

# ArcFace: Additive Angular Margin Loss for Deep Face Recognition

Jiankang Deng<sup>\* 1,2,3</sup>Jia Guo<sup>\* 2</sup>Niannan Xue<sup>1</sup>Stefanos Zafeiriou<sup>1,3</sup><sup>1</sup>Imperial College London<sup>2</sup>InsightFace<sup>3</sup>FaceSoft

{j.deng16, n.xue15, s.zafeiriou}@imperial.ac.uk, guojia@gmail.com

## 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 can enhance the **discriminative power**. **Centre loss** **penalises** the distance between 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 the angular space and therefore penalises the angles between 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 its exact correspondence to **geodesic distance** on a **hypersphere**. We present arguably the most extensive experimental evaluation against all recent state-of-the-art face recognition methods on ten face recognition benchmarks which includes a new large-scale image database with trillions 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. To facilitate future research, code has been made available at: <https://github.com/deepinsight/insightface>

3차원보다 큰 차원으로 확장한 것

## 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

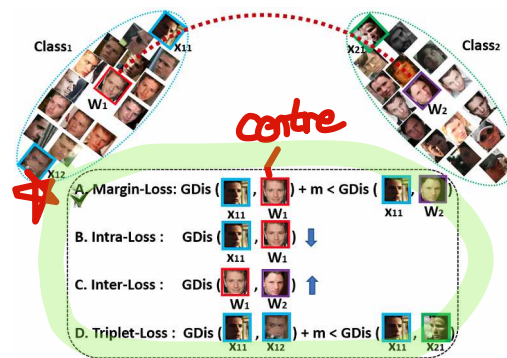


Figure 1. Based on the centre [15] and feature [35] normalisation, all identities are distributed on a hypersphere. To enhance intra-class compactness and inter-class discrepancy, we consider four kinds of Geodesic Distance (GDis) constraint. (A) Margin-Loss: insert a geodesic distance margin between 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.

feature that should have small intra-class and large inter-class distance.

There are two main lines of research to train DCNNs for face recognition. Some train a **multi-class classifier** which can separate different identities in the training set, such by using a **softmax classifier** [31, 22, 3], and the others 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 [3] 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)

\* Equal contributions.

InsightFace is a nonprofit Github project for 2D and 3D face analysis.

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, 6, 43, 15, 35, 33, 4, 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 [15, 16] 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 [15] 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 [5], Pytorch [23] and Tensorflow [2]. Furthermore, contrary to the works in [15, 16], 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:

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

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 [36, 43, 15, 35].  $W_j \in \mathbb{R}^d$  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 [22, 3]. 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 [28, 44] and age gaps [19, 45]) and large-scale test scenarios (*e.g.* million [12, 37, 18] or trillion pairs [1]).

For simplicity, we fix the bias  $b_j = 0$  as in [15]. 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 [15, 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  $s$ .

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

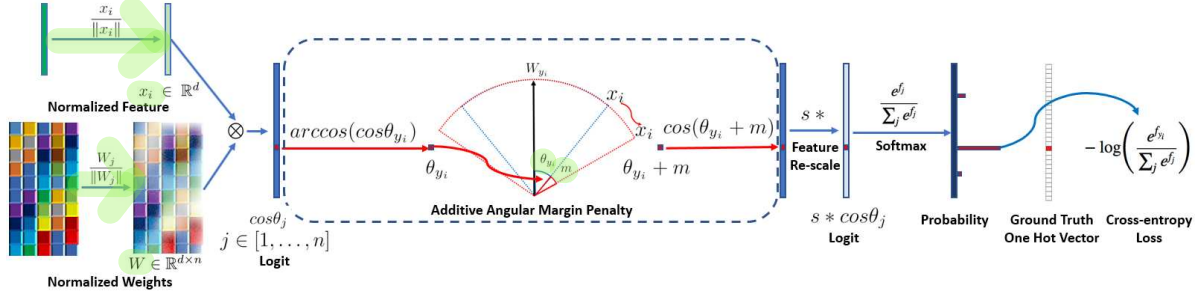


Figure 2. Training a DCNN for face recognition supervised by the ArcFace loss. Based on the feature  $x_i$  and weight  $W$  normalisation, we get the  $\cos \theta_j$  (logit) for each class as  $W_j^T x_i$ . We calculate the  $\arccos \theta_{y_i}$  and get the angle between the feature  $x_i$  and 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  $\theta_{y_i}$ . 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.  $x = \text{mx.symbol.L2Normalization}(x, \text{mode} = \text{'instance'})$
2.  $W = \text{mx.symbol.L2Normalization}(W, \text{mode} = \text{'instance'})$
3.  $\text{fc7} = \text{mx.sym.FullyConnected}(\text{data} = x, \text{weight} = W, \text{no\_bias} = \text{True}, \text{num\_hidden} = n)$
4.  $\text{original\_target\_logit} = \text{mx.sym.pick}(\text{fc7}, \text{gt}, \text{axis} = 1)$
5.  $\text{theta} = \text{mx.sym.arccos}(\text{original\_target\_logit})$
6.  $\text{marginal\_target\_logit} = \text{mx.sym.cos}(\text{theta} + m)$
7.  $\text{one\_hot} = \text{mx.sym.one\_hot}(\text{gt}, \text{depth} = n, \text{on\_value} = 1.0, \text{off\_value} = 0.0)$
8.  $\text{fc7} = \text{fc7} + \text{mx.sym.broadcast\_mul}(\text{one\_hot}, \text{mx.sym.expand\_dims}(\text{marginal\_target\_logit} - \text{original\_target\_logit}, 1))$
9.  $\text{fc7} = \text{fc7} * s$

**Output:** Class-wise affinity score  $\text{fc7}$ .

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.

## 2.2. Comparison with SphereFace and CosFace

**Numerical Similarity.** In SphereFace [15, 16], 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 mar-

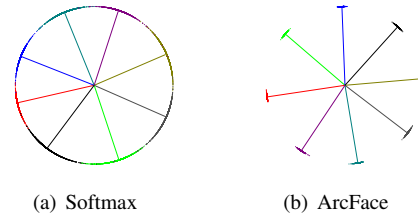


Figure 3. Toy examples under the softmax and ArcFace loss on 8 identities with 2D features. Dots indicate samples and lines refer to the centre direction of each identity. Based on the feature normalisation, all face features are pushed to the arc space with a fixed radius. The geodesic distance gap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

gin  $m_3$ , respectively. From the view of numerical analysis, different margin penalties, no matter add on the angle [15] 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  $W_{y_i}$  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 perfor-

mance, *i.e.* the starting point, the end point and the slope.

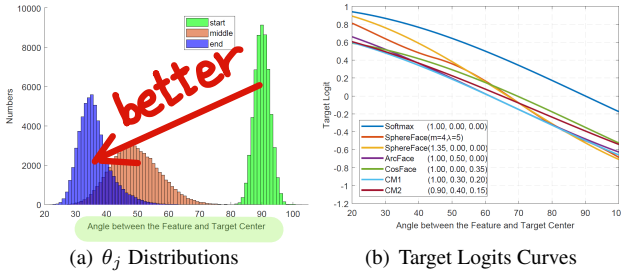


Figure 4. Target logit analysis. (a)  $\theta_j$  distributions from start to end during ArcFace training. (2) Target logit curves for softmax, SphereFace, ArcFace, CosFace and combined margin penalty ( $\cos(m_1\theta + m_2) - m_3$ ).

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-motivated 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.

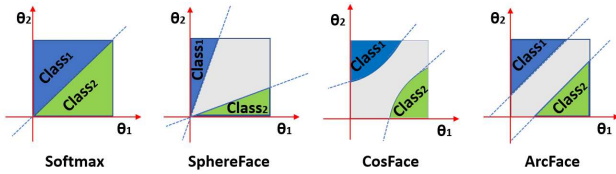


Figure 5. Decision margins of different loss functions under binary classification case. The dashed line represents the decision boundary, and the grey areas are the decision margins.

The minor difference in margin designs can have “butterfly effect” on the model training. For example, the original SphereFace [15] 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 [14]. In [14], 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 [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 [3], MS1MV2 and DeepGlint-Face (including MS1M-DeepGlint and Asian-DeepGlint) [1] 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 [7]. To best of our knowledge, we are the first to employ ethnicity-specific annotators for large-scale face image 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 [10], CFP-FP [28], AgeDB-30 [19]) to check the improvement from different settings. Besides the most widely used LFW [10] and YTF [38] datasets, we also report the performance of ArcFace on the recent large-pose and large-age datasets(*e.g.*



Datasets	#Identity	#Image/Video
CASIA [41]	10K	0.5M
VGGFace2 [3]	9.1K	3.3M
MS1MV2	85K	5.8M
MS1M-DeepGlint [1]	87K	3.9M
Asian-DeepGlint [1]	94 K	2.83M
LFW [10]	5,749	13,233
CFP-FP [28]	500	7,000
AgeDB-30 [19]	568	16,488
CPLFW [44]	5,749	11,652
CALFW [45]	5,749	12,174
YTF [38]	1,595	3,425
MegaFace [12]	530 (P)	1M (G)
IJB-B [37]	1,845	76.8K
IJB-C [18]	3,531	148.8K
Trillion-Pairs [1]	5,749 (P)	1.58M (G)
iQIYI-VID [17]	4,934	172,835

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

CPLFW [44] and CALFW [45]). We also extensively test the proposed ArcFace on large-scale image datasets (e.g. MegaFace [12], IJB-B [37], IJB-C [18] and Trillion-Pairs [1]) and video datasets (iQIYI-VID [17]).

**Experimental Settings.** For data preprocessing, we follow the recent papers [15, 35] to generate the normalised face crops ( $112 \times 112$ ) by utilising five facial points. For the embedding network, we employ the widely used CNN architectures, ResNet50 and ResNet100 [9, 8]. After the last convolutional layer, we explore the BN [11]-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 [5]. 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 ( $14.29^\circ$ ), but there is an obvious deviation ( $44.26^\circ$ ) between  $W_j$  and the embedding 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

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)	<b>99.53</b>	<b>95.56</b>	<b>95.15</b>
ArcFace (0.55)	99.41	95.32	95.05
SphereFace [15]	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 (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\*]).

	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.

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.

### 3.3. Evaluation Results

**Results on LFW, YTF, CALFW and CPLFW.** LFW [10] 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 [15] and CosFace [35]) by a significant margin on both LFW and YTF, which shows

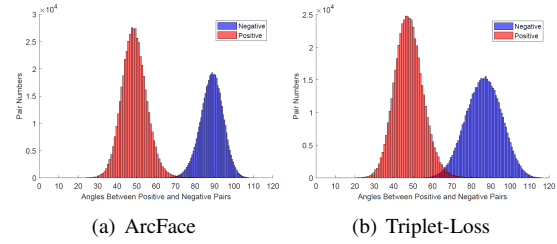


Figure 6. Angle distributions of all positive pairs and random negative pairs ( $\sim 0.5M$ ) from LFW. Red area indicates positive pairs while blue indicates negative pairs. All angles are represented in degree. ([CASIA, ResNet50, loss\*]).

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 [13]	1.3M	99.13	-
Center Loss [36]	0.7M	99.28	94.9
Range Loss [43]	5M	99.52	93.70
Marginal Loss [6]	3.8M	99.48	95.98
SphereFace [15]	0.5M	99.42	95.0
SphereFace+ [14]	0.5M	99.47	-
CosFace [35]	5M	99.73	97.6
MS1MV2, R100, ArcFace	5.8M	<b>99.83</b>	<b>98.02</b>

Table 4. Verification performance (%) of different methods on LFW and YTF.

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 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.

**Results on MegaFace.** The MegaFace dataset [12] includes 1M images of 690K 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

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 [15]	99.27	90.30	81.40
VGGFace2 [3]	99.43	90.57	84.00
MS1MV2, R100, ArcFace	<b>99.82</b>	<b>95.45</b>	<b>92.08</b>

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

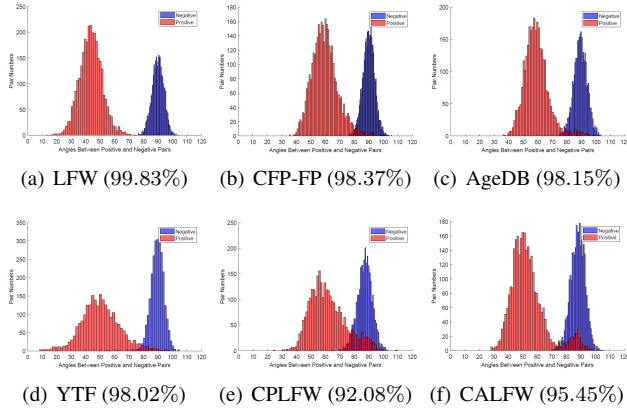


Figure 7. Angle distributions of both positive and negative pairs on LFW, CFP-FP, AgeDB-30, YTF, CPLFW and CALFW. Red area indicates positive pairs while blue indicates negative pairs. All angles are represented in degree. ([MS1MV2, ResNet100, ArcFace])

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 [15] and CosFace [35]) but also outperforming other published methods [36, 14].

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]

Methods	Id (%)	Ver (%)
Softmax [15]	54.85	65.92
Contrastive Loss[15, 30]	65.21	78.86
Triplet [15, 27]	64.79	78.32
Center Loss[36]	65.49	80.14
SphereFace [15]	72.729	85.561
CosFace [35]	77.11	89.88
AM-Softmax [33]	72.47	84.44
SphereFace+ [14]	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

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.

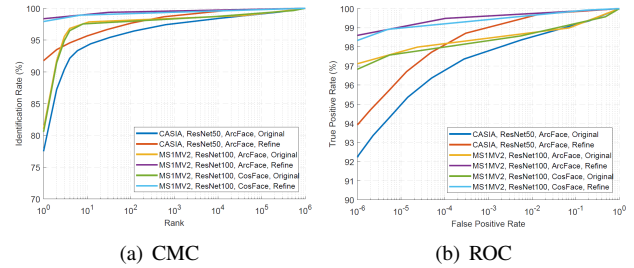


Figure 8. CMC and ROC curves of different models on MegaFace. Results are evaluated on both original and refined MegaFace dataset.

contains 1,845 subjects with 21.8K still images and 55K 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 [37] 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 15,639K 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 [3, 40, 39]. In Table 7, we compare the TAR (@FAR=1e-4) of ArcFace with the previous state-of-the-art models [3, 40, 39]. ArcFace can obviously boost the performance on both IJB-B and IJB-C (about 3 ~ 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

Method	IJB-B	IJB-C
ResNet50 [3]	0.784	0.825
SENet50 [3]	0.800	0.840
ResNet50+SENet50 [3]	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	<b>0.942</b>	<b>0.956</b>

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

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, and ArcFace achieves impressive performance even at FAR=1e-6 setting a new baseline.

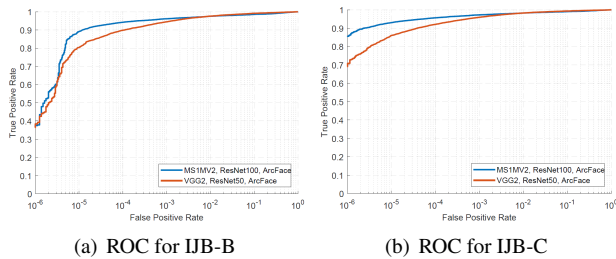


Figure 9. ROC curves of 1:1 verification protocol on the IJB-B and IJB-C dataset.

**Results on Trillion-Pairs.** The Trillion-Pairs dataset [1] provides 1.58M images from Flickr as the gallery set and 274K images from 5.7k LFW [10] 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.

**Results on iQIYI-VID.** The iQIYI-VID challenge [17] 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

<https://github.com/deepinsight/insightface/tree/master/Evaluation/IJB>

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	<b>84.840</b> (1st)	80.540
CIGIT_IRSEC	84.234 (2nd)	<b>81.558</b> (1st)

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

Method	MAP(%)
MS1MV2+Asian, R100, ArcFace	79.80
+ MLP	86.40
+ Ensemble	88.26
+ Context	<b>88.65</b> (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.

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 [20], our final result surpasses the runner-up by a clear margin (0.99%).

## 4. Conclusions

In this paper, we propose an Additive Angular Margin Loss function, which can effectively enhance the discriminative power of feature embeddings learned via DCNNs for face recognition. We demonstrate that our method consistently outperforms the state of the art through the most comprehensive experiments. Codes with detailed explanations are released to facilitate reproducibility of the results reported in this paper.

**Acknowledgements.** Jiankang Deng acknowledges financial support from the Imperial President’s PhD Scholarship and GPU donations from NVIDIA. Stefanos Zafeiriou acknowledges support from EPSRC Fellowship DEFORM (EP/S010203/1), FACER2VM (EP/N007743/1) and a Google Faculty Fellowship.



## References

- [1] <http://trillionpairs.deepglint.com/overview>. 2, 4, 5, 8
- [2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467*, 2016. 2
- [3] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018. 1, 2, 4, 5, 7, 8
- [4] Binghui Chen, Weihong Deng, and Junping Du. Noisy softmax: improving the generalization ability of dcnn via postponing the early softmax saturation. In *CVPR*, 2017. 2
- [5] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274*, 2015. 2, 5
- [6] Jiankang Deng, Yuxiang Zhou, and Stefanos Zafeiriou. Marginal loss for deep face recognition. In *CVPR Workshop*, 2017. 2, 6
- [7] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016. 4
- [8] Dongyoon Han, Jiwon Kim, and Junmo Kim. Deep pyramidal residual networks. *arXiv:1610.02915*, 2016. 5
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5
- [10] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, 2007. 4, 5, 6, 8
- [11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5
- [12] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016. 2, 5, 6
- [13] Jingtuo Liu, Yafeng Deng, Tao Bai, Zhengping Wei, and Chang Huang. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv:1506.07310*, 2015. 6
- [14] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, 2018. 4, 6, 7
- [15] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 1, 2, 3, 4, 5, 6, 7
- [16] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. 2, 3
- [17] Yuanliu Liu, Peipei Shi, Bo Peng, He Yan, Yong Zhou, Bing Han, Yi Zheng, Chao Lin, Jianbin Jiang, and Yin Fan. iqiivid: A large dataset for multi-modal person identification. *arXiv:1811.07548*, 2018. 5, 8
- [18] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, and Jordan Cheney. Iarpa janus benchmark-c: Face dataset and protocol. In *ICB*, 2018. 2, 5
- [19] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected in-the-wild age database. In *CVPR Workshop*, 2017. 2, 4, 5
- [20] MxNetModels. <http://data.mxnet.io/models/>. 8
- [21] Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2014. 6
- [22] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015. 1, 2, 6
- [23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NeurIPS Workshop*, 2017. 2
- [24] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv:1701.06548*, 2017. 2, 3
- [25] Xianbiao Qi and Lei Zhang. Face recognition via centralized coordinate learning. *arXiv:1801.05678*, 2018. 2
- [26] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv:1703.09507*, 2017. 2
- [27] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 1, 4, 6, 7
- [28] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016. 2, 4, 5
- [29] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JML*, 2014. 5
- [30] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *NeurIPS*, 2014. 1, 6, 7
- [31] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014. 1, 6
- [32] Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. *arXiv:1803.02988*, 2018. 2
- [33] Feng Wang, Weiyang Liu, Haijun Liu, and Jian Cheng. Additive margin softmax for face verification. *SPL*, 2018. 2, 3, 7
- [34] Feng Wang, Xiang Xiang, Jian Cheng, and Alan L Yuille. Normface:  $l_2$  hypersphere embedding for face verification. *arXiv:1704.06369*, 2017. 2
- [35] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jingchao Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018. 1, 2, 3, 5, 6, 7
- [36] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 2, 6, 7

- [37] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn C Adams, Tim Miller, Nathan D Kalka, Anil K Jain, James A Duncan, and Kristen Allen. Iarpa janus benchmark-b face dataset. In *CVPR Workshop*, 2017. 2, 5, 7
- [38] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR*, 2011. 4, 5, 6
- [39] Weidi Xie, Shen Li, and Andrew Zisserman. Comparator networks. In *ECCV*, 2018. 7, 8
- [40] Weidi Xie and Andrew Zisserman. Multicolumn networks for face recognition. In *BMVC*, 2018. 7, 8
- [41] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014. 4, 5
- [42] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *SPL*, 2016. 1
- [43] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tail. In *ICCV*, 2017. 2, 6
- [44] Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Technical Report*, 2018. 2, 5, 6
- [45] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv:1708.08197*, 2017. 2, 5, 6