ArcFace: Additive Angular Margin Loss for Deep Face Recognition

Jiankang Deng *
Imperial College London
UK
j.deng16@imperial.ac.uk

Jia Guo *
DeepInSight
China
guojia@gmail.com

Stefanos Zafeiriou
Imperial College London
UK
s.zafeiriou@imperial.ac.uk

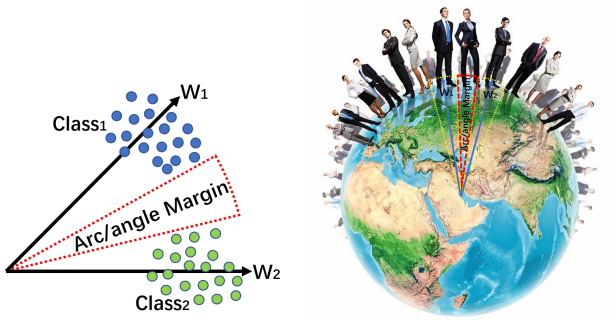
Abstract

Convolutional neural networks have significantly boosted the performance of face recognition in recent years due to its high capacity in learning discriminative features. To enhance the discriminative power of the Softmax loss, multiplicative angular margin [23] and additive cosine margin [44, 43] incorporate angular margin and cosine margin into the loss functions, respectively. In this paper, we propose a novel supervisor signal, additive angular margin (ArcFace), which has a better geometrical interpretation than supervision signals proposed so far. Specifically, the proposed ArcFace $\cos(\theta + m)$ *directly maximise decision boundary in angular (arc) space based on the L2 normalised weights and features. Compared to multiplicative angular margin* $\cos(m\theta)$ *and additive cosine margin* $\cos\theta - m$*, ArcFace can obtain more discriminative deep features. We also emphasise the importance of network settings and data refinement in the problem of deep face recognition. Extensive experiments on several relevant face recognition benchmarks, LFW, CFP and AgeDB, prove the effectiveness of the proposed ArcFace. Most importantly, we get state-of-art performance in the MegaFace Challenge in a totally reproducible way. We make data, models and training/test code public available* [1].

1. Introduction

Face representation through the deep convolutional network embedding is considered the state-of-the-art method for face verification, face clustering, and face recognition [42, 35, 31]. The deep convolutional network is responsible for mapping the face image, typically after a pose normalisation step, into an embedding feature vector such that features of the same person have a small distance while features of different individuals have a considerable distance.

The various face recognition approaches by deep convolutional network embedding differ along three primary attributes.



(a) ArcFace (b) Geodesic Correspondence

Figure 1. Geometrical interpretation of ArcFace. (a) Blue and green points represent embedding features from two different classes. ArcFace can directly impose angular (arc) margin between classes. (b) We show an intuitive correspondence between angle and arc margin. The angular margin of ArcFace corresponds to arc margin (geodesic distance) on the hypersphere surface.

The first attribute is the training data employed to train the model. The identity number of public available training data, such as VGG-Face [31], VGG2-Face [7], CAISA-WebFace [48], UMDFaces [6], MS-Celeb-1M [11], and MegaFace [21], ranges from several thousand to half million. Although MS-Celeb-1M and MegaFace have a significant number of identities, they suffer from annotation noises [47] and long tail distributions [50]. By comparison, private training data of Google [35] even has several million identities. As we can check from the latest performance report of Face Recognition Vendor Test (FRVT) [4], Yitu, a start-up company from China, ranks first based on their private 1.8 billion face images [5]. Due to orders of magnitude difference on the training data scale, face recognition models from industry perform much better than models from academia. The difference of training data also makes some deep face recognition results [2] not fully reproducible.

The second attribute is the network architecture and settings. High capacity deep convolutional networks, such

*denotes equal contribution to this work.
[1] https://github.com/deepinsight/insightface

as ResNet [14, 15, 46, 50, 23] and Inception-ResNet [40, 3], can obtain better performance compared to VGG network [37, 31] and Google Inception V1 network [41, 35]. Different applications of deep face recognition prefer different trade-off between speed and accuracy [16, 51]. For face verification on mobile devices, real-time running speed and compact model size are essential for slick customer experience. For billion level security system, high accuracy is as important as efficiency.

The third attribute is the design of the loss functions.

(1) Euclidean margin based loss.

In [42] and [31], a Softmax classification layer is trained over a set of known identities. The feature vector is then taken from an intermediate layer of the network and used to generalise recognition beyond the set of identities used in training. Centre loss [46] Range loss [50] and Marginal loss [10] add extra penalty to compress intra-variance or enlarge inter-distance to improve the recognition rate, but all of them still combine Softmax to train recognition models. However, the classification-based methods [42, 31] suffer from massive GPU memory consumption on the classification layer when the identity number increases to million level, and prefer balanced and sufficient training data for each identity.

The contrastive loss [39] and the Triplet loss [35] utilise pair training strategy. The contrastive loss function consists of positive pairs and negative pairs. The gradients of the loss function pull together positive pairs and push apart negative pairs. Triplet loss minimises the distance between an anchor and a positive sample and maximises the distance between the anchor and a negative sample from a different identity. However, the training procedure of the contrastive loss [39] and the Triplet loss [35] is tricky due to the selection of effective training samples.

(2) Angular and cosine margin based loss.

Liu *et al*. [24] proposed a large margin Softmax (L-Softmax) by adding multiplicative angular constraints to each identity to improve feature discrimination. SphereFace $\cos(m\theta)$ [23] applies L-Softmax to deep face recognition with weights normalisation. Due to the non-monotonicity of the cosine function, a piece-wise function is applied in SphereFace to guarantee the monotonicity. During training of SphereFace, Softmax loss is combined to facilitate and ensure the convergence. To overcome the optimisation difficulty of SphereFace, additive cosine margin [44, 43] $\cos(\theta)-m$ moves the angular margin into cosine space. The implementation and optimisation of additive cosine margin are much easier than SphereFace. Additive cosine margin is easily reproducible and achieves state-of-the-art performance on MegaFace (TencentAILab_FaceCNN_v1) [2]. Compared to Euclidean margin based loss, angular and cosine margin based loss explicitly adds discriminative constraints on a hypershpere manifold, which intrinsically matches the prior that human face lies on a manifold.

As is well known that the above mentioned three attributes, data, network and loss, have a high-to-low influence on the performance of face recognition models. In this paper, we contribute to improving deep face recognition from all of these three attributes.

Data. We refined the largest public available training data, MS-Celeb-1M [11], in both automatic and manual way. We have checked the quality of the refined MS1M dataset with the Resnet-27 [14, 50, 10] network and the marginal loss [10] on the NIST Face Recognition Prize Challenge [2]. We also find that there are hundreds of overlap face images between the MegaFace one million distractors and the FaceScrub dataset, which significantly affects the evaluation results. We manually find these overlap face images from the MegaFace distractors. Both the refinement of training data and test data will be public available.

Network. Taking VGG2 [7] as the training data, we conduct extensive contrast experiments regarding the convolutional network settings and report the verification accuracy on LFW, CFP and AgeDB. The proposed network settings have been confirmed robust under large pose and age variations. We also explore the trade-off between the speed and accuracy based on the most recent network structures.

Loss. We propose a new loss function, additive angular margin (ArcFace), to learn highly discriminative features for robust face recognition. As shown in Figure 1, the proposed loss function $\cos(\theta + m)$ directly maximise decision boundary in angular (arc) space based on the L2 normalised weights and features. We show that ArcFace not only has a more clear geometrical interpretation but also outperforms the baseline methods, *e.g*. multiplicative angular margin [23] and additive cosine margin [44, 43]. We innovatively explain why ArcFace is better than Softmax, SphereFace [23] and CosineFace [44, 43] from the view of semi-hard sample distributions.

Performance. The proposed ArcFace achieves state-of-the-art results on the MegaFace Challenge [21], which is the largest public face benchmark with one million faces for recognition. We make these results totally reproducible with data, trained models and training/test code public available.

2. From Softmax to ArcFace

2.1. Softmax

The most widely used classification loss function, Softmax loss, is presented as follows:

$$L_1 = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \qquad (1)$$

[2] http://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8197.pdf

where $x_i \in \mathbb{R}^d$ denotes the deep feature of the i-th samples, belonging to the y_i-th class. The feature dimension d is set as 512 in this paper following [46, 50, 23, 43]. $W_j \in \mathbb{R}^d$ denotes the j-th column of the weights $W \in \mathbb{R}^{d \times n}$ in the last fully connected layer and $b \in \mathbb{R}^n$ is the bias term. The batch size and the class number is m and n, respectively. Traditional Softmax loss is widely used in deep face recognition [31, 7]. However, the Softmax loss function does not explicitly optimise the features to have higher similarity score for positive pairs and lower similarity score for negative pairs, which leads to a performance gap.

2.2. Weights Normalisation

For simplicity, we fix the bias $b_j = 0$ as [23]. Then, we transform the target logit [32] as follows:

$$W_j^T x_i = \|W_j\| \, \|x_i\| \cos\theta_j, \tag{2}$$

Following [23, 43, 45], we fix $\|W_j\| = 1$ by L2 normalisation, which makes the predictions only depend on the angle between the feature vector and the weight.

$$L_2 = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{\|x_i\| \cos(\theta_{y_i})}}{e^{\|x_i\| \cos(\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{\|x_i\| \cos\theta_j}}. \tag{3}$$

In the experiments of SphereFace, L2 weight normalisation only improves little on performance.

2.3. Multiplicative Angular Margin

In SphereFace [23, 24], angular margin m is introduced by multiplication on the angle.

$$L_3 = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{\|x_i\| \cos(m\theta_{y_i})}}{e^{\|x_i\| \cos(m\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{\|x_i\| \cos\theta_j}}, \tag{4}$$

where $\theta_{y_i} \in [0, \pi/m]$. In order to remove this restriction, $\cos(m\theta_{y_i})$ is substituted by a piece-wise monotonic function $\psi(\theta_{y_i})$. The SphereFace is formulated as:

$$L_4 = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{\|x_i\| \psi(\theta_{y_i})}}{e^{\|x_i\| \psi(\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{\|x_i\| \cos\theta_j}}, \tag{5}$$

where $\psi(\theta_{y_i}) = (-1)^k \cos(m\theta_{y_i}) - 2k, \theta_{y_i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right], k \in [0, m-1]$, $m \geqslant 1$ is the integer that controls the size of angular margin. However, during the implementation of SphereFace, Softmax supervision is incorporated to guarantee the convergence of training, and the weight is controlled by a dynamic hyper-parameter λ. With the additional Softmax loss, $\psi(\theta_{y_i})$ in fact is:

$$\psi(\theta_{y_i}) = \frac{(-1)^k \cos(m\theta_{y_i}) - 2k + \lambda \cos(\theta_{y_i})}{1 + \lambda}. \tag{6}$$

where λ is a additional hyper-parameter to facilitate the training of SphereFace. λ is set to 1,000 at beginning and decreases to 5 to make the angular space of each class more compact [23]. This additional dynamic hyper-parameter λ makes the training of SphereFace relatively tricky.

2.4. Feature Normalisation

Feature normalisation is widely used for face verification, *e.g.* L2-normalised Euclidean distance and cosine distance [29]. Parde *et al.* [30] observe that the L2-norm of features learned using Softmax loss is informative of the quality of the face. Features for good quality frontal faces have a high L2-norm while blurry faces with extreme pose have low L2-norm. Ranjan *et al.* [33] add the L2-constraint to the feature descriptors and restrict features to lie on a hypersphere of a fixed radius. L2 normalisation on features can be easily implemented using existing deep learning frameworks and significantly boost the performance of face verification. Wang *et al.* [44] point out that gradient norm may be extremely large when the feature norm from low-quality face image is very small, which potentially increases the risk of gradient explosion. The advantages of feature normalisation are also revealed in [25, 26, 43, 45] and the feature normalisation is explained from analytic, geometric and experimental perspectives.

As we can see from above works, L2 normalisation on features and weights is an important step for hypersphere metric learning. The intuitive insight behind feature and weight normalisation is to remove the radial variation and push every feature to distribute on a hypersphere manifold.

Following [33, 43, 45, 44], we fix $\|x_i\|$ by L2 normalisation and re-scale $\|x_i\|$ to s, which is the hypersphere radius and the lower bound is give in [33]. In this paper, we use $s = 64$ for face recognition experiments [33, 43]. Based on feature and weight normalisation, we can get $W_j^T x_i = \cos\theta_j$.

If the feature normalisation is applied to SphereFace, we can get the feature normalised SphareFace, denoted as SphereFace-FNorm

$$L_5 = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{s\psi(\theta_{y_i})}}{e^{s\psi(\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}. \tag{7}$$

2.5. Additive Cosine Margin

In [44, 43], the angular margin m is removed to the outside of $\cos\theta$, thus they propose the cosine margin loss function:

$$L_6 = -\frac{1}{m}\sum_{i=1}^{m} \log \frac{e^{s(\cos(\theta_{y_i}) - m)}}{e^{s(\cos(\theta_{y_i}) - m)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}. \tag{8}$$

In this paper, we set the cosine margin m as 0.35 [44, 43]. Compared to SphereFace, additive cosine margin (Cosine-

Face) has three advantages: (1) extremely easy to implement without tricky hyper-parameters; (2) more clear and able to converge without the Softmax supervision; (3) obvious performance improvement.

2.6. Additive Angular Margin

Although the cosine margin in [44, 43] has a one-to-one mapping from the cosine space to the angular space, there is still a difference between these two margins. In fact, the angular margin has a more clear geometric interpretation compared to cosine margin, and the margin in angular space corresponds to the arc distance on the hypersphere manifold.

We add an angular margin m within $\cos\theta$. Since $\cos(\theta+m)$ is lower than $\cos(\theta)$ when $\theta \in [0, \pi - m]$, the constraint is more stringent for classification. We define the proposed ArcFace as:

$$L_7 = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}, \tag{9}$$

subject to

$$W_j = \frac{W_j}{\|W_j\|}, x_i = \frac{x_i}{\|x_i\|}, \cos\theta_j = W_j^T x_i. \tag{10}$$

If we expand the proposed additive angular margin $\cos(\theta+m)$, we get $\cos(\theta+m) = \cos\theta\cos m - \sin\theta\sin m$. Compared to the additive cosine margin $\cos(\theta) - m$ proposed in [44, 43], the proposed ArcFace is similar but the margin is dynamic due to $\sin\theta$.

In Figure 2, we illustrate the proposed ArcFace, and the angular margin corresponds to the arc margin. Compared to SphereFace and CosineFace, our method has the best geometric interpretation.

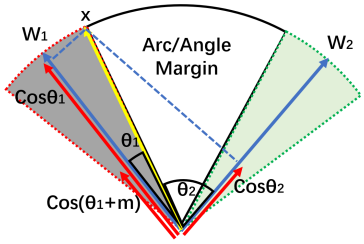


Figure 2. Geometrical interpretation of ArcFace. Different colour areas represent feature spaces from distinct classes. ArcFace can not only compress the feature regions but also correspond to the geodesic distance on the hypersphere surface.

2.7. Comparison under Binary Case

To better understand the process from Softmax to the proposed ArcFace, we give the decision boundaries under binary classification case in Table 1 and Figure 3. Based on the weights and features normalisation, the main difference among these methods is where we put the margin.

Loss Functions	Decision Boundaries
Softmax	$(W_1 - W_2)\,x + b_1 - b_2 = 0$
W-Norm Softmax	$\|x\|\,(\cos\theta_1 - \cos\theta_2) = 0$
SphereFace [23]	$\|x\|\,(\cos m\theta_1 - \cos\theta_2) = 0$
F-Norm SphereFace	$s(\cos m\theta_1 - \cos\theta_2) = 0$
CosineFace [44, 43]	$s(\cos\theta_1 - m - \cos\theta_2) = 0$
ArcFace	$s(\cos(\theta_1 + m) - \cos\theta_2) = 0$

Table 1. Decision boundaries for class 1 under binary classification case. Note that, θ_i is the angle between W_i and x, s is the hypersphere radius, and m is the margin.

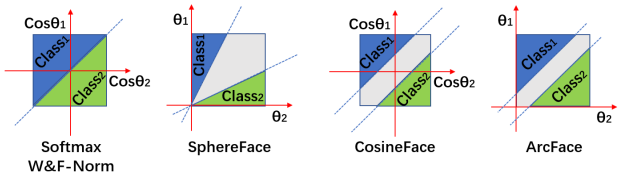


Figure 3. Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.

2.8. Target Logit Analysis

To investigate why the face recognition performance can be improved by SphereFace, CosineFace and ArcFace, we analysis the target logit curves and the θ distributions during training. Here, we use the LResNet34E-IR (refer to Sec. 3.2) network and the refined MS1M dataset (refer to Sec. 3.1).

In Figure 4(a), we plot the target logit curves for Softmax, SphereFace, CosineFace and ArcFace. For SphereFace, the best setting is $m = 4$ and $\lambda = 5$, which is similar to the curve with $m = 1.5$ and $\lambda = 0$. However, the implementation of SphereFace requires the m to be an integer. When we try the minimum multiplicative margin, $m = 2$ and $\lambda = 0$, the training can not converge. Therefore, decreasing the target logit curve slightly from Softmax is able to increase the training difficulty and improve the performance, but decreasing too much may cause the training divergence.

Both CosineFace and ArcFace follow this insight. As we can see from Figure 4(a), CosineFace moves the target logit curve along the negative direction of y-axis, while ArcFace moves the target logit curve along the negative direction of x-axis. Now, we can easily understand the performance improvement from Softmax to CosineFace and ArcFace.

For ArcFace with the margin $m = 0.5$, the target logit curve is not monotonic decreasing when $\theta \in [0, 180°]$. In fact, the target logit curve increases when $\theta > 151.35°$. However, as shown in Figure 4(c), the θ has a Gaussian distribution with the centre at $90°$ and the largest angle below $105°$ when starting from the randomly initialised network. The increasing interval of ArcFace is almost never reached

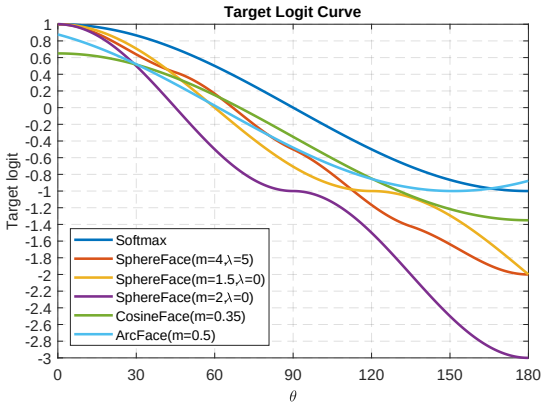
during training. Therefore, we do not need to deal with this explicitly.

In Figure 4(c), we show the θ distributions of CosineFace and ArcFace in three phases of training, *e.g.* start, middle and end. The distribution centres gradually move from 90° to $35^\circ - 40^\circ$. In Figure 4(a), we find the target logit curve of ArcFace is lower than that of CosineFace between 30° to 90°. Therefore, the proposed ArcFace puts more strict margin penalty compared to CosineFace in this interval. In Figure 4(b), we show the target logit converge curves estimated on training batches for Softmax, CosineFace and ArcFace. We can also find that the margin penalty of ArcFace is heavier than that of CosineFace at the beginning, as the red dotted line is lower than the blue dotted line. At the end of training, ArcFace converges better than CosineFace, as the histogram of θ is in the left (Figure 4(c)) and the target logit converge curve is higher (Figure 4(b)). From Figure 4(c), we can find that almost all of the θs are smaller than 60° at the end of training. The samples beyond this field are the hardest samples as well as the noise samples of the training dataset. Even though CosineFace puts more strict margin penalty when $\theta < 30^\circ$ (Figure 4(a)), this field is seldom reached even at the end of training (Figure 4(c)). Therefore, we can also understand why SphereFace can obtain very good performance even with a relatively small margin in this section.

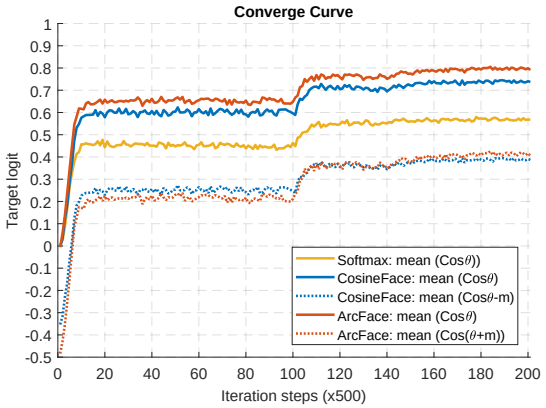
In conclusion, adding too much margin penalty when $\theta \in [60^\circ, 90^\circ]$ may cause training divergence, *e.g.* SphereFace ($m = 2$ and $\lambda = 0$). Adding margin when $\theta \in [30^\circ, 60^\circ]$ can potentially improve the performance, because this section corresponds to the most effective semi-hard negative samples [35]. Adding margin when $\theta < 30^\circ$ can not obviously improve the performance, because this section corresponds to the easiest samples. When we go back to Figure 4(a) and rank the curves between $[30^\circ, 60^\circ]$, we can understand why the performance can improve from Softmax, SphereFace, CosineFace to ArcFace under their best parameter settings. Note that, 30° and 60° here are the roughly estimated thresholds for easy and hard training samples.

3. Experiments

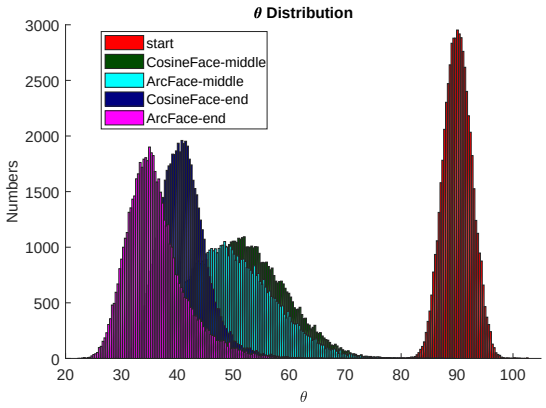
In this paper, we target to obtain state-of-the-art performance on MegaFace Challenge [21], the largest face identification and verification benchmark, in a totally reproducible way. We take Labelled Faces in the Wild (LFW) [19], Celebrities in Frontal Profile (CFP) [36], Age Database (AgeDB) [27] as the validation datasets, and conduct extensive experiments regarding network settings and loss function designs. The proposed ArcFace achieves state-of-the-art performance on all of these four datasets.



(a) Target Logit Curves



(b) Target Logit Converge Curves



(c) θ Distributions during Training

Figure 4. Target logit analysis. (a) Target logit curves for Softmax, SphereFace, CosineFace and ArcFace. (b) Target logit converge curves estimated on training batches for Softmax, CosineFace and ArcFace. (c) θ distributions move from large angles to small angles during training (start, middle and end). Better to view by zoom in.

3.1. Data

3.1.1 Training data

We use two datasets, VGG2 [7] and MS-Celeb-1M [11], as our training data.

VGG2. VGG2 dataset contains a training set with 8,631 identities (3,141,890 images) and a test set with 500 identities (169,396 images). VGG2 has large variations in pose, age, illumination, ethnicity and profession. Since VGG2 is a high-quality dataset, we use it directly without data refinement.

MS-Celeb-1M. The original MS-Celeb-1M dataset contains about 100k identities with 10 million images. To decrease the noise of MS-Celeb-1M and get a high-quality training data, we rank all face images of each identity by their distances to the identity centre. For a particular identity, the face image whose feature vector is too far from the identity's feature centre is automatically removed [10]. We further manually check the face images around the threshold of the first automatic step for each identity. Finally, we obtain a dataset which contains 3.8M images of 85k unique identities. To facilitate other researchers to reproduce all of the experiments in this paper, we make the refined MS1M dataset public available within a binary file, but please cite the original paper [11] and follow the original license [11] when using this dataset. Our contribution here is only training data refinement, not release.

3.1.2 Validation data

We employ Labelled Faces in the Wild (LFW) [19], Celebrities in Frontal Profile (CFP) [36] and Age Database (AgeDB) [27] as the validation datasets.

LFW. [19] LFW dataset contains $13,233$ web-collected images from 5749 different identities, with large variations in pose, expression and illuminations. Following the standard protocol of *unrestricted with labelled outside data*, we give the verification accuracy on $6,000$ face pairs.

CFP. [36]. CFP dataset consists of 500 subjects, each with 10 frontal and 4 profile images. The evaluation protocol includes frontal-frontal (FF) and frontal-profile (FP) face verification, each having 10 folders with 350 same-person pairs and 350 different-person pairs. In this paper, we only use the most challenging subset, CFP-FP, to report the performance.

AgeDB. [27, 10] AgeDB dataset is an in-the-wild dataset with large variations in pose, expression, illuminations, and age. AgeDB contains $12,240$ images of 440 distinct subjects, such as actors, actresses, writers, scientists, and politicians. Each image is annotated with respect to the identity, age and gender attribute. The minimum and maximum ages are 3 and 101, respectively. The average age range for each subject is 49 years. There are four groups of test data with different year gaps (5 years, 10 years, 20 years and 30 years, respectively) [10]. Each group has ten split of face images, and each split contains 300 positive examples and 300 negative examples. The face verification evaluation metric is the same as LFW. In this paper, we only use the most challenging subset, AgeDB-30, to report the performance.

3.1.3 Test data

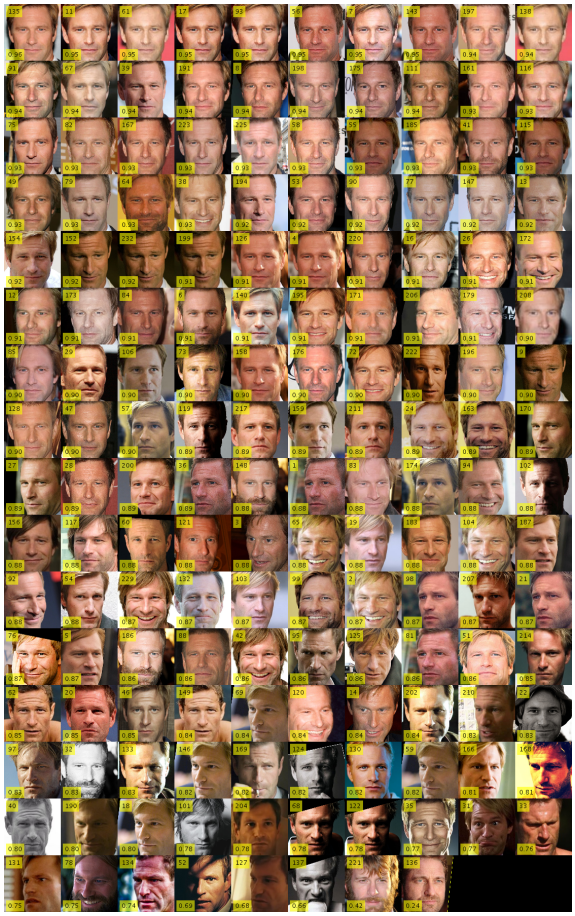
MegaFace. MegaFace datasets [21] are released as the largest public available testing benchmark, which aims at evaluating the performance of face recognition algorithms at the million scale of distractors. MegaFace datasets include gallery set and probe set. The gallery set, a subset of Flickr photos from Yahoo, consists of more than one million images from 690k different individuals. The probe sets are two existing databases: FaceScrub [28] and FGNet [1]. FaceScrub is a publicly available dataset that containing 100k photos of 530 unique individuals, in which $55,742$ images are males, and $52,076$ images are females. FGNet is a face ageing dataset, with 1002 images from 82 identities. Each identity has multiple face images at different ages (ranging from 1 to 69).

It is quite understandable that data collection of MegaFace is very arduous and time-consuming thus data noise is inevitable. For FaceScrub dataset, all of the face images from one particular identity should have the same identity. For the one million distractors, there should not be any overlap with the FaceScrub identities. However, we find noisy face images not only exist in FaceScrub dataset but also exist in the one million distractors, which significantly affect the performance.

In Figure 5, we give the noisy face image examples from the Facesrub dataset. As shown in Figure 8(c), we rank all of the faces according to the cosine distance to the identity centre. In fact, face image 221 and 136 are not Aaron Eckhart. We manually clean the FaceScrub dataset and finally find 605 noisy face images. During testing, we change the noisy face to another right face, which can increase the identification accuracy by about 1%. In Figure 6(b), we give the noisy face image examples from the MegaFace distractors. All of the four face images from the MegaFace distractors are Alec Baldwin. We manually clean the MegaFace distractors and finally find 707 noisy face images. During testing, we add one additional feature dimension to distinguish these noisy faces, which can increase the identification accuracy by about 15%.

Even though the noisy face images are double checked by seven annotators who are very familiar with these celebrities, we still can not promise these images are 100% noisy. We put the noise lists of the FaceScrub dataset and the MegaFace distractors online. We believe the masses have sharp eyes and we will update these lists based on other

researchers' feedback.



(a) Aaron Eckhart

(b) noise face 221

(c) noise face 136

Figure 5. Noisy face image examples from the FaceScrub dataset. In (a), the image id is put in top left and cosine distance to the identity centre is put in bottom left.

3.2. Network Settings

We first evaluate the face verification performance based on different network settings by using VGG2 as the training data and Softmax as the loss function. All experiments in this paper are implemented by MxNet [8]. We set the batch size as 512 and train models on four or eight NVIDIA Tesla P40 (24GB) GPUs. The learning rate is started from 0.1 and divided by 10 at the 100k, 140k, 160k iterations. Total iteration step is set as 200k. We set momentum at 0.9 and weight decay at $5e-4$ (Table 5).

(a) Alec Baldwin

(b) Distractors Noise

Figure 6. Noisy face image examples from the MegaFace distractors. (a) is used for annotators to learn the identity from the FaceScrub dataset. (b) shows the selected overlap faces from the MegaFace distractors.

3.2.1 Input setting

Following [46, 23], we use five facial landmarks (eye centres, nose tip and mouth corners) [49] for similarity transformation to normalise the face images. The faces are cropped and resized to 112×112, and each pixel (ranged between $[0, 255]$) in RGB images is normalised by subtracting 127.5 then divided by 128.

As most of the convolutional networks are designed for the Image-Net [34] classification task, the input image size is usually set as 224×224 or larger. However, the size of our face crops is only 112×112. To preserve higher feature map resolution, we use $conv3 \times 3$ and $stride = 1$ in the first convolutional layer instead of using $conv7 \times 7$ and $stride = 2$. For these two settings, the output size of the convolutional networks is 7×7 (denoted as "L" in front of the network names) and 3×3, respectively.

3.2.2 Output setting

In last several layers, some different options can be investigated to check how the embedding settings affect the model performance. All feature embedding dimension is set to 512 expect for Option-A, as the embedding size in Option-A is determined by the channel size of last convolutional layer.

- Option-A: Use global pooling layer(GP).

- Option-B: Use one fully connected (FC) layer after GP.
- Option-C: Use FC-Batch Normalisation (BN) [20] after GP.
- Option-D: Use FC-BN-Parametric Rectified Linear Unit (PReLu) [13] after GP.
- Option-E: Use BN-Dropout [38]-FC-BN after the last convolutional layer.

During testing, the score is computed by the Cosine Distance of two feature vectors. Nearest neighbour and threshold comparison are used for face identification and verification tasks.

3.2.3 Block Setting

Besides the original ResNet [14] unit, we also investigate a more advanced residual unit setting [12] for the training of face recognition model. In Figure 7, we show the improved residual unit (denoted as "IR" in the end of model names), which has a BN-Conv-BN-PReLu-Conv-BN structure. Compared to the residual unit proposed by [12], we set $stride = 2$ for the second convolutional layer instead of the first one. In addition, PReLu [13] is used to substitute the original ReLu.

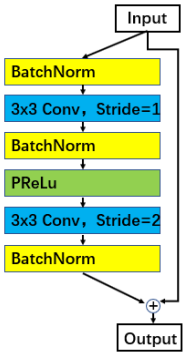


Figure 7. Improved residual unit: BN-Conv-BN-PReLu-Conv-BN.

3.2.4 Backbones

Based on recent advances on the model structure designs, we also explore MobileNet [16], Inception-Resnet-V2 [40], Densely connected convolutional networks (DenseNet) [18], Squeeze and excitation networks (SE) [17] and Dual path Network (DPN) [9] for deep face recognition. In this paper, we compare the differences between these networks from the aspects of accuracy, speed and model size.

3.2.5 Network Setting Conclusions

Input selects L. In Table 2, we compare two networks with and without the setting of "L". When using $conv3 \times 3$ and $stride = 1$ as the first convolutional layer, the network output is 7×7. By contrast, if we use $conv7 \times 7$ and $stride = 2$ as the first convolutional layer, the network output is only 3×3. It is obvious from Table 2 that choosing larger feature maps during training obtains higher verification accuracy.

Networks	LFW	CFP-FP	AgeDB-30
SE-ResNet50D	99.38	94.58	91.00
SE-LResNet50D	**99.6**	**96.04**	**92.68**
SE-ResNet50E	99.26	94.11	90.85
SE-LResNet50E	**99.71**	**96.38**	**92.98**

Table 2. Verification accuracy (%) under different input sittings (Softmax@VGG2).

Output selects E. In Table 3, we give the detailed comparison between different output settings. The option E (BN-Dropout-FC-BN) obtains the best performance. In this paper, the dropout parameter is set as 0.4. Dropout can effectively act as the regularisation term to avoid over-fitting and obtain better generalisation for deep face recognition.

Networks	LFW	CFP-FP	AgeDB-30
SE-LResNet50A	99.51	95.81	92.60
SE-LResNet50B	99.46	94.90	91.85
SE-LResNet50C	99.56	95.81	92.61
SE-LResNet50D	99.6	96.04	92.68
SE-LResNet50E	**99.71**	**96.38**	**92.98**
SE-LResNet50A-IR	99.58	95.90	92.63
SE-LResNet50D-IR	99.61	96.51	92.68
SE-LResNet50E-IR	**99.78**	**96.82**	**93.83**

Table 3. Verification accuracy (%) under different output settings (Softmax@VGG2).

Block selects IR. In Table 4, we give the comparison between the original residual unit and the improved residual unit. As we can see from the results, the proposed BN-Conv(stride=1)-BN-PReLu-Conv(stride=2)-BN unit can obviously improve the verification performance.

Networks	LFW	CFP-FP	AgeDB-30
SE-LResNet50E	99.71	96.38	92.98
SE-LResNet50E-IR	**99.78**	**96.82**	**93.83**

Table 4. Verification accuracy (%) comparison between the original residual unit and the improved residual unit (Softmax@VGG2).

Backbones Comparisons. In Table 8, we give the verification accuracy, test speed and model size of different back-

bones. The running time is estimated on the P40 GPU. As the performance on LFW is almost saturated, we focus on the more challenging test sets, CFP-FP and AgeDB-30, to compare these network backbones. The Inception-Resnet-V2 network obtains the best performance with long running time ($53.6ms$) and largest model size ($642MB$). By contrast, MobileNet can finish face feature embedding within $4.2ms$ with a model of $112MB$, and the performance only drops slightly. As we can see from Table 8, the performance gaps between these large networks, *e.g.* ResNet-100, Inception-Resnet-V2, DenseNet, DPN and SE-Resnet-100, are relatively small. Based on the trade-off between accuracy, speed and model size, we choose LResNet100E-IR to conduct experiments on the Megaface challenge.

Weight decay. Based on the SE-LResNet50E-IR network, we also explore how the weight decay (WD) value affects the verification performance. As we can see from Table 5, when the weight decay value is set as $5e-4$, the verification accuracy reaches the highest point. Therefore, we fix the weight decay at $5e-4$ in all other experiments.

WD values	LFW	CFP-FP	AgeDB-30
5e-6	99.11	94.52	90.43
5e-5	99.56	95.74	92.95
5e-4	**99.78**	**96.82**	**93.83**
1e-3	99.71	96.60	93.53

Table 5. Verification performance (%) of different weight decay (WD) values (SE-LResNet50E-IR,Softmax@VGG2).

3.3. Loss Setting

Since the margin parameter m plays an important role in the proposed ArcFace, we first conduct experiments to search the best angular margin. By varying m from 0.2 to 0.8, we use the LMobileNetE network and the ArcFace loss to train models on the refined MS1M dataset. As illustrated in Table 6, the performance improves consistently from $m = 0.2$ on all datasets and gets saturated at $m = 0.5$. Then, the verification accuracy turns to decrease from $m = 0.5$. In this paper, we fix the additive angular margin m as 0.5.

Based on the LResNet100E-IR network and the refined MS1M dataset, we compare the performance of different loss functions, *e.g.* Softmax, SphereFace [23], CosineFace [44, 43] and ArcFace. In Table 7, we give the detailed verification accuracy on the LFW, CFP-FP, and AgeDB-30 datasets. As LFW is almost saturated, the performance improvement is not obvious. We find that (1) Compared to Softmax, SphereFace, CosineFace and ArcFace improve the performance obviously, especially under large pose and age variations. (2) CosineFace and ArcFace obviously outperform SphereFace with much easier implementation. Both

m	LFW	CFP-FP	AgeDB-30
0.2	99.23	87.23	95.25
0.3	99.40	88.15	96.00
0.4	99.48	87.85	96.00
0.5	**99.50**	**88.50**	**96.06**
0.6	99.46	87.23	95.68
0.7	99.46	87.48	95.80
0.8	99.40	86.74	95.68

Table 6. Verification performance (%) of ArcFace with different angular margins m (LMobileNetE,ArcFace@MS1M).

CosineFace and ArcFace can converge easily without additional supervision from Softmax. By contrast, additional supervision from Softmax is indispensable for SphereFace to avoid divergence during training. (3) ArcFace is slightly better than CosineFace. However, ArcFace is more intuitive and has a more clear geometric interpretation on the hypersphere manifold as shown in Figure 1.

Loss	LFW	CFP-FP	AgeDB-30
Softmax	99.7	91.4	95.56
SphereFace (m=4, $\lambda = 5$)	99.76	93.7	97.56
CosineFace (m=0.35)	99.80	94.4	97.91
ArcFace(m=0.4)	99.80	**94.5**	98.0
ArcFace(m=0.5)	**99.83**	94.04	**98.08**

Table 7. Verification performance (%) for different loss functions (LResNet100E-IR@MS1M).

3.4. MegaFace Challenge1 on FaceScrub

For the experiments on the MegaFace challenge, we use the LResNet100E-IR network and the refined MS1M dataset as the training data. In both Table 9 and 10, we give the identification and verification results on the original MegaFace dataset and the refined MegaFace dataset.

In Table 9, we use the whole refined MS1M dataset to train models. We compare the performance of the proposed ArcFace with related baseline methods, *e.g.* Softmax, Triplet, SphereFace, and CosineFace. The proposed ArcFace obtains the best performance before and after the distractors refinement. After the overlapped face images are removed from the one million distractors, the identification performance significantly improves. We believe that the results on the manually refined MegaFace dataset are more reliable, and the performance of face identification under million distractors is better than we think [2].

To strictly follow the evaluation instructions on MegaFace, we need to remove all of the identities appearing in the FaceScrub dataset from our training data. We calculate the feature centre for each identity in the refined MS1M dataset and the FaceScrub dataset. We find that 578 identi-

Backbones	LFW (%)	CFP-FP (%)	AgeDB-30 (%)	Speed(ms)	Model-Size(MB)
LResNet50E-IR	99.75	96.58	93.53	8.9	167
SE-LResNet50E-IR	**99.78**	96.82	93.83	13.0	169
LResNet100E-IR	99.75	96.95	94.4	15.4	250
SE-LResNet100E-IR	99.71	97.01	94.23	23.8	252
LResNet101(Bottle Neck)E-IR	99.76	96.72	93.68	49.2	294
LMobileNetE	99.63	95.81	91.85	**4.2**	**112**
LDenseNet161E	99.71	96.51	93.68	29.3	315
LDPN92E	99.71	96.82	94.18	38.1	393
LDPN107E	99.76	96.94	94.9	58.8	581
LInception-ResNet-v2	99.75	**97.15**	**95.35**	53.6	642

Table 8. Accuracy (%), speed (ms) and model size (MB) comparison between different backbones (Softmax@VGG2)

Methods	Rank1@10^6	VR@FAR10^{-6}	Rank1@10^6 (R)	VR@FAR10^{-6} (R)
Softmax	78.89	94.95	91.43	94.95
Softmax-pretrain,Triplet-finetune	80.6	94.65	94.08	95.03
Softmax-pretrain@VGG2, Triplet-finetune	78.87	95.43	93.96	95.07
SphereFace(m=4, λ=5)	82.95	97.66	97.43	97.66
CosineFace(m=0.35)	82.75	98.41	98.33	98.41
ArcFace(m=0.4)	82.29	98.20	98.10	97.83
ArcFace(m=0.5)	**83.27**	**98.48**	**98.36**	**98.48**

Table 9. Identification and verification results of different methods on MegaFace Challenge1 (LResNet100E-IR@MS1M). "Rank 1" refers to the rank-1 face identification accuracy and "VR" refers to face verification TAR (True Accepted Rate) at 10^{-6} FAR (False Accepted Rate). (R) denotes the refined version of MegaFace dataset.

ties from the refined MS1M dataset have a close distance (cosine similarity is higher than 0.45) with the identities from the FaceScrub dataset. We remove these 578 identities from the refined MS1M dataset and compare the proposed ArcFace to other baseline methods in Table 10. ArcFace still outperforms CosineFace with a slight performance drop compared to Table 9. But for Softmax, the identification rate drops obviously from 78.89% to 73.66% after the suspectable overlap identities are removed from the training data. On the refined MegaFace testset, the verification result of CosineFace is slightly higher than that of ArcFace. This is because we read the verification results which are closest to FAR=1e-6 from the outputs of the devkit. As we can see from Figure 8, the proposed ArcFace always outperforms CosineFace under both identification and verification metric.

3.5. Further Improvement by Triplet Loss

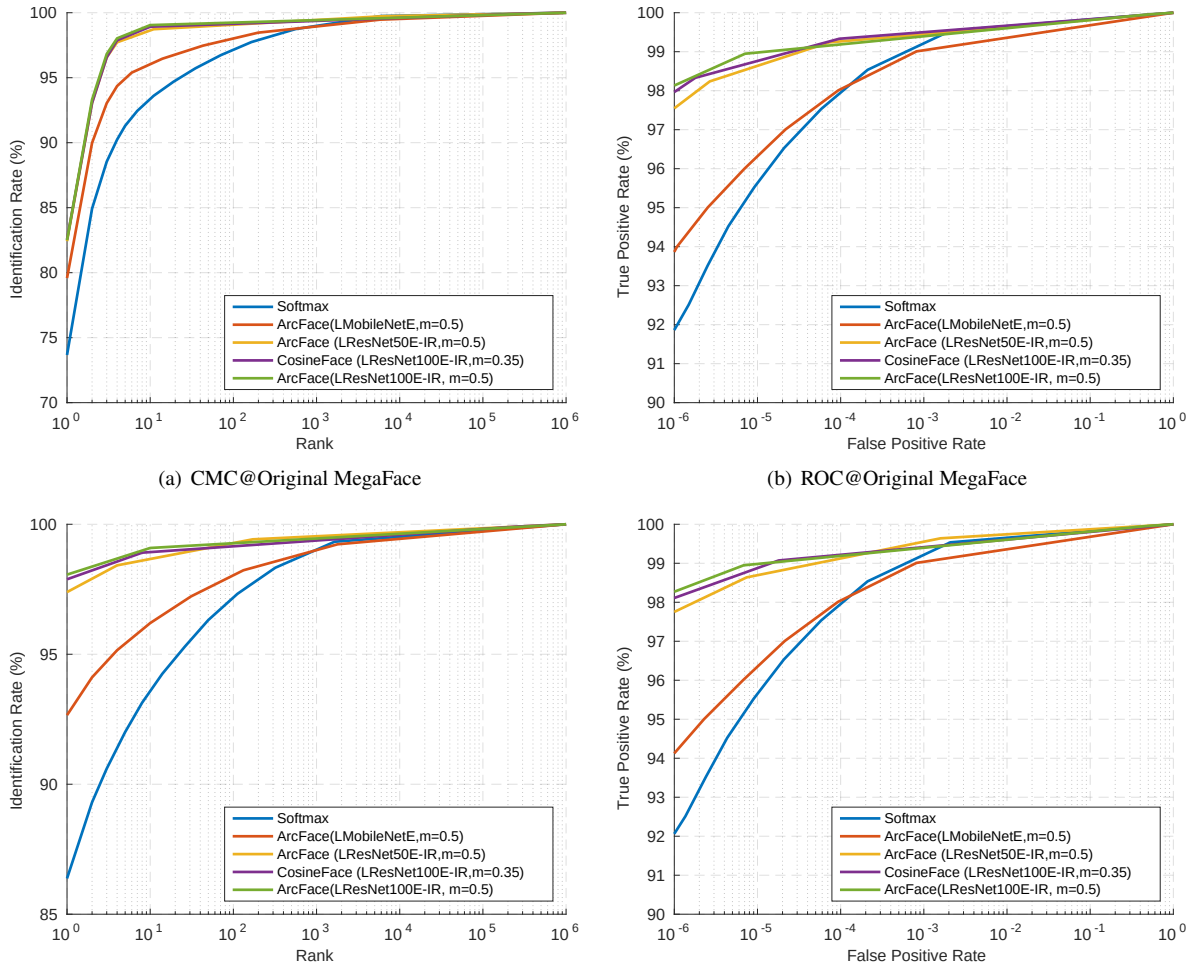
Due to the limitation of GPU memory, it is hard to train Softmax-based methods,*e.g.* SphereFace, CosineFace and ArcFace, with millions of identities. One practical solution is to employ metric learning methods, and the most widely used method is the Triplet loss [35, 22]. However, the converging speed of Triplet loss is relatively slow. To this end, we explore Triplet loss to fine-turn exist face recognition models which are trained with Softmax based methods.

For Triplet loss fine-tuning, we use the LResNet100E-IR network and set learning rate at 0.005, momentum at 0 and weight decay at $5e-4$. As shown in Table 11, we give the verification accuracy by Triplet loss fine-tuning on the AgeDB-30 dataset. We find that (1) The Softmax model trained on a dataset with fewer identity numbers (*e.g.* VGG2 with 8,631 identities) can be obviously improved by Triplet loss fine-tuning on a dataset with more identity numbers (*e.g.* MS1M with 85k identities). This improvement confirms the effectiveness of the two-step training strategy, and this strategy can significantly accelerate the whole model training compared to training Triplet loss from scratch. (2) The Softmax model can be further improved by Triplet loss fine-tuning on the same dataset, which proves that the local refinement can improve the global model. (3) The excellence of margin improved Softmax methods, *e.g.* SphereFace, CosineFace, and ArcFace, can be kept and further improved by Triplet loss fine-tuning, which also verifies that local metric learning method, *e.g.* Triplet loss, is complementary to global hypersphere metric learning based methods.

As the margin used in Triplet loss is the Euclidean distance, we will investigate Triplet loss with the angular margin recently.

Methods	Rank1@10^6	VR@FAR10^{-6}	Rank1@10^6 (R)	VR@FAR10^{-6} (R)
Softmax	73.66	91.5	86.37	91.5
CosineFace(m=0.35)	82.49	97.95	97.88	**98.07**
ArcFace(LMobileNetE,m=0.5)	79.58	93.0	92.65	94.0
ArcFace(LResNet50E-IR,m=0.5)	82.42	97.23	97.39	97.63
ArcFace(LResNet50E-IR,m=0.5)	**82.55**	**98.33**	**98.06**	97.94

Table 10. Identification and verification results of different methods on MegaFace Challenge1 (Methods@ MS1M - FaceScrub). "Rank 1" refers to the rank-1 face identification accuracy and "VR" refers to face verification TAR (True Accepted Rate) at 10^{-6} FAR (False Accepted Rate). (R) denotes the refined version of MegaFace dataset.



(a) CMC@Original MegaFace

(b) ROC@Original MegaFace

(c) CMC@Refined MegaFace

(d) ROC@Refined MegaFace

Figure 8. (a) and (c) report CMC curves of different methods with 1M distractors on MegaFace Set 1. (b) and (d) give the ROC curves of different methods with 1M distractors on MegaFace Set 1. (a) and (b) are eveluated on the original MegaFace dataset, while (c) and (d) are evaluated on the refined MegaFace Dataset.

4. Conclusions

In this paper, we contribute to improving deep face recognition from data refinement, network settings and loss function designs. We have (1) refined the largest public available training dataset (MS1M) and test dataset (MegaFace); (2) explored different network settings and analysed the trade-off between accuracy and speed; (3) proposed a geometrically interpretable loss function called ArcFace and explained why the proposed ArcFace is better than Softmax, SphereFace and CosineFace from the view of semi-hard sample distributions; (4) obtained state-of-the-art performance on the MegaFace dataset in a totally repro-

Dataset@Loss	AgeDB-30
VGG2@Softmax	94.4
VGG2@Softmax, MS1M@Triplet	97.5
MS1M@Softmax	95.56
MS1M@Softmax, MS1M@Triplet	97.16
MS1M@SphereFace	97.56
MS1M@SphereFace, MS1M@Triplet	97.85
MS1M@CosineFace	97.91
MS1M@CosineFace, MS1M@Triplet	97.98
MS1M@ArcFace	98.08
MS1M@ArcFace, MS1M@Triplet	**98.15**

Table 11. Improve verification accuracy by Triplet loss fine-tuning (LResNet100E-IR).

ducible way.

References

[1] Fg-net aging database, www-prima.inrialpes.fr/fgnet/.2002. 6

[2] http://megaface.cs.washington.edu/results/facescrub.html. 1, 2, 9

[3] https://github.com/davidsandberg/facenet. 2

[4] https://www.nist.gov/programs-projects/face-recognition-vendor-test-frvt-ongoing. 1

[5] http://www.yitutech.com/intro/. 1

[6] A. Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa. Umdfaces: An annotated face dataset for training deep networks. *arXiv:1611.01484v2*, 2016. 1

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. *arXiv:1710.08092*, 2017. 1, 2, 3, 6

[8] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274*, 2015. 7

[9] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng. Dual path networks. In *Advances in Neural Information Processing Systems*, pages 4470–4478, 2017. 8

[10] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal loss for deep face recognition. In *CVPRW*, 2017. 2, 6

[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 *European Conference on Computer Vision*, pages 87–102. Springer, 2016. 1, 2, 6

[12] D. Han, J. Kim, and J. Kim. Deep pyramidal residual networks. *arXiv:1610.02915*, 2016. 8

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 8

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2, 8

[15] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016. 2

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017. 2, 8

[17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv:1709.01507*, 2017. 8

[18] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *CVPR*, 2016. 8

[19] 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. 5, 6

[20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. 8

[21] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016. 1, 2, 5, 6

[22] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*, 2015. 10

[23] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. *CVPR*, 2017. 1, 2, 3, 4, 7, 9

[24] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, pages 507–516, 2016. 2, 3

[25] W. Liu, Y.-M. Zhang, X. Li, Z. Yu, B. Dai, T. Zhao, and L. Song. Deep hyperspherical learning. In *Advances in Neural Information Processing Systems*, pages 3953–3963, 2017. 3

[26] Y. Liu, H. Li, and X. Wang. Rethinking feature discrimination and polymerization for large-scale recognition. *arXiv:1710.00870*, 2017. 3

[27] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and 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*, 2017. 5, 6

[28] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 343–347. IEEE, 2014. 6

[29] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV*, pages 709–720, 2010. 3

[30] C. J. Parde, C. Castillo, M. Q. Hill, Y. I. Colon, S. Sankaranarayanan, J.-C. Chen, and A. J. O'Toole. Deep convolutional neural network features and the original image. *arXiv:1611.01751*, 2016. 3

[31] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 1, 2, 3

[32] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv:1701.06548*, 2017. 3

[33] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv:1703.09507*, 2017. 3

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 7

[35] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 2, 5, 10

[36] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, pages 1–9, 2016. 5, 6

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[38] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014. 8

[39] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014. 2

[40] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017. 2, 8

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2

[42] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 1, 2

[43] TencentAILab. Facecnn v1. 9/21/2017. 1, 2, 3, 4, 9

[44] F. Wang, W. Liu, H. Liu, and J. Cheng. Additive margin softmax for face verification. In *arXiv:1801.05599*, 2018. 1, 2, 3, 4, 9

[45] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: l_2 hypersphere embedding for face verification. *arXiv:1704.06369*, 2017. 3

[46] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016. 2, 3, 7

[47] X. Wu, R. He, Z. Sun, and T. Tan. A light cnn for deep face representation with noisy labels. *arXiv preprint arXiv:1511.02683*, 2015. 1

[48] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 1

[49] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 7

[50] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tail. *ICCV*, 2017. 1, 2, 3

[51] X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *arXiv:1707.01083*, 2017. 2