ElasticFace: Elastic Margin Loss for Deep Face Recognition

Fadi Boutros[1,2], Naser Damer[1,2], Florian Kirchbuchner[1], Arjan Kuijper[1,2]

[1]Fraunhofer Institute for Computer Graphics Research IGD, Darmstadt, 64283 Germany
[2]Department of Computer Science, TU Darmstadt, Darmstadt, 64289 Germany
Email: fadi.boutros@igd.fraunhofer.de

Abstract

Learning discriminative face features plays a major role in building high-performing face recognition models. The recent state-of-the-art face recognition solutions proposed to incorporate a fixed penalty margin on commonly used classification loss function, softmax loss, in the normalized hypersphere to increase the discriminative power of face recognition models, by minimizing the intra-class variation and maximizing the inter-class variation. Marginal penalty softmax losses, such as ArcFace and CosFace, assume that the geodesic distance between and within the different identities can be equally learned using a fixed penalty margin. However, such a learning objective is not realistic for real data with inconsistent inter-and intra-class variation, which might limit the discriminative and generalizability of the face recognition model. In this paper, we relax the fixed penalty margin constrain by proposing elastic penalty margin loss (ElasticFace) that allows flexibility in the push for class separability. The main idea is to utilize random margin values drawn from a normal distribution in each training iteration. This aims at giving the decision boundary chances to extract and retract to allow space for flexible class separability learning. We demonstrate the superiority of our ElasticFace loss over ArcFace and CosFace losses, using the same geometric transformation, on a large set of mainstream benchmarks. From a wider perspective, our ElasticFace has advanced the state-of-the-art face recognition performance on seven out of nine mainstream benchmarks. All training codes, pre-trained models, training logs will be publicly released [1].

1. Introduction

Face recognition technologies are increasingly deployed to enhance the security and convenience of processes involving identity verification, such as border control and financial services. The typical pipeline of a face recog-

---

[1]https://github.com/fdbtrs/ElasticFace

nition system involves mapping the face image (after detection and alignment [31]) into a feature vector (embedding) [2, 4, 27]. Two face images are then compared by comparing their relative embeddings and therefore, measuring the degree of identity similarity between both faces. Knowing that it is intuitive that such embeddings should ideally have small intra-class and large inter-class variation, with the class here being an identity. This corresponds to a face recognition system that still makes correct genuine decisions (same identity) even when face images are largely varied (pose, age, expression, etc.), and make correct imposter (not same identity) decision even when the appearance of the face image pair of different identities is very similar. To achieve that, different solutions opted to train deep neural networks by either directly learning the embedding (e.g. Triplet loss [23]) or by learning an identity classification problem (e.g. Softmax loss [2]). One of the main challenges for training with metric-based learning such as Triple [23], n-pair [25], or contrastive [3] losses, is training the model with a large-scale dataset as the number of possible triplets explodes with the number of samples. Alternatively, classification-based losses such as softmax loss can be easily adopted for training a face recognition model as it does not pose that issue. However, the softmax loss does not directly optimize the feature embedding needed for face verification. Liu et al. [17] proposed a large-margin softmax (L-Softmax) by incorporating angular margin constraints on softmax loss to encourage intra-class compactness and inter-class separability between learned features. SphereFace [16] extended L-Softmax by normalizing the weights of the last full-connected layer and deploying multiplicative angular penalty margin between the deep features and their corresponding weights. Different from SphereFace, CosFace [27] proposed additive cosine margin on the cosine angle between the deep features and their corresponding weights. CosFace also proposed to fix the norm of the deep features and their corresponding weights to 1, then scaling the deep feature norm to a constant s, achieving better performance on mainstream face recognition benchmarks. Later, ArcFace [4] proposed addi-

tive angular margin by deploying angular penalty margin on the angle between the deep features and their corresponding weights. The great success of softmax loss with penalty margin motivated several works to propose a novel variant of softmax loss [1, 5, 10, 12, 14, 15, 19, 26]. All these solutions achieved notable accuracies on mainstream benchmarks [9, 18, 24, 28] for face recognition. Huang et al. [10] proposed an Adaptive Curriculum Learning loss based on margin-based softmax loss. The proposed loss targets the easy samples at an early stage of training and the hard ones at a later stage of training. Jiao et al. [12] proposed Dyn-arcface based on ArcFace loss [4] by replacing the fixed margin value of ArcFace with an adaptive one. The margin value of Dyn-arcface is adjusted based on the distance between each class center and the other class centers. However, this might not reflect the real properties of the class separability, but rather their separability in the current stage of the model training. Kim et al. [14] proposed to enrich the feature representation learned by ArcFace loss with group-aware representations. UniformFace [5] suggested to equalize distances between all the classes centers by adding a new loss function to SphereFace loss [16]. A recent work by An et al. [1] presented an efficient distributed sampling algorithm (Partial-FC). The Partial-FC method is based on randomly sampling a small subset of the complete training set of classes for the softmax-based loss function. Thus, it enables the training of the face recognition model on a massive number of identities. The authors experimentally proved that training with only 10% of training samples using CosFace [27] and ArcFace [4] can achieve comparable results on mainstream benchmarks to the case when training is performed on a complete set of classes. MagFace [19] deployed magnitude-aware margin on ArcFace loss to enhance intra-class compactness by pulling high-quality samples close to class centers while pushing low-quality samples away. However, this is based on the weak assumption of optimal face quality (utility) estimation. Moreover, this might prevent the model from convergence when the most of training samples in the training dataset are of low quality.

The main challenge for the majority of the previously listed works is the fine selection of the ideal margin penalty value. In this work, we propose the ElasticFace loss that relaxes the fixed single margin value by deploying a random margin drawn from a normal distribution. We additionally extended this concept by guiding the assignment of the drawn margin values to put more attention on hardly classified samples. We provided a simple toy example with an 8-class classification problem to demonstrate the enhanced separability and robustness induced by our ElasticFace loss. To experimentally demonstrate the effect of our ElasticFace loss on face recognition accuracy, we report the results on nine different benchmarks. The achieved results are compared to the results reported in the recent state-of-the-art. In

a detailed comparison, compared to fixed margin penalties and recent state-of-the-art, our ElasticFace loss enhanced the face recognition accuracy on most of the considered benchmarks, consequently extending state-of-the-art face recognition performance on seven out of nine benchmarks and scoring close to the state-of-the-art in the remaining two. This is especially the case in the benchmarks where the intra-class variation is extremely high, such as frontal-to-profile face verification (CFP-FP [24]) and large age gap face verification (AgeDB-30 [20]), which points to the generalizability induced by the proposed ElasticFace.

In the rest of this paper, we will first introduce our proposed ElasticFace loss by building up to its definition starting from the basic softmax loss. This rationalization will include an experimental toy example demonstrating the effect of the proposed loss. Later on, the experimental setup and implementation details are introduced. This is followed by a detailed comparative discussion of the achieved results and a final conclusion.

2. ElasticFace loss

We propose in this work a novel learning loss strategy, ElasticFace loss, aiming at improving the accuracy of face recognition by targeting enhanced intra-class compactness and inter-class discrepancy in a flexible manner. Unlike previous works [4, 16, 27] that utilize a fixed penalty margin value, our proposed ElasticFace loss accommodates flexibility through relaxing this constraint by randomly drawing the margin value from a Gaussian distribution. Our proposed ElasticFace loss targets giving the model flexibility in optimizing the separability between and within the classes as it incorporates random margin values for each sample in each training iteration. The randomized margin penalty can be easily integrated into any of the angular margin-based softmax losses, which we demonstrate on two state-of-the-art margin-based softmax losses. The angular margin-based losses and our ElasticFace loss extend over the softmax loss by manipulating the decision boundary to enhance intra-class compactness and inter-class discrepancy. Therefore, in this section, we first revisit the conventional softmax loss. Then, we present the modified version of softmax loss and the angular margin-based softmax loss. This leads up to presenting our proposed ElasticFace loss and an extended definition, the ElasticFace+, where the assignment of the drawn margins to training samples is linked to their proximity to their class centers.

Softmax Loss The widely used multi-class classification loss, softmax loss [17], refers to applying cross-entropy loss between the output of the logistic function (softmax activation function) and the ground-truth. Assume $x_i \in R^d$ is a feature representation of the i-th sample z_i and y_i is its corresponding class label (y_i integer in the range $[1, c]$). Given

that c is the number of classes (identities), the output of the softmax activation function is defined as follows:

$$softmax(x_i, y_i) = \frac{e^{f_{y_i}}}{\sum\limits_{j=1}^{c} e^{f_j}} = \frac{e^{x_i W_{y_i}^T + b_{y_i}}}{\sum\limits_{j=1}^{c} e^{x_i W_j^T + b_j}}, \quad (1)$$

where f_{y_i} is the activation of the last fully-connected layer with weight vector W_{y_i} and bias b_{y_i}. W_{y_i} is the y_i-th column of weights $W \in R_c^d$ and b_{y_i} is the corresponding bias offset. The output of the softmax activation function is the probability of x_i being correctly classified as y_i. Given a mini-batch of size N, the cross-entropy loss function that measures the divergence between the model output and the ground-truth labels can be defined as follows:

$$L_{CE} = \frac{1}{N} \sum_{i \in N} -log \frac{e^{x_i W_{y_i}^T + b_{y_i}}}{\sum\limits_{j=1}^{c} e^{x_i W_j^T + b_j}}. \quad (2)$$

In a simple binary class classification, assuming that the input z_i belong to class 1, the model will correctly classify z_i if $W_1^T x_i + b1 > W_2^T x_i + b2$ and z_i will be classified as class 2 if $W_2^T x_i + b2 > W_1^T x_i + b1$. Therefore, the decision boundary of softmax loss is $x(W_1^T - W_2^T) + b1 - b2 = 0$. One of the main limitations of using softmax loss for learning face embeddings is that softmax loss does not explicitly optimize the feature representation needed for face verification as there is no restriction on the minimum distance between the class centers. Thus, training with softmax loss is less than optimal for achieving the maximum inter-class distances and the minimum intra-class distances. To mitigate this limitation, margin penalty-based cosine softmax loss was proposed as an alternative to the conventional softmax loss and it became a popular loss function for training face recognition models [4, 16, 27]. To get there, [16] has proposed a modified softmax loss (Cosine softmax loss) that optimized the angle cosine between features and the weights $cos(\theta)$ and then, incorporates a margin penalty on $cos(\theta)$.

Cosine Softmax Loss Following [4, 16, 17, 27], the bias offset, for simplicity, can be fixed to $b_{y_i} = 0$. The logit f_{y_i}, in this case, can be reformulated as: $x_i W_{y_i}^T = \|x_i\| \|W_{y_i}\| cos(\theta_{y_i})$, where θ_{y_i} is the angle between the weights of the last fully-connected layer W_{y_i} and the feature representation x_i. By fixing the weights norm and the feature norm to $\|W_{y_i}\| = 1$ and $\|x_i\| = 1$, respectively, and rescaling the $\|x_i\|$ to the constant s [27], the output of the softmax activation function is subject to the cosine of the angle θ_{y_i}. The modified softmax loss (L_{ML}) can be defined, as stated in [16, 27], as follows:

$$L_{ML} = \frac{1}{N} \sum_{i \in N} -log \frac{e^{s(cos(\theta_{y_i}))}}{e^{s(cos(\theta_{y_i}))} + \sum\limits_{j=1, j \neq y_i}^{c} e^{s(cos(\theta_j))}}. \quad (3)$$

In the previous binary example, assume that the input z_i belong to the class 1, z_i will be correctly classified if $cos(\theta1) > cos(\theta2)$. The decision boundary, in this case, is $cos(\theta1) - cos(\theta2) = 0$. Therefore, training with the modified (cosine) softmax loss emphasizes that the model prediction depends on the angle cosine between the features and the weights. However, and similar to conventional softmax loss, modified softmax loss does not explicitly optimize the feature representation needed for face verification. This motivated the introduction of the angular margin penalty-based losses [4, 16, 27].

Angular Margin Penalty-based Loss In recent works [4, 16, 27], different types of margin penalty are proposed to push the decision boundary of softmax, and thus enhance intra-class compactness and inter-class discrepancy aiming at improving the accuracy of face recognition. The general angular margin penalty-based loss (L_{AML}) is defined as follows:

$$L_{AML} = \frac{1}{N} \sum_{i \in N} -log \frac{e^{s(cos(m_1 \theta_{y_i} + m_2) - m_3)}}{e^{s(cos(m_1 \theta_{y_i} + m_2) - m_3)} + \sum\limits_{j=1, j \neq y_i}^{c} e^{s(cos(\theta_j))}}, \quad (4)$$

where m_1, m_2 and $m3$ are the margin penalty parameters proposed by SphereFace [16], ArcFace [4] and CosFace [27], respectively. In SphereFace [16], multiplicative angular margin penalty is deployed by multiplying θ by $m_1 = \alpha$ and setting $m_2 = 0$ and $m_3 = 0$ ($\alpha > 1.0$). The decision boundary of SphereFace is then $cos(m_1 \theta_{y_i}) - cos(\theta_j) = 0$. Differently, CosFace [27] proposed additive cosine margin penalty by setting $m_1 = 1$, $m_2 = 0$ and $m_3 = \alpha$ ($0 < \alpha < 1 - cos(\frac{\pi}{4})$). The decision boundary of CosFace is then $cos(\theta_{y_i}) - cos(\theta_j) - m3 = 0$. Later, ArcFace [4] proposed additive angular margin penalty by setting up $m_1 = 1$, $m_2 = \alpha$ and $m_3 = 0$ ($0 < \alpha < 1.0$). The decision boundary of ArcFace is then $cos(\theta_{y_i} + m2) - cos(\theta_j) = 0$.

Even though, ArcFace [4], CosFace [27] and SphereFace [16] introduced the important concept of angular margin penalty on softmax loss, selecting a single optimal margin value (α) is a critical issue in these works. By setting up $m_1 = 1$, $m_2 = 0$ and $m_3 = 0$, ArcFace, CosFace and SphereFace are equivalent to the modified softmax loss. A reasonable choice could be selecting a large margin value that is close to the margin upper bound to enable higher separability between the classes. However, when the margin value is too large, the model fails to converge, as stated in [27]. ArcFace, CosFace, and SphereFace selected the
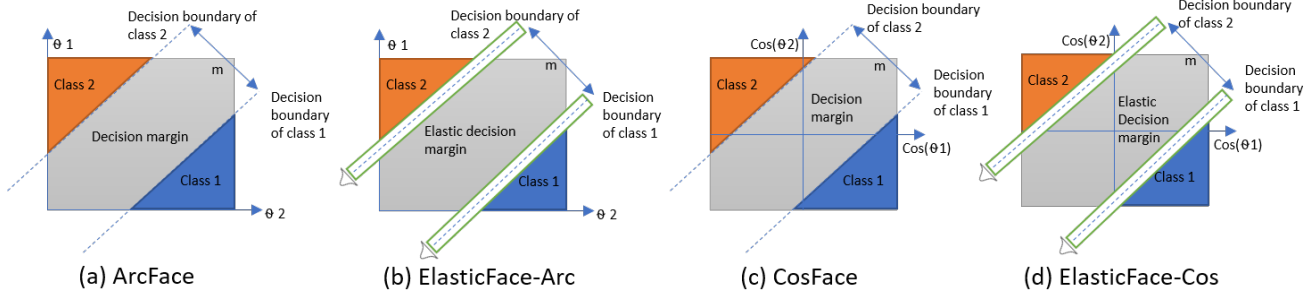
Figure 1. Decision boundary of (a) ArcFace, (b) ElasticFace-Arc, (c) CosFace, and (d) ElasticFace-Cos for binary classification. The dashed blue line is the decision boundary. The gray area illustrates the decision margin.

margin value through trial and error assuming that the samples are equally distributed in geodesic space around the class centers. However, this assumption could not be held when there are largely different intra-class variations leading to less than optimal discriminative feature learning, especially when there are large variations between the samples/classes in the training dataset. This motivated us to propose ElasitcFace loss by utilizing random margin penalty values drawn from a Gaussian distribution aiming at giving the model space for flexible class separability learning.

Elastic Angular Margin Penalty-based Loss (Elastic-Face) The proposed ElasticFace loss is extended over the angular margin penalty-based loss by deploying random margin penalty values drawn from a Gaussian distribution. Formally, the probability density function of a normal distribution is defined as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \tag{5}$$

where μ is the mean of the distribution and σ is its standard deviation. To demonstrate and prove our proposed elastic margin, we chose to integrate the randomized margin penalty in ArcFace (noted as ElasticFace-Arc) and CosFace (noted as ElasticFace-Cos) as they proved to have clearer geometric interpretation and achieved higher accuracy on mainstream benchmarks than the earlier SphereFace. ElasticFace-Arc (L_{EArc}) can be defined as follows:

$$L_{EArc} = \frac{1}{N} \sum_{i \in N} -log \frac{e^{s(cos(\theta_{y_i} + E(m,\sigma)))}}{e^{s(cos(\theta_{y_i} + E(m,\sigma)))} + \sum\limits_{j=1, j \neq y_i}^{c} e^{s(cos(\theta_j))}}, \tag{6}$$

and ElasticFace-Cos (L_{ECos}) can be defined as follows:

$$L_{ECos} = \frac{1}{N} \sum_{i \in N} -log \frac{e^{s(cos(\theta_{y_i}) - E(m,\sigma))}}{e^{s(cos(\theta_{y_i}) - E(m,\sigma))} + \sum\limits_{j=1, j \neq y_i}^{c} e^{s(cos(\theta_j))}}, \tag{7}$$

where $E(m, \sigma)$ is a normal function that return a random value from a Gaussian distribution with the mean m and the standard deviation σ.

The decision boundaries of ElasticFace-Arc and ElasticFace-Cos are $cos(\theta_{y_i} + E(m, \sigma)) - cos(\theta_j) = 0$ and $cos(\theta_{y_i}) - cos(\theta_j) - E(m, \sigma) = 0$, respectively. Figure 1 illustrates the decision boundary of ArcFace, ElasticFace-Arc, CosFace and ElasticFace-Cos. The sample push towards its center during training using ElasticFace-Arc and ElasticFace-Cos varies between training samples, based on the margin penalty drawn from $E(m, \sigma)$. During the training phase, a new random margin is generated for each sample in each training iteration. This aims at giving the model flexibility in the push for class separability. When σ is 0, our ElasticFace-Arc and ElasticFace-Cos are equivalent to ArcFace and CosFace, respectively.

ElasticFace+ We propose an extension to our Elastic-Face, the ElasticFace+, that observes the intra-class variation during each training iteration and use this observation to assign a margin value to each sample based on its proximity to its class center. This causes the samples that are relatively far from their class center to be pushed with a larger penalty margin to their class center. This aims at giving the model space to push the samples that are relatively far from their class center to be closer to their centers while giving less penalty attention to the samples that are already close to their center. To achieve that, we sort (descending) the output of the Gaussian distribution function (Equation 5) based on $cos(\theta_{y_i})$ value. Thus, the sample with small $cos(\theta_{y_i})$ will be pushed with large value from $E(m, \sigma)$ function, and vice versa.

Parameter Selection The probability density function has its peak around m [22]. Thus, when ElasticFace is integrated into ArcFace [4], we select the best margin value (as a single value) by training three instances of ResNet-50 [8] on CASIA [30] with ArcFace loss using margins equal to 0.45, 0.50 and 0.55, respectively, to assure the advised margin in [4]. Then, based on the sum of the performance ranking Borda count on LFW [9], AgeDB-30 [20], CALFW [34], CPLFW [33], and CFP-FP [24], we select the

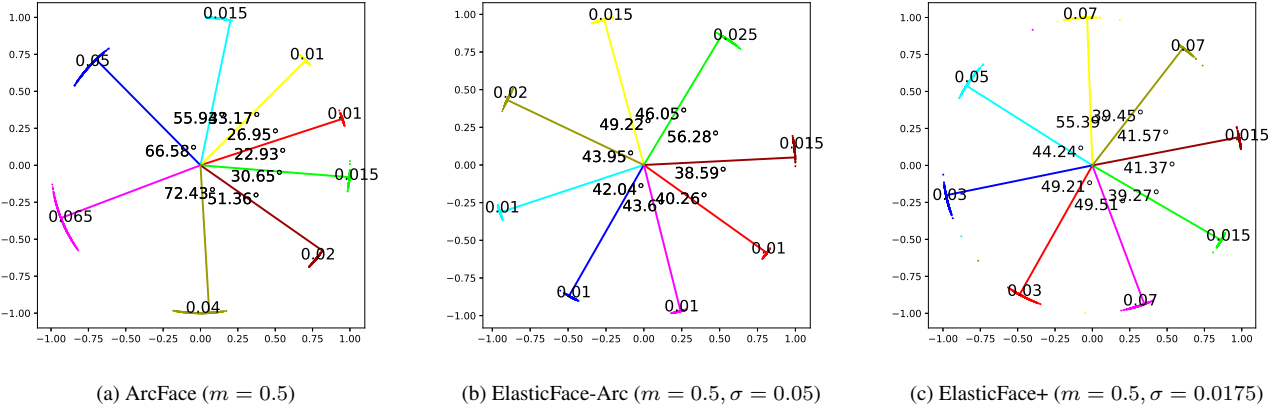| (a) ArcFace ($m = 0.5$) | (b) ElasticFace-Arc ($m = 0.5, \sigma = 0.05$) | (c) ElasticFace+ ($m = 0.5, \sigma = 0.0175$) |

Figure 2. Toy example of 3 ResNet-18 networks trained under different experimental settings. The 2-D features are normalized. Thus, the feature embeddings are allocated around the class centers in the arc space with a fixed radius. The numbers next to each class center indicate the mean of the standard deviation of each class feature embeddings. The angle in degree are calculated between each two consecutive classes to illustrate the decision margin between the classes. One can noticed that feature produced by ElasticFace and ElasticFace+ are more equally distributed around the class centers than ArcFace, in the arc space. Same colors always indicates same class across plots.

Loss	m	σ	LFW		AgeDB-30		CALFW		CPLFW		CFP-FP		-
			Acc (%)	BC	Acc(%)	BC	Acc (%)	BC	Acc(%)	BC	Acc(%)	BC	Sum BC
ArcFace	0.55	-	99.52	3	94.58	1	93.82	2	89.05	1	95.24	1	8
ArcFace	0.5	-	99.46	2	94.83	3	93.88	3	89.72	3	95.36	2	13
ArcFace	0.45	-	99.43	1	94.66	2	93.80	1	89.42	2	95.53	3	9
ElasticFace-Arc	0.5	0.0125	99.53	4	94.80	1	93.68	2	89.72	3	95.43	1	11
ElasticFace-Arc	0.5	0.0175	99.47	1	95.13	4	93.67	1	89.53	2	95.54	3	11
ElasticFace-Arc	0.5	0.025	99.52	3	94.95	3	93.78	3	89.50	1	95.44	2	12
ElasitcFace-Arc	0.5	0.05	99.52	3	94.82	2	93.90	4	89.79	4	95.59	4	17
ElasticFace-Arc+	0.5	0.0125	99.53	4	95.00	2	93.68	1	89.58	4	95.40	2	13
ElasticFace-Arc+	0.5	0.0175	99.53	4	95.07	3	93.95	3	89.37	1	95.67	4	15
ElasticFace-Arc+	0.5	0.0225	99.42	1	95.15	4	93.73	2	89.48	2	95.36	1	10
ElasitcFace-Arc+	0.5	0.05	99.45	2	94.83	1	94.00	4	89.50	3	95.56	3	13

Table 1. Parameter selection for ElasticFace-Arc and ElasticFace-Arc+. The Borda count (BC) is reported separately for each of training settings (ArcFace, ElasticFace-Arc and ElasticFace-Arc+) and each of the evaluation benchmarks. The final σ and m parameters are selected based on the highest BC sum. In all settings, the used architecture is ResNet-50 trained on CASIA [30].

Loss	m	σ	LFW		AgeDB-30		CALFW		CPLFW		CFP-FP		-
			Acc (%)	BC	Acc (%)	BC	Acc (%)	BC	Acc (%)	BC	Acc(%)	BC	Sum BC
CosFace	0.4	-	99.42	1	94.65	3	93.45	1	90.38	3	95.30	1	9
CosFace	0.35	-	99.55	3	94.55	2	93.78	3	89.95	1	95.31	2	11
CosFace	0.3	-	99.45	2	94.45	1	93.46	2	90.12	2	95.39	3	10
ElasticFace-Cos	0.35	0.0125	99.45	2	94.72	1	93.83	1	90.12	2	95.47	3	9
ElasticFace-Cos	0.35	0.0175	99.50	3	94.77	3	93.97	4	90.10	1	95.30	2	13
ElasticFace-Cos	0.35	0.025	99.42	1	94.85	4	93.88	2	90.20	3	95.21	1	11
ElasticFace-Cos	0.35	0.05	99.52	4	94.77	3	93.93	3	90.38	4	95.52	4	18
ElasticFace-Cos+	0.35	0.0125	99.38	1	94.50	2	93.67	3	89.85	1	95.20	1	8
ElasticFace-Cos+	0.35	0.0175	99.45	2	94.97	4	93.48	1	89.98	2	95.23	2	11
ElasticFace-Cos+	0.35	0.025	99.55	4	94.63	3	93.65	2	90.28	4	95.47	4	17
ElasticFace-Cos+	0.35	0.05	99.48	3	94.45	1	93.77	4	90.01	3	95.26	3	14

Table 2. Parameter selection for ElasticFace-Cos and ElasticFace-Cos+. The Borda count (BC) is reported separately for each of training settings (ArcFace, ElasticFace-Cos and ElasticFace-Cos+) and each of the evaluation benchmarks. The final σ and m parameters are selected based on the highest BC sum. In all settings, the used architecture is ResNet-50 trained on CASIA [30].

margin that achieved the highest Borda count sum and set it as m for $E(m, \sigma)$ function, where our goal is to use the most optimal margin. The best margin observed in our experiment, in this case, is 0.5 (Table 1). To select the σ value for $E(m, \sigma)$ function, we conducted additional experiments on four instances of ResNet-50 trained on CASIA [30] with our proposed ElasticFace-Arc by setting up the σ to one of these

values 0.0125, 0.015, 0.025 and 0.05. Then, we rank these models based on the sum of the performance ranking Borda count across all datasets. Finally, the σ value is chosen based on the highest Borda count sum. The best σ observed in our experiment, in this case, is 0.05 (Table 1). Similarly, we follow the same procedure to select the parameters (m and σ) for ElasticFace-Cos. We first choose the best margin value by training three different instances of ResNet-50 on CASIA [30] with CosFace using a margin equal to 0.3, 0.35, and 0.40. The best m observed in our experiment based on the sum of the performance ranking Borda count across all evaluated datasets, in this case, is 0.035 (Table 2). Similar to σ selection approach of ElasticFace-Arc, we train four instance of ElasticFace-Cos to choose the best σ for $E(m, \sigma)$ function. The best observed σ in our experiment, in this case, is 0.05 (Table 2). For ElasticFace-Cos+ and ElasticFace-Arc+, we followed the exact approach of parameter selection for ElasticFace-Arc and ElasticFace-Cos. The best observed σ for ElasticFace-Arc+ is 0.0175 and the best observed one for ElasticFace-Cos+ is 0.025 (Table 1 and 2). These selected parameters are used to train our solutions (training details in Section 3) evaluated in Section 4.

Toy example To demonstrate the robustness and the class separability induced by our proposed ElasticFace and ElasticFace+, we present a simple toy example by training three ResNet-18 networks [8] to classify eight different identities and produce 2-D feature embeddings. All the networks are trained with a small batch size of 128 for 11200 iterations with stochastic gradient descent (SGD) and an initial learning rate of 0.1. The learning rate is reduced by a factor of 10 after 1680, 2800, 3360, and 8400 training

iterations. To demonstrate a classification case where the classes are not identically varied, these eight identities are selected to have four identities with small intra-class variation and another four identities with a large intra-class variation (measured as the average of all intra-class comparison scores for each identity). These identities were chosen from all the identities with more than 400 images per identity in the MS1MV2 dataset [4], we note this selected subset as MS1MV2-400. From these identities, we select the four identities with the highest intra-class variation and the four with the lowest intra-class variation. The features for this selection were extracted using an open-source [2] ResNet-100 [8] model trained with ArcFace loss [4], and the comparison is performed by a cosine similarity. The set of the selected eight identities is noted as MS1MV2-8. We use MS1MV2-8 to train the toy networks with ArcFace (m=0.5), ElasticArcFace (m=0.5, σ=0.05), and ElasticArcFace+ (m=0.5, σ=0.0175), based on our parameter selection. Figure 2 shows the classification of MS1MV2-8 for each of the experimental settings. In each of the plots in Figure 2a, 2b and 2c, we calculate the angle between each consecutive identities to demonstrate the separability between the identities in the arc space (inter-class discrepancy). The optimal inter-class discrepancy may be achieved if the angle, in degree, between each of consecutive identities is close to 45 degrees i.e. 360 / 8. Also, we calculate the mean of the standard deviation of each class feature embeddings to illustrate intra-class compactness induced by ArcFace, ElasticFace, and ElasticFace+. The smaller standard deviation (shown at the edge of each class in Figure 2), in this case, indicates higher intra-class compactness. It can be noticed that our EalsticFace and EalsticFace+ achieved better intra-class compactness and inter-class discrepancy than ArcFace, while the differences in inter-class variation between EalsticFace and EalsticFace+ are minor (Figures 2a 2c, and 2b).

3. Experimental setup

Training settings: The network architecture we used to demonstrate our ElasticFace is the ReseNet-100 [8]. This was motivated by the wide use of this architecture in the state-of-the-art face recognition solutions [1,4,5,10,26]. We follow the common setting [1, 4, 10] to set the scale parameter s to 64. We set the mini-batch size to 512 and train our model on one Linux machine (Ubuntu 20.04.2 LTS) with Intel(R) Xeon(R) Gold 5218 CPU 2.30GHz, 512 G RAM, and 4 Nvidia GeForce RTX 6000 GPUs. The proposed models in this paper are implemented using Pytorch [21]. All models are trained with Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 1e-1. We set the momentum to 0.9 and the weight decay to 5e-4.

---
[2] https://github.com/deepinsight/insightface

The learning rate is divided by 10 at 80k, 140k, 210k, and 280k training iterations. The total number of training iteration is 295K, which corresponds to the number of margin sampling from the normal distribution. During the training, we use random horizontal flipping with a probability of 0.5 for data augmentation. The networks are trained (and evaluated) on images of the size $112 \times 112 \times 3$ to produce $512 - d$ feature embeddings. These images are aligned and cropped using the Multi-task Cascaded Convolutional Networks (MTCNN) [31] following [4]. All the training and testing images are normalized to have pixel values between -1 and 1.

Training dataset: We follow the trend in recent works [1,4,10,19] in using the MS1MV2 dataset [4] to train the investigated models with the proposed ElasticFace loss. This enables a direct comparison with the state-of-the-art as will be shown in Section 4. The MS1MV2 is a refined version [4] of the MS-Celeb-1M [7] containing 5.8M images of 85K identities.

Evaluation benchmarks and metrics: To demonstrate the effect of our proposed ElasticFace on face recognition accuracy and enable a wide comparison to state-of-the-art, we report the achieved results on nine benchmarks. These benchmarks are of a diverse nature, where some represent a special vulnerabilities of face recognition. The nine benchmarks are 1) Labeled Faces in the Wild (LFW) [9], 2) AgeDB-30 [20], 3) Cross-age LFW (CALFW) [34], 4) Cross-Pose LFW (CPLFW) [33], 5) Celebrities in Frontal-Profile in the Wild (CFP-FP) [24], 6) IARPA Janus Benchmark-B (IJB-B) [28], 7) IARPA Janus Benchmark-C (IJB-C) [18], 8) MegaFace [13], and 9) MegaFace (R) [4]. The face recognition performance on LFW, AgeDB-30, CALFW, CPLFW, and CFP-FP is reported as verification accuracy, following their evaluation protocol. The performance on IJB-C and IJB-B is reported (as defined in [18, 28]) as true acceptance rates (TAR) at false acceptance rates (FAR) of 1e-4. The MegaFace and MegaFace(R) benchmarks report the face recognition performance as Rank-1 correct identification rate and as TAR at FAR=1e–6 verification accuracy.

We acknowledge the verification and identification performance evaluation metrics reported in ISO/IEC 19795-1 [11]. However, to enhance the reproducibility and comparability, we follow the evaluation protocols and metrics used in each of the benchmarks as listed above.

4. Results

Tables 3 and 4 presents the achieved results on the nine considered benchmarks. The main observation is that our proposed ElasticFace solutions scored beyond the state-of-

Method	Training Dataset	LFW Accuracy (%)	AgeDB-30 Accuracy (%)	CALFW Accuracy (%)	CPLFW Accuracy (%)	CFP-FP Accuracy (%)
ArcFace [4] (CVPR2019)	MS1MV2 [4, 7]	99.82 (3)	98.15	95.45	92.08	98.27
CosFace [27] (CVPR2018)	private	99.73	-	-	-	-
Dynamic-AdaCos [32] (CVPR2019)	clean MS1M [7, 32] + CASIA [30]	99.73	-	-	-	-
AdaptiveFace [15] (CVPR2019)	clean MS1M [7, 29]	99.62	-	-	-	-
UniformFace [5] (CVPR2019)	clean MS1M [4, 7] + VGGFace2 [2]	99.8	-	-	-	-
GroupFace [14] (CVPR2020)	clean MS1M [4, 7]	**99.85** (1)	98.28 (3)	**96.20** (1)	93.17	98.63
CircleLoss [26] (CVPR2020)	clean MS1M [7, 26]	99.73	-	-	-	96.02
CurricularFace [10] (CVPR2020)	MS1MV2 [4, 7]	99.80	98.32 (2)	**96.20** (1)	93.13	98.37
Dyn-arcFace [12] (MTAP2021)	clean MS1M [4, 7]	99.80	97.76	-	-	94.25
MagFace [19] (CVPR2021)	MS1MV2 [4, 7]	99.83 (2)	98.17	96.15	92.87	98.46
Partial-FC-ArcFace [1] (ICCVW2021)	MS1MV2 [4, 7]	99.83 (2)	98.20	96.18 (2)	93.00	98.45
Partial-FC-CosFace [1] (ICCVW2021)	MS1MV2 [4, 7]	99.83 (2)	98.03	**96.20** (1)	93.10	98.51
ElasticFace-Arc (ours)	MS1MV2 [4, 7]	99.80	**98.35** (1)	96.17 (3)	93.27 (2)	98.67 (2)
ElasticFace-Cos (ours)	MS1MV2 [4, 7]	99.82 (3)	98.27	96.03	93.17	98.61 (3)
ElasticFace-Arc+ (ours)	MS1MV2 [4, 7]	99.82 (3)	**98.35** (1)	96.17 (3)	**93.28** (1)	98.60
ElasticFace-Cos+ (ours)	MS1MV2 [4, 7]	99.80	98.28 (3)	96.18 (2)	93.23 (3)	**98.73** (1)

Table 3. The achieved results on the LFW, AgeDB-30, CALFW, CPLFW, and CFP-FP benchmarks. On large age gape (AgeDB-30) and frontal-to-profile face comparisons (CFP-FP), the ElasticFace solutions consistently extend state-of-the-art performances. ElasticFace scores very close to the state-of-the-art on LFW and CALFW. All decimal points are provided as reported in the respective works. The top performance in each benchmark is in bold. The top three performances in each benchmark are noted with rank number between parentheses (1,2 or 3).

Method	Training Dataset	IJB-B TAR at FAR1e–4 (%)	IJB-C TAR at FAR1e–4 (%)	MegaFace (R) Rank-1 (%)	MegaFace (R) TAR at FAR1e–6 (%)	MegaFace Rank-1 (%)	MegaFace TAR at FAR1e–6 (%)
ArcFace [4] (CVPR2019)	MS1MV2 [4, 7]	94.2	95.6	98.35	98.48	81.03	96.98
CosFace [27] (CVPR2018)	private	-	-	-	-	**82.72** (1)	96.65
Dynamic-AdaCos [32] (CVPR2019)	clean MS1M [7, 32] + CASIA [30]	-	92.40	97.41	-	-	-
AdaptiveFace [15] (CVPR2019)	clean MS1M [7, 29]	-	-	95.02	95.61	-	-
UniformFace [5] (CVPR2019)	clean MS1M [4, 7] + VGGFace2 [2]	-	-	-	-	79.98	95.36
GroupFace [14] (CVPR2020)	clean MS1M [4, 7]	94.93	96.26	98.74 (3)	98.79	81.31 (2)	97.35 (2)
CircleLoss [26] (CVPR2020)	clean MS1M [7, 26]	-	93.95	98.50	98.73	-	-
CurricularFace [10] (CVPR2020)	MS1MV2 [4, 7]	94.8	96.1	98.71	98.64	81.26 (3)	97.26
Dyn-arcFace [12] (MTAP2021)	clean MS1M [4, 7]	-	-	-	-	-	-
MagFace [19] (CVPR2021)	MS1MV2 [4, 7]	94.51	95.97	-	-	-	-
Partial-FC-ArcFace [1] (ICCVW2021)	MS1MV2 [4, 7]	94.8	96.2	98.31	98.59	-	-
Partial-FC-CosFace [1] (ICCVW2021)	MS1MV2 [4, 7]	95.0	96.4	98.36	98.58	-	-
ElasticFace-Arc (ours)	MS1MV2 [4, 7]	95.22 (3)	96.49 (3)	**98.81** (1)	**98.92** (1)	80.76	97.30
ElasticFace-Cos (ours)	MS1MV2 [4, 7]	95.30 (2)	96.57 (2)	98.70	98.75	81.01	97.31 (3)
ElasticFace-Arc+ (ours)	MS1MV2 [4, 7]	95.09	96.40	98.80 (2)	98.83 (3)	80.68	**97.44** (1)
ElasticFace-Cos+ (ours)	MS1MV2 [4, 7]	**95.43** (1)	**96.65** (1)	98.62	98.85 (2)	80.08	97.29

Table 4. The achieved results on the IJB-B, IJB-C, MegaFace (R), and MegaFace benchmarks. On the earlier three, and the verification accuracy of the fourth, the ElasticFace solutions consistently extend state-of-the-art performances. ElasticFace scores very close to the state-of-the-art on MegaFace. MegaFace has been refined in [4] to MegaFace (R) as it contains many face images with wrong labels. All decimal points are provided as reported in the respective works. The top performance in each benchmark is in bold. The top three performances in each benchmark are noted with rank number between parentheses (1,2 or 3).

the-art in seven out of the nine benchmarks, and very close to the state-of-the-art in the remaining two. When possible, and to build a fair comparison, the results of previous works are reported when trained on the MS1MV2 [4, 7] (or a refined variant of MS1M [7]) as the ElasticFace results are based on training on this dataset. The proposed ElasticFace ranked first in comparison to the state-of-the-art on the benchmarks AgeDB-30, CPLFW, CFP-FP, IJB-B, IJB-C, MegaFace (R), and MegaFace (verification). In the remaining benchmarks, ElasticFace solutions ranked second on CALFW, third on LFW, and fourth on MegaFace (identification).

A main outcome of the evaluation is concerning the databases with very large intra-user variations. These are the large age gape benchmark (AgeDB-30) and the frontal-to-profile face verification benchmark (CFP-FP). On AgeDB-30, our ElasticFace-Arc solution scored an accuracy of 98.35%, while the top state-of-the-art performance was 98.32% scored by the CurricularFace [10]. On CFP-FP, our ElasticFace-Arc+ solution scored an accuracy of 98.73% and our ElasticFace-Arc scored an accuracy of 98.67%, while the top state-of-the-art performances were 98.51% scored by the Partial-FC-CosFace [1] solution and 98.46% scored by the MagFace [19]. This significantly enhanced performance in the extreme intra-class variation scenarios points out the generalizability induced by the ElasticFace loss. CALFW and CPLFW also considered age gaps and pose variation, however, with a lower variation

than AgeDB-30 and CFP-FP. In CALFW, ElasticFace-Cos+ scored a close second with 96.18% accuracy, with the lead going to the CurricularFace [10] with 96.20% accuracy. In CPLFW, our ElasticFace-Arc+ is ranked first with 93.28% accuracy, while the top state-of-the-art performance was 93.17% accuracy scored by the GroupFace [14]. On the LFW benchmark [9], which is one of the oldest and nearly saturated benchmarks reported in the recent works, our ElasticFace-Cos and ElasticFace-Arc+ solutions scored an accuracy of 98.82%, very close behind the GroupFace [14] with 99.85%.

In Table 4, on IJB-B benchmark, our ElasticFace-Cos+ scored a TAR at FAR1e–4 of 95.43%, far ahead of the Partial-FC-CosFace [1] and the GroupFace [14] with 95.0% and 94.93%, respectively. Similarly, on the IJB-C benchmark, our ElasticFace-Cos+ scored a TAR at FAR1e–4 of 96.65%, ahead of the Partial-FC-CosFace [1] and the GroupFace [14] with 96.4% and 96.36% respectively. On the MegaFace (R), our ElasticFace-Arc scored 98.81% Rank-1 identification rate and 98.92% TAR at FAR1e–6, ahead of the previous lead solution, the GroupFace [14] with 98.74% and 98.79%, respectively. On the MegaFace benchmark, our ElasticFace-Cos scored Rank-1 identification rate of 81.01%, close to the state-of-the-art 82.72% score by CosFace [27], noting that CosFace was trained on a private dataset. On the same benchmark (MegaFace), our ElasticFace-Arc+ ranked first with 97.44% TAR at FAR1e–6, while the top state-of-the-art performances were 97.35% scored by the GroupFace [14]. It must be mentioned that the MegaFace benchmark has been refined in [4] to MegaFace (R) as it contains many face images with wrong labels as reported in [4].

In comparison to the closely defined losses in ArcFace [4], CosFace [27], and Partial-FC [1] solutions, our ElasticFace models did prove to provide a strong performance edge by scoring higher recognition performance on most benchmarks. When it comes to comparing ElasticFace and ElasticFace+, the ElasticFace-Arc and ElasticFace-Arc+ did achieve very close performances when considering all benchmarks. On the other hand, the ElasticFace-Cos+ did outperform ElasticFace-Cos on most benchmarks.

We acknowledge that the Partial-FC [1] solution reported additional performance rates when trained on their new collected database, the Glint360K [1]. However, we could not acquire this database as it requires an account on a cloud platform, that in itself requires a SIM card registered in a specific country, which is very restrictive and we do not have access to. Therefore, and for a fair comparison, we opted to compare our results with the Partial-FC results when trained on the same dataset that our ElasticFace solution is using, the MS1MV2 [4, 7] dataset.

The slightly increased training computational cost is a minor limitation of our proposed ElasticFace. Training the ResNet-100 model on MS1MV2 dataset with CosFace or ArcFace using the specified machine and training details described in Section 3 requires around 57 hours. This training time is increased by around one minute for ElasticFace and by 11 hours for ElasticFace+. The minor increase in the ElasticFace training time is caused by the sampling of the margin values, while the larger increase in ElasticFace+ training time is additionally caused by the sorting algorithms.

On a less technical note, we stress that our efforts in the advancement of face recognition are aimed at enhancing the security, convenience, and life quality of the members of society, e.g. enabling convenient access to financial and health services [6] and enhancing the security of border checks within clear legal frameworks and users consent. We acknowledge and reject the possible malicious or illegal use of this and other technologies.

5. Conclusion

In this paper, we propose an elastic margin penalty loss (ElasticFace) that avoids setting a single constant penalty margin. Our motivation considers that real training data is inconsistent in terms of inter and intra-class variation, and thus the assumption made by many margin softmax losses that the geodesic distance between and within the different identities can be equally learned using a fixed margin is less than optimal. We, therefore, relax this fixed margin constrain by using a random margin value drawn from a normal distribution in each training iteration. In an extended definition, the assignment of these margin values to training samples corresponds to their proximity to their class centers. We evaluated our ElasticFace loss, in comparison to state-of-the-art face recognition approaches, on nine different benchmarks. This evaluation demonstrated that our ElasticFace solution consistently extended state-of-the-art face recognition performance on most benchmarks (seven out of nine). This was specifically apparent in the challenging benchmarks with large intra-class variations, such as large age gaps and frontal-to-profile face comparisons.

Acknowledgment

References

[1] Xiang An, Xuhan Zhu, Yuan Gao, Yang Xiao, Yongle Zhao, Ziyong Feng, Lan Wu, Bin Qin, Ming Zhang, Debing Zhang, and Ying Fu. Partial fc: Training 10 million identities on a single machine. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1445–1449, October 2021. 2, 6, 7, 8

[2] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, Xi'an, China, May 15-19, 2018*, pages 67–74. IEEE Computer Society, 2018. 1, 7

[3] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 539–546. IEEE Computer Society, 2005. 1

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1, 2, 3, 4, 6, 7, 8

[5] Yueqi Duan, Jiwen Lu, and Jie Zhou. Uniformface: Learning deep equidistributed representation for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3415–3424. Computer Vision Foundation / IEEE, 2019. 2, 6, 7

[6] e-Aadhaar - Unique Identification Authority of India. https://eaadhaar.uidai.gov.in/, 2015. 8

[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 Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III*, volume 9907 of *Lecture Notes in Computer Science*, pages 87–102. Springer, 2016. 6, 7, 8

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. 4, 5, 6

[9] 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 07-49, University of Massachusetts, Amherst, October 2007. 2, 4, 6, 8

[10] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: Adaptive curriculum learning loss for deep face recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 5900–5909. IEEE, 2020. 2, 6, 7, 8

[11] ISO/IEC JTC1 SC37 Biometrics. ISO/IEC 19795-1:2021 Information technology — Biometric performance testing and reporting — Part 1: Principles and framework. International Organization for Standardization, 2021. 6

[12] Jichao Jiao, Weilun Liu, Yaokai Moand Jian Jiao, Zhongliang Deng, and Xinping Chen. Dyn-arcface: dynamic additive angular margin loss for deep face recognition. *Multim. Tools Appl.*, 2021. 2, 7

[13] Ira Kemelmacher-Shlizerman, Steven M. Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4873–4882. IEEE Computer Society, 2016. 6

[14] Yonghyun Kim, Wonpyo Park, Myung-Cheol Roh, and Jongju Shin. Groupface: Learning latent groups and constructing group-based representations for face recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 5620–5629. IEEE, 2020. 2, 7, 8

[15] Hao Liu, Xiangyu Zhu, Zhen Lei, and Stan Z. Li. Adaptiveface: Adaptive margin and sampling for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11947–11956. Computer Vision Foundation / IEEE, 2019. 2, 7

[16] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6738–6746. IEEE Computer Society, 2017. 1, 2, 3

[17] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 507–516. JMLR.org, 2016. 1, 2, 3

[18] Brianna Maze, Jocelyn C. Adams, James A. Duncan, Nathan D. Kalka, Tim Miller, Charles Otto, Anil K. Jain, W. Tyler Niggel, Janet Anderson, Jordan Cheney, and Patrick Grother. IARPA janus benchmark - C: face dataset and protocol. In *2018 International Conference on Biometrics, ICB 2018, Gold Coast, Australia, February 20-23, 2018*, pages 158–165. IEEE, 2018. 2, 6

[19] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Virtual, June 19-25, 2020*. IEEE, 2021. 2, 6, 7

[20] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*

2017, Honolulu, HI, USA, July 21-26, 2017, pages 1997–2005. IEEE Computer Society, 2017. 2, 4, 6

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[22] Peyton Z Peebles. *Probability, random variables, and random signal principles*. McGraw Hill, 1987. 4

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 815–823. IEEE Computer Society, 2015. 1

[24] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Domingo Castillo, Vishal M. Patel, Rama Chellappa, and David W. Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, Lake Placid, NY, USA, March 7-10, 2016*, pages 1–9. IEEE Computer Society, 2016. 2, 4, 6

[25] Kihyuk Sohn. Improved deep metric learning with multiclass n-pair loss objective. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1849–1857, 2016. 1

[26] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6397–6406. IEEE, 2020. 2, 6, 7

[27] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5265–5274. IEEE Computer Society, 2018. 1, 2, 3, 7, 8

[28] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn C. Adams, Tim Miller, Nathan D. Kalka, Anil K. Jain, James A. Duncan, Kristen Allen, Jordan Cheney, and Patrick Grother. IARPA janus benchmark-b face dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 592–600. IEEE Computer Society, 2017. 2, 6

[29] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light CNN for deep face representation with noisy labels. *IEEE Trans. Inf. Forensics Secur.*, 13(11):2884–2896, 2018. 7

[30] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch. *CoRR*, abs/1411.7923, 2014. 4, 5, 7

[31] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 1, 6

[32] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10823–10832. Computer Vision Foundation / IEEE, 2019. 7

[33] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Technical Report 18-01, Beijing University of Posts and Telecommunications, February 2018. 4, 6

[34] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments. *CoRR*, abs/1708.08197, 2017. 4, 6