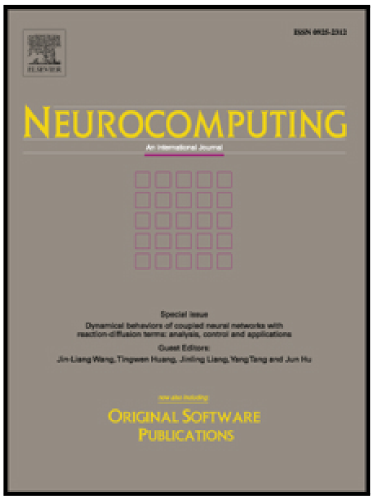


Systematic Evaluation of Deep Face Recognition Methods

Mingyu You, Xuan Han, Yangliu Xu, Li Li

PII:	S0925-2312(20)30055-2
DOI:	https://doi.org/10.1016/j.neucom.2020.01.023
Reference:	NEUCOM 21767
To appear in:	*Neurocomputing*
Received date:	22 July 2019
Revised date:	29 November 2019
Accepted date:	6 January 2020

Available online at www.sciencedirect.com

ScienceDirect

Neurocomputing 00 (2020) 1–??

Systematic Evaluation of Deep Face Recognition Methods

Mingyu You*[a,b], Xuan Han[a], Yangliu Xu[a], Li Li[a,b]

[a]*Department of Control Science & Engineering, Tongji University, Shanghai, China*
[b]*Shanghai Institute of Intelligent Science & Technology, Tongji University, Shanghai, China*

Abstract

Face recognition is an important task in both academia and industry. With the development of deep convolutional neural networks, many deep face recognition methods have been proposed and have achieved remarkable results. However, these methods show great diversity among their datasets, network architectures, loss functions, and parameter learning strategies. For those who want to apply these technologies to establish a deep face recognition system, it is bewildering to evaluate which improvements are more suitable and effective.

This study systematically summarizes and evaluates the state-of-the-art face recognition methods. However, unlike general reviews, on the basis of a survey, this study presents a comprehensive evaluation framework and measures the effects of multifarious settings in five components, including data augmentation, network architecture, loss function, network training, and model compression.

Based on the experimental results, the influences of these five components on the deep face recognition are summarized. In terms of the datasets, a high sample-identity ratio is conducive to generalization, but it leads to increased difficulty for the training to converge. For the network architecture, deep ResNet has an advantage over other designs. Various normalization operations in the network are also necessary. For the loss function, whose performance is closely related to network design and training conditions. The angle-margin loss has a higher upper bound performance, but the traditional Euclidean-margin loss has a stable performance in limited training condition and shallower network. In terms of the training strategy, the step-declining learning rate and large batch size are recommended for recognition tasks. Furthermore, this study compares several popular model compression methods and shows that MobileNet has advantages over the others in terms of both compression ratio and robustness. Finally, a detailed list of recommended settings is provided.

Keywords: Deep Face Recognition, Facial Model Evaluation, Model Design Recommendation

1. Introduction

Face recognition (FR) has a wide range of applications, such as security and electronic payments. It has drawn much attention in computer vision in recent decades. In the early stage, many traditional methods encounter bottlenecks in performance due to the limitations of computing power and model capability[1, 2]. With the advent of deep convolutional neural networks (DCNNs) and increased hardware capability, these restrictions have been rapidly eliminated, and many DCNN-based FR methods have been proposed[3–7].

However, these DCNN-based methods show great diversity among their implementation settings, which makes it difficult to determine which settings of a specific method are worth learning. For instance, Parkhi et al.[8] designed a VGGNet-based model trained with face images from 2,622 identities, and Schroff et al. purposed the GoogLeNet-based FaceNet[9], but trained with images from 8M different identities. Although the latter model achieved a better result than the former, we cannot simply assert that GoogLeNet-based[10] models are more suitable for face feature extraction than VGGNet-based models. Moreover, even within a single method, the effectiveness of operations is difficult to confirm. Taking Arcface[11] as an example, the original paper stated that Arcface Loss was favorable to network training, but it lacked comparison experiments. Thus, it remains questionable whether that loss function is better than the conventional Center Loss[12].

A complete FR system has a few fixed components. Figure 1 shows the general pipeline, including the detection, alignment, feature extraction, and similarity calculation[13, 14]. In this article, we focus on the feature extraction, which is a key factor for improving the performance of FR systems (Figure 1 can describe both traditional and DCNN-based recognition systems. We primarily discusses the latter in this paper. Unless otherwise specified, the term “model” in the following refers to the DCNN model).

In this paper, we analyze the process of model design, and summarize five components that have great influences on the final performance, including data augmentation, network architecture, loss functions, network training, and model compression. Each component has various alternative settings or

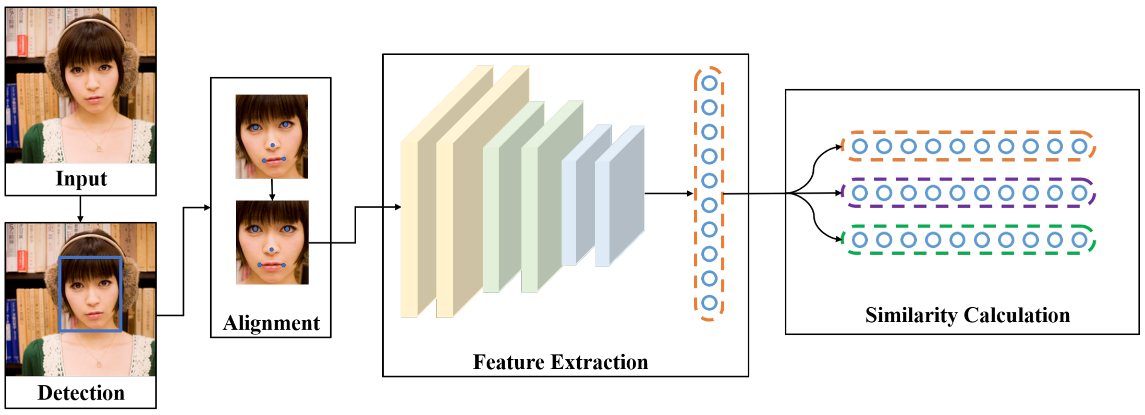


Figure 1. Pipeline of face recognition

operations. E.g., for loss functions, there are candidates such as the original Softmax Loss[15], Center Loss, and Sphereface Loss[16]. By systematically summarizing and evaluating the start-of-the-art works, we filter out the popular settings for each. We then design an evaluation framework that formulates the default settings regarding all five influencing factors. This framework provides a baseline for comparison experiments. In each experiment, except for the discussed factors, all the settings remain unchanged.

The contributions of this paper are summarized as follows:

- We conduct a systematic evaluation of a broad spectrum of FR methods for the first time and evaluate different candidates with respect to the components of the deep FR pipeline.
- By investigating the state-of-the-art methods in this field, we summarize five components that affect the recognition accuracy and design an evaluation framework.
- After synthesizing all the experiment results, we suggest a series of recommended settings. In the screening process, we consider both the performance accuracy and the stability of the training stage.

The remainder of this paper is organized as follows: In the second section, we review related works of FR reported in recent years. The third section explains the five components and introduces the evaluation framework. The fourth section demonstrates and analyzes all the experimental results. In the last section, we summarize this work and draw conclusions.

2. Related Works

FR has been studied for decades. Some traditional works used hand-crafted features such as Local Binary Patterns(LBP) and Histogram of Oriented Gradient(HOG). These methods worked well under constrained conditions. But the hyperparameters in these methods had to be rigorously tuned to achieve acceptable results. Instead of using traditional handcrafted features, researchers currently tend to use DCNN-based features. In the following, we briefly review the existing studies according to the five aforementioned components.

Dataset Many studies[17] have committed to aggregating datasets of face images. CASIA-WebFace[18], maintained by Yi et al., contains 494,414 images from 10,575 identities. UMD-Faces[19] is another popular dataset with 367,920 images of 8,501 identities. VGG-Face, used by VGGNet-based networks, is made up of images from 2,622 identities, each with 1,000 samples. Recently, researchers have increasingly trained their networks with MS-Celeb-1M[20] or MegaFace[21, 22]. MS-Celeb-1M is the largest public face dataset to date, which contains 10M images of 10K celebrities, whereas MegaFace has the largest number of identities (approximately 672K). In general, researchers tend to use as much data as possible to guarantee the final performance of their models, which has resulted in works from different times being based on datasets of different sizes. Without the same training and testing data, comparisons among methods would be unfair.

Apart from the varied training databases, in order to verify the discrimination ability of the face recognition system in different scenario, many distinctive testing datasets have been proposed. Huang et al. proposed LFW (Labeled Faces in the Wild)[23] in 2008, which was the most typical face verification set. LFW contains 6000 pairs of images, 3000 pairs are the same identity (positive samples) and the other 3000 of which are not (negative samples). The testing task is to verify whether the images in one pair are from the same identity. YTF (YouTube Faces)[24] is another important testing dataset. Other than other image-to-image set, YTF consists of 3425 independent videos. And the final task is to verify whether the persons in two videos are the same one. Besides the universal face verification set like LFW and YTF, There are also a few testing databases focusing on the special scenario. CACD-VS (Cross-Age Celebrity Dataset Validation Set) [25] was released in 2014, whose final task is to determine whether two faces with great difference between ages are from the same person. It contains 2000 positive image pairs and 2000 negative pairs. Another wildly noted scenario is cross-angle validation set. The most popular dataset in this field is CFP (Celebrities in Frontal-Profile in the Wild) [26], which has 7000 pairs of

Frontal-Frontal and 7000 pairs of Frontal- Profile face images.

Network Architecture As with the case of the datasets, the works focusing on reforming the network architecture are also disordered[27–29]. The work of Parkhi et al. was based on VGGNet-16. It resulted in a more discriminative decision function and fewer parameters and achieved 98.95% accuracy in LFW. FaceNet from Schroff et al. was based on GoogLeNet, which was capable of processing visual information at various scales. Hasnat et al.[30] designed an architecture called Face-ResNet, which contained 11 residual modules, and the accuracy reached 99.63% in LFW. Most of the works in recent years used deep ResNet. ArcFace used ResNet-100, and CosFace[31] utilized ResNet-64. Apart from using a single network to extract the embeddings, some works combined the outputs of multiple networks. The most representative one of these is DeepID[32, 33] series. In DeepID, the authors trained 60 relatively shallow networks that shared the same architecture. Each processed different face regions with different sizes. Features extracted by these subnetworks were then combined as the final embedding. On the basis of DeepID, DeepID2 reduced the number of subnetworks, and used Softmax Loss and Contrastive Loss to jointly supervise the training, achieving an accuracy of 99.15% on LFW. The newest generation in this series, DeepID3[34], replaced shallow networks with deep networks, which were rebuilt from VGGNet and GoogLeNet. Compared to DeepID2, DeepID3's accuracy has increased by nearly 0.4%. It must be pointed out that although these studies have been tested on the same test set, we still cannot simply assert the strengths of the network structures according to the accuracies. Because these studies differed in many of their experimental settings, VGGFace performed 0.68% lower on LFW than FaceNet. The former used 2.6M images to train, whereas the latter used Google's internal dataset, which contains 500M samples. It is therefore unfair to compare them directly.

Loss Function In the field of deep FR, loss functions draw more research attention than any other topics. This is because FR has stricter requirements on the separability and discriminability of deep embedding space[35, 36] than many other image classification tasks and the loss function is the most direct constraint of that aspect. Sun et al. combined Softmax Loss with Contrastive Loss[37], which not only retained the feature identifiability provided by Softmax Loss, but also incorporated the advantages of the expanding intraclass from Contrastive Loss. Schroff et al. used Triplet Loss[38], which enforced the margin between different faces. Contrastive Loss and Triplet Loss require constructing binary or ternary sample sets. Wen et al. proposed Center Loss, which could penalize the distances between features by learning the center of each class. Zhang et al. proposed Range Loss[39], which was specifically designed for training data with a long-tail distribution. In all these works, only Zhang et al. compared the results with other loss functions, although the comparison was under their own long-tail distribution training set. Although the loss functions described above attempted to improve the ability of constraining features for Softmax Loss, they still failed to replace it. In most cases, either of these loss functions must be used jointly with Softmax Loss to make the model converge. In contrast, the A-Softmax series[11, 16, 31, 40] have a unique approach. They put the module length of the feature aside, and emphasize the aggregation and dispersion of features in terms. It has been proved by practice that the A-Softmax series loss can replace Softmax supervision training.

Training Strategy The learning rate policy and batch size are known collectively as the training strategy. For the learning rate, decreasing it with respect to the training round is widely used, although recent studies have explored periodic learning rates to achieve better results[41, 42]. As for the batch size, it is generally believed that large mini-batch improves the utilization of computing resources[43], while small mini-batch will bring more randomness to the training procedure and improve the generalization performance[44, 45].

Model Compression As mentioned in Section 1, many FR systems are now required to be built on mobile or embedded platforms, and thus compressing models to fit these platforms is becoming an important issue. Currently, there are two popular schemes, SqueezeNet[46] and MobileNet[47]. Both networks can be trained from scratch. And both methods reduce the number of parameters by changing the structure of the convolutional layer.

With several years of development, this field has accumulated rich research results, but the chaos of the experimental conditions obstructs the convenient application of such models. In this case, we intend to systematically summarize the state-of-the-art methods and evaluate the effects of different settings.

3. Evaluation Framework

This section describes the evaluation framework. In the first subsection, we present the five components that influence the performance of FR systems. These components are summarized according to the sequential logic. As shown in Figure 2, they cover the entire process of building an FR system, from design to deployment. This forms the basis of our evaluation framework, and the subsequent evaluations will follow this order. In the second subsection, we formulate the default settings regarding all the influencing factors.

3.1. Components

We list the five components in the first column of Table 1. In particular, the network structure and training strategy both contain several subparts, which are listed in the second column. They are sorted as follows:

- **Dataset:** The dataset plays a significant role in model training. Its influence on the performance originates from the dataset size. Mishkin et al.[48] proved that a larger dataset could improve the performance of DCNNs, directly comparing datasets with significantly different sizes is meaningless and unfair. Moreover, datasets are not only diverse with respect to their sizes, but also to their sample–identity ratios (SIRs), which reflect the data richness in every category. The SIR is relevant to the training difficulty and generalization. In FR system implementation,

Table 1. Factors and operations compared in this work.

Components	**Sub-components**	**Candidates**
Dataset	-	More Identities Less Samples; Less Identities More Samples;
Network Architecture	Widely Used Network	GoogLeNet; VGG-16; Face-ResNet; ResNet-50;
	Activation Function	ReLU; PReLU; ELU; LeakyReLU;
	Feature Normalization	With; Without;
	Batch Normalization	Before Activation; After Activation;
	SE Block	With; Without;
	Downsample	Max Pooling; Average Pooling; Stochastic Pooling; Strided Convolution;
Loss Function	-	Softmax Loss; Contrastive Loss; Triplet Loss; Center Loss; Range Loss; SphereFace Loss; ArcFace Loss; CosFace Loss; Ring Loss;
Training Strategy	Learning Rate	Step; Sqrt; Square; Linear;
	Batch Size	16; 32; 64; 128; 256; 512;
Model Compression	-	SqueezeNet; MobileNet;

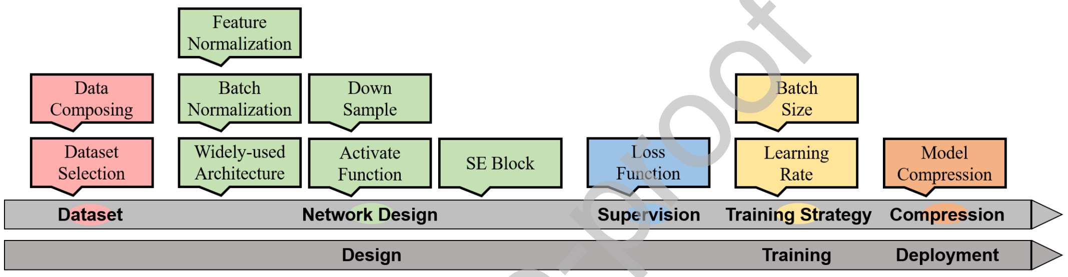


Figure 2. Components to be evaluated and their positions in the FR system building procedure.

the builders are usually required to collect additional training data to finetune the model. Collecting adequate samples for every identity is of great significance. In this case, we focus on the correlation between SIR and model performance.

- **Network Structure:** A deep neural network is a hierarchical model whose structure is easily adjusted by stacking or removing certain layers. In practice, most studies restructure the network to adapt to specific application. Although there are numerous designs of DCNNs for facial feature extraction, they still belong to several popular styles, such as VGG, GoogLeNet, and ResNet. In this part, the evaluation of network styles will be our priority. Apart from the style, the network structure involves many other detailed factors, such as the activation function, batch normalization[49], feature normalization, down-sample layer, and squeeze-and-excitation (SE) block. These can be regarded as special layers or blocks in the deep model, which will also be discussed in the next section.

- **Loss Function:** The design of the loss function is a significant topic. Facial-related tasks have extremely high requirements for the separability and discriminability of features. First, the differences among heterogeneous faces are relatively small. Second, a DCNN-based model usually needs to identify tens of thousands of individuals. The loss function embodies the penalizing rule of models. For FR tasks, we hope that the intra-category gap of features is small, and the inter-category gap is as large as possible. Almost all loss functions are based on this principle. Many studies on loss functions have emerged in recent years. In this work, we evaluate eight representative loss functions: Contrastive Loss, Triplet Loss, Center Loss, Range Loss, SphereFace Loss, ArcFace Loss, CosFace Loss and Ring Loss.

- **Training Strategy:** After determining the dataset, network structure, and loss function, we enter the model training process. This involves the initial value of the learning rate and the shift strategy, the setting of the batch size, and so on. The choice of these parameters affects the entire optimization path. The proper settings of these hyperparameters make the model converge precisely and rapidly. In this article, we focus on the learning rate shift strategy and batch size.

- **Model Compression:** In the last part, we consider the choice of model compression methods. For model compression, layer restructuring methods are the most popular choices. Here, we compare two methods: SqueezeNet and MobileNet. Both methods reduce the number of parameters by replacing the convolutional layers with specific blocks.

This study evaluates several available settings for every component. Because of time constraints, the widely used settings are prioritized. All the candidates to be discussed are listed in the third column of Table 1. Detailed descriptions of these settings are given in the next section.

3.2. Default Experimental Settings

The purpose of this research is to compare the validity of different operations of facial feature extraction, and thus the basic experimental settings, including database, pre-processing method and training parameters, are extremely important. Except for the network architecture comparison, all the experiments use the Face-ResNet network, which is introduced in the next section.

3.2.1. Training Set and Validation Set

In Section 2, we list several common datasets. Every dataset is maintained by a professional team and has complete annotation information. These datasets vary in size. MS-Celeb-1M includes approximately 10M images from 100K identities, and the latter includes approximately 4.7M images from 67K identities. Because our work involves many experiments, training with large datasets would severely increase the time cost, and we finally employ UMD-Faces and CASIA-WebFace. UMD-Faces contains 367888 face images of 8277 identities and CASIA-WebFace contains 494414 face images of 10575 identities (approximately 100 times smaller than MS-Celeb-1M and MegaFace), which balance data diversity with efficiency. In addition, they are comparable in terms of SIR, which makes the union of the two datasets easier. We combine UMD-Faces and CASIA-WebFace as UMD-CASIA, and remove the identities overlapping with our testing sets. UMD-CASIA is divided into training and validation at a ratio of 9:1.

3.2.2. Testing Set

LFW, YTF, CACD-VS and CFP Frontal-Frontal are selected as the testing sets. The distances between extracted features are measured cosine similarity.

Generally, testing on YTF is more complex than other three image-pair validation sets, because face verification[50, 51] on LFW, CACD and CFP-FF measures image-to-image similarities, whereas YTF calculates set-to-set similarities. For each video sequence in YTF, we truncate the first 100 frames for testing. Cosine similarity is utilized here. We take the average accuracy of five models as the actual score, which are randomly selected after the accuracy on the validation set stops increasing. Moreover, we check the variance to ensure that there are no abnormal results.

3.2.3. Pre-processing

Pre-processing can be segmented into two steps: face detection and face alignment. Face detection is to locate the face in an original image and crop it separately. MTCNN[52] is used in this work for detection. It was published in 2016, and has recently become the most popular detection method. It consists of three cascaded networks capable of predicting the bounding boxes and five landmarks of each face.

After the faces are detected, we use a five-point similarity transformation to align the image. The gestures of the faces are varied, with different angles and expressions. To some extent, face alignment erases the influence of gesture variance and benefits the facial feature extraction. The five points we use are provided by MTCNN, which are the center of the left eye, center of the right eye, tip of the nose, left corner of the mouth, and right corner of the mouth. The similarity transformation is to adjust these five points to be as close as possible to the given five reference points by rotation and scaling.

All the images are aligned to 112*96*3 RGB and their mirrored versions are also sent to the network while training, and all the images are shuffled after each training epoch.

3.2.4. Training Parameters

To ensure fairness, we keep the training hyperparameters consistent across the experiment groups. We use the stochastic gradient descent (SGD) optimizer, with the momentum set to 0.9. The default batch size is set to 64. The learning rate is initially set to 10^{-2} and then decreases by a factor of 10 when the validation accuracy stops increasing. We keep training the network until its learning rate decreases to 10^{-6}. All the networks are trained from scratch.

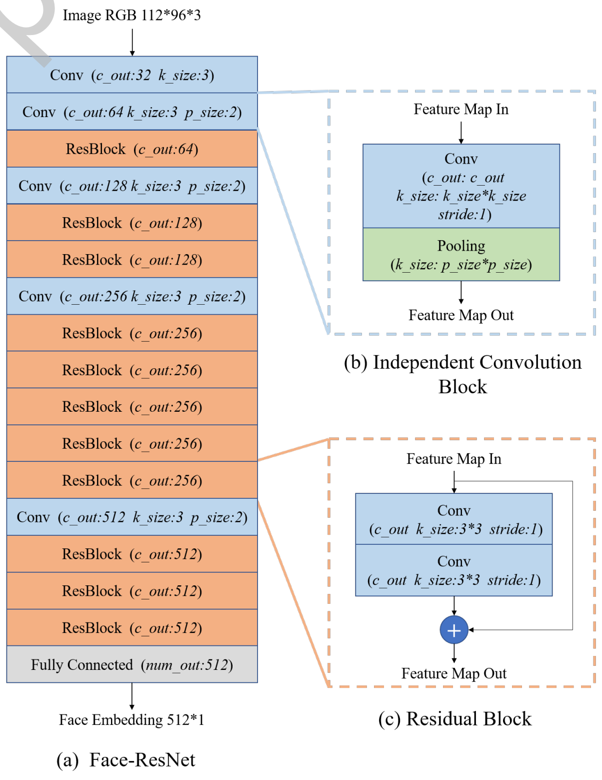


Figure 3. Architecture of Face-ResNet and its two sub-blocks types, independent convolution block and residual block. *c_out* represents the channel number of the output. *k_size* means kernel size. *p_size* is the size of the pooling kernel.

3.2.5. Face-ResNet

We choose Face-ResNet as the default network in the following contrast experiments. It was proposed by Wen in 2016

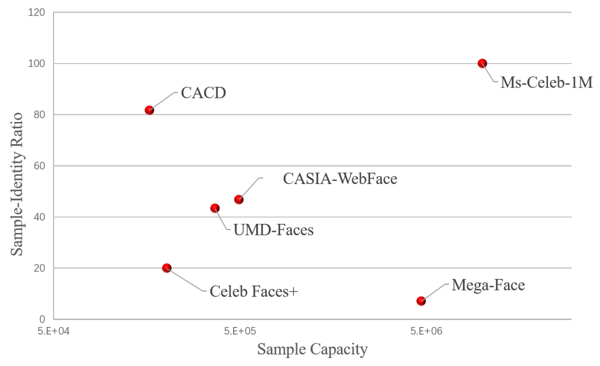
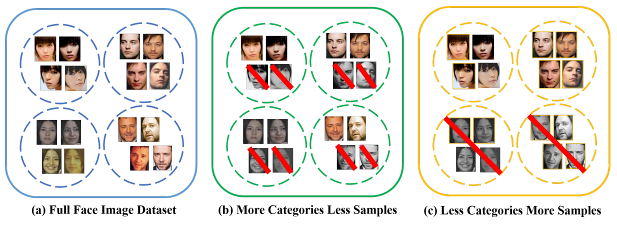


Figure 4. Visualization of sample capacities and sample–identity ratios of datasets. (CACD was proposed by [25], and Celeb Faces+ was from [55].)



(a) Full Face Image Dataset
(b) More Categories Less Samples
(c) Less Categories More Samples

Figure 5. (a) Full dataset, (b) MSLI, (c) LSMI. The images surrounded by a dotted circle are from same identity and the red diagonal lines represent abandoned images.

along with Center Loss. The structure of Face-ResNet is shown in Figure 3(a). Face-ResNet contains 11 residual blocks and 5 independent convolutional blocks. Their interior structures are illustrated in Figure 3(b) and (c). Moreover, each convolution operation is followed by a PReLU activation function.

Among all the alternative FR networks, Face-ResNet is not the best-performing one. However, those better-preforming models such as ResNet-50[53, 54] are too deep to experiment frequently, and their extremely high baseline would submerge the effect gaps among various settings or operations. Moreover, the training of Face-ResNet is steadier than other candidates. Therefore, we finally choose Face-ResNet as our default network.

4. Evaluation Results and Analyses

4.1. Data Augmentation

We do not compare these datasets by experiments directly. First, there are significant differences among the numbers of images, which sometimes reach a factor of 100 or more. Second, most images in the public datasets are photographs of celebrities crawled from websites, which have high repetition rates among them. In most instances, researchers do not make choices, but rather combine them.

As shown in Figure 4, there is significant imparity among the mean SIRs(sampleidentity ratios) of the datasets. For MS-Celeb-1M, it is approximately 100:1, and it is approximately 70:1 for MegaFace. As for UMD-Faces and CASIA-WebFace, their mean SIRs are almost 45:1. In general, richer samples can provide more references to grasp the core characteristics of the identity, but do redundant samples hinder the convergence of the model? To answer this question, we design and implement the following evaluation.

Table 2. Results of datasets with MSLI and LSMI.

Train Set	SIR	LFW (%)	YTF (%)	CACD (%)	CFP (%)
UMD-	45.21	96.46	90.08	94.20	**94.35**
CASIA	22.61	**96.62**	**90.78**	**94.28**	93.71
UMD-	43.28	**93.60**	87.60	89.70	**91.75**
Faces	21.64	93.47	**88.40**	**90.35**	91.52
CASIA-	46.75	**94.85**	88.28	92.08	92.86
WebFace	23.38	94.82	**88.36**	**92.85**	**93.16**

The model is trained with three datasets: UMD-Faces, CASIA-WebFace, and UMD-CASIA, where UMD-CASIA is a union of the other two. As shown in Figure 5, there are two compare tests for each group. In the first, we only use half of the identities but keep all the samples of each. We call it More Samples Less Identities (MSLI). In the second, we keep all the identities but randomly select only half of the samples. This is called Less Samples More Identities (LSMI). The total number of images is unchanged across the contrastive tests. The results are shown in Table 2.

Owing to the advantage of its large data size, UMD-CASIA achieves an higher accuracy than the others, which verifies the previous idea that a larger dataset benefits the performance of model. The results of YTF show more deviations than the others. In YTF, the average accuracy of LSMI is 0.53% higher than that of MSLI. Because YTF is a typical single-identity-multi-sample face image test set, it is theoretically expected to achieve a better performance with a high-SIR training set. However, the MSLI training set significantly reduces the score. This means that the sample richness and the model's generalization are not positively correlated as we envisioned. Therefore, it is risky to blindly increase the samples of the same individual. For example, the sample numbers of the identities in MegaFace are highly nonuniform (minimum 3 and maximum 2000+). When we use the rich categories, it is recommended to prune them.

More samples of the same identity provide more references for efficiently capturing features, but the various useless details in the extra samples make the training more challenging. These results also demonstrate that the expansion of categories is more efficient than that of samples.

4.2. Network Architecture

Widely used networks VGG-16, GoogLeNet, ResNet-50, and Face-ResNet have all been applied in FR tasks. They represent three popular styles of DCNN design, VGG-style, GoogLeNet-style, and ResNet-style, respectively. ResNet-50 and Face-ResNet share the same type but differ in depth. VGG-style networks are relatively unchanged, with a strict hierarchical form without any branch or skip. VGGNet-16 contains 16 convolution layers and 5 pooling layers. It replaces the 7×7

and 11 × 11 receptive fields in AlexNet[56] with a 3 × 3 receptive field. GoogLeNet-style focuses on the horizontal expansion of networks. GoogLeNet was applied to recognize faces from 2015. The author of the original paper designed five improved versions to verify the validity. In this work, we use the version named NN2, in which the inception modules concatenate several convolution layers' output to get different feature granularity. Finally, ResNet-style introduces residual links into the network to facilitate vertical expansion. Here, ResNet-50 is included to test the influence of depth and residual principles on the network. The original pooling layers of GoogLeNet and ResNet-50 are replaced by global pooling to handle rectangular images, and the other components remain unchanged.

Table 3. Results of different network architectures.

Network	Train Set	LFW (%)	YTF (%)	CACD (%)	CFP (%)
VGG-16	CASIA-WebFace	96.52	90.92	94.97	95.17
	UMD-CASIA	97.41	92.98	95.79	96.10
GoogLe-Net	CASIA-WebFace	94.92	87.36	89.78	90.17
	UMD-CASIA	96.22	90.36	91.60	92.55
Face-ResNet	CASIA-WebFace	96.77	90.52	95.25	95.67
	UMD-CASIA	97.70	91.98	95.90	96.05
ResNet-50	CASIA-WebFace	97.72	91.00	96.25	95.51
	UMD-CASIA	**98.52**	**94.16**	**97.28**	**96.93**

The experimental results are shown in Table 3. ResNet-50 achieves the highest accuracy among the four networks in all validation set, and GoogLeNet obtained the lowest. For each group, Face-ResNet's performance is always below that of ResNet-50, and sometimes lower than VGG-16. We attribute this to the structural incompleteness of Face-ResNet. In the original paper, Face-ResNet was followed by a feature normalization layer. The accuracy gaps between Face-ResNet and GoogLeNet reach 2–3% in LFW and YTF, and are more apparent in CACD and CFP. The inception module of GoogLeNet has not brought the expected increase in accuracy, but rather a decline compared to the single-track model VGG-16. This implies that vertical expansion of a network can be more efficient than horizontal expansion for facial feature extraction.

Batch normalization Batch normalization (BN) is one of the most efficient methods to optimize the training of a neural network. The basic principle can be described as follows. For each hidden layer neuron, the inputs gradually approach to the saturation region of the activation function. Normalizing them to a standard normal distribution before feeding to a specific layer makes their values fall into the linear area of the activation function, thereby avoiding the problem of gradient disappearance.

Table 4. Results of different networks with different placements of batch normalization.

Network	Place	LFW (%)	YTF (%)	CACD (%)	CFP (%)
VGG-16	No	97.41	92.98	95.79	96.10
	After	-	-	-	-
	Before	**98.94**	**94.78**	**98.75**	**98.57**
GoogLe-Net	No	96.22	90.39	91.60	92.55
	After	-	-	-	-
	Before	**98.16**	**93.55**	**93.40**	**94.63**
Face-ResNet	No	97.70	**91.98**	**95.90**	96.05
	After	95.42	89.40	94.45	92.29
	Before	**97.76**	91.90	95.78	**96.28**
ResNet-50	No	-	-	-	-
	After	-	-	-	-
	Before	**98.52**	**94.16**	**97.28**	**96.93**

BN is usually placed before or after the activation function. Because it is a common configuration of DCNNs, we only focus on its location.

The results in Table 4 show that the BN layer does have a positive effect on the final accuracy, especially for ResNet-50, which fails to converge when all the BN layers are removed. However, the influence of the BN location is also beyond our expectations. For GoogLeNet and VGG-16, an incorrect location makes the training process unable to converge, and for Face-ResNet, an incorrect location leads to a decline in the accuracy. Therefore, it is necessary to add BN to overcome the gradient disappearance and accelerate the optimization, but the location of the BN layer should be considered. If a network with BN layers cannot converge after many epochs, it is recommended to adjust the location of the BN layers.

Table 5. Results of different network architectures with or without feature normalization.

Network	FN Layer	LFW (%)	YTF (%)	CACD (%)	CFP (%)
VGG-16	Without	97.41	92.98	95.79	96.10
	With	**99.00**	**94.29**	**98.29**	**98.25**
GoogLe-Net	Without	96.22	90.39	91.60	92.55
	With	**97.58**	**92.72**	**95.48**	**95.09**
Face-ResNet	Without	97.70	91.98	95.90	96.05
	With	**99.37**	**95.08**	**98.65**	**98.83**
ResNet-50	Without	98.52	94.16	97.28	96.93
	With	**99.13**	**94.99**	**97.91**	**97.87**

Feature normalization Feature normalization (FN) is usually located at the end of the network. It normalizes the elements of feature embeddings to ensure they have an equal contribution to the loss function. FN has been used in DCNN-based FR since DeepVisage[30] and has shown remarkable performance.

As shown in Table 5, the increase is clear for all four networks after adding an FN layer. For LFW, the accuracies increase by 1.30% on average, for YTF, the increase is approx-

Table 6. Results of Face-ResNet with different activation functions

Activation Function	Formula	Parameter	LFW (%)	YTF (%)	CACD (%)	CFP (%)
ReLU	$y = max(x, 0)$	-	**97.78**	91.77	95.58	96.14
PReLU	$y = max(x, \alpha x)$	α is learnable	97.70	**91.98**	95.90	96.05
ELU	$y = \begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	$\alpha = 0.01$	96.17	90.16	92.00	91.70
		$\alpha = 0.1$	95.83	90.16	92.18	92.91
		$\alpha = 1$	96.54	90.29	93.63	94.00
LeakyReLU	$y = max(x, \alpha x)$	$\alpha = 0.01$	97.48	91.62	95.47	96.22
		$\alpha = 0.1$	97.60	91.74	**96.28**	**96.59**
		$\alpha = 1$	94.24	81.50	88.27	85.38

imately 1.89%, while CACD and CFP get 2.44% and 2.10%. Moreover, the increases of Face-ResNet are remarkable. With FN, Face-ResNet surpasses VGG-16 on each test set, which also proves that Face-ResNet and an FN layer have excellent fitness.

Activation function In neural networks, the activation function introduces nonlinearity to the model. The Sigmoid has gradually been abandoned because of the vanishing gradient problem. Currently, most studies tend to use the ReLU series. In this part, we compare several popular activation functions, including ReLU[57], PReLU, ELU[58], and LeakyReLU[59]. They all remain linear over an interval greater than zero. To ensure the attributes of nonlinearity, they are segmented. Their mathematical expressions are shown in Table 6.

As we can see, the curve of PReLU is the same as that of LeakyReLU, but the parameter of PReLU is learnable. When the coefficients of PReLU, ELU, and LeakyReLU are zero, they degenerate to ReLU. For functions with handcrafted parameters, we set several contrastive groups with different values.

The results in Table 6 show that ReLU and PReLU (both without handcrafted parameters) obtain stable performance in four validation sets and occupy the top two in the accuracy rankings of LFW and YTF. The LeakyReLU reaches the highest score when we set α to 0.1, and ELU reaches its maximum when $\alpha = 1$. When LeakyReLU's α is set to 1, the mean accuracy of four datasets decreases by approximately 7.97%. This phenomenon also verifies the importance of the nonlinearity of the activation function. To summarize, if a model has other hyperparameters that need to be adjusted, it is recommended to use activation functions such as ReLU and PReLU without parameters or self-tuning parameters. But it is undeniable that some of the manual-tuning activation functions can achieve better results.

Table 7. Results of Face-ResNet with different pooling layers.

Pooling	LFW (%)	YTF (%)	CACD (%)	CFP (%)
Max pooling	**97.70**	91.98	**95.90**	96.05
Average pooling	97.57	91.76	95.58	96.11
Stochastic pooling	93.93	89.41	87.45	87.12
Strided convolution	97.63	**92.00**	95.68	**96.35**

Downsample Pooling is the most classic downsampling operation in DCNNs, which produces highly compact feature maps by discarding some high-frequency information. In this part, we discuss four downsample strategies: max pooling, average pooling, stochastic pooling[60], and strided convolution. Here, we do not change the window and stride.

The results in Table 7 show that there is a clear gap between stochastic pooling and other pooling operations. It can be speculated that for FR, a classification task requires great image detail, the uncertainty in stochastic pooling may bring some negative effects. In addition, we also try the sum of max pooling and average pooling, but Face-ResNet fail to converge with this method.

SE block General convolutional layers express local spatial connectivity patterns, which means they are restricted to local receptive fields. To this end, Hu et al.[61] designed SE block to utilize global information. It achieved the highest precision in the ILSVRC 2017 classification task. In this work, we replace the residual modules in Face-ResNet with SE-ResNet modules and vary the reduction ratio r to verify its effectiveness.

The results in Table 8 imply that the improvement in FR with SE block is disappointing. Owing to the concomitant decline in performance, it is not recommended to apply.

Table 8. Results of Face-ResNet with or without SE Block.

SE-Block	Param	LFW (%)	YTF (%)	CACD (%)	CFP (%)
Without	-	**97.70**	**91.98**	**95.90**	**96.05**
With	r = 4	97.38	91.45	95.65	95.90
	r = 8	97.34	90.99	94.97	95.34
	r = 16	97.49	91.06	95.43	95.66
	r = 32	97.43	91.46	95.70	95.93

4.3. Loss Function

Contrastive Loss is the most direct penalty for the distance between classes and the distance within the class. Its mathematical expression is shown in Equation (1). d represents the Euclidean distance between two sample features, and y indicates whether the two samples belong to the same class. When $y = 1$, $L = \frac{1}{2N} \sum_{n=1}^{N} yd^2$. When $y = 0$, $L = max(margin - d, 0)^2$, where *margin* is artificially defined and represents a penalty threshold for the distance between classes. The loss is only calculated if the distance between different types of samples is less than *margin*.

$$L_{Contrastive} = \frac{1}{2N}\sum_{n=1}^{N} yd^2 + (1-y)max(margin - d, 0)^2 \quad (1)$$

Triplet Loss was proposed based on the principle that the distance between different samples should be greater than the distance between the same samples. Its expression is shown in Equation (2).

$$L_{Triplet} = \sum_{i}^{N}\left[\left\|f(x_i^a) - f(x_i^p)\right\|_2^2 - \left\|f(x_i^a) - f(x_i^n)\right\|_2^2 + \alpha\right] \quad (2)$$

In this formula, a, p, and n index three samples in a triplet group, where a and p belong to the same identity, n belongs to another, and $f(*)$ represents their feature vector. The geometrical meaning of this formula is: for one triplet group, the same distance should be at least α units smaller than the heterogeneous distance. Were it to be satisfied, the loss would be zero.

Both Contrastive Loss and Triplet Loss need to construct the sample group. Especially with Triplet Loss, one must ensure that each batch contains a certain number of samples from the same identity, which actually destroys the conditions of an independent identical distribution. In addition, they are sensitive to the strategy of constructing the sample group. Taking Contrastive Loss as an example, if we follow the original batch generation strategy, the number of positive sample groups is much smaller than the number of negative sample groups, such that the aggregation within the class is also difficult to achieve. Center Loss and Range Loss were proposed to solve this problem. The formula for Center Loss is shown in Equation (3).

$$L_{Center} = \frac{1}{2}\sum_{i=1}^{m}\left\|x_i - c_{y_i}\right\|_2^2 \quad (3)$$

The principle of Center Loss is intuitive. It learns a feature center for each class and uses the Euclidean distance of each sample relative to its center as the loss value. C in Equation (3) represents a center for class Y_i. Center Loss does not constrain the distance between classes. It is only responsible for the aggregation within the class.

The mathematical formula for Range Loss is shown in Equation (4.3):

$$L_{Range} = \alpha L_{R_{intra}} + \beta L_{R_{inter}} \quad (4)$$

$$L_{R_{intra}} = \sum_{i \subseteq I} L^i_{R_{intra}} = \sum_{i \subseteq I} \frac{k}{\sum_{j=1}^{k}\frac{1}{D_j}}$$

$$L_{R_{inter}} = max(m - D_{Center}, 0) = max(m - \left\|\overline{X}_Q - \overline{X}_R\right\|_2^2, 0)$$

Range Loss is divided into two parts. The first part is the intra-class loss LR-intra, which is the harmonic mean of the k largest Euclidean distances among features of the same class, and the second part is the inter-class loss LR-inter, which is calculated according to distance between class centers. For the sake of fairness, we maintain the parameter settings and sample group construction methods of the original paper.

As mentioned in Section 2, the A-Softmax series introduce the discriminant angle into the FR loss. Starting with the form of the basic SphereFace Loss, an improved form, ArcFace Loss and CosFace Loss, have been proposed. The formulas are shown in Equation (5) (6) (7).

$$L_{SphereFace} = -log\left(\frac{e^{\|f(x_i)\|cos(m\theta_{y_i},i)}}{e^{\|f(x_i)\|cos(m\theta_{y_i},i)} + \sum_j e^{\|f(x_i)\|cos(\theta_j,i)}}\right) \quad (5)$$

$$L_{ArcFace} = -log\left(\frac{e^{\|f(x_i)\|cos(\theta_{y_i}+m,i)}}{e^{\|f(x_i)\|cos(\theta_{y_i}+m,i)} + \sum_j e^{\|f(x_i)\|cos(\theta_j,i)}}\right) \quad (6)$$

$$L_{CosFace} = -log\left(\frac{e^{\|f(x_i)\|(cos(\theta_{y_i},i)+m)}}{e^{\|f(x_i)\|(cos(\theta_{y_i},i)+m)} + \sum_j e^{\|f(x_i)\|cos(\theta_j,i)}}\right) \quad (7)$$

where θ_{y_i} represents the angle between W_{y_i} and x_i, and m is an artificially set parameter. Comparing them with the original Softmax, it can be seen that A-Softmax Serial Loss is similar to Softmax. And their principles are the same as that of SphereFace: by limiting the angle between the feature and its class vector, features are more angularly aggregated, and heterogeneous features remain different in angle.

Ring Loss was proposed in 2018 and its expression is shown in Equation (8):

$$L_{Ring} = \frac{1}{2N}\sum_{i=1}^{N} \|\|\|f(x_i)\| - R\|^2 \quad (8)$$

where $\|f(x_i)\|$ represents the modulus of the feature vector and R is the center of the modulus, which is similar to C_i of Center Loss. R is a learnable parameter, except that the former corresponds to a specific class of samples and the latter is shared by all samples. The geometric meaning of Equation (8) is that the modulus lengths of all sample features should be concentrated on R.

The original authors gave an explanation for the design of Ring Loss. In Softmax, the penalty for small modular length is heavier than for large modular length classes, which makes the final discriminant angle unbalanced[62]. Adding constraints on the feature length can theoretically solve this problem, and subsequent experiments have confirmed the effectiveness of Ring Loss.

In this part, we design two contrast experiments. The first one follows the settings of our evaluation framework strictly, which means all these non-softmax loss functions will be utilized together with Softmax Loss in Face-ResNet (without BN). The aim is to evaluate the performance of different supervisions in the case of relatively limited training condition and shallower network. As shown in Table 9, among eight loss functions, the gain achieved by Center Loss is the most obvious, followed by

Table 9. Results of Face-ResNet(w/o BN) with different loss functions.

Loss Function	LFW (%)	YTF (%)	CACD (%)	CFP (%)
Softmax Loss	97.70	91.98	95.90	96.05
Contrastive Loss	98.23	92.28	97.36	97.47
Triplet Loss	98.20	93.22	96.88	97.07
Center Loss	**99.23**	**94.60**	**98.41**	**98.57**
Range Loss	98.92	93.83	97.85	97.89
SphereFace Loss	98.32	92.94	97.68	97.41
ArcFace Loss	98.17	92.70	96.87	96.99
CosFace Loss	98.82	93.70	97.93	97.79
Ring Loss	98.88	93.62	97.42	97.67

Table 10. Results of ResNet-64(w/ BN) with different loss functions.

Loss Function	LFW (%)	YTF (%)	CACD (%)	CFP (%)
Softmax Loss	97.85	92.84	96.25	96.51
Contrastive Loss	98.63	93.10	97.26	97.67
Triplet Loss	98.56	93.24	97.12	97.12
Center Loss	99.32	94.36	98.44	98.19
Range Loss	99.02	93.34	97.62	97.97
SphereFace Loss	99.38	94.46	**98.68**	**98.23**
ArcFace Loss	98.95	93.52	97.56	97.71
CosFace Loss	**99.50**	**95.00**	98.20	98.21
Ring Loss	99.24	93.34	97.98	97.97

Range Loss, Ring Loss and CosFace Loss. Contrastive Loss and Triplet Loss only yields a lower improvement. In other words, the well-constructed inter-class distances of other supervision schemes fail to achieve the desired results. During the training, in order to obtain more positive sample pairs to strengthen the intra-class distance constraint, we must make provisions for the choice of the mini-batch, which destroys the independent and identical distribution conditions, and leads to an optimization path offset.

To our surprise, in the first contrast experiment, the A-Softmax serial loss functions fail to show their excellent nature in supervising. The mean accuracies of ArcFace, SphereFace and CosFace are 1.52%, 1.12% and 0.64% lower than Center Loss respectively, which seems to conflict to the results of the original paper. This phenomenon has strongly drawn our attention. We speculate it is partly result from that Center Loss and Face-ResNet were proposed together and well-adapted with each other. To be fair, we design the second contrast experiment, in which the ResNet-64 (with BN) is employed as the backbone network. And following the recommended setting, we separate all three A-softmax Losses from original Softmax. The results of the second contrast experiment are shown in Table 10. As we can see, in ResNet-64 (with BN) the superiority of A- softmax serial lossW is fully reflected. Compared with the results in Face-ResNet(without BN), CosFace and SphereFace get a big boosting in performance, which surpass the Center Loss at least 0.13% in mean accuracy, meanwhile the gap between ArcFace and Center Loss reduces to 0.64%. It indicates that to show the angle-margin losses' advantages, a deeper network is required.

Combining the results of the above two contrast experiments, we can present some recommendations about the loss function selection. Firstly, the inter-intra-class margin losses like Contrastive Loss and Triplet Loss require special mini-batch sampling method, which will damage the i.i.d conditions to some extent. So it is recommended to employ them in finetune than training from scratch. Secondly, the effectiveness of different supervision is strongly associated with the training condition and network design. To our best knowledge, the advanced angle-margin losses have higher upper limit in performance, and their advantages are more likely to emerge when they work with a more updated (like with BN Layer) and deeper (Like ResNet-64) network. It can be partially explained that the constraint of the angle margin is more difficult to meet than in the Euclidean margin. To sum up, the angle-margin losses like Sphereface have a higher upper bound performance, but the traditional Euclidean-margin losses like Center Loss do have a steady nature in the limited training condition and the shallower network.

4.4. Training Strategy

Learning rate In this study, we use the SGD optimizer with momentum 0.9 in the experiments. Here, we compare four learning rate shift strategies: Step, Sqrt, Liner, and Square. The formulas for these strategies are shown in Table 11, where L_0 is the initial learning rate, s is the step size for Step policy, and m is the number of iterations. γ represents the decrease factor of the learning rate. Here, $L_0 = 0.01$, $s = 2 \times 10^4$, and $m = 10^5$, which make all the learning rates drop to zero after 100,000 iterations. We compare the networks trained from scratch, as well as the networks finetuned from a well-trained model with a smaller learning rate.

The results are shown in Table 11. When training starts from scratch. It is worth noting that Step has the highest average accuracy, while the Square has the lowest. As for the finetuning training method, Step obtained an accuracy higher than the others. These results imply the characteristics and application scopes of the different shift modes. For randomly initialized networks, the parameters in the network should be significantly adjusted, and thus the strategies with higher mean learning rate tend to get better results, because they tend to make larger adjustments in the same iteration round, such as Sqrt. Conversely, for a pre-trained network, the parameter adjustment space is limited, and a strategy with a lower mean learning rate is required.

There are a few different learning rate scheduling schemes. Many studies have presented other ideas. GoogLeNet and SqueezeNet chose to use a polynomial learning rate, which gradually reduces the learning rate after each iteration of training. ResNet-110 warmed up by training with a low learning rate and then continues training with a higher one.

Batch size The batch size has a decisive effect on the optimization path. Owing to the limitation of resources, we cannot put the whole training set into the forward operation at every

Table 11. Results of different learning rate policies with UMD-CASIA.

Training Method	Learning Rate	Formula	LFW (%)	YTF (%)	CACD (%)	CFP (%)
From Scratch	Step	$Lr_i = L_0 \gamma floor(i/s)$	97.70	**91.98**	**95.90**	**96.05**
	Sqrt	$Lr_i = L_0 \sqrt{(1 - i/m)}$	**98.00**	91.74	94.62	94.79
	Linear	$Lr_i = L_0(1 - i/m)$	97.88	91.60	93.96	93.95
	Square	$Lr_i = L_0(1 - i/m)^2$	97.40	91.36	93.68	94.00
Finetune	Step	$Lr_i = L_0 \gamma floor(i/s)/10$	**97.56**	**92.14**	**96.01**	**96.25**
	Sqrt	$Lr_i = L_0 \sqrt{(1 - i/m)}/10$	96.36	90.03	92.91	92.51
	Linear	$Lr_i = L_0(1 - i/m)/10$	95.64	90.23	92.15	91.31
	Square	$Lr_i = L_0(1 - i/m)^2/10$	95.52	89.62	91.80	92.06

Table 12. Results of different batch sizes trained with UMD-CASIA.

Batch Size	LFW (%)	YTF (%)	CACD (%)	CFP (%)
16	97.43	91.64	95.40	95.52
32	**97.76**	91.24	95.17	**96.35**
64	97.50	91.31	95.06	95.39
128	97.20	90.24	93.71	94.81
256	97.70	**91.98**	**95.90**	96.05
512	97.53	90.84	94.70	95.14

step. Therefore, we divide the data into mini-batches, calculate the loss value, and update the parameters. With mini-batch training, the parameters are not updated directly, but rather in a zigzag manner. In theory, however, if each batch is independent and identically distributed (iid) with the training set, the direction of the update does not change. Therefore, when training the network, some researchers tend to choose a larger batch size to try to pursue the iid condition.

In order to verify the effectiveness of a large batch size, we adjust the batch size from 16 to 512. The results are shown in Table 12. We do not find an obvious tendency of the accuracy. In particular, the difference between the large and small batches is not expected. The accuracy of the 512 batch size is actually lower than those of the 16 and 32 batch sizes. However, a large batch size tends to cooperate more with a high initial learning rate. With batch sizes of 256 and 512, the networks trained with an initial learning rate of 0.1 can still converge, but the networks trained with lower initial learning rates fail to converge. Therefore, it is recommended to increase the learning rate appropriately.

4.5. Model Compression

Figure 6 shows their improvement. After replacing the convolutional layer of Face-ResNet with the methods of SqueezeNet and MobileNet, the model is greatly compressed. Table 13 shows a comparison of the parameters before and after compression. When using SqueezeNet, the parameters reduce by a factor of 6.02 from 22.29M to 3.70M, and when using MobileNet, the reduction ratio is 8.78. Moreover, there are BN layers in MobileNet but not in SqueezeNet. To eliminate their in-

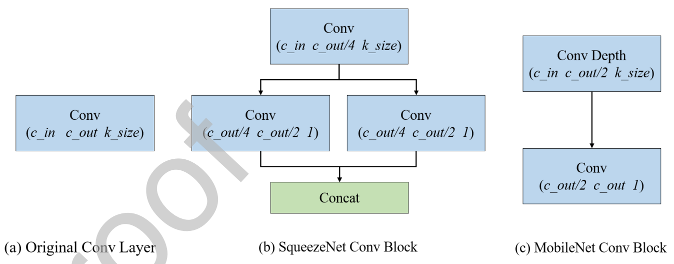


Figure 6. Model compression methods tested in this study. (a) is a normal convolution layer. (b) and (c) represent its substitutions in SqueezeNet and MobileNet. *c_in* is the number of input channels. *c_out* is the number of output channels. *k_size* represents the weight and height of the convolution kernel. Conv Depth is a depthwise convolution layer designed for MobileNet.

fluence, we tried MobileNet-based and SqueezeNet-based networks with or without BN.

In addition to the mentioned changes, it can be seen that model compression inevitably results in a decrease in the performance. In the experimental group without the BN layer, the amplitude of the decline is more obvious. The mean accuracy in four test sets of the SqueezeNet-based network is 2.24% lower than the original network, and that of the MobileNet-based network is 3.33% lower. For the FR model, these declines are large enough to be considered. In the group with the BN layer, the SqueezeNet-based network does not converge after adding the BN layer. The complete MobileNet-based network with the BN layer achieves great performance, with an average accuracy 2.94% higher than that of without, and only 0.41% lower than the original network. From all the experiments, the MobileNet-based network has advantages not only in the degree of parameter compression, but also in model performance.

5. Conclusion

In this study, we systematically summarize and evaluate the existing state-of-the-art FR studies, and compare the effects of different settings and operations with respect to five significant components, including data augmentation, network architecture, loss function, training strategy, and model compression. Unlike general review articles, we design an evaluation framework and measure the effects of multifarious settings with extensive experiments.

Table 13. Results of Face-ResNet with different compression methods.

Method	BN	LFW (%)	YTF (%)	CACD (%)	CFP (%)	Params (M)
Baseline	w/o	**97.70**	**91.98**	**95.90**	**96.05**	22.29
SqueezeNet	w/o	95.45	89.52	93.80	93.92	3.70
MobileNet	w/o	94.43	88.24	92.67	92.98	**2.54**
Baseline	w/	**97.76**	**91.90**	**95.78**	**96.28**	22.29
SqueezeNet	w/	-	-	-	-	3.70
MobileNet	w/	97.34	91.39	95.43	95.92	**2.54**

Table 14. All the results after the evaluation. Recommended factors are represented with bold text.

Components	**Sub-components**	**Candidates**
Dataset	-	**More Identities Less Samples;** Less Identities More Samples;
Network architecture	Widely used Network	GoogLeNet; **VGG-16; Face-ResNet; ResNet-50;**
	Activation Function	**ReLU; PReLU;** ELU; LeakyReLU;
	Feature Normalization	**With;** Without;
	Batch Normalization	**Before Activation Function;** After Activation Function;
	SE Block	With; **Without;**
	Downsample	**Max Pooling; Average Pooling;** Stochastic Pooling; Strided Convolution;
Loss Function	-	Softmax Loss; Contrastive Loss; Triplet Loss; **Center Loss;** Range Loss; SphereFace Loss; ArcFace Loss; **CosFace** Loss; Ring Loss;
Network Training	Learning Rate	**Step;** Sqrt; Square; Linear;
	Batch Size	16; 32; 64; 128; 256; **512;**
Model Compression	-	SqueezeNet; **MobileNet;**

The recommended settings are bold in Table 14. With respect to selecting the dataset, those with high SIR are theoretically conducive to generalization, but they also make convergence more difficult for the training. For network architecture, deep ResNet series networks such as ResNet-50 still have an advantage over other designs. Moreover, BN and FN layers also benefit the model's performance. In terms of the loss function, combining weak-constraint loss functions, such as Center Loss and Ring Loss, with Softmax Loss or A-Softmax Loss would be a stable choice. For the training strategy, the Step decline strategy rate and large batch size are recommended. In terms of model compression, many existing compression methods can effectively reduce the number of parameters, but the resulting accuracy decline must be carefully considered. Finally, based on the experimental results, we present the following suggestions:

- It is risky to blindly increase the samples of the same individual, which makes training more challenging and has less of an effect on model performance. When the SIR of the training set is too high, it is recommended to prune it properly.
- A deeper network that contains residual blocks well facilitates the extraction of facial features. ResNet-style networks have an advantage over the others, which implies that the vertical expansion of a network is far more efficient than horizontal expansion for FR systems.
- Both BN and FN are effective methods to optimize network performance. However, particular networks are sensitive to the location of the BN layer.
- There is no particular benefit to using one activation function over the other, but ReLU and PReLU are more convenient because they have no artificial hyperparameters.
- For an FR model, there is no need to replace max pooling layers with other downsample layers, such as stochastic pooling in particular, because its uncertainty may have some negative effects.
- SE block has no obvious effect on FR and is not recommended to be implemented.
- The choosing of loss function is greatly related to the network design and training condition. According to our experiment results, the angle-margin losses like Sphereface have a higher upper bound performance, but the traditional Euclidean-margin loss like Center Loss have a steady nature in limited training condition and shallower network.
- The learning rate reduction strategy should be selected according to the current task. If the model starts training from scratch, the learning rate should not decrease rapidly. If the model is being finetuned, the learning rate should be maintained at a low level.

- It is better to use a large batch size with other parameters, such as the learning rate. A large batch size should be used along with a high initial learning rate.
- MobileNet-based networks have advantages not only in parameter compression, but also in model performance.

Acknowledgement

This research has been supported by the Shanghai Natural Science Foundation under Grant 18ZR1442600, the National Natural Science Foundation of China under grant no.51475334 and the Fundamental Research Funds for the Central Universities.

References

[1] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local binary patterns, in: European conference on computer vision, Springer, 2004, pp. 469–481.

[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: international Conference on computer vision & Pattern Recognition (CVPR'05), volume 1, IEEE Computer Society, 2005, pp. 886–893.

[3] M. Yang, F. Huang, X. Lv, A feature learning approach for face recognition with robustness to noisy label based on top-n prediction, Neurocomputing 330 (2019) 48 – 55.

[4] L. Du, H. Hu, Nuclear norm based adapted occlusion dictionary learning for face recognition with occlusion and illumination changes, Neurocomputing 340 (2019) 133 – 144.

[5] J.-X. Mi, Z. Luo, L.-F. Zhou, F. Zhong, Bilateral structure based matrix regression classification for face recognition, Neurocomputing 348 (2019) 107 – 119. Advances in Data Representation and Learning for Pattern Analysis.

[6] M. M. Zhang, K. Shang, H. Wu, Learning deep discriminative face features by customized weighted constraint, Neurocomputing 332 (2019) 71 – 79.

[7] J.-J. Lv, J.-S. Huang, X.-D. Zhou, X. Zhou, Y. Feng, Latent face model for across-media face recognition, Neurocomputing 216 (2016) 735 – 745.

[8] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al., Deep face recognition., in: bmvc, volume 1, 2015, p. 6.

[9] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[11] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, arXiv preprint arXiv:1801.07698 (2018).

[12] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: European conference on computer vision, Springer, 2016, pp. 499–515.

[13] L. Chi, H. Zhang, M. Chen, End-to-end face detection and recognition, arXiv preprint arXiv:1703.10818 (2017).

[14] C. Ding, D. Tao, Robust face recognition via multimodal deep face representation, IEEE Transactions on Multimedia 17 (2015) 2049–2058.

[15] Y. Wu, J. Li, Y. Kong, Y. Fu, Deep convolutional neural network with independent softmax for large scale face recognition, in: Proceedings of the 24th ACM international conference on media, ACM, 2016, pp. 1063–1067.

[16] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Sphereface: Deep hypersphere embedding for face recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 212–220.

[17] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, Vggface2: A dataset for recognising faces across pose and age, in: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), IEEE, 2018, pp. 67–74.

[18] D. Yi, Z. Lei, S. Liao, S. Z. Li, Learning face representation from scratch, arXiv preprint arXiv:1411.7923 (2014).

[19] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, R. Chellappa, Umdfaces: An annotated face dataset for training deep networks, in: 2017 IEEE International Joint Conference on Biometrics (IJCB), IEEE, 2017, pp. 464–473.

[20] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, Ms-celeb-1m: Challenge of recognizing one million celebrities in the real world, Electronic Imaging 2016 (2016) 1–6.

[21] D. Miller, E. Brossard, S. Seitz, I. Kemelmacher-Shlizerman, Megaface: A million faces for recognition at scale, arXiv preprint arXiv:1505.02108 (2015).

[22] A. Nech, I. Kemelmacher-Shlizerman, Level playing field for million scale face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7044–7053.

[23] G. B. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, in: Workshop on faces in'Real-Life'Images: detection, alignment, and recognition, 2008.

[24] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, IEEE, 2011.

[25] B.-C. Chen, C.-S. Chen, W. H. Hsu, Cross-age reference coding for age-invariant face recognition and retrieval, in: European Conference on Computer Vision, Springer, 2014, pp. 768–783.

[26] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, D. W. Jacobs, Frontal to profile face verification in the wild, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2016, pp. 1–9.

[27] S. Bell, K. Bala, Learning visual similarity for product design with convolutional neural networks, ACM Transactions on Graphics (TOG) 34 (2015) 98.

[28] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Z. Li, T. Hospedales, When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition, in: Proceedings of the IEEE international conference on computer vision workshops, 2015, pp. 142–150.

[29] S. Minaee, A. Abdolrashidi, Y. Wang, Face recognition using scattering convolutional network, in: 2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB), IEEE, 2017, pp. 1–6.

[30] A. Hasnat, J. Bohné, J. Milgram, S. Gentric, L. Chen, Deepvisage: Making face recognition simple yet with powerful generalization skills, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1682–1691.

[31] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, Cosface: Large margin cosine loss for deep face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5265–5274.

[32] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Advances in neural information processing systems, 2014, pp. 1988–1996.

[33] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2892–2900.

[34] Y. Sun, D. Liang, X. Wang, X. Tang, Deepid3: Face recognition with very deep neural networks, arXiv preprint arXiv:1502.00873 (2015).

[35] K. Huang, D. Dai, C. Ren, Z. Lai, Learning kernel extended dictionary for face recognition, IEEE Transactions on Neural Networks and Learning Systems 28 (2017) 1082–1094.

[36] C.-X. Ren, Z. Lei, D.-Q. Dai, S. Z. Li, Enhanced local gradient order features and discriminant analysis for face recognition, IEEE transactions on cybernetics 46 (2016) 26562669.

[37] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, IEEE, 2006, pp. 1735–1742.

[38] H. Oh Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: Proceedings of the IEEE Confer-

ence on Computer Vision and Pattern Recognition, 2016, pp. 4004–4012.
[39] X. Zhang, Z. Fang, Y. Wen, Z. Li, Y. Qiao, Range loss for deep face recognition with long-tailed training data, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5409–5418.
[40] X. Zhang, R. Zhao, Y. Qiao, X. Wang, H. Li, Adacos: Adaptively scaling cosine logits for effectively learning deep face representations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10823–10832.
[41] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983 (2016).
[42] L. N. Smith, Cyclical learning rates for training neural networks, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2017, pp. 464–472.
[43] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, J. Sun, Megdet: A large mini-batch object detector, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6181–6189.
[44] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T. P. Tang, On large-batch training for deep learning: Generalization gap and sharp minima, arXiv preprint arXiv:1609.04836 (2016).
[45] D. Masters, C. Luschi, Revisiting small batch training for deep neural networks, arXiv preprint arXiv:1804.07612 (2018).
[46] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, K. Keutzer, Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size, arXiv preprint arXiv:1602.07360 (2016).
[47] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).
[48] D. Mishkin, N. Sergievskiy, J. Matas, Systematic evaluation of cnn advances on the imagenet. arxiv 1606: 02228 [cs], arXiv preprint arXiv:1606.02228 (2016).
[49] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 (2015).
[50] J.-C. Chen, V. M. Patel, R. Chellappa, Unconstrained face verification using deep cnn features, in: 2016 IEEE winter conference on applications of computer vision (WACV), IEEE, 2016, pp. 1–9.
[51] A. Bansal, C. Castillo, R. Ranjan, R. Chellappa, The do's and don'ts for cnn-based face verification, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2545–2554.
[52] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Processing Letters 23 (2016) 1499–1503.
[53] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.
[54] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[55] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.
[56] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.
[57] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 315–323.
[58] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus), arXiv preprint arXiv:1511.07289 (2015).
[59] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. icml, volume 30, 2013, p. 3.
[60] M. D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, arXiv preprint arXiv:1301.3557 (2013).
[61] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
[62] Y. Zheng, D. K. Pal, M. Savvides, Ring loss: Convex feature normalization for face recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5089–5097.
[63] X. Yin, X. Liu, Multi-task convolutional neural network for pose-invariant face recognition, IEEE Transactions on Image Processing 27 (2018) 964–975.
[64] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, S. Zafeiriou, Agedb: the first manually collected, in-the-wild age database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop, volume 2, 2017, p. 5.

Biography of the authors

Mingyu You received the B.S. and Ph.D. degrees from Zhejiang University, China in 2002 and 2007. She is currently an Associate Professor with the College of Electronics and Information Engineering, Tongji University. Her research interests are multimedia understanding, pattern recognition and imitation learning.

Xuan Han received the B.S. degree of Automation from Tongji University, China in 2018. He is studying for a master's degree of Pattern Recognition and Intelligent System in the College of Electronics and Information Engineering, Tongji University. His research interests are face recognition and knowledge distillation.

Yangliu Xu received the Master's degree in Control Science and Engineering from Tongji University, China in 2018. She is currently an algorithm engineer in Hikvision Research Institute. Her researches are about face recognition and text recognition.

Li Li received the B.S. and M.S. degrees from Shengyang Agriculture University, China in1996 and 1999, respectively, and the Ph.D. degree from Shenyang Institute of Automation, Chinese Academy of Science, in 2003. She joined Tongji University, Shanghai, China, in 2003, and is presently a Professor of Control Science and Engineering. Her research interests are in data-driven modeling and optimization, computational intelligence.

CRediT

Mingyu You: Conceptualization, Methodology, Writing - Review & Editing, Validation, Funding acquisition.
Xuan Han: Software, Writing-Original draft preparation, Formal analysis.
Yangliu Xu: Investigation, Writing-Original draft preparation, Software.
Li Li: Supervision, Resources, Project administration.

Declaration of interests

☑ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Mingyu You, Xuan Han,
Yanglou Xu, Li Li

Nov. 29, 2019.