ArcFace: Additive Angular Margin Loss for Deep Face Recognition

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

Jia Guo *
InsightFace
guojia@gmail.com

Niannan Xue
Imperial College London
n.xue15@imperial.ac.uk

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

Abstract

One of the main challenges in feature learning using Deep Convolutional Neural Networks (DCNNs) for large-scale face recognition is the design of appropriate loss functions that enhance discriminative power. Centre loss penalises the distance between the deep features and their corresponding class centres in the Euclidean space to achieve intra-class compactness. SphereFace assumes that the linear transformation matrix in the last fully connected layer can be used as a representation of the class centres in an angular space and penalises the angles between the deep features and their corresponding weights in a multiplicative way. Recently, a popular line of research is to incorporate margins in well-established loss functions in order to maximise face class separability. In this paper, we propose an Additive Angular Margin Loss (ArcFace) to obtain highly discriminative features for face recognition. The proposed ArcFace has a clear geometric interpretation due to the exact correspondence to the geodesic distance on the hypersphere. We present arguably the most extensive experimental evaluation of all the recent state-of-the-art face recognition methods on over 10 face recognition benchmarks including a new large-scale image database with trillion level of pairs and a large-scale video dataset. We show that ArcFace consistently outperforms the state-of-the-art and can be easily implemented with negligible computational overhead. We release all refined training data, training codes, pre-trained models and training logs, which will help reproduce the results in this paper.

1. Introduction

Face representation using Deep Convolutional Neural Network (DCNN) embedding is the method of choice for face recognition [30, 31, 27, 22]. DCNNs map the face image, typically after a pose normalisation step [42], into a feature that has small intra-class and large inter-class distance.

There are two main lines of research to train DCNNs for face recognition. Those that train a multi-class classifier which can separate different identities in the training set, such by using a softmax classifier [31, 22, 5], and those that learn directly an embedding, such as the triplet loss [27]. Based on the large-scale training data and the elaborate DCNN architectures, both the softmax-loss-based methods [5] and the triplet-loss-based methods [27] can obtain excellent performance on face recognition. However,
both the softmax loss and the triplet loss have some drawbacks. For the softmax loss: (1) the size of the linear transformation matrix $W \in \mathbb{R}^{d \times n}$ increases linearly with the identities number $n$; (2) the learned features are separable for the closed-set classification problem but not discriminative enough for the open-set face recognition problem. For the triplet loss: (1) there is a combinatorial explosion in the number of face triplets especially for large-scale datasets, leading to a significant increase in the number of iteration steps; (2) semi-hard sample mining is a quite difficult problem for effective model training.

Several variants [36, 8, 43, 17, 35, 33, 6, 32, 25] have been proposed to enhance the discriminative power of the softmax loss. Wen et al. [36] pioneered the centre loss, the Euclidean distance between each feature vector and its class centre, to obtain intra-class compactness while the inter-class dispersion is guaranteed by the joint penalisation of the softmax loss. Nevertheless, updating the actual centres during training is extremely difficult as the number of face classes available for training has recently dramatically increased.

By observing that the weights from the last fully connected layer of a classification DCNN trained on the softmax loss bear conceptual similarities with the centres of each face class, the works in [17, 18] proposed a multiplicative angular margin penalty to enforce extra intra-class compactness and inter-class discrepancy simultaneously, leading to a better discriminative power of the trained model. Even though Sphereface [17] introduced the important idea of angular margin, their loss function required a series of approximations in order to be computed, which resulted in an unstable training of the network. In order to stabilise training, they proposed a hybrid loss function which includes the standard softmax loss. Empirically, the softmax loss dominates the training process, because the integer-based multiplicative angular margin makes the target logit curve very precipitous and thus hinders convergence. CosFace [35, 33] directly adds cosine margin penalty to the target logit, which obtains better performance compared to SphereFace but admits much easier implementation and relieves the need for joint supervision from the softmax loss.

In this paper, we propose an Additive Angular Margin Loss (ArcFace) to further improve the discriminative power of the face recognition model and to stabilise the training process. As illustrated in Figure 2, the dot product between the DCNN feature and the last fully connected layer is equal to the cosine distance after feature and weight normalisation. We utilise the arc-cosine function to calculate the angle between the current feature and the target weight. Afterwards, we add an additive angular margin to the target angle, and we get the target logit back again by the cosine function. Then, we re-scale all logits by a fixed feature norm, and the subsequent steps are exactly the same as in the softmax loss. The advantages of the proposed ArcFace can be summarised as follows:

**Engaging.** ArcFace directly optimises the geodesic distance margin by virtue of the exact correspondence between the angle and arc in the normalised hypersphere. We intuitively illustrate what happens in the $512$-$D$ space via analysing the angle statistics between features and weights.

**Effective.** ArcFace achieves state-of-the-art performance on ten face recognition benchmarks including large-scale image and video datasets.

**Easy.** ArcFace only needs several lines of code as given in Algorithm 1 and is extremely easy to implement in the computational-graph-based deep learning frameworks, e.g. MxNet [7], Pytorch [23] and Tensorflow [4]. Furthermore, contrary to the works in [17, 18], ArcFace does not need to be combined with other loss functions in order to have stable performance, and can easily converge on any training datasets.

**Efficient.** ArcFace only adds negligible computational complexity during training. Current GPUs can easily support millions of identities for training and the model parallel strategy can easily support many more identities.

### 2. Proposed Approach

#### 2.1. ArcFace

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

$$L_1 = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{W_{yi}^T x_i + b_{yi}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}},$$

where $x_i \in \mathbb{R}^d$ denotes the deep feature of the $i$-th sample, belonging to the $y_i$-th class. The embedding feature dimension $d$ is set to $512$ in this paper following [36, 43, 17, 35]. $W_j \in \mathbb{R}^d$ directly adds cosine margin penalty to the target logit, which obtains better performance compared to SphereFace but admits much easier implementation and relieves the need for joint supervision from the softmax loss.

For simplicity, we fix the bias $b_j = 0$ as in [17]. Then, we transform the logit [24] as $W_j^T x_i = \| W_j \| \| x_i \| \cos \theta_j$, where $\theta_j$ is the angle between the weight $W_j$ and the feature $x_i$. Following [17, 35, 34], we fix the individual weight $\| W_j \| = 1$ by $l_2$ normalisation. Following [26, 35, 34, 33], we also fix the embedding feature $\| x_i \|$ by $l_2$ normalisation and re-scale it to $s$. The normalisation step on features and
weights makes the predictions only depend on the angle between the feature and the weight. The learned embedding features are thus distributed on a hypersphere with a radius of $s$.

$$L_2 = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{s \cos \theta_{y_i}}}{\sum_{j=1, j \neq y_i}^{n} e^{s \cos \theta_j}}. \quad (2)$$

As the embedding features are distributed around each feature centre on the hypersphere, we add an additive angular margin penalty $m$ between $x_i$ and $W_{y_i}$ to simultaneously enhance the intra-class compactness and inter-class discrepancy. Since the proposed additive angular margin penalty is equal to the geodesic distance margin penalty in the normalised hypersphere, we name our method as ArcFace.

$$L_3 = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{s (\cos \theta_{y_i} + m)}}{\sum_{j=1, j \neq y_i}^{n} e^{s \cos \theta_j}}. \quad (3)$$

We select face images from 8 different identities containing enough samples (around 1,500 images/class) to train 2-D feature embedding networks with the softmax and ArcFace loss, respectively. As illustrated in Figure 3, the softmax loss provides roughly separable feature embedding but produces noticeable ambiguity in decision boundaries, while the proposed ArcFace loss can obviously enforce a more evident gap between the nearest classes.

2.2. Comparison with SphereFace and CosFace

**Numerical Similarity.** In SphereFace [17, 18], ArcFace, and CosFace [35, 33], three different kinds of margin penalty are proposed, e.g. multiplicative angular margin $m_1$, additive angular margin $m_2$, and additive cosine margin $m_3$, respectively. From the view of numerical analysis, different margin penalties, no matter add on the angle [17] or cosine space [35], all enforce the intra-class compactness
and inter-class diversity by penalising the target logit [24]. In Figure 4(b), we plot the target logit curves of SphereFace, ArcFace and CosFace under their best margin settings. We only show these target logit curves within $[20^\circ, 100^\circ]$ because the angles between $W_i$ and $x_i$ start from around $90^\circ$ (random initialisation) and end at around $30^\circ$ during ArcFace training as shown in Figure 4(a). Intuitively, there are three factors in the target logit curves that affect the performance, i.e. the starting point, the end point and the slope.

SphereFace [17] employs an annealing optimisation strategy. To avoid divergence at the beginning of training, joint supervision from softmax is used in SphereFace to weaken the multiplicative margin penalty. We implement a new version of SphereFace without the integer requirement on the margin by employing the arc-cosine function instead of using the complex double angle formula. In our implementation, we find that $m = 1.35$ can obtain similar performance compared to the original SphereFace without any convergence difficulty.

2.3. Comparison with Other Losses

Other loss functions can be designed based on the angular representation of features and weight-vectors. For examples, we can design a loss to enforce intra-class compactness and inter-class discrepancy on the hypersphere. As shown in Figure 1, we compare with three other losses in this paper.

**Intra-Loss** is designed to improve the intra-class compactness by decreasing the angle/arc between the sample and the ground truth centre.

\[
L_5 = L_2 + \frac{1}{\pi N} \sum_{i=1}^{N} \theta_{y_i}. \tag{5}
\]

**Inter-Loss** targets at enhancing inter-class discrepancy by increasing the angle/arc between different centres.

\[
L_6 = L_2 - \frac{1}{\pi N (n-1)} \sum_{i=1}^{N} \sum_{j=1, j \neq y_i}^{n} \arccos(W_{y_i}^T W_j). \tag{6}
\]

The Inter-Loss here is a special case of the Minimum Hyper-spherical Energy (MHE) method [16]. In [16], both hidden layers and output layers are regularised by MHE. In the MHE paper, a special case of loss function was also proposed by combining the SphereFace loss with MHE on the last layer of the network.

**Triplet-loss** aims at enlarging the angle/arc margin between triplet samples. In FaceNet [27], Euclidean margin is applied on the normalised features. Here, we employ the triplet-loss by the angular representation of our features as $\arccos(x_{i}^{pos} x_{i}) + m \leq \arccos(x_{i}^{neg} x_{i})$.

3. Experiments

3.1. Implementation Details

**Datasets.** As given in Table 1, we separately employ CASIA [41], VGGFace2 [5], MS1MV2 and DeepGlint-Face (including MS1M-DeepGlint and Asian-DeepGlint) [3] as our training data in order to conduct fair comparison with other methods. Please note that the proposed MS1MV2 is a semi-automatic refined version of the MS-Celeb-1M dataset [9]. To best of our knowledge, we are the first to employ ethnicity-specific annotators for large-scale face image
| Datasets           | #Identity | #Image/Video |
|--------------------|-----------|--------------|
| CASIA [41]         | 10K       | 0.5M         |
| VGGFace2 [5]       | 9.1K      | 3.3M         |
| MS1MV2             | 85K       | 5.8M         |
| MS1M-DeepGlint [3] | 87K       | 3.9M         |
| Asian-DeepGlint    | 94K       | 2.83M        |

Table 1. Face datasets for training and testing. “(P)” and “(G)” refer to the probe and gallery set, respectively.

annotations, as the boundary cases (e.g. hard samples and noisy samples) are very hard to distinguish if the annotator is not familiar with the identity. During training, we explore efficient face verification datasets (e.g. LFW [12], CFP-FP [28], AgeDB-30 [20]) to check the improvement from different settings. Besides the most widely used LFW [12] and YTF [38] datasets, we also report the performance of ArcFace on the recent large-pose and large-age datasets (e.g. CPLFW [44] and CALFW [45]). We also extensively test the proposed ArcFace on large-scale image datasets (e.g. MegaFace [14], IJB-B [37], IJB-C [19] and Trillion-Pairs [3]) and video datasets (iQIYI-VID [1]).

**Experimental Settings.** For data preprocessing, we follow the recent papers [17, 35] to generate the normalised face crops (112 × 112) by utilising five facial points. For the embedding network, we employ the widely used CNN architectures, ResNet50 and ResNet100 [11, 10]. After the last convolutional layer, we explore the BN [13]-Dropout [29]-FC-BN structure to get the final 512-D embedding feature. In this paper, we use (training dataset, network structure, loss) to facilitate understanding of the experimental settings.

We follow [35] to set the feature scale s to 64 and choose the angular margin m of ArcFace at 0.5. All experiments in this paper are implemented by MXNet [7]. We set the batch size to 512 and train models on four NVIDIA Tesla P40 (24GB) GPUs. On CASIA, the learning rate starts from 0.1 and is divided by 10 at 20K, 28K iterations. The training process is finished at 32K iterations. On MS1MV2, we divide the learning rate at 100K, 160K iterations and finish at 180K iterations. We set momentum to 0.9 and weight decay to 5e − 4. During testing, we only keep the feature embedding network without the fully connected layer (160MB for ResNet50 and 250MB for ResNet100) and extract the 512-D features (8.9 ms/face for ResNet50 and 15.4 ms/face for ResNet100) for each normalised face. To get the embedding features for templates (e.g. IJB-B and IJB-C) or videos (e.g. YTF and iQIYI-VID), we simply calculate the feature centre of all images from the template or all frames from the video. Note that, overlap identities between the training set and the test set are removed for strict evaluations, and we only use a single crop for all testing.

### 3.2. Ablation Study on Losses

In Table 2, we first explore the angular margin setting for ArcFace on the CASIA dataset with ResNet50. The best margin observed in our experiments was 0.5. Using the proposed combined margin framework in Eq. 4, it is easier to set the margin of SphereFace and CosFace which we found to have optimal performance when setting at 1.35 and 0.35, respectively. Our implementations for both SphereFace and CosFace can lead to excellent performance without observing any difficulty in convergence. The proposed ArcFace achieves the highest verification accuracy on all three test sets. In addition, we performed extensive experiments with the combined margin framework (some of the best performance was observed for CM1 (1, 0.3, 0.2) and CM2 (0.9, 0.4, 0.15)) guided by the target logit curves in Figure 4(b). The combined margin framework led to better performance than individual SphereFace and CosFace but upper-bounded by the performance of ArcFace.

Besides the comparison with margin-based methods, we conduct a further comparison between ArcFace and other losses which aim at enforcing intra-class compactness (Eq. 5) and inter-class discrepancy (Eq. 6). As the baseline we have chosen the softmax loss and we have observed performance drop on CFP-FP and AgeDB-30 after weight and feature normalisation. By combining the softmax with the intra-class loss, the performance improves on CFP-FP and AgeDB-30. However, combining the softmax with the inter-class loss only slightly improves the accuracy. The fact that Triplet-loss outperforms Norm-Softmax loss indicates the importance of margin in improving the performance. However, employing margin penalty within triplet samples is less effective than inserting margin between samples and centres as in ArcFace. Finally, we incorporate the Intra-loss, Inter-loss and Triplet-loss into ArcFace, but no improvement is observed, which leads us to believe that ArcFace is already enforcing intra-class compactness, inter-class discrepancy and classification margin.

To get a better understanding of ArcFace’s superiority, we give the detailed angle statistics on training data (CASIA) and test data (LFW) under different losses in Table 3. We find that (1) $W_j$ is nearly synchronised with embedding feature centre for ArcFace (14.29°), but there is an obvious deviation (44.26°) between $W_j$ and the em-
bedding feature centre for Norm-Softmax. Therefore, the angles between $W_j$ cannot absolutely represent the inter-class discrepancy on training data. Alternatively, the embedding feature centres calculated by the trained network are more representative. (2) Intra-Loss can effectively compress intra-class variations but also brings in smaller inter-class angles. (3) Inter-Loss can slightly increase inter-class discrepancy on both $W$ (directly) and the embedding network (indirectly), but also raises intra-class angles. (4) ArcFace already has very good intra-class compactness and inter-class discrepancy. (5) Triplet-Loss has similar intra-class compactness but inferior inter-class discrepancy compared to ArcFace. In addition, ArcFace has a more distinct margin than Triplet-Loss on the test set as illustrated in Figure 6.

| Loss Functions | LFW | CFP-FP | AgeDB-30 |
|----------------|-----|--------|----------|
| ArcFace (0.4)  | 99.53 | 95.41  | 94.98    |
| ArcFace (0.45) | 99.46 | 95.47  | 94.93    |
| ArcFace (0.5)  | 99.53 | 95.56  | 95.15    |
| ArcFace (0.55) | 99.41 | 95.32  | 95.05    |
| SphereFace [17] | 99.42 | -      | -        |
| SphereFace (1.35) | 99.11 | 94.38  | 91.70    |
| CosFace [35]   | 99.33 | -      | -        |
| CosFace (0.35) | 99.51 | 95.44  | 94.56    |
| CM1 (0.3,0.2)  | 99.48 | 95.12  | 94.38    |
| CM2 (0.9, 0.4, 0.15) | 99.50 | 95.24  | 94.86    |
| Softmax        | 99.08 | 94.39  | 92.33    |
| Norm-Softmax (NS) | 98.56 | 89.79  | 88.72    |
| NS+Intra       | 98.75 | 93.81  | 90.92    |
| NS+Inter       | 98.68 | 90.67  | 89.50    |
| NS+Intra+Inter | 98.73 | 94.00  | 91.41    |
| Triplet (0.35) | 98.98 | 91.90  | 89.98    |
| ArcFace+Intra  | 99.45 | 95.37  | 94.73    |
| ArcFace+Inter  | 99.43 | 95.25  | 94.55    |
| ArcFace+Intra+Inter | 99.43 | 95.42  | 95.10    |
| ArcFace+Triplet | 99.50 | 95.51  | 94.40    |

Table 2. Verification results (%) of different loss functions ([ICA-
SIA, ResNet50, loss*]).

3.3. Evaluation Results

Results on LFW, YTF, CALFW and CPLFW. LFW [12] and YTF [38] datasets are the most widely used benchmark for unconstrained face verification on images and videos. In this paper, we follow the unrestricted with labelled outside data protocol to report the performance. As reported in Table 4, ArcFace trained on MS1MV2 with ResNet100 beats the baselines (e.g. SphereFace [17] and CosFace [35]) by a significant margin on both LFW and YTF, which shows that the additive angular margin penalty can notably enhance the discriminative power of deeply learned features, demonstrating the effectiveness of ArcFace.

Besides on LFW and YTF datasets, we also report the performance of ArcFace on the recently introduced datasets (e.g. CPLFW [44] and CALFW [45]) which show higher pose and age variations with same identities from LFW. Among all of the open-sourced face recognition models, the ArcFace model is evaluated as the top-ranked face recognition model as shown in Table 5, outperforming counterparts by an obvious margin. In Figure 7, we illustrate the angle distributions (predicted by ArcFace model trained

![Figure 6](image_url)  
(a) ArcFace  
(b) Triplet-Loss  

Figure 6. Angle distributions of all positive pairs and random negative pairs (~ 0.5M) from LFW. Red area indicates positive pairs while blue indicates negative pairs. All angles are represented in degree. ([ICA-
SIA, ResNet50, loss*]).

Table 3. The angle statistics under different losses ([CASIA-
SIA, ResNet50, loss*]). Each column denotes one particular loss. “W-
EC” refers to the mean of angles between $W_i$ and the correspond-
ing embedding feature centre. “W-Inter” refers to the mean of min-
umimum angles between $W_j$’s. “Inter1” and “Inter2” refer to the mean of angles between $x_i$ and the embedding feature centre on
CASIA and LFW, respectively. “Intra1” and “Intra2” refer to the mean of minimum angles between embedding feature centres on
CASIA and LFW, respectively.

| Method               | #Image | LFW   | YTF   |
|----------------------|--------|-------|-------|
| DeepID [30]          | 0.2M   | 99.47 | 93.20 |
| Deep Face [31]       | 4.4M   | 97.35 | 91.4  |
| VGG Face [22]        | 2.6M   | 98.95 | 97.30 |
| FaceNet [27]         | 200M   | 99.63 | 95.10 |
| Baidu [15]           | 1.3M   | 99.13 | -     |
| Center Loss [36]     | 0.7M   | 99.28 | 94.9  |
| Range Loss [43]      | 5M     | 99.52 | 93.70 |
| Marginal Loss [8]    | 3.8M   | 99.48 | 95.98 |
| SphereFace [17]      | 0.5M   | 99.42 | 95.0  |
| SphereFace+ [16]     | 0.5M   | 99.47 | -     |
| CosFace [35]         | 5M     | 99.73 | 97.6  |
| MS1MV2, R100, ArcFace | 5.8M   | 99.83 | 98.02 |

Table 4. Verification performance (%) of different methods on LFW and YTF.
Table 5. Verification performance (%) of open-sourced face recognition models on LFW, CALFW and CPLFW.

| Method                       | LFW  | CALFW | CPLFW |
|------------------------------|------|-------|-------|
| HUMAN-Individual             | 97.27| 82.32 | 81.21 |
| HUMAN-Fusion                 | 99.85| 86.50 | 85.24 |
| Center Loss [36]             | 98.75| 85.48 | 77.48 |
| SphereFace [17]              | 99.27| 90.30 | 81.40 |
| VGGFace2 [5]                 | 99.43| 90.57 | 84.00 |
| MS1MV2, R100, ArcFace        | 99.82| 95.45 | 92.08 |

Table 6. Face identification and verification evaluation of different methods on MegaFace Challenge1 using FaceScrub as the probe set. “Id” refers to the rank-1 face identification accuracy with 1M distractors, and “Ver” refers to the face verification TAR at 10−6 FAR. “R” refers to data refinement on both probe set and 1M distractors. ArcFace obtains state-of-the-art performance under both small and large protocols.

| Methods                     | Id (%) | Ver (%) |
|-----------------------------|--------|---------|
| Softmax [17]                | 54.85  | 65.92   |
| Contrastive Loss [17, 30]   | 65.21  | 78.86   |
| Triplet [17, 27]            | 64.79  | 78.32   |
| Center Loss [36]            | 65.49  | 80.14   |
| SphereFace [17]             | 72.729 | 85.561  |
| CosFace [35]                | 77.11  | 89.88   |
| AM-Softmax [33]             | 72.47  | 84.44   |
| SphereFace+ [16]            | 73.03  | -       |
| CASIA, R50, ArcFace         | 77.50  | 92.34   |
| CASIA, R50, ArcFace, R      | 91.75  | 93.69   |
| FaceNet [27]                | 70.49  | 86.47   |
| CosFace [35]                | 82.72  | 96.65   |
| MS1MV2, R100, ArcFace       | 81.03  | 96.98   |
| MS1MV2, R100, CosFace       | 80.56  | 96.56   |
| MS1MV2, R100, ArcFace, R    | 98.35  | 98.48   |
| MS1MV2, R100, CosFace, R    | 97.91  | 97.91   |

Results on MegaFace. The MegaFace dataset [14] includes 1M images of 69K different individuals as the gallery set and 100K photos of 530 unique individuals from FaceScrub [21] as the probe set. On MegaFace, there are two testing scenarios (identification and verification) under two protocols (large or small training set). The training set is defined as large if it contains more than 0.5M images. For the fair comparison, we train ArcFace on CAISA and MS1MV2 under the small protocol and large protocol, respectively. In Table 6, ArcFace trained on CAISA achieves the best single-model identification and verification performance, not only surpassing the strong baselines (e.g. SphereFace [17] and CosFace [35]) but also outperforming other published methods [36, 16].

As we observed an obvious performance gap between identification and verification, we performed a thorough manual check in the whole MegaFace dataset and found many face images with wrong labels, which significantly affects the performance. Therefore, we manually refined the whole MegaFace dataset and report the correct performance of ArcFace on MegaFace. On the refined MegaFace, ArcFace still clearly outperforms CosFace and achieves the best performance on both verification and identification.

Under large protocol, ArcFace surpasses FaceNet [27] by a clear margin and obtains comparable results on identification and better results on verification compared to CosFace [35]. Since CosFace employs a private training data, we retrain CosFace on our MS1MV2 dataset with ResNet100. Under fair comparison, ArcFace shows superiority over CosFace and forms an upper envelope of CosFace under both identification and verification scenarios as shown in Figure 8.

Results on IJB-B and IJB-C. The IJB-B dataset [37] contains 1,845 subjects with 21.8K still images and 55K
Table 7. 1:1 verification TAR (@FAR=1e-4) on the IJB-B and IJB-C datasets.

| Method                  | IJB-B | IJB-C |
|-------------------------|-------|-------|
| ResNet50 [5]            | 0.784 | 0.825 |
| SENet50 [5]             | 0.800 | 0.840 |
| ResNet50+SENet50 [5]    | 0.800 | 0.841 |
| MN-v [40]               | 0.818 | 0.852 |
| MN-vc [40]              | 0.831 | 0.862 |
| ResNet50+DCN(Kpts) [39]| 0.850 | 0.867 |
| ResNet50+DCN(Divs) [39]| 0.841 | 0.880 |
| SENet50+DCN(Kpts) [39]  | 0.846 | 0.874 |
| SENet50+DCN(Divs) [39]  | 0.849 | 0.885 |
| VGG2, R50, ArcFace      | 0.898 | 0.921 |
| MS1MV2, R100, ArcFace   | 0.942 | 0.956 |

Table 8. Identification and verification results (%) on the Trillion-Pairs dataset. ([Dataset*, ResNet100, ArcFace])

| Method                  | Id (@FPR=1e-3) | Ver(@FPR=1e-9) |
|-------------------------|----------------|----------------|
| CASIA                   | 26.643         | 21.452         |
| MS1MV2                  | 80.968         | 78.600         |
| DeepGlint-Face          | 80.331         | 78.586         |
| MS1MV2+Asian            | 84.840 (1st)   | 80.540         |
| CIGIT_IRSEC             | 84.234 (2nd)   | 81.558 (1st)   |

Table 9. MAP of our method on the iQIYI-VID test set. “MLP” refers to a three-layer fully connected network trained on the iQIYI-VID training data.

Results on iQIYI-VID. The iQIYI-VID challenge [1] contains 565,372 video clips (training set 219,677, validation set 172,860, and test set 172,835) of 4934 identities from iQIYI variety shows, films and television dramas. The length of each video ranges from 1 to 30 seconds. This dataset supplies multi-modal cues, including face, cloth, voice, gait and subtitles, for character identification. The iQIYI-VID dataset employs MAP@100 as the evaluation indicator. MAP (Mean Average Precision) refers to the overall average accuracy rate, which is the mean of the average accuracy rate of the corresponding videos of person ID retrieved in the test set for each person ID (as the query) in the training set.

As shown in Table 9, ArcFace trained on combined MS1MV2 and Asian datasets with ResNet100 sets a high baseline (MAP=(79.80%). Based on the embedding feature for each training video, we train an additional three-layer fully connected network with a classification loss to get the customised feature descriptor on the iQIYI-VID dataset. The MLP learned on the iQIYI-VID training set significantly boosts the MAP by 6.60%. Drawing support from the model ensemble and context features from the off-the-shelf object and scene classifier [2], our final result surpasses the runner-up by a clear margin (0.99%).

Results on Trillion-Pairs. The Trillion-Pairs dataset [3] provides 1.58M images from Flickr as the gallery set and 274K images from 5.7k LFW [12] identities as the probe set. Every pair between gallery and probe set is used for evaluation (0.4 trillion pairs in total). In Table 8, we compare the performance of ArcFace trained on different datasets. The proposed MS1MV2 dataset obviously boosts the performance compared to CASIA and even slightly outperforms the DeepGlint-Face dataset, which has a double identity number. When combining all identities from MS1MV2 and Asian celebrities from DeepGlint, ArcFace achieves the best identification performance 84.840% (@FPR=1e-3) and comparable verification performance compared to the most recent submission (CIGIT_IRSEC) from the lead-board.
4. Conclusions

In this paper, we proposed an Additive Angular Margin Loss function, which can effectively enhance the discriminative power of feature embeddings learned via DCNNs for face recognition. In the most comprehensive experiments reported in the literature we demonstrate that our method consistently outperforms the state-of-the-art. Code and details will provided publicly available.

References

[1] http://challenge.ai.iqiyi.com/ 5, 8
[2] http://data.mxnet.io/models/ 8
[3] [17] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In ICML, 2016. 2, 3
[4] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song. Learning towards minimum hyperspherical energy. In NIPS, 2018. 4, 6, 7
[5] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Spherenet: Deep hypersphere embedding for face recognition. In CVPR, 2017. 1, 2, 3, 4, 5, 6, 7
[6] W. Liu, Y. Wen, Z. Yu, and M. Yang. Large-margin softmax loss for convolutional neural networks. In ICML, 2016. 2, 3
[7] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, and J. Cheney. Iarpa janus benchmark–c: Face dataset and protocol. In ICB, 2018. 2, 5
[8] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kosia, and S. Zafeiriou. Agedb: The first manually collected in-the-wild age database. In CVPR Workshop, 2017. 2, 5
[9] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In ICIP, 2014. 7
[10] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In BMVC, 2015. 1, 2, 6
[11] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In NIPS Workshop, 2017. 2
[12] G. Pereyra, G. Tucker, J.chorowski, L. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. arXiv:1701.06548, 2017. 2, 3
[13] X. Qi and L. Zhang. Face recognition via centralized coordinate learning. arXiv:1801.05678, 2018. 2
[14] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. arXiv:1703.09507, 2017. 2
[15] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In CVPR, 2015. 1, 4, 6, 7
[16] S. Sengupta, J.-C. Chen, C. Castillo, V. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In WACV, 2016. 2, 5
[17] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. JML, 2014. 5
[18] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In NIPS, 2014. 1, 6, 7
[19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Learning from unstructured biometric data. In CVPR, 2014. 1, 6
[20] W. Wan, Y. Zhong, T. Li, and J. Chen. Rethinking feature distribution for loss functions in image classification. arXiv:1803.02988, 2018. 2
[21] F. Wang, W. Liu, H. Liu, and J. Cheng. Adaptive margin softmax for face verification. IEEE Signal Processing Letters, 2018. 2, 3, 7
[22] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: L2 hypersphere embedding for face verification. arXiv:1704.06369, 2017. 2
[23] Y. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In CVPR, 2018. 1, 2, 3, 5, 6, 7
[24] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In ECCV, 2016. 2, 6, 7
[25] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. C. Adams, T. Miller, N. D. Kalka, A. K. Jain, J. A. Duncan, and
K. Allen. Iarpa janus benchmark-b face dataset. In CVPR Workshop, 2017.

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

[39] W. Xie, S. Li, and A. Zisserman. Comparator networks. In ECCV, 2018.

[40] W. Xie and A. Zisserman. Multicolumn networks for face recognition. In BMVC, 2018.

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

[42] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 2016.

[43] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tail. In ICCV, 2017.

[44] T. Zheng and W. Deng. Cross-pose Ifw: A database for studying cross-pose face recognition in unconstrained environments. Technical Report, 2018.

[45] T. Zheng, W. Deng, and J. Hu. Cross-age Ifw: A database for studying cross-age face recognition in unconstrained environments. arXiv:1708.08197, 2017.