An open source deep face recognition sdk. *Frontiers of Computer Science*, 2016.

[30] X. Liu, L. Song, X. Wu, and T. Tan. Transferring deep representation for NIR-VIS heterogeneous face recognition. In *ICB*, 2016.

[31] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou. Multi-manifold deep metric learning for image set classification. In *CVPR*, 2015.

[32] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.

[33] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

[34] V. Mnih and G. E. Hinton. Learning to label aerial images from noisy data. In *ICML*, 2012.

[35] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[36] H. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2014.

[37] E. G. Ortiz, A. Wright, and M. Shah. Face recognition in movie trailers via mean sequence sparse representation-based classification. In *CVPR*, 2013.

[38] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

[39] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[40] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR Workshop*, 2015.

[41] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[43] S. Sukhbaatar, J. Bruna, M. Paluri, and L. B. R. Fergus. Training convolutional networks with noisy labels. In *ICLR Workshop*, 2015.

[44] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.

[45] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *CVPR*, 2013.

[46] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2015.

[47] Z. Sun and T. Tan. Ordinal measures for iris recognition. *TPAMI*, 31(12):2211–2226, 2009.

[48] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[49] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Web-scale training for face identification. In *CVPR*, 2015.

[50] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, 2017.

[51] M. Uzair, F. Shafait, B. Ghanem, and A. S. Mian. Representation learning with deep extreme learning machines for efficient image set classification. *CoRR*, abs/1503.02445, 2015.

[52] D. Wang, X. Du, L. Yin, C. Lin, H. Ma, W. Ren, H. Wang, X. Wang, S. Xie, L. Wang, Z. Liu, T. Wang, Z. Pu, G. Ding, M. Zhu, L. Yang, R. Guo, Z. Zhang, X. Lin, J. Hao, Y. Yang, W. Sun, F. Zhou, N. Xiao, Q. Cui, and X. Wang. Mapu: A novel mathematical computing architecture. In *ISHPCA*, 2016.

[53] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*. Springer, 2016.

[54] D. R. Wilson and T. R. Martinez. Instance pruning techniques. In *ICML*, 1997.

[55] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011.

[56] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. In *CVPR*, 2017.

[57] D. Yi, Z. Lei, and S. Z. Li. Shared representation learning for heterogenous face recognition. In *FG*, 2015.

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

[59] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim. Rotating your face using multi-task deep neural network. In *CVPR*, 2015.

[60] X. Yin and X. Liu. Multi-task convolutional neural network for face recognition. *CoRR*, abs/1702.04710, 2017.

[61] X. Yuan, X. Liu, and S. Yan. Visual classification with multitask joint sparse representation. *IEEE TIP*, 21(10):4349–4360, 2012.

[62] M. Zhang, R. He, D. Cao, Z. Sun, and T. Tan. Simultaneous feature and sample reduction for image-set classification. In *AAAI*, 2016.

[63] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *ICCV*, 2013.

[64] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *NIPS*, 2014.