arXiv:1801.07698v3 [cs.CV] 9 Feb 2019

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[1], 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 [32, 33, 29, 24]. DCNNs map the face image, typically after a pose normalisation step [45], 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 [33, 24, 6], and those that learn directly an embedding, such as the triplet loss [29]. Based on the large-scale training data and the elaborate DCNN architectures, both the softmax-loss-based methods [6] and the triplet-loss-based methods [29] can obtain excellent performance on face recognition. However,

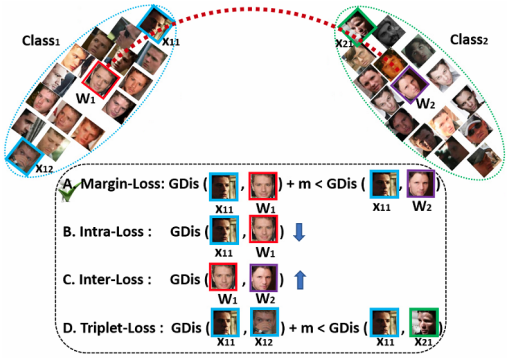


Figure 1. Based on the centre [18] and feature [37] normalisation, all identities are distributed on a hypersphere. To enhance intra-class compactness and inter-class discrepancy, we consider four kinds of Geodesic Distance (GDis) constraint. (A) Margin-Loss: insert a geodesic distance margin between the sample and centres. (B) Intra-Loss: decrease the geodesic distance between the sample and the corresponding centre. (C) Inter-Loss: increase the geodesic distance between different centres. (D) Triplet-Loss: insert a geodesic distance margin between triplet samples. In this paper, we propose an Additive Angular Margin Loss (ArcFace), which is exactly corresponded to the geodesic distance (Arc) margin penalty in (A), to enhance the discriminative power of face recognition model. Extensive experimental results show that the strategy of (A) is most effective.

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

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 [38, 9, 46, 18, 37, 35, 7, 34, 27] have been proposed to enhance the discriminative power of the softmax loss. Wen *et al*. [38] 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 [18, 19] 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 [18] 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 [37, 35] 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 [8], Pytorch [25] and Tensorflow [4]. Furthermore, contrary to the works in [18, 19], 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:

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

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 [38, 46, 18, 37]. $W_j \in \mathbb{R}^d$ denotes the j-th column of the weight $W \in \mathbb{R}^{d\times n}$ and $b_j \in \mathbb{R}^n$ is the bias term. The batch size and the class number are N and n, respectively. Traditional softmax loss is widely used in deep face recognition [24, 6]. However, the softmax loss function does not explicitly optimise the feature embedding to enforce higher similarity for intra-class samples and diversity for inter-class samples, which results in a performance gap for deep face recognition under large intra-class appearance variations (*e.g.* pose variations [30, 48] and age gaps [22, 49]) and large-scale test scenarios (*e.g.* million [15, 39, 21] or trillion pairs [2]).

For simplicity, we fix the bias $b_j = 0$ as in [18]. Then, we transform the logit [26] as $W_j^T x_i = \|W_j\|\,\|x_i\|\cos\theta_j$, where θ_j is the angle between the weight W_j and the feature x_i. Following [18, 37, 36], we fix the individual weight $\|W_j\| = 1$ by l_2 normalisation. Following [28, 37, 36, 35], we also fix the embedding feature $\|x_i\|$ by l_2 normalisation and re-scale it to s. The normalisation step on features and

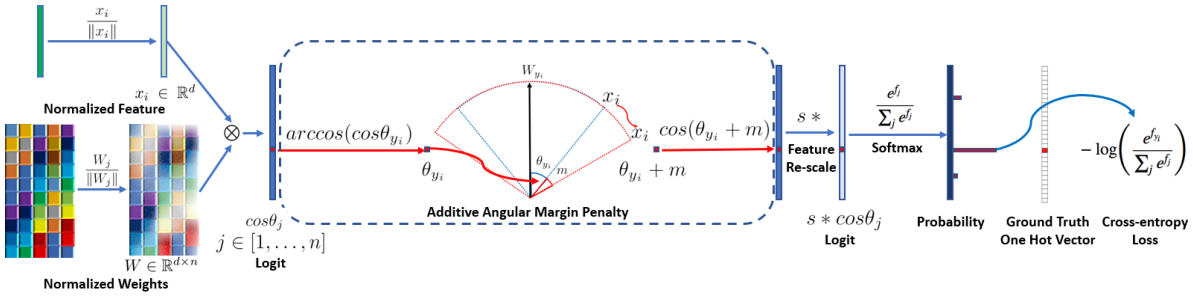


Figure 2. Training a DCNN for face recognition supervised by the ArcFace loss. Based on the feature x_i and weight W normalisation, we get the $\cos\theta_j$ (logit) for each class as $W_j^T x_i$. We calculate the $arccos\theta_{y_i}$ and get the angle between the feature x_i and the ground truth weight W_{y_i}. In fact, W_j provides a kind of centre for each class. Then, we add an angular margin penalty m on the target (ground truth) angle θ_{y_i}. After that, we calculate $\cos(\theta_{y_i} + m)$ and multiply all logits by the feature scale s. The logits then go through the softmax function and contribute to the cross entropy loss.

Algorithm 1 The Pseudo-code of ArcFace on MxNet

Input: Feature Scale s, Margin Parameter m in Eq. 3, Class Number n, Ground-Truth ID gt.

1. x = mx.symbol.L2Normalization (x, mode = 'instance')
2. W = mx.symbol.L2Normalization (W, mode = 'instance')
3. fc7 = mx.sym.FullyConnected (data = x, weight = W, no_bias = True, num_hidden = n)
4. original_target_logit = mx.sym.pick (fc7, gt, axis = 1)
5. theta = mx.sym.arccos (original_target_logit)
6. marginal_target_logit = mx.sym.cos (theta + m)
7. one_hot = mx.sym.one_hot (gt, depth = n, on_value = 1.0, off_value = 0.0)
8. fc7 = fc7 + mx.sym.broadcast_mul (one_hot, mx.sym.expand_dims (marginal_target_logit - original_target_logit, 1))
9. fc7 = fc7 * s

Output: Class-wise affinity score $fc7$.

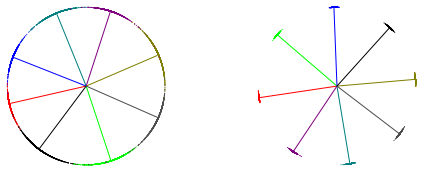
weights makes the predictions only depend on the angle between the feature and the weight. The learned embedding features are thus distributed on a hypersphere with a radius of s.

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

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.

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

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.



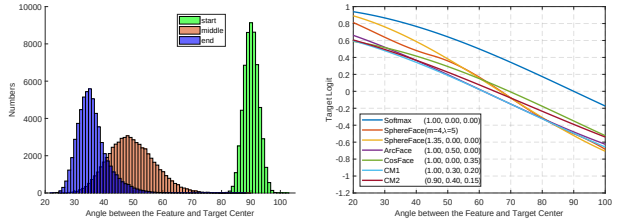
(a) Softmax (b) ArcFace

Figure 3. Toy examples under the softmax and ArcFace loss on 8 identities with 2D features. Dots indicate samples and lines refer to the centre direction of each identity. Based on the feature normalisation, all face features are pushed to the arc space with a fixed radius. The geodesic distance gap between closest classes becomes evident as the additive angular margin penalty is incorporated.

2.2. Comparison with SphereFace and CosFace

Numerical Similarity. In SphereFace [18, 19], ArcFace, and CosFace [37, 35], 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 [18] or cosine space [37], all enforce the intra-class compactness

and inter-class diversity by penalising the target logit [26]. 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 $[20°, 100°]$ because the angles between W_{y_i} and x_i start from around $90°$ (random initialisation) and end at around $30°$ 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.



(a) θ_j Distributions (b) Target Logits Curves

Figure 4. Target logit analysis. (a) θ_j distributions from start to end during ArcFace training. (2) Target logit curves for softmax, SphereFace, ArcFace, CosFace and combined margin penalty $(\cos(m_1\theta + m_2) - m_3)$.

By combining all of the margin penalties, we implement SphereFace, ArcFace and CosFace in an united framework with m_1, m_2 and m_3 as the hyper-parameters.

$$L_4 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}. \quad (4)$$

As shown in Figure 4(b), by combining all of the above-motioned margins $(\cos(m_1\theta + m_2) - m_3)$, we can easily get some other target logit curves which also have high performance.

Geometric Difference. Despite the numerical similarity between ArcFace and previous works, the proposed additive angular margin has a better geometric attribute as the angular margin has the exact correspondence to the geodesic distance. As illustrated in Figure 5, we compare the decision boundaries under the binary classification case. The proposed ArcFace has a constant linear angular margin throughout the whole interval. By contrast, SphereFace and CosFace only have a nonlinear angular margin.

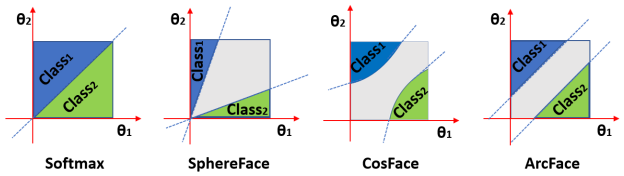


Figure 5. Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.

The minor difference in margin designs can have "butterfly effect" on the model training. For example, the original SphereFace [18] 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.

$$L_5 = L_2 + \frac{1}{\pi N}\sum_{i=1}^{N}\theta_{y_i}. \quad (5)$$

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

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

The Inter-Loss here is a special case of the Minimum Hyper-spherical Energy (MHE) method [17]. In [17], 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 loss on the last layer of the network.

Triplet-loss aims at enlarging the angle/arc margin between triplet samples. In FaceNet [29], 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 [43], VGGFace2 [6], MS1MV2 and DeepGlint-Face (including MS1M-DeepGlint and Asian-DeepGlint) [2] 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 [10]. To best of our knowledge, we are the first to employ ethnicity-specific annotators for large-scale face image

Datasets	#Identity	#Image/Video
CASIA [43]	10K	0.5M
VGGFace2 [6]	9.1K	3.3M
MS1MV2	85K	5.8M
MS1M-DeepGlint [2]	87K	3.9M
Asian-DeepGlint [2]	94 K	2.83M
LFW [13]	5,749	13,233
CFP-FP [30]	500	7,000
AgeDB-30 [22]	568	16,488
CPLFW [48]	5,749	11,652
CALFW [49]	5,749	12,174
YTF [40]	1,595	3,425
MegaFace [15]	530 (P)	1M (G)
IJB-B [39]	1,845	76.8K
IJB-C [21]	3,531	148.8K
Trillion-Pairs [2]	5,749 (P)	1.58M (G)
iQIYI-VID [20]	4,934	172,835

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 [13], CFP-FP [30], AgeDB-30 [22]) to check the improvement from different settings. Besides the most widely used LFW [13] and YTF [40] datasets, we also report the performance of ArcFace on the recent large-pose and large-age datasets(*e.g*. CPLFW [48] and CALFW [49]). We also extensively test the proposed ArcFace on large-scale image datasets (*e.g*. MegaFace [15], IJB-B [39], IJB-C [21] and Trillion-Pairs [2]) and video datasets (iQIYI-VID [20]).

Experimental Settings. For data prepossessing, we follow the recent papers [18, 37] 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 [12, 11]. After the last convolutional layer, we explore the BN [14]-Dropout [31]-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 [37] 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 [8]. 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.

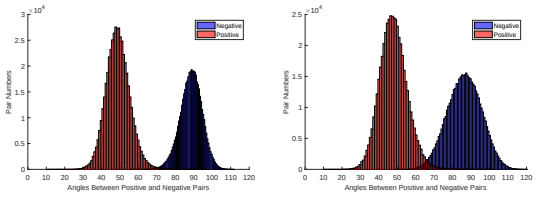
Besides the comparison with margin-based methods, we conduct a further comparison between ArcFace and other losses which aim at enforcing intra-class compactness (Eq. 5) and inter-class discrepancy (Eq. 6). As the baseline we have chosen the softmax loss and we have observed performance drop on CFP-FP 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-

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 [18]	99.42	-	-
SphereFace (1.35)	99.11	94.38	91.70
CosFace [37]	99.33	-	-
CosFace (0.35)	99.51	95.44	94.56
CM1 (1, 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 ([CASIA, ResNet50, loss*]).

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.



(a) ArcFace (b) Triplet-Loss

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

	NS	ArcFace	IntraL	InterL	TripletL
W-EC	44.26	14.29	8.83	46.85	-
W-Inter	69.66	71.61	31.34	75.66	-
Intra1	50.50	38.45	17.50	52.74	41.19
Inter1	59.23	65.83	24.07	62.40	50.23
Intra2	33.97	28.05	12.94	35.38	27.42
Inter2	65.60	66.55	26.28	67.90	55.94

Table 3. The angle statistics under different losses ([CASIA, ResNet50, loss*]). Each column denotes one particular loss. "W-EC" refers to the mean of angles between W_j and the corresponding embedding feature centre. "W-Inter" refers to the mean of minimum angles between W_j's. "Intra1" and "Intra2" refer to the mean of angles between x_i and the embedding feature centre on CASIA and LFW, respectively. "Inter1" and "Inter2" refer to the mean of minimum angles between embedding feature centres on CASIA and LFW, respectively.

Method	#Image	LFW	YTF
DeepID [32]	0.2M	99.47	93.20
Deep Face [33]	4.4M	97.35	91.4
VGG Face [24]	2.6M	98.95	97.30
FaceNet [29]	200M	99.63	95.10
Baidu [16]	1.3M	99.13	-
Center Loss [38]	0.7M	99.28	94.9
Range Loss [46]	5M	99.52	93.70
Marginal Loss [9]	3.8M	99.48	95.98
SphereFace [18]	0.5M	99.42	95.0
SphereFace+ [17]	0.5M	99.47	-
CosFace [37]	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.

3.3. Evaluation Results

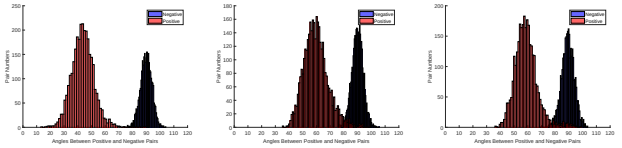
Results on LFW, YTF, CALFW and CPLFW. LFW [13] and YTF [40] datasets are the most widely used benchmark for unconstrained face verification on images and videos. In this paper, we follow the *unrestricted with labelled outside data* protocol to report the performance. As reported in Table 4, ArcFace trained on MS1MV2 with ResNet100 beats the baselines (*e.g.* SphereFace [18] and CosFace [37]) 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 [48] and CALFW [49]) 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

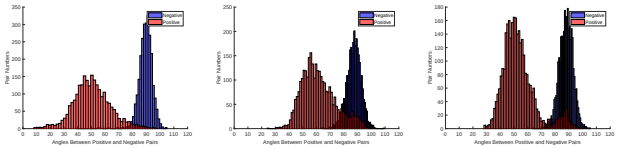
Method	LFW	CALFW	CPLFW
HUMAN-Individual	97.27	82.32	81.21
HUMAN-Fusion	99.85	86.50	85.24
Center Loss [38]	98.75	85.48	77.48
SphereFace [18]	99.27	90.30	81.40
VGGFace2 [6]	99.43	90.57	84.00
MS1MV2, R100, ArcFace	**99.82**	**95.45**	**92.08**

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

on MS1MV2 with ResNet100) of both positive and negative pairs on LFW, CFP-FP, AgeDB-30, YTF, CPLFW and CALFW. We can clearly find that the intra-variance due to pose and age gaps significantly increases the angles between positive pairs thus making the best threshold for face verification increasing and generating more confusion regions on the histogram.



(a) LFW (99.83%) (b) CFP-FP (98.37%) (c) AgeDB (98.15%)



(d) YTF (98.02%) (e) CPLFW (92.08%) (f) CALFW (95.45%)

Figure 7. Angle distributions of both positive and negative pairs on LFW, CFP-FP, AgeDB-30, YTF, CPLFW and CALFW. Red area indicates positive pairs while blue indicates negative pairs. All angles are represented in degree. ([MS1MV2, ResNet100, ArcFace])

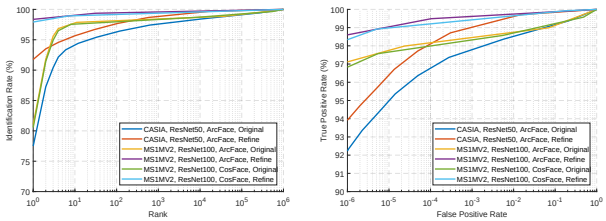
Results on MegaFace. The MegaFace dataset [15] includes 1M images of 690K different individuals as the gallery set and 100K photos of 530 unique individuals from FaceScrub [23] as the probe set. On MegaFace, there are two testing scenarios (identification and verification) under two protocols (large 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 CASIA achieves the best single-model identification and verification performance, not only surpassing the strong baselines (*e.g.* SphereFace [18] and CosFace [37]) but also outperforming other published methods [38, 17].

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.

Methods	Id (%)	Ver (%)
Softmax [18]	54.85	65.92
Contrastive Loss[18, 32]	65.21	78.86
Triplet [18, 29]	64.79	78.32
Center Loss[38]	65.49	80.14
SphereFace [18]	72.729	85.561
CosFace [37]	77.11	89.88
AM-Softmax [35]	72.47	84.44
SphereFace+ [17]	73.03	-
CASIA, R50, ArcFace	77.50	92.34
CASIA, R50, ArcFace, R	91.75	93.69
FaceNet [29]	70.49	86.47
CosFace [37]	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

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.

Under large protocol, ArcFace surpasses FaceNet [29] by a clear margin and obtains comparable results on identification and better results on verification compared to CosFace [37]. 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.



(a) CMC (b) ROC

Figure 8. CMC and ROC curves of different models on MegaFace. Results are evaluated on both original and refined MegaFace dataset.

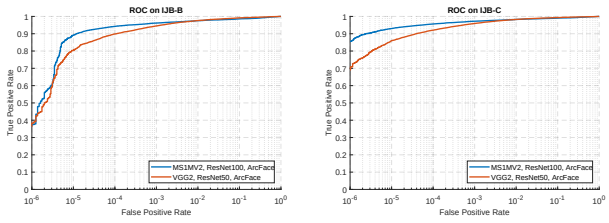
Results on IJB-B and IJB-C. The IJB-B dataset [39] contains 1,845 subjects with 21.8K still images and 55K

Method	IJB-B	IJB-C
ResNet50 [6]	0.784	0.825
SENet50 [6]	0.800	0.840
ResNet50+SENet50 [6]	0.800	0.841
MN-v [42]	0.818	0.852
MN-vc [42]	0.831	0.862
ResNet50+DCN(Kpts) [41]	0.850	0.867
ResNet50+DCN(Divs) [41]	0.841	0.880
SENet50+DCN(Kpts) [41]	0.846	0.874
SENet50+DCN(Divs) [41]	0.849	0.885
VGG2, R50, ArcFace	0.898	0.921
MS1MV2, R100, ArcFace	**0.942**	**0.956**

Table 7. 1:1 verification TAR (@FAR=1e-4) on the IJB-B and IJB-C dataset.

frames from $7,011$ videos. In total, there are $12,115$ templates with $10,270$ genuine matches and 8M impostor matches. The IJB-C dataset [39] is a further extension of IJB-B, having $3,531$ subjects with 31.3K still images and 117.5K frames from $11,779$ videos. In total, there are $23,124$ templates with $19,557$ genuine matches and $15,639$K impostor matches.

On the IJB-B and IJB-C datasets, we employ the VGG2 dataset as the training data and the ResNet50 as the embedding network to train ArcFace for the fair comparison with the most recent methods [6, 42, 41]. In Table 7, we compare the TAR (@FAR=1e-4) of ArcFace with the previous state-of-the-art models [6, 42, 41]. ArcFace can obviously boost the performance on both IJB-B and IJB-C (about $3 \sim 5\%$, which is a significant reduction in the error). Drawing support from more training data (MS1MV2) and deeper neural network (ResNet100), ArcFace can further improve the TAR (@FAR=1e-4) to 94.2% and 95.6% on IJB-B and IJB-C, respectively. In Figure 9, we show the full ROC curves of the proposed ArcFace on IJB-B and IJB-C [2], and ArcFace achieves impressive performance even at FAR=1e-6 setting a new baseline.



(a) ROC for IJB-B (b) ROC for IJB-C

Figure 9. ROC curves of 1:1 verification protocol on the IJB-B and IJB-C dataset.

Results on Trillion-Pairs. The Trillion-Pairs dataset [2] provides 1.58M images from Flickr as the gallery set and 274K images from 5.7k LFW [13] 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.

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

Results on iQIYI-VID. The iQIYI-VID challenge [20] 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 [1], our final result surpasses the runner-up by a clear margin (0.99%).

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

[2] https://github.com/deepinsight/insightface/tree/master/Evaluation/IJB

Method	MAP(%)
MS1MV2+Asian, R100, ArcFace	79.80
+ MLP	86.40
+ Ensemble	88.26
+ Context	**88.65 (1st)**
Other Participant	87.66 (2nd)

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.

consistently outperforms the state-of-the-art. Code and details have been released under the MIT license.

5. Appendix

5.1. Parallel Acceleration

Can we apply ArcFace on large-scale identities? Yes, millions of identities are not a problem.

The concept of Centre (W) is indispensable in ArcFace, but the parameter size of Centre (W) is proportional to the number of classes. When there are millions of identities in the training data, the proposed ArcFace confronts with substantial training difficulties, *e.g.* excessive GPU memory consumption and massive computational cost, even at a prohibitive level.

In our implementation [3], we employ a parallel acceleration strategy [44] to relieve this problem. We optimise our training code to easily and efficiently support **million** level identities on a single machine by parallel acceleration on both feature x (it known as the general data parallel strategy) and centre W (we named it as the centre parallel strategy). As shown in Figure 10, our parallel acceleration on both feature x and centre W can significantly decrease the GPU memory consumption and accelerate the training speed. Even for one million identities trained on 8*1080ti (11GB), our implementation (ResNet 50, batch size 8*64, feature dimension 512 and float point 32) can still run at 800 samples per second. Compared to the approximate acceleration method proposed in [47], our implementation has no performance drop.

In Figure 11, we illustrate the main calculation steps of the parallel acceleration by simple matrix partition, which can be easily grasped and reproduced by beginners [3].

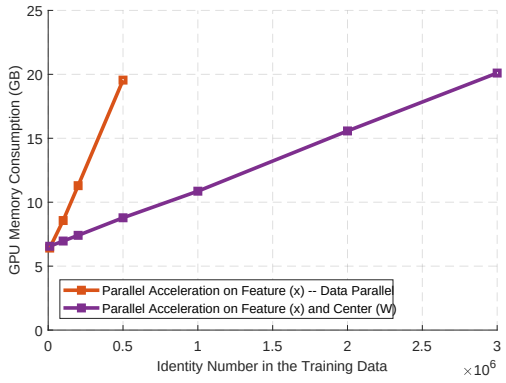
(1) Get feature (x). Face embedding features are aggregated into one feature matrix (batch size 8*64 $\times$ feature dimension 512) from 8 GPU cards. The size of the aggregated feature matrix is only 1MB, and the communication cost is negligible when we transfer the feature matrix.

(2) Get similarity score matrix ($score = xW$). We copy the feature matrix into each GPU, and concurrently multiply the feature matrix by the centre sub-matrix (feature dimension 512 $\times$ identity number 1M/8) to get the similarity score sub-matrix (batch size 512 $\times$ identity number 1M/8) on each GPU. The similarity score matrix goes forward to calculate the ArcFace loss and the gradient. Here, we conduct a simple matrix partition on the centre matrix and the similarity score matrix along the identity dimension, and there is no communication cost on the centre and similarity score matrix. Both the centre sub-matrix and the similarity score sub-matrix are only 256MB on each GPU.

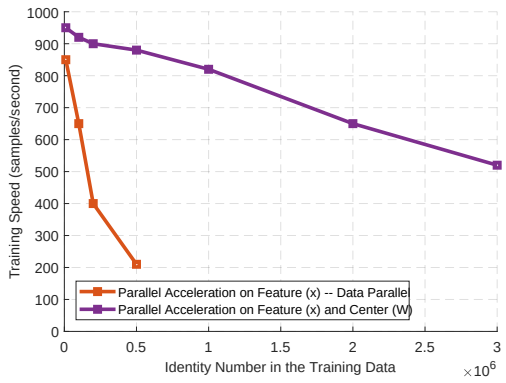
(3) Get gradient on centre (dW). We transpose the feature matrix on each GPU, and concurrently multiply the transposed feature matrix by the gradient sub-matrix of the similarity score.

(4) Get gradient on feature (x). We concurrently multiply the gradient sub-matrix of similarity score by the transposed centre sub-matrix and sum up the outputs from 8 GPU cards to get the gradient on feature x.

Considering the communication cost (MB level), our implementation of ArcFace can be easily and efficiently trained on millions of identities by clusters.



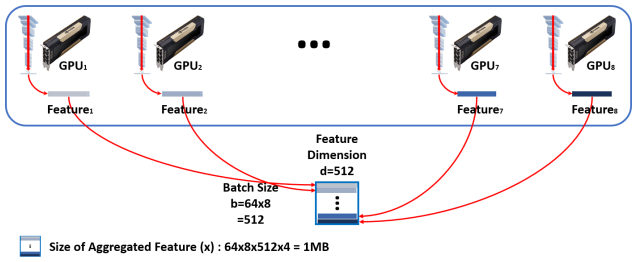
(a) GPU Memory



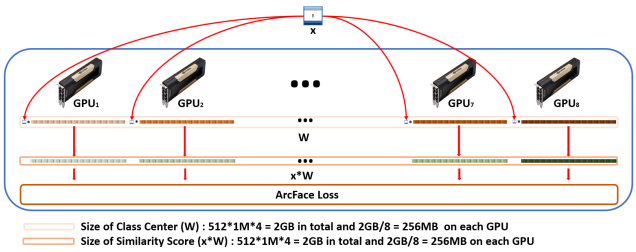
(b) Training Speed

Figure 10. Parallel acceleration on both feature x and centre W. Setting: ResNet 50, batch size 8*64, feature dimension 512, float point 32, GPU 8*P40 (24GB).

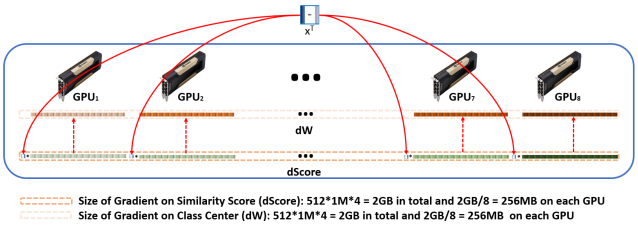
[3] https://github.com/deepinsight/insightface/tree/master/recognition



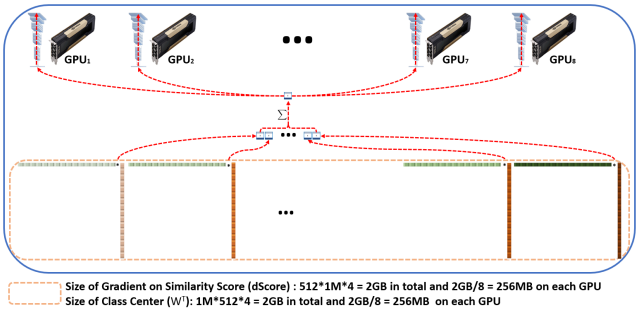
(a) x



(b) $score = xW$



(c) $dW = x^T dscore$



(d) $dx = dscore W^T$

Figure 11. Parallel calculation by simple matrix partition. Setting: ResNet 50, batch size 8*64, feature dimension 512, float point 32, identity number 1 Million, GPU 8 * 1080ti (11GB). **Communication cost: 1MB (feature x). Training speed: 800 samples/second**.

5.2. Feature Space Analysis

Is the 512-d hypersphere space large enough to hold large-scale identities? Theoretically, Yes.

We assume that the identity centre W_j's follow a realistically spherical uniform distribution, the expectation of the nearest neighbour separation[5] is

$$\mathbb{E}[\theta(W_j)] \to n^{-\frac{2}{d-1}}\Gamma(1+\frac{1}{d-1})(\frac{\Gamma(\frac{d}{2})}{2\sqrt{\pi}(d-1)\Gamma(\frac{d-1}{2})})^{-\frac{1}{d-1}}, \tag{7}$$

where d is the space dimension, n is the identity number, and $\theta(W_j) = \min_{1\leq i,j\leq n, i\neq j} \arccos(W_i, W_j) \forall i, j$. In Figure 12, we give $\mathbb{E}[\theta(W_j)]$ in the 128-d, 256-d and 512-d space with the class number ranging from $10K$ to $100M$. The high-dimensional space is so large that $\mathbb{E}[\theta(W_j)]$ decreases slowly when the class number increases exponentially.

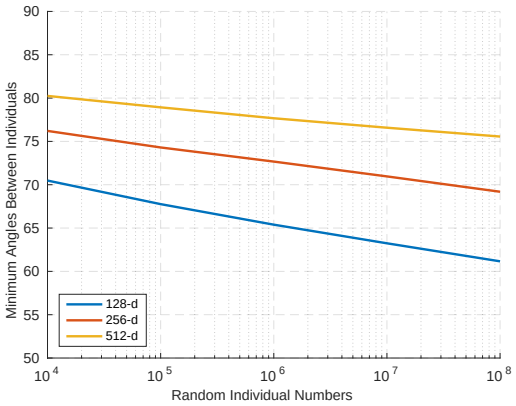


Figure 12. The high-dimensional space is so large that the mean of the nearest angles decreases slowly when the class number increases exponentially.

References

[1] http://data.mxnet.io/models/. 8

[2] http://trillionpairs.deepglint.com/overview. 2, 4, 5, 8

[3] Stanford cs class cs231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/neural-networks-case-study/. 9

[4] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467*, 2016. 2

[5] J. S. Brauchart, A. B. Reznikov, E. B. Saff, I. H. Sloan, Y. G. Wang, and R. S. Womersley. Random point sets on the spherehole radii, covering, and separation. *Experimental Mathematics*, 2018. 10

[6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018. 1, 2, 4, 5, 7, 8

[7] B. Chen, W. Deng, and J. Du. Noisy softmax: improving the generalization ability of dcnn via postponing the early softmax saturation. In *CVPR*, 2017. 2

[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. 2, 5

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

[10] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016. 4

[11] D. Han, J. Kim, and J. Kim. Deep pyramidal residual networks. *arXiv:1610.02915*, 2016. 5

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[13] 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, 2007. 5, 6, 8

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[15] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016. 2, 5, 7

[16] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv:1506.07310*, 2015. 6

[17] 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

[18] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7

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

[20] Y. Liu, P. Shi, B. Peng, H. Yan, Y. Zhou, B. Han, Y. Zheng, C. Lin, J. Jiang, and Y. Fan. iqiyi-vid: A large dataset for multi-modal person identification. *arXiv:1811.07548*, 2018. 5, 8

[21] 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

[22] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou. Agedb: The first manually collected in-the-wild age database. In *CVPR Workshop*, 2017. 2, 5

[23] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2014. 7

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

[25] 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

[26] G. Pereyra, G. Tucker, J. Chorowski, Ł. Kaiser, and G. Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv:1701.06548*, 2017. 2, 3

[27] X. Qi and L. Zhang. Face recognition via centralized coordinate learning. *arXiv:1801.05678*, 2018. 2

[28] R. Ranjan, C. D. Castillo, and R. Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv:1703.09507*, 2017. 2

[29] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 4, 6, 7

[30] 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*, 2016. 2, 5

[31] 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

[32] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014. 1, 6, 7

[33] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 1, 6

[34] W. Wan, Y. Zhong, T. Li, and J. Chen. Rethinking feature distribution for loss functions in image classification. *arXiv:1803.02988*, 2018. 2

[35] F. Wang, W. Liu, H. Liu, and J. Cheng. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 2018. 2, 3, 7

[36] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: l_2 hypersphere embedding for face verification. *arXiv:1704.06369*, 2017. 2

[37] H. 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

[38] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 2, 6, 7

[39] 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. 2, 5, 7, 8

[40] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011. 5, 6

[41] W. Xie, S. Li, and A. Zisserman. Comparator networks. In *ECCV*, 2018. 8

[42] W. Xie and A. Zisserman. Multicolumn networks for face recognition. In *BMVC*, 2018. 8

[43] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014. 4, 5

[44] D. Zhang. A distributed training solution for face recognition. *DeepGlint*, 2018. 9

[45] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016. 1

[46] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao. Range loss for deep face recognition with long-tail. In *ICCV*, 2017. 2, 6

[47] X. Zhang, L. Yang, J. Yan, and D. Lin. Accelerated training for massive classification via dynamic class selection. In *AAAI*, 2018. 9

[48] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Technical Report*, 2018. 2, 5, 6

[49] T. Zheng, W. Deng, and J. Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv:1708.08197*, 2017. 2, 5, 6