

SFace: Sigmoid-constrained Hypersphere Loss for Robust Face Recognition

Yaoyao Zhong, Weihong Deng, Jiani Hu, Dongyue Zhao, Xian Li, Dongchao Wen

Abstract—Deep face recognition has achieved great success due to large-scale training databases and rapidly developing loss functions. The existing algorithms devote to realizing an ideal idea: minimizing the intra-class distance and maximizing the inter-class distance. However, they may neglect that there are also low quality training images which should not be optimized in this strict way. Considering the imperfection of training databases, we propose that intra-class and inter-class objectives can be optimized in a moderate way to mitigate overfitting problem, and further propose a novel loss function, named sigmoid-constrained hypersphere loss (SFace). Specifically, SFace imposes intra-class and inter-class constraints on a hypersphere manifold, which are controlled by two sigmoid gradient re-scale functions respectively. The sigmoid curves precisely re-scale the intra-class and inter-class gradients so that training samples can be optimized to some degree. Therefore, SFace can make a better balance between decreasing the intra-class distances for clean examples and preventing overfitting to the label noise, and contributes more robust deep face recognition models. Extensive experiments of models trained on CASIA-WebFace, VGGFace2, and MS-Celeb-1M databases, and evaluated on several face recognition benchmarks, such as LFW, MegaFace and IJB-C databases, have demonstrated the superiority of SFace.

I. INTRODUCTION

DEEP face recognition has obtained surprising improvement recent years [1], [2], [3], [4], [5], [6], [7], [8], [9]. The pipeline for deep face recognition has been widely used for its practical usage [10], [4], [5], [8]. That is, deep face recognition models are trained on web-collected databases [11], [12], [13], [14], [15], and work as deep feature extractors to evaluate on other testing databases [16], [17], [18], [19], [20], [21], [22].

The large-scale training databases [11], [12], [13], [14], [15] are fundamental for the success of deep face recognition. For training databases of deep face recognition, we can never expect to obtain a “perfect” training database which should include, but not limited to, sufficient numbers of identities, and adequate images of each identity. Considering the copyright and privacy protection, the number of identities in the web-collected training databases is limited compared with the global population, and celebrities of web-collected databases may be far from the testing settings in daily life [10]. In addition, we can hardly collect images with full intra-class

variation to model the large pose, face expressions and illumination variance of each identity [23], [22], therefore there are a significant portion of under-represented identities [24], [25], [26], [27]. Considering the open-set protocol and the limitations of training databases, current research focus is trying to make best use of the training databases, and improve the ability of loss functions to obtain a more discriminative feature extractor. One of the most effective loss functions is the large margin loss function [5], [6], [7], [8], [9]. They incorporate large margins to softmax loss to encourage the intra-class compactness and the inter-class orthogonality, which has alleviated the aforementioned quantity limitation and imbalance problem of identities to some degree.

Existing mainstream methods devote to minimizing the intra-class distance and maximizing the inter-class distance. Despite the success, they may neglect that, in addition to the high quality training images, there are also low quality training images such as misaligned images, low-resolution images, and label noise, which cannot provide effective information for distinguishing the labeled identity. Even human annotations are not reliable as we thought, because humans often struggle to distinguish between hard examples and low quality training images, and they have already been surpassed by deep face recognition models a few years ago [28]. For this reason, although training databases have been elaborated by semi-automatic data cleaning algorithms [11], [14], [15], [7], there still exists noise inevitably. Due to the imperfection of training databases, strictly minimizing the intra-class distance and maximizing the inter-class distance would lead to overfitting. Therefore, our aim is to design a new loss function, which can increase the possibility of finding the best compromise between underfitting and overfitting to a specific training database, in order to obtaining better generalization ability.

Considering the imperfection of the training databases, formally, we abandon the softmax-based loss while start from the primary and fundamental idea: optimize intra-class and inter-class distances to some extent, to improve the generalization ability of models. Furthermore, we propose a novel loss function, named sigmoid-constrained hypersphere loss (SFace), to implement this idea. SFace imposes intra-class and inter-class constraints on a hypersphere manifold. The intra-class and inter-class constraints are controlled by two sigmoid curves. The sigmoid curves precisely re-scale intra-class and inter-class gradients so that intra-class and inter-class distances are optimized to some extent. As illustrated in Figure 1, for the deep feature \mathbf{x}_i of a training sample, the optimizing direction is always along the tangent of the hypersphere while the moving speed is controlled by the designed gradients precisely.

Yaoyao Zhong, Weihong Deng, and Jiani Hu are with the Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: zhongyaoyao@bupt.edu.cn; whdeng@bupt.edu.cn; jnhu@bupt.edu.cn). Weihong Deng is the corresponding author.

Dongyue Zhao, Xian Li, and Dongchao Wen are with Canon Information Technology (Beijing) Co., Ltd. (e-mail: zhaodongyue@canon-ib.com.cn; lixian@canon-ib.com.cn; wendongchao@canon-ib.com.cn).

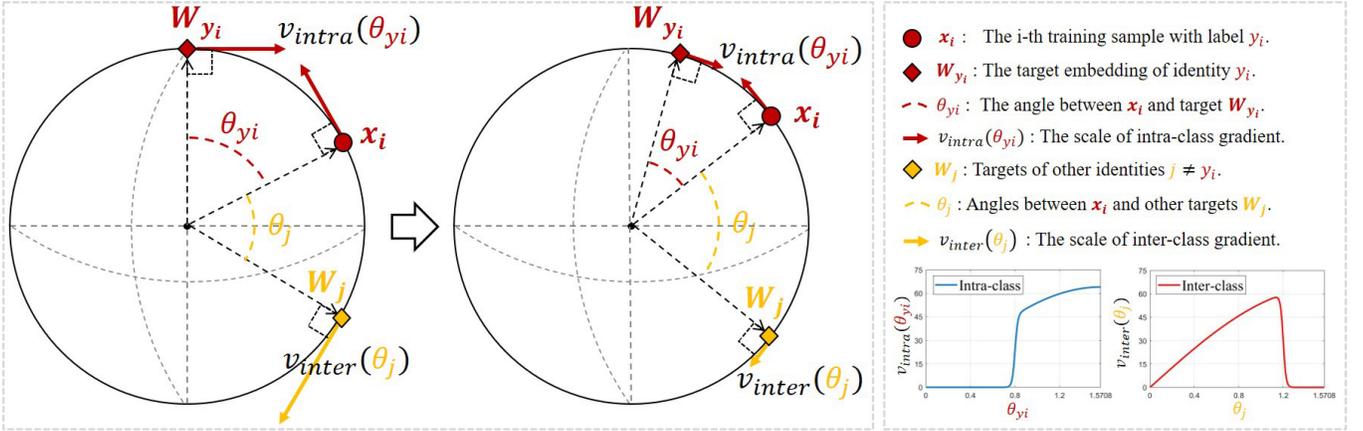


Fig. 1. Schematic illustration of the sigmoid-constrained hypersphere loss, which imposes intra-class and inter-class constraints on a hypersphere manifold. The optimizing directions of samples and target embedding are always along the tangent of the hypersphere while the moving speed is controlled by two sigmoid curves respectively. Specifically, the moving speed of the deep feature x_i and its target center W_{y_i} decreases gradually as they approaching to each other, while the moving speed of x_i and other target centers W_j increases rapidly as they start approaching to each other.

Specifically, the moving speed of the deep feature x_i and its target center W_{y_i} decreases gradually as they approaching to each other, while the moving speed of x_i and other target centers W_j increases rapidly as they start approaching to each other.

Compared with optimizing training samples strictly, the advantage of SFace is that it provides a relatively better balance between overfitting and underfitting, for the reason that SFace adopts sigmoid functions of intra-class and inter-class gradient re-scale terms to achieve excellent control respectively. We give a simple and easy example in Figure 2 for understanding. Under the label noise setting, the model would overfit to the label noise by strictly dragging the noisy samples to the wrong labeled identities. In contrast, SFace can mitigate this problem in some degree because it optimizes noisy samples in a moderate way. With the precisely control, the clean training samples are optimized earlier and more easily, while the label noise can be left behind.

Our major contributions can be summarized as follows:

- Considering the imperfection of face training databases, we introduce a new idea: optimizing intra-class and inter-class objectives in a moderate way to mitigate overfitting problem to face training databases.
- Under the guidance of this idea, we propose a new loss function, named sigmoid-constrained hypersphere loss (SFace), which can increase the possibility of finding the best compromise between underfitting and overfitting, in order to obtaining better generalization ability.
- Our method is evaluated on three training databases including CASIA-WebFace [11], VGGFace2 [14] and MS-Celeb-1M [12], and consistently outperforms the state-of-the-art methods on several benchmarks including LFW [16], YTF [17], CALFW [18], CPLFW [19], MegaFace [20], IJB-A [21] and IJB-C [22] databases.

The remainder of the paper is organized as follows. Section II briefly reviews the related deep face recognition works. In Section III, we first give a general introduction to the proposed sigmoid-constrained hypersphere loss (SFace). Then,

we detail the gradient re-scale function of SFace. Finally, we discuss the relationship between SFace and softmax based loss functions. Experimental settings and results are presented in Section IV. Section V summarizes the conclusions.

II. RELATED WORK

In this section, we discuss and compare the loss functions in deep face recognition, which are almost entirely around the idea of minimizing the intra-class distance and maximizing the inter-class distance. There are mainly two types.

The first type applies metric learning method in deep learning [1], [2], [3], which maps face images to a deep feature space and directly optimizes distances, so that the inter-class distance is larger than the intra-class distance. The contrastive loss [1], triplet loss [2] and N-pair loss [29] are early methods to enhance the discrimination ability of deep features, which optimize intra-class and inter-class variance by using face pairs. Combined with softmax loss, centerloss [3] obtains promising performance by simultaneously learns a center for deep features of each class and minimizes the distances between training samples and their corresponding class centers. Then, range loss [24] minimizes overall intra-personal differences and maximizes inter-personal differences in one mini-batch. Marginal loss [30] is further proposed to maximize the inter-class distance and minimize the intra-class distance simultaneously by focusing on the marginal samples.

The second type makes modification on cross-entropy loss (usually referred to as “softmax loss”) to learn more discriminative features [4], [5], [7]. Some early works incorporate weights or features normalization [31], [4], [32]. L2-softmax [31] is proposed to add an L2-constraint to the deep features and restrict them to lie on a hypersphere of a fixed radius. NSoftmax [4] is proposed to normalize both features and weights of the last inner-product layer. Ring loss [32] applies soft normalization by gradually learning to constrain the norm to the scaled unit circle while preserving convexity. Then, based on previous works [31], [4], the large margin [33], [6], [7] is introduced to obtain better discriminative

power by further enforcing the extra intra-class compactness and inter-class discrepancy simultaneously. L-Softmax [33] first incorporates a large margin to softmax loss to learn discriminative face features by strictly separating the hard samples. Instead of the multiplicative margin, CosFace [6] and ArcFace [7] introduce the additive margin to guarantee the convergence, which is easy for implementation. However, AdaCos [8] and P2SGrad [9] point that the inflexible form of softmax based loss functions lacks the ability to precisely supervise the cosine distances, and they improve the large margin angular loss functions by setting the direct mapping relation between classification probability and cosine distances, which can further decrease the intra-class angles of training databases. MV-Softmax [34] is proposed to improve softmax based loss functions by mining the mis-classified samples and emphasizing them to guide the discriminative feature learning. CurricularFace [35] further develops MV-Softmax by incorporating curriculum learning, which automatically emphasizes easy samples first and hard samples later. Recent works [7], [36] also point that inter-class and intra-class objectives of softmax based loss functions would interact and lead to relaxation on each other. Although recent works have pointed out some shortcomings of softmax based loss functions, overall, weight/feature normalization softmax-based loss functions and large margin softmax based loss functions have significantly boosted the performance of deep face recognition.

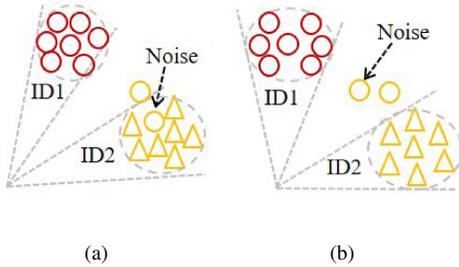


Fig. 2. (a) The model would overfit to the label noise by strictly dragging the noisy samples to the wrong labeled identities. (b) In contrast, SFace can mitigate this problem in some degree because it optimizes samples in a moderate way.

Our method can be categorized as the first type method in the form of metric learning, which directly optimizes the intra-class and inter-class distances. However, it also has a close connection to the second type based on softmax loss, which we will discuss in details in Section III-C. In addition, there are also some works [37], [38] aiming to solve the noise-robust training in deep face recognition, which usually use training databases with high-level label noise to obtain comparable performance with the model trained with clean databases. While our work is devoted to improving performance of models trained on clean databases which have been refined by semi-automatic data cleaning algorithms [11], [14], [7].

III. METHODOLOGY

A. Sigmoid-constrained Hypersphere Loss

In this section, we introduce the proposed loss function. First, we give some denotations and descriptions. The deep

face recognition models embeds an image into a d -dimensional Euclidean space. $\mathbf{x}_i \in \mathbb{R}^d$ denotes the embedding feature of the i -th training image, and y_i is the label of \mathbf{x}_i . $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_C\} \in \mathbb{R}^{d \times C}$ denotes the weight of the last fully connected layer, where C denotes the number of identities in the training database. $\mathbf{W}_{y_i} \in \mathbb{R}^d$ is seen as the target center feature of identity y_i .

Recent works [4], [5], [6], [7] have empirically demonstrated the superiority of constraining deep face features to be discriminative on a hypersphere manifold, where gradients are restricted in the tangent of the hypersphere. We also map deep face features to the hypersphere manifold and optimize cosine similarity to restrict directions of gradients. To help understanding, we illustrate it in Figure 1. With the restricted directions of gradients, the moving directions of samples and target centers are always along the tangent of the hypersphere.

The aim is to decrease the intra-class distance and increase the inter-class distance in a moderate way. Therefore, the sigmoid-constrained hypersphere loss (SFace) of \mathbf{x}_i can be formulated as $L_{SFace} = L_{intra}(\theta_{y_i}) + L_{inter}(\theta_j)$, where θ_{y_i} is the angular distance between $\mathbf{x}_i / \|\mathbf{x}_i\|$ and $\mathbf{W}_{y_i} / \|\mathbf{W}_{y_i}\|$, and θ_j ($j \neq y_i$) is the angular distance between $\mathbf{x}_i / \|\mathbf{x}_i\|$ and $\mathbf{W}_j / \|\mathbf{W}_j\|$. Specifically, $L_{intra}(\theta_{y_i})$ and $L_{inter}(\theta_j)$ are formulated as follows:

$$\begin{aligned} L_{intra}(\theta_{y_i}) &= -[r_{intra}(\theta_{y_i})]_b \cos(\theta_{y_i}), \\ L_{inter}(\theta_j) &= \sum_{j=1, j \neq y_i}^C [r_{inter}(\theta_j)]_b \cos(\theta_j). \end{aligned} \quad (1)$$

In the above equations, $\cos(\theta_{y_i}) = \mathbf{W}_{y_i}^T \mathbf{x}_i / (\|\mathbf{W}_{y_i}\| \|\mathbf{x}_i\|)$, and $\cos(\theta_j) = \mathbf{W}_j^T \mathbf{x}_i / (\|\mathbf{W}_j\| \|\mathbf{x}_i\|)$, $j \neq y_i$. Since the goal is to obtain precisely control of the optimization degree, we design functions $r_{intra}(\theta_{y_i})$ and $r_{inter}(\theta_j)$ to re-scale intra-class and inter-class objectives respectively to further restrict the optimizing speed. $[\cdot]_b$ is the block gradient operator, which prevents the contribution of its inputs to be taken into account for computing gradients. In the forward propagation process of SFace,

$$\begin{aligned} L_{SFace} &= \\ &= -[r_{intra}(\theta_{y_i})]_b \cos(\theta_{y_i}) + \sum_{j=1, j \neq y_i}^C [r_{inter}(\theta_j)]_b \cos(\theta_j). \end{aligned} \quad (2)$$

While in the backward propagation process,

$$\begin{aligned} \frac{\partial L_{SFace}}{\partial \mathbf{x}_i} &= \\ &= -[r_{intra}(\theta_{y_i})]_b \frac{\partial \cos(\theta_{y_i})}{\partial \mathbf{x}_i} + \sum_{j=1, j \neq y_i}^C [r_{inter}(\theta_j)]_b \frac{\partial \cos(\theta_j)}{\partial \mathbf{x}_i}, \\ \frac{\partial L_{SFace}}{\partial \mathbf{W}_{y_i}} &= -[r_{intra}(\theta_{y_i})]_b \frac{\partial \cos(\theta_{y_i})}{\partial \mathbf{W}_{y_i}}, \\ \frac{\partial L_{SFace}}{\partial \mathbf{W}_j} &= [r_{inter}(\theta_j)]_b \frac{\partial \cos(\theta_j)}{\partial \mathbf{W}_j}, \end{aligned} \quad (3)$$

where

$$\begin{aligned}
\frac{\partial \cos(\theta_{y_i})}{\partial \mathbf{x}_i} &= \frac{1}{\|\mathbf{x}_i\|} \left(\frac{\mathbf{W}_{y_i}}{\|\mathbf{W}_{y_i}\|} - \cos(\theta_{y_i}) \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} \right), \\
\frac{\partial \cos(\theta_j)}{\partial \mathbf{x}_i} &= \frac{1}{\|\mathbf{x}_i\|} \left(\frac{\mathbf{W}_j}{\|\mathbf{W}_j\|} - \cos(\theta_j) \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} \right), \\
\frac{\partial \cos(\theta_{y_i})}{\partial \mathbf{W}_{y_i}} &= \frac{1}{\|\mathbf{W}_{y_i}\|} \left(\frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} - \cos(\theta_{y_i}) \frac{\mathbf{W}_{y_i}}{\|\mathbf{W}_{y_i}\|} \right), \\
\frac{\partial \cos(\theta_j)}{\partial \mathbf{W}_j} &= \frac{1}{\|\mathbf{W}_j\|} \left(\frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} - \cos(\theta_j) \frac{\mathbf{W}_j}{\|\mathbf{W}_j\|} \right).
\end{aligned} \tag{4}$$

B. Gradient Re-scale Function

The optimization gradients are always along the tangent direction, because $\langle \frac{\partial \cos(\theta_{y_i})}{\partial \mathbf{x}_i}, \mathbf{x}_i \rangle = 0$, $\langle \frac{\partial \cos(\theta_j)}{\partial \mathbf{x}_i}, \mathbf{x}_i \rangle = 0$, $\langle \frac{\partial \cos(\theta_{y_i})}{\partial \mathbf{W}_{y_i}}, \mathbf{W}_{y_i} \rangle = 0$, and $\langle \frac{\partial \cos(\theta_j)}{\partial \mathbf{W}_j}, \mathbf{W}_j \rangle = 0$ (refer to the illustration in Figure 3). In addition, $\|\mathbf{x}_i\|$, $\|\mathbf{W}_{y_i}\|$ and $\|\mathbf{W}_j\|$ almost remain unchanged in the training process, for the reason that there are no components of gradients in the radial direction. Function $r_{intra}(\theta_{y_i})$ and $r_{inter}(\theta_j)$ are designed to re-scale intra-class and inter-class objectives respectively. These two terms actually re-scale the gradient, *i.e.* control the moving speed of samples and target centers in Figure 1. Therefore we name $r_{intra}(\theta_{y_i})$ and $r_{inter}(\theta_j)$ as the gradient re-scale functions. Since the original gradient scales of intra-class and inter-class objectives are proportional to $\sin\theta_{y_i}$ and $\sin\theta_j$ (refer to Function (4) and Figure 3), the final gradient scales are proportional to $v_{intra}(\theta_{y_i}) = r_{intra}(\theta_{y_i}) \sin\theta_{y_i}$ and $v_{inter}(\theta_j) = r_{inter}(\theta_j) \sin\theta_j$.

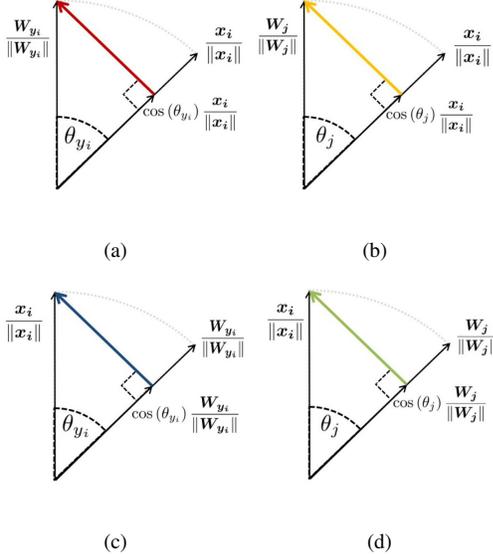


Fig. 3. Illustration of Function (4), which means that optimization gradients are along the tangent direction. (a)(b)(c)(d) interprets the four orthogonality relationships of Function (4) respectively.

At the beginning of training, the initial angular distances θ_{y_i} and θ_j are all about $\frac{\pi}{2}$. The intra-class loss function decreases θ_{y_i} gradually while the inter-class loss function prevents θ_j from being decreased. Therefore, the ideal functions of $v_{intra}(\theta_{y_i})$ and $v_{inter}(\theta_j)$ should satisfy at least three

properties as follows: (1) The function $v_{intra}(\theta_{y_i})$ should be non-negative and monotonically increasing on the interval $[0, \frac{\pi}{2}]$, so that the moving speed of \mathbf{x}_i and \mathbf{W}_{y_i} decreases gradually as they approaching to each other. (2) The function $v_{inter}(\theta_j)$ should be non-negative on the interval $[0, \frac{\pi}{2}]$, so that the moving speed of \mathbf{x}_i and \mathbf{W}_j increases rapidly as they start approaching to each other. (3) Considering the imperfection of training databases, there should be two flexible intervals to suppress the moving speed, one is around $\theta_{y_i} \approx 0$ of $v_{intra}(\theta_{y_i})$ and the other is around $\theta_j \approx \frac{\pi}{2}$ of $v_{inter}(\theta_j)$, so that both intra-class and inter-class objectives can be optimized with a moderate target rather than be minimized or maximized strictly.

Eventually, we choose sigmoid functions as the gradient re-scale functions. The specific forms are,

$$\begin{aligned}
r_{intra}(\theta_{y_i}) &= \frac{s}{1 + e^{-k*(\theta_{y_i}-a)}}, \\
r_{inter}(\theta_j) &= \frac{s}{1 + e^{k*(\theta_j-b)}}.
\end{aligned} \tag{5}$$

s is the upper asymptote of two sigmoid curves as the initial scale of gradient, and k is the control the slope of sigmoid curves. Hyperparameters a and b decide the horizontal intercept of two sigmoid curves and actually control the flexible interval to suppress the moving speed. Therefore a and b are vital parameters should be selected according to characteristics of a specific training database, which we will discuss later. The sigmoid curve functions of $r_{intra}(\theta_{y_i})$ and $r_{inter}(\theta_j)$ are illustrated in (a) of Figure 4. With the gradient re-scale functions, scales of intra-class gradient and inter-class gradient in theory are proportional to $v_{intra}(\theta_{y_i}) = r_{intra}(\theta_{y_i}) \sin\theta_{y_i}$ and $v_{inter}(\theta_j) = r_{inter}(\theta_j) \sin\theta_j$, shown in (b) of Figure 4. The entire training process of SFace is summarized in Algorithm 1, which is easy for implementation.

Algorithm 1: SFace

Input: Embedding feature \mathbf{x}_i with label y_i , parameters of the embedding network Θ , parameters of the last fully-connected layer \mathbf{W} (composed of \mathbf{W}_{y_i} and \mathbf{W}_j ($j \neq y_i$)), SFace parameters s and k , a and b , the number of iteration $i = 0$, learning rate $\lambda^{(i)}$,

```

1 while not converged do
2    $i = i + 1$ ;
3   Compute the intra-distance by
    $\theta_{y_i} = \arccos(\mathbf{W}_{y_i}^T \mathbf{x}_i / \|\mathbf{W}_{y_i}\| \|\mathbf{x}_i\|)$ ;
4   Compute the inter-distance by
    $\theta_j = \arccos(\mathbf{W}_j^T \mathbf{x}_i / \|\mathbf{W}_j\| \|\mathbf{x}_i\|)$ ,  $j \neq y_i$ ;
5   Compute gradient re-scale functions by Equation 5;
6   Compute the loss by Equation 2;
7   Compute the gradients of  $\mathbf{x}_i$  and  $\mathbf{W}$  by Equation 3;
8   Update parameters  $\mathbf{W}$  and  $\Theta$  by
    $\mathbf{W} = \mathbf{W} - \lambda^{(i)} \frac{\partial LS_{Face}}{\partial \mathbf{W}}$ ,  $\Theta = \Theta - \lambda^{(i)} \frac{\partial LS_{Face}}{\partial \Theta} \frac{\partial \mathbf{x}_i}{\partial \Theta}$ ;
9 end
Output:  $\mathbf{W}$ ,  $\Theta$ .

```

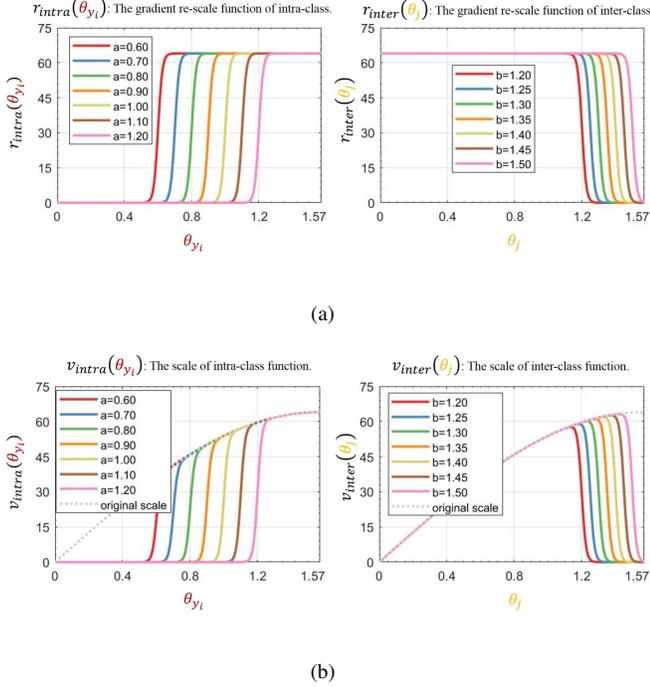


Fig. 4. (a) The sigmoid curves of intra-class gradient re-scale function $r_{intra}(\theta_{y_i})$ and inter-class gradient re-scale function $r_{inter}(\theta_j)$ of SFace. (b) The final scale curves of intra-class gradient $v_{intra}(\theta_{y_i})$ and inter-class gradient $v_{inter}(\theta_j)$ of SFace.

C. Relation to Softmax Based Loss

We have mentioned in Section II that SFace in form can be categorized as the metric learning method, but it has a close connection to the softmax based loss functions. In this section, we discuss this relation in details.

We start from the original softmax loss function. For each embedding feature \mathbf{x}_i , the softmax loss can be formulated as:

$$L = -\log P_{y_i} = -\log \frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i + \mathbf{b}_{y_i}}}{\sum_{j=1}^C e^{\mathbf{W}_j^T \mathbf{x}_i + \mathbf{b}_j}}. \quad (6)$$

$\mathbf{x}_i \in \mathbb{R}^d$ denotes the embedding feature of the i -th training image, and y_i is the label of \mathbf{x}_i . P_{y_i} is the predicted probability of assigning \mathbf{x}_i to class y_i . C is the number of identities, $\mathbf{W}_j \in \mathbb{R}^d$ is the j -th column of the weight of the last fully connected layer, $\mathbf{b}_j \in \mathbb{R}^C$ is the bias. Softmax based loss functions [4], [5], [6], [7] remove the bias term and transform $\mathbf{W}_j^T \mathbf{x}_i = s \cos \theta_j$. To further improve the performance, large margin is adopted in the $\cos \theta_{y_i}$ term [5], [6], [7]. Therefore, softmax based loss functions can be formulated as:

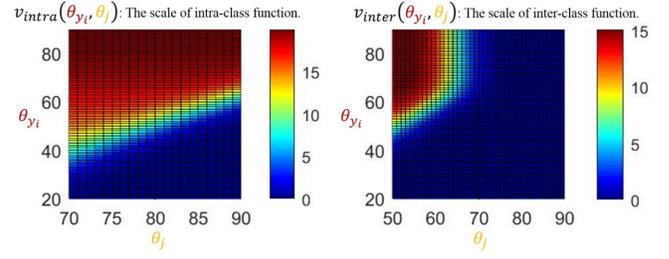
$$L = -\log P_{y_i} = -\log \frac{e^{s f(\theta_{y_i})}}{e^{s f(\theta_{y_i})} + \sum_{j=1, j \neq y_i}^C e^{s \cos \theta_j}}, \quad (7)$$

where $f(\theta_{y_i}) = \cos \theta_{y_i}$ in NSoftmax [4], $f(\theta_{y_i}) = \cos \theta_{y_i} - m$ in CosFace [6], and $f(\theta_{y_i}) = \cos(\theta_{y_i} + m)$ in ArcFace [7]. With the influence of the loss function, θ_{y_i} is decreased and

θ_j is increased in theory. In the backward propagation process,

$$\begin{aligned} \frac{\partial L}{\partial \cos \theta_{y_i}} &= s(P_{y_i} - 1) \frac{\partial f(\theta_{y_i})}{\partial \cos \theta_{y_i}} \\ &= -\frac{s \sum_{j=1, j \neq y_i}^C e^{s \cos \theta_j}}{e^{s f(\theta_{y_i})} + \sum_{j=1, j \neq y_i}^C e^{s \cos \theta_j}} \frac{\partial f(\theta_{y_i})}{\partial \cos \theta_{y_i}}, \quad (8) \\ \frac{\partial L}{\partial \cos \theta_j} &= s P_j = \frac{s e^{s \cos \theta_j}}{e^{s f(\theta_{y_i})} + \sum_{k=1, k \neq y_i}^C e^{s \cos \theta_k}}, \end{aligned}$$

where $\frac{\partial f(\theta_{y_i})}{\partial \cos \theta_{y_i}} = 1$ in NSoftmax [4] and CosFace [6], and $\frac{\partial f(\theta_{y_i})}{\partial \cos \theta_{y_i}} = \frac{\sin(\theta_{y_i} + m)}{\sin \theta_{y_i}}$ in ArcFace [7].



(a) NSoftmax [4]

(b) CosFace [6]

(c) ArcFace [7]

Fig. 5. Under some ideal assumptions, the scale of intra-class gradient $v_{intra}(\theta_{y_i}, \theta_j)$ and inter-class gradient $v_{inter}(\theta_{y_i}, \theta_j)$ of (a) NSoftmax [4], (b) CosFace [6], and (c) ArcFace [7]. Softmax based loss functions [4], [6], [7] can be understood as a kind of special metric learning method with specific speed constraints decided by the intra-class distance θ_{y_i} and the inter-class distances θ_j ($j \neq y_i$).

Further, the softmax based functions are equivalent to the following loss functions for training face models

$$L = -[r_{intra}(\theta_{y_i}, \theta_j)]_b \cos(\theta_{y_i}) + \sum_{j=1, j \neq y_i}^C [r_{inter}(\theta_{y_i}, \theta_j)]_b \cos(\theta_j), \quad (9)$$

where gradient re-scale functions are,

$$r_{intra}(\theta_{y_i}, \theta_j) = \frac{s \sum_{j=1, j \neq y_i}^C e^{s \cos \theta_j} \frac{\partial f(\theta_{y_i})}{\partial \cos \theta_{y_i}}}{e^{s f(\theta_{y_i})} + \sum_{j=1, j \neq y_i}^C e^{s \cos \theta_j}}, \quad (10)$$

and

$$r_{inter}(\theta_{y_i}, \theta_j) = \frac{s e^{s \cos \theta_j}}{e^{s f(\theta_{y_i})} + \sum_{k=1, k \neq y_i}^C e^{s \cos \theta_k}}. \quad (11)$$

Since only backward propagation have influence on the network parameters of deep face recognition models, and the backward propagation function (8) of softmax based loss functions and function (9) are the same. Therefore loss function (7) are equivalent to loss function (9) in the training process.

Now from equations (8)(9)(10)(11), we can see that softmax based loss functions can be understood as a kind of special metric learning method with the speed constraints on a hypersphere. However, both the gradient re-scale functions (speed constraints) of intra-class and inter-class are decided by the intra-class distance θ_{y_i} and the inter-class distances θ_j ($j \neq y_i$). To better understanding of the optimization of softmax based loss functions, we hypothesize that all the inter-class distances θ_j ($j \neq y_i$) are the same ideally, and plot scale curves of the intra-class gradient $v_{intra}(\theta_{y_i}, \theta_j) = r_{intra}(\theta_{y_i}, \theta_j) \sin \theta_{y_i}$ and the inter-class gradient $v_{inter}(\theta_{y_i}, \theta_j) = r_{inter}(\theta_{y_i}, \theta_j) \sin \theta_j$ of (a) NSoftmax [4], (b) CosFace [6], and (c) ArcFace [7] in Figure 5. At the beginning of training, the intra-class distance θ_{y_i} and inter-class distances θ_j is about 90 degrees ($\frac{\pi}{2}$). We can see that, from the intra-class sub-figure (left) of Figure 5, with the high intra-class gradient v_{intra} , the intra-class distance θ_{y_i} will decrease gradually. While at the same time, as the intra-class distance θ_{y_i} decreases, from the inter-class sub-figure (right) of Figure 5, the inter-class gradient v_{inter} will decrease, which will relax the inter-class constraints and decrease the inter-class distance θ_j . Then, we come back to the intra-class sub-figure (left) of Figure 5, as the inter-class distances θ_j decrease, the change curve of intra-class gradient v_{intra} vs θ_{y_i} will also changed.

In the optimization of softmax based loss, the intra-class and inter-class distance will always have influence on each other. Therefore, in conclusion, softmax based loss functions actually lack the ability to control intra-class and inter-class optimizations precisely. However, compared with softmax based loss functions, both intra-class and inter-class distance of SFace (Figure 4) can be constrained to a designed degree therefore can be optimized in a moderate way, which is exactly the advantage of SFace.

IV. EXPERIMENTS

A. Experimental settings

We separately train models on training databases including CASIA-WebFace [11], VGGFace2 [14], MS1MV2 [12] databases, which have been elaborated by semi-automatic data cleaning algorithms, to evaluate our methods and conduct fair comparison with state-of-the-art loss functions. The compared loss functions include softmax, NSoftmax [4], SphereFace [5],

CosFace [6], ArcFace [7], Combined loss [7], D-softmax [36] and so on.

Evaluation Databases. We evaluate on LFW [16], YTF [17], CFP-FP [39], AgeDB-30 [40], CALFW [18], CPLFW [19], MegaFace [20], IJB-A [21] and IJB-C [22] databases.

LFW [16] database contains 13,233 face images from 5,749 different identities. YTF [17] is a database of face video collected from YouTube, which consists of 3,425 videos of 1,595 different people. CFP-FP database [39] is built for facilitating large pose variation in unconstrained settings. AgeDB-30 database [40] is a manually collected cross-age database in unconstrained settings. Cross-Age LFW (CALFW) [18] and Cross-Pose LFW (CPLFW) [19] databases are constructed based on LFW database, to emphasize cross-age challenge and cross-pose challenge in face recognition.

MegaFace [20] is a large public available testing benchmark, which evaluates the performance of face models at the million scale distractors. We use FaceScrub database [41] as the probe set, which contains 106,863 images from 530 celebrities. The gallery set is a subset of Flickr photos and it consists of more than one million images. Recently, research [7] points out that there are many wrong labels in the MegaFace database and the noise significantly affects the performance. Therefore, for comparison, in this paper we report experimental results on both the original MegaFace database and the refined version [7].

IJB-A [21] and IJB-C [22] databases address the unconstrained face recognition, which contain both still images and video frames. IJB-A database contains 500 subjects with 5,396 still images and 20,395 video frames. IJB-C database further increases emphasis on occlusion and diversity of subject occupation and geographic origin population, containing 3,531 subjects with 31.3K still images and 117.5K frames from 11,779 videos. We evaluate the models on the standard verification setting (matching between the Mixed Media probes and two galleries) and identification protocol (1:N Mixed Media probes across two galleries).

Training and Testing. We use MxNet [42] to implement all the experiments. For the fair comparison, the CNN architecture used in our work is the same ResNet [43] networks as [7], which applies the ‘‘BN [44]-Dropout [45]-FC-BN’’ structure to get the final 512- D embedding feature. The data preprocessing follows settings of insightface [7]. That is, horizontally flip with a probability of 50% is used for training data augmentation. In addition, all the images are normalized by subtracting 127.5 and dividing by 128. All the models are trained with stochastic gradient descent (SGD) algorithm from scratch. Models trained on CASIA-WebFace database are trained on 2 GPUs and the total batch size is 256. The learning rate is started from 0.1 and divided by 10 at the 100k, 140k, 160k iterations. Models trained on MS1MV2 database are trained on 4 GPUs and the total batch size is 512. The learning rate is started from 0.1 and divided by 10 at the 100k, 160k, 220k iterations. Models trained on VGGFace2 database are trained on 4 GPUs and the total batch size is 512. The learning rate is started from 0.1 and divided by 10 at the 80k, 100k, 160k iterations. The parameter s for SFace is set to 64, k is set to

80. The intra-class and inter-class parameters a and b control the optimization and should be decided according to specific training databases, which will be introduced later.

B. Experiment on the CASIA-WebFace Database

CASIA-WebFace database [11] contains 0.49M images from 10,575 celebrities, which is the first widely used large training database in deep face recognition. While recently it has been seen as a relatively small-scale database compared with other Million-scale ones [12], [14]. According to the research [15], there are 9.3%-13.0% label noise in CASIA-WebFace database. That is, the original CASIA-WebFace database is exactly the database using semi-automatic annotation with low level noise. We use the arcface version [7] with 0.49M images from 10,572 identities. We first implement our method on it and compare with the state-of-art loss functions. Then, we experiment on the noise-controlled WebFace database to further evaluate our method under training databases with different noise levels, and study the choice of hyper-parameters.

1) *Experiment on the CASIA-WebFace Database:* We train face models on CASIA-WebFace database supervised by softmax, NSoftmax [4], SphereFace [5], CosFace [6], ArcFace [7], Combined loss [7] with combined margin $\cos(m_1\theta + m_2) - m_3$, D-softmax [36], and SFace respectively. The source codes of most compared methods can be downloaded from the github. In addition, we implement D-softmax [36] by ourselves. Since the performance of all the above loss functions is sensitive to the choice of hyper-parameters, we list them in the Table I, which are determined according to the suggestion. All the models are trained on the ResNet50 which we have mentioned above. For SFace, we choose intra-class and inter-class hyper-parameters a and b by taking reference to the experience of the final models of large margin loss functions, and then tuning them. In the experiment, both intra-class and inter-class parameters have crucial influence. Table I lists the experimental results, our method is compared with the recent advanced loss functions. As shown, under the same training and test settings, our method significantly improves the results on several evaluation benchmarks, especially TAR at very low FAR on the well-known challenging IJB-C database, which demonstrates the superiority of our method on a semi-automatic annotated face training database with low level noise.

From Table I, we select three models trained supervised by SFace and two classic methods, NSoftmax and ArcFace, respectively, and analyze these models. We extract the deep features of images in the training database, and calculate intra-class and inter-class angles (distances) statistics. Specifically, using the manual refined image list [46] released by [4], we can split the training database (0.49M images) into clean images (0.45M) and label noises (0.04M). Therefore, the mean angles (distances) between embedding feature x_i and the embedding feature W_{y_i} of clean images and label noise can be calculated respectively. In addition, we calculate the mean angles between different W_j . The results are listed in Table II. We can see that, compared with NSoftmax and SFace, ArcFace

optimizes training samples in a more strict way. That is, the intra-class angles (distances) of ArcFace are smaller. The decrease of intra-class angles (distances) of clean images is a good trend. However, the intra-class angles (distances) of label noise are also decreased, which is not a good phenomenon. While SFace keep a better balance between decreasing the intra-class angles (distances) and preventing overfitting to label noise. The reason may be that with the precisely control to a cutoff point, the clean training samples are optimized earlier and more easily, while the label noise can be left behind to prevent them close to the wrong labeled targets. At the same time, the inter-class class optimization guarantees that different identities still remain to be orthogonal to each other.

To evaluate the proposed gradient re-scale function of SFace, we compare face models trained on loss function (1) with three different gradient re-scale functions: constant value (no gradient re-scale), the piecewise functions, and the sigmoid functions (SFace). Specifically, the piecewise function can be seen as the “steep version” of the sigmoid functions, formulated as follows,

$$\begin{aligned} r_{intra}(\theta_{y_i}) &= s * \text{sign}(\max(\theta_{y_i} - a, 0)), \\ r_{inter}(\theta_j) &= s * \text{sign}(\max(b - \theta_j, 0)), \end{aligned} \quad (12)$$

where $\text{sign}(\ast)$ is the sign function to extract the sign of a real number. For the piecewise and sigmoid functions, the hyper-parameters a , b are set as the same, 0.9 and 1.3. The experimental results are listed in Table III. We can see that, the proposed sigmoid gradient re-scale function has better performance than the constant value and the piecewise version.



Fig. 6. Images of an identity in WebFace-Clean, WebFace-ArcFace and WebFace-Noisy databases. The WebFace-Clean database is a manually cleaned version [46], [7]. The noise in WebFace-ArcFace [7] database is from the label noise that derive from the collection process of the CASIA-WebFace database [11]. Based on WebFace-ArcFace database, we add images from MS-Celeb-1M database [12] evenly across each identity of WebFace-ArcFace database, which means that we incorporate outliers in WebFace-Noisy database.

2) *Experiment on the Noise-Controlled WebFace Database:* To further evaluate our method on the training databases with low level noise, we train deep face models under noise-controlled settings. Specifically, we use three databases with different noise level. (1) Since we have the manual refined image list released by [46], we first clean the ArcFace version [7]

TABLE I

COMPARISON OF DIFFERENT LOSS FUNCTIONS WITH SFACE. MODELS ARE TRAINED ON CASIA-WebFace [11] USING RESNET50. THE COMBINED LOSS [7] ADOPTS THE COMBINED MARGIN $\cos(m_1\theta + m_2) - m_3$. THE EVALUATION BENCHMARK CONTAINS IJB-C [22] (TAR@FAR=1E-5, 1E-4, 1E-3), YTF [17] (%) DATABASES, AND AVERAGE PERFORMANCE (%) ON LFW [16], CFP-FP [39], AGEDB-30 [40], CALFW [18] AND CPLFW [19] DATABASES.

| Method | IJB-C | | | YTF | Avg. | LFW | CFP-FP | AgeDB-30 | CPFLW | CALFW |
|---------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | FAR=1e-5 | FAR=1e-4 | FAR=1e-3 | | | | | | | |
| softmax | 64.57 | 77.57 | 88.03 | 95.60 | 93.82 | 99.25 | 95.10 | 93.28 | 88.97 | 92.48 |
| NSoftmax [4] (s=20.0) | 67.82 | 79.88 | 89.33 | 95.54 | 93.72 | 99.23 | 95.00 | 93.17 | 88.82 | 92.40 |
| SphereFace [5] (m=1.35) | 46.73 | 61.54 | 76.10 | 93.18 | 92.99 | 99.17 | 94.76 | 92.60 | 86.50 | 91.93 |
| CosFace [6] (m=0.35) | 75.58 | 85.03 | 92.00 | 95.76 | 94.91 | 99.53 | 95.50 | 95.23 | 90.32 | 93.97 |
| ArcFace [7] (m=0.3) | 73.55 | 84.60 | 91.90 | 95.80 | 94.65 | 99.57 | 95.26 | 94.40 | 90.10 | 93.93 |
| ArcFace [7] (m=0.4) | 72.49 | 83.76 | 91.21 | 96.06 | 94.91 | 99.52 | 95.76 | 95.00 | 90.43 | 93.87 |
| ArcFace [7] (m=0.5) | 70.15 | 81.48 | 90.26 | 95.66 | 94.83 | 99.52 | 95.60 | 95.30 | 89.97 | 93.77 |
| Combined [7] (m = 0.9,0.4,0.15) | 73.99 | 83.91 | 91.63 | 95.86 | 94.90 | 99.48 | 95.56 | 94.97 | 90.68 | 93.82 |
| D-softmax [36] (d=0.9) | 71.48 | 83.56 | 91.23 | 95.42 | 94.29 | 99.50 | 95.44 | 93.95 | 89.60 | 92.95 |
| SFace (a=0.87, b=1.20) | 77.13 | 86.38 | 92.52 | 95.82 | 94.93 | 99.50 | 95.81 | 95.10 | 90.18 | 94.07 |
| SFace (a=0.90, b=1.20) | 76.77 | 85.95 | 92.37 | 95.86 | 94.88 | 99.57 | 95.67 | 95.00 | 90.22 | 93.95 |
| SFace (a=0.93, b=1.20) | 77.77 | 86.38 | 92.52 | 96.08 | 94.88 | 99.48 | 95.81 | 94.87 | 90.28 | 93.97 |
| SFace (a=0.90, b=1.30) | 76.92 | 87.27 | 93.11 | 96.00 | 94.80 | 99.57 | 95.26 | 94.82 | 90.68 | 93.70 |

TABLE II

THE ANGLES (DISTANCES) STATISTICS UNDER DIFFERENT LOSS FUNCTIONS (NSOFTMAX [4], ARCFACE [7] AND SFACE MODELS TRAINED ON WebFace DATABASE (0.49M IMAGES)). EACH COLUMN DENOTES ONE LOSS FUNCTION. "CLEAN-INTRA" AND "NOISE-INTRA" REFERS TO CALCULATE THE MEAN ANGLES (DISTANCES) BETWEEN EMBEDDING FEATURE x_i AND THE EMBEDDING FEATURE W_{y_i} OF CLEAN IMAGES AND LABEL NOISE, RESPECTIVELY. WE USE THE MANUAL REFINED IMAGE LIST RELEASED BY [46] TO SPLIT THE TRAINING DATABASE (0.49M IMAGES) INTO CLEAN IMAGES (0.45M) AND LABEL NOISES (0.04M). "DELTA-INTRA" IS THE DIFFERENCE BETWEEN "NOISE-INTRA" AND "CLEAN-INTRA". "INTER" REFERS TO THE MEAN ANGLES BETWEEN DIFFERENT W_j .

| | NSoftmax [4] | ArcFace [7] | SFace |
|-------------|--------------|-------------|------------|
| Clean-Intra | 44.42 | 35.31 | 39.68 |
| Noise-Intra | 50.85 | 40.09 | 47.30 |
| Delta-Intra | 6.43 | 4.78 | 7.62 |
| Inter | 89.75±5.55 | 89.99±4.73 | 89.96±4.67 |

TABLE III

COMPARISON OF THREE DIFFERENT GRADIENT RE-SCALE FUNCTIONS: CONSTANT VALUE (NO GRADIENT RE-SCALE), THE PIECEWISE FUNCTIONS, AND THE SIGMOID FUNCTIONS (SFACE). MODELS ARE TRAINED ON CASIA-WebFace [11] USING RESNET50. THE AVERAGE PERFORMANCE (%) ON LFW [16], CFP-FP [39], AGEDB-30 [40], CALFW [18] AND CPLFW [19] DATABASES IS USED FOR EVALUATION.

| Method | Avg. | LFW | CFP-FP | AgeDB-30 | CPLFW | CALFW |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
| Constant | 90.05 | 98.30 | 90.46 | 89.55 | 83.15 | 88.78 |
| Piecewise | 94.64 | 99.45 | 94.90 | 94.73 | 90.08 | 94.03 |
| Sigmoid | 94.80 | 99.57 | 95.26 | 94.82 | 90.68 | 93.70 |

of CASIA-WebFace database. Finally, we obtain a manually cleaned version of CASIA-WebFace database (0.45M images from 10,572 identities). This database is named as WebFace-Clean. (2) Then, ArcFace version [7] of CASIA-WebFace database (0.49M images from 10,572 identities) is used as first noise level database. We name this database as WebFace-ArcFace. (3) Finally, we augment the ArcFace version [7] of CASIA-WebFace database with synthesis images. We add images from MS-Celeb-1M database [12] evenly across each identity of WebFace-ArcFace database. That is to say, we

TABLE IV

STUDY ON THE CHOICE OF HYPE-PARAMETERS a AND b OF SFACE (RESNET34). AS THE NOISE LEVEL INCREASES, PARAMETER a SHOULD BE LARGER, *i.e.* $v_{intra}(\theta_{y_i})$ CURVES SHOULD MOVE TO THE RIGHT, WHICH INDICATES THAT THE SPEED OF INTRA-CLASS IS DECREASED MORE EARLY TO PREVENT OVERFITTING.

| Noise | Parameters | | IJB-C | |
|--|-------------|-------------|--------------|--------------|
| | a | b | FAR=1e-4 | FAR=1e-3 |
| WebFace-Clean (Noise Level \approx 0%) | 0.81 | 1.28 | 84.70 | 91.71 |
| | 0.80 | 1.28 | 85.72 | 92.52 |
| | 0.80 | 1.25 | 83.99 | 91.17 |
| WebFace-ArcFace (Noise Level \approx 10%) | 0.80 | 1.28 | 85.39 | 92.09 |
| | 0.82 | 1.28 | 86.30 | 92.43 |
| | 0.82 | 1.25 | 84.17 | 91.32 |
| WebFace-Noisy (Noise Level \approx 20%) | 0.82 | 1.28 | 83.97 | 91.36 |
| | 0.84 | 1.28 | 84.80 | 91.84 |
| | 0.84 | 1.25 | 84.09 | 91.43 |

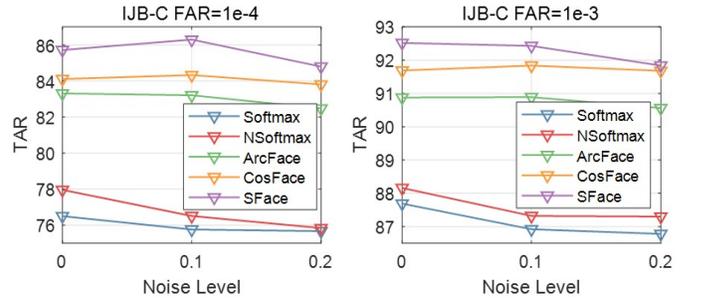


Fig. 7. Comparison of verification TAR@FAR=1e-4 and TAR@FAR=1e-3 results on the IJB-C database [22] of softmax, NSoftmax, CosFace, ArcFace and SFace models (ResNet34) which are trained with databases of different noise level (WebFace-Clean (\approx 0%), WebFace-ArcFace (\approx 10%), and WebFace-Noisy (\approx 20%).

incorporate outliers in this training database. The database is referred to as WebFace-Noisy. We use this setting because in practice, outliers noise is a more common type of label noise than label flip noise. The noise level of WebFace-Clean, WebFace-ArcFace and WebFace-Noisy is approximately 0%, 10% and 20%, respectively. Some identities of WebFace-

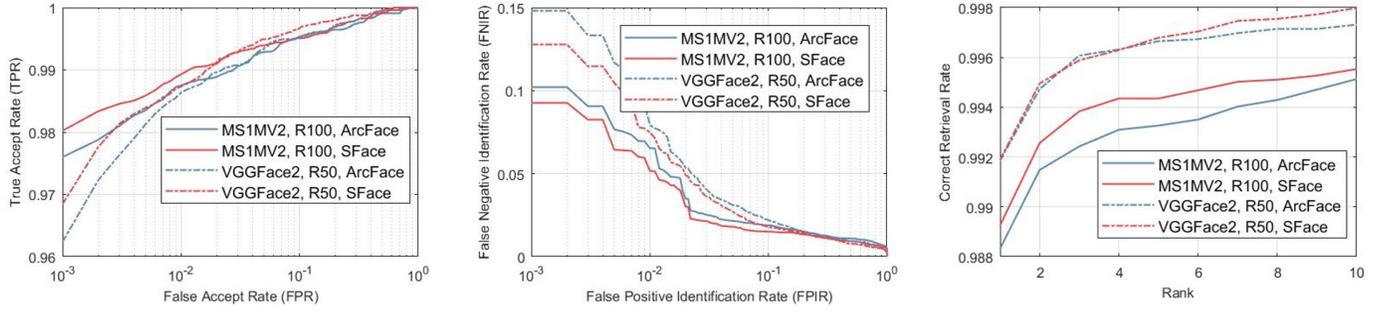


Fig. 8. Comparison of ArcFace and SFace models on the IJB-A database [21]. Left: ROC (higher is better). Middle: DET (lower is better). Right: CMC (higher is better). Our method is represented using red color.

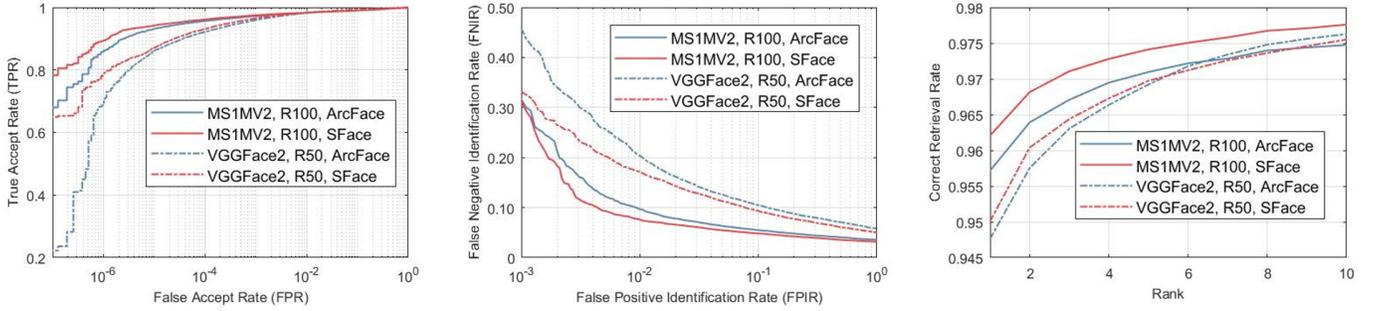


Fig. 9. Comparison of ArcFace and SFace models on the IJB-C database [22]. Left: ROC (higher is better). Middle: DET (lower is better). Right: CMC (higher is better). Our method is represented using red color.

TABLE V
VERIFICATION PERFORMANCE ON LFW [16] AND YTF [17] DATABASES.
THE STATE-OF-ART MODELS IN FACE RECOGNITION COMMUNITY ARE
LISTED FOR COMPARISON.

| Method | #Images | LFW | YTF |
|---------------------------|---------|--------------|--------------|
| DeepID [1] | 0.2M | 99.47 | 93.20 |
| DeepFace [47] | 4.4M | 97.35 | 91.4 |
| VGG Face [48] | 2.6M | 98.95 | 97.30 |
| FaceNet [2] | 200M | 99.63 | 95.10 |
| Baidu [49] | 1.3M | 99.13 | - |
| Center Loss [3] | 0.7M | 99.28 | 94.9 |
| Range Loss [24] | 5M | 99.52 | 93.70 |
| Marginal Loss [30] | 3.8M | 99.48 | 95.98 |
| SphereFace [5] | 0.5M | 99.42 | 95.0 |
| SphereFace+ [50] | 0.5M | 99.47 | - |
| CosFace [6] | 5M | 99.73 | 97.6 |
| MS1MV2, R100, ArcFace [7] | 5.8M | 99.83 | 98.02 |
| MS1MV2, R100, SFace | 5.8M | 99.82 | 98.06 |

TABLE VI
VERIFICATION PERFORMANCE ON ON LFW [16], CALFW [18] AND
CPLFW [19] DATABASES. THE SECOND CELL LISTS RESULTS OF THE
OPEN-SOURCED FACE RECOGNITION MODELS OF STATE-OF-ART
METHODS. IN THE THIRD CELL, OUR METHOD IS EVALUATED STRICTLY
FOLLOWING ARCFACE [7].

| Method | LFW | CALFW | CPLFW |
|---------------------------|--------------|--------------|--------------|
| HUMAN-Individual | 97.27 | 82.32 | 81.21 |
| HUMAN-Fusion | 99.85 | 86.50 | 85.24 |
| Center Loss [3] | 98.75 | 85.48 | 77.48 |
| SphereFace [5] | 99.27 | 90.30 | 81.40 |
| VGGFace2 [14] | 99.43 | 90.57 | 84.00 |
| MS1MV2, R100, ArcFace [7] | 99.82 | 95.45 | 92.08 |
| MS1MV2, R100, SFace | 99.82 | 96.07 | 93.28 |

TABLE VII
FACE IDENTIFICATION AND VERIFICATION EVALUATION ON MEGAFACE
CHALLENGE 1 [20] USING FACESCRUB [41] AS THE PROBE SET. "ACC."
REFERS TO THE RANK-1 FACE IDENTIFICATION ACCURACY WITH 1M
DISTRACTORS, AND "VER." REFERS TO THE FACE VERIFICATION
TAR@FAR=1E-6. "R" REFERS TO DATA REFINEMENT ON BOTH PROBE
SET AND 1M DISTRACTORS FOLLOWING [7]. IN THE SECOND AND THIRD
CELL, METHODS ARE COMPARED IN THE SAME SETTING WITH
RESNET100 MODELS TRAINED ON MS1MV2 DATABASE [12].

| Method | Protocol | Acc. | Ver. |
|------------------------------|----------|--------------|--------------|
| FaceNet [2] | Large | 70.49 | 86.47 |
| CosFace [6] | Large | 82.72 | 96.65 |
| AdaptiveFace [26], R | Large | 95.023 | 95.608 |
| P2SGrad [9], R | Large | 97.25 | - |
| AdaCos [8], R | Large | 97.41 | - |
| MS1MV2, R100, CosFace [6] | Large | 80.56 | 96.56 |
| MS1MV2, R100, ArcFace [7] | Large | 81.03 | 96.98 |
| MS1MV2, R100, SFace | Large | 81.15 | 97.11 |
| MS1MV2, R100, CosFace [6], R | Large | 97.91 | 97.91 |
| MS1MV2, R100, ArcFace [7], R | Large | 98.35 | 98.48 |
| MS1MV2, R100, SFace, R | Large | 98.50 | 98.61 |

ArcFace and WebFace-Noisy databases are shown in Figure 6. Note that the 10% noise in WebFace-ArcFace database is from the label noise that derive from the collection process of the CASIA-WebFace database. While the 20% label noise in WebFace-Noisy contains 10% noise in WebFace-ArcFace and other 10% synthetic outliers.

We train ResNet34 models on WebFace-Clean, WebFace-ArcFace and WebFace-Noisy databases supervised by softmax, NSoftmax, CosFace, ArcFace and SFace. The experimental results are shown in Figure 7, which demonstrates the ro-

TABLE VIII

FACE IDENTIFICATION AND VERIFICATION EVALUATION OF DIFFERENT METHODS ON THE IJB-A [21] DATABASE. IN THE FIRST CELL, EXPERIMENTAL RESULTS ARE READ FROM ORIGINAL PAPERS. FOR COMPARISON, WE IMPLEMENT EXPERIMENTAL RESULTS IN THE SECOND CELL USING ARCFACE AND SFACE TRAINED ON VGGFACE2 AND MS-CELEB-1M DATABASES, RESPECTIVELY.

| Method | 1:1 | | | 1:N | | | | |
|----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | FAR=1e-3 | FAR=1e-2 | FAR=1e-1 | FPIR=0.01 | FPIR=0.1 | Rank-1 | Rank-5 | Rank-10 |
| VGGFace [48] | 62.00 | 83.40 | 95.40 | 45.40 | 74.80 | 92.50 | 97.20 | 98.30 |
| Template Adaption [51] | 83.60 | 93.90 | 97.90 | 77.40 | 88.20 | 92.80 | 97.70 | 98.60 |
| NAN [52] | 88.10 | 94.10 | 97.80 | 81.70 | 91.70 | 95.80 | 98.00 | 98.60 |
| VGGFace2 [14] | 92.10 | 96.80 | 99.00 | 88.30 | 94.60 | 98.20 | 99.30 | 99.40 |
| FTL [25] | 91.20 | 95.30 | - | - | - | 96.00 | 98.30 | 98.70 |
| UniformFace [53] | 92.30 | 96.90 | - | - | - | 97.90 | 98.80 | - |
| L2-Face [31] | 94.30 | 97.00 | 98.40 | 91.50 | 95.60 | 97.30 | - | 98.80 |
| Crystal Loss [54] | 94.90 | 96.90 | 98.40 | 91.80 | 95.90 | 97.20 | - | 98.80 |
| VGGFace2, R50, ArcFace [7] | 96.24 | 98.64 | 99.51 | 92.07 | 97.80 | 99.19 | 99.67 | 99.73 |
| VGGFace2, R50, SFace | 96.85 | 98.74 | 99.67 | 92.51 | 98.19 | 99.19 | 99.68 | 99.80 |
| MS1MV2, R100, ArcFace [7] | 97.60 | 98.75 | 99.53 | 93.47 | 98.11 | 98.83 | 99.33 | 99.51 |
| MS1MV2, R100, SFace | 98.02 | 98.93 | 99.51 | 94.84 | 98.50 | 98.93 | 99.44 | 99.55 |

TABLE IX

FACE IDENTIFICATION AND VERIFICATION EVALUATION OF DIFFERENT METHODS ON THE IJB-C DATABASE [22]. EXPERIMENTAL RESULTS IN THE FIRST CELL ARE READ FROM ORIGINAL PAPERS, AND ALL THE MODELS ARE TRAINED ON VGGFACE2 DATABASE [14] OR MS-CELEB-1M DATABASE [12].

FOR COMPARISON, WE IMPLEMENT EXPERIMENTAL RESULTS IN THE SECOND CELL USING ARCFACE AND SFACE TRAINED ON VGGFACE2 AND MS-CELEB-1M DATABASES, RESPECTIVELY.

| Method | 1:1 | | | | | 1:N | | | | |
|-------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | FAR=1e-5 | FAR=1e-4 | FAR=1e-3 | FAR=1e-2 | FAR=1e-1 | FPIR=0.01 | FPIR=0.1 | Rank-1 | Rank-5 | Rank-10 |
| VGGFace2, ResNet50 [14] | 73.40 | 82.50 | 90.00 | 95.00 | 98.00 | 73.50 | 83.00 | 89.80 | 93.90 | 95.30 |
| VGGFace2, SENet50 [14] | 74.70 | 84.00 | 91.00 | 96.00 | 98.70 | 74.60 | 84.20 | 91.20 | 94.90 | 96.20 |
| VGGFace2, MN-v [55] | 75.50 | 85.20 | 92.00 | 96.50 | 98.80 | - | - | - | - | - |
| VGGFace2, MN-vc [55] | 77.10 | 86.20 | 92.70 | 96.80 | 98.90 | - | - | - | - | - |
| VGGFace2, ResNet50+DCN(Kpts) [56] | - | 86.70 | 94.00 | 97.90 | 99.70 | - | - | - | - | - |
| VGGFace2, ResNet50+DCN(Divs) [56] | - | 88.00 | 94.40 | 98.10 | 99.80 | - | - | - | - | - |
| VGGFace2, SENet50+DCN(Kpts) [56] | - | 87.40 | 94.40 | 98.10 | 99.80 | - | - | - | - | - |
| VGGFace2, SENet50+DCN(Divs) [56] | - | 88.50 | 94.70 | 98.30 | 99.80 | - | - | - | - | - |
| MS1M, Inception-ResNet, P2SGrad [9] | 87.84 | 92.25 | 95.58 | 97.79 | 99.03 | - | - | - | - | - |
| MS1M, Inception-ResNet, AdaCos [8] | 88.03 | 92.40 | 95.65 | 97.72 | 99.06 | - | - | - | - | - |
| VGGFace2, R50, ArcFace [7] | 86.03 | 92.12 | 95.93 | 98.23 | 99.34 | 79.50 | 89.53 | 94.75 | 96.94 | 97.64 |
| VGGFace2, R50, SFace | 87.08 | 93.12 | 96.50 | 98.34 | 99.25 | 82.84 | 90.69 | 95.01 | 96.97 | 97.55 |
| MS1MV2, R100, ArcFace [7] | 93.15 | 95.65 | 97.20 | 98.18 | 99.01 | 90.32 | 94.52 | 95.72 | 97.10 | 97.47 |
| MS1MV2, R100, SFace | 94.21 | 96.11 | 97.50 | 98.33 | 99.00 | 92.41 | 95.17 | 96.21 | 97.41 | 97.76 |

bustness of SFace to low level label noise. We also list the choice of hyper-parameters of SFace in Table IV. We can conclude that parameter a should be larger, *i.e.* $v_{intra}(\theta_{y_i})$ curves should move to the right, as the noise level increases, which indicates that the speed of intra-class is decreased more early to prevent overfitting. Although inter-class parameters b are also important for training, we find the optimal groups of them are the same for the three training databases, the reason may be that noisy data is relatively balanced across all identities. Another interesting phenomenon is that the model have similar performance on WebFace-Clean and WebFace-ArcFace. This result indicates that the manual cleaned data by human annotations actually has limited influence on these face models.

C. Evaluation Results on Several Benchmarks

We first evaluate our method on LFW [16] and YTF [17]. For fair comparison, we train models using ResNet100 on MS1MV2 database [12], strictly following the settings in [7]. MS1MV2 database is a refined version of MS-Celeb-1M database [12], cleaned by insightface [7]. MS1MV2 database

contains 5.8M images of 85,742 celebrities. We use this semi-artificial cleaned face database as a large-scale training database to further evaluate our method. For SFace trained on MS1MV2 database, the hyper-parameters a , b are set as 0.90 and 1.20. The experimental results on LFW and YTF are shown in Table V. SFace model trained on MS1MV2 database with ResNet100 obtains comparable results as the baseline method such as CosFace [6] and ArcFace [7]. We report the performance on CALFW [18] and CPLFW [19] databases in Table VI. As shown in Table VI, SFace outperforms both human performance and the advanced deep face models on CALFW and CPLFW databases by a significant margin.

Then, we evaluate our method on MegaFace database [20] including both the original MegaFace database and the refined version [7]. We report the rank-1 face identification accuracy with 1M distractors, and the face verification TAR@FAR=1e-6, shown in Table VII. In the second and third cell, methods are compared in the same setting with ResNet100 models trained on MS1MV2 database. As reported in Table VII, our method shows superiority over CosFace and ArcFace on both identification and verification settings on MegaFace challenge.

Finally, we evaluate our method on IJB-A [21] and IJB-C [22] databases on both identification and verification settings. Our method is compared with ArcFace using the same databases and models, other results are cited from the original papers. For fair comparison, we also train ResNet50 models on VGGFace2 database [14] following [7]. VGGFace2 training database has 3.13 million images of 8,631 identities, and has large variations in pose, age, illumination, ethnicity and profession. For SFace model trained on VGGFace2 database, the hyper-parameters a , b are set as 0.88 and 1.25. The results on IJB-A database are exhibited in Table VIII and Figure 8. The results on IJB-C database are shown in Table IX and Figure 9. For verification, we report TAR@FAR (ROC curves, higher is better). For identification, the performance is reported using TPIR@FPIR (DET curve, lower is better) and Rank-N accuracy (CMC curve, higher is better). Compared with ArcFace models trained on both VGGFace2 and MS1MV2 databases, our method performs better in both identification and verification settings, especially the TAR at very low FAR, which demonstrates the effectiveness and superiority of SFace.

V. CONCLUSION

In this paper, different from previous works which minimize the intra-class distances and maximize the inter-class distance, we introduce a new idea which aims to optimize intra-class and inter-class distance to some extent for the purpose of mitigating overfitting problems to the imperfect training databases. To carry out this idea, we propose a new loss function SFace to improve the performance of models in the robust unconstrained face recognition. SFace imposes intra-class and inter-class constraints on a hypersphere manifold with precisely controlled intra-class and inter-class gradients so that intra-class and inter-class distances are optimized to some extent. To promote further understanding of SFace, we explain the relationship to softmax based loss functions, and show that, compared with softmax based loss, the advantage of SFace is the precisely control ability of both intra-class and inter-class optimization. The proposed SFace makes a better balance between underfitting and overfitting, and further improves the generalization ability of deep face models. Experiments on several benchmarks including LFW, YTF, CALFW, CPLFW, MegaFace, IJB-A and IJB-C databases, have demonstrated the effectiveness and superiority of our method.

ACKNOWLEDGMENT

This work was supported by Canon Information Technology (Beijing) Co., Ltd. under Grant No. OLA19023, and supported by BUPT Excellent Ph.D. Students Foundation CX2020201.

REFERENCES

- [1] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, pp. 1988–1996.
- [2] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
- [3] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.
- [4] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1041–1049.
- [5] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.
- [6] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.
- [7] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.
- [8] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "Adacos: Adaptively scaling cosine logits for effectively learning deep face representations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10823–10832.
- [9] X. Zhang, R. Zhao, J. Yan, M. Gao, Y. Qiao, X. Wang, and H. Li, "P2sgd: Refined gradients for optimizing deep face models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9906–9914.
- [10] E. Zhou, Z. Cao, and Q. Yin, "Naive-deep face recognition: Touching the limit of lfw benchmark or not?" *arXiv preprint arXiv:1501.04690*, 2015.
- [11] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.
- [12] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European conference on computer vision*. Springer, 2016, pp. 87–102.
- [13] A. Nech and I. Kemelmacher-Shlizerman, "Level playing field for million scale face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7044–7053.
- [14] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 67–74.
- [15] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. Change Loy, "The devil of face recognition is in the noise," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 765–780.
- [16] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.
- [17] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR 2011*. IEEE, 2011, pp. 529–534.
- [18] T. Zheng, W. Deng, and J. Hu, "Cross-age LFW: A database for studying cross-age face recognition in unconstrained environments," *arXiv:1708.08197*, 2017.
- [19] T. Zheng and W. Deng, "Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments," Beijing University of Posts and Telecommunications, Tech. Rep. 18-01, February 2018.
- [20] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4873–4882.
- [21] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1931–1939.
- [22] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 158–165.
- [23] W. Deng, J. Hu, N. Zhang, B. Chen, and J. Guo, "Fine-grained face verification: Fglfw database, baselines, and human-dcmn partnership," *Pattern Recognition*, vol. 66, pp. 63–73, 2017.
- [24] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, "Range loss for deep face recognition with long-tailed training data," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5409–5418.

- [25] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, "Feature transfer learning for face recognition with under-represented data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5704–5713.
- [26] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, "Adaptiveface: Adaptive margin and sampling for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11947–11956.
- [27] Y. Zhong, W. Deng, M. Wang, J. Hu, J. Peng, X. Tao, and Y. Huang, "Unequal-training for deep face recognition with long-tailed noisy data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7812–7821.
- [28] C. Lu and X. Tang, "Surpassing human-level face verification performance on lfw with gaussian face," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 3811–3819.
- [29] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in neural information processing systems*, 2016, pp. 1857–1865.
- [30] J. Deng, Y. Zhou, and S. Zafeiriou, "Marginal loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 60–68.
- [31] R. C. Rajevee Ranjan, Carlos D. Castillo, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.
- [32] Y. Zheng, D. K. Pal, and M. Savvides, "Ring loss: Convex feature normalization for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5089–5097.
- [33] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *ICML*, vol. 2, no. 3, 2016, p. 7.
- [34] X. Wang, S. Zhang, S. Wang, T. Fu, H. Shi, and T. Mei, "Misclassified vector guided softmax loss for face recognition," *arXiv preprint arXiv:1912.00833*, 2019.
- [35] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang, "Curricularface: adaptive curriculum learning loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5901–5910.
- [36] L. He, Z. Wang, Y. Li, and S. Wang, "Softmax dissection: Towards understanding intra-and inter-clas objective for embedding learning," *arXiv preprint arXiv:1908.01281*, 2019.
- [37] W. Hu, Y. Huang, F. Zhang, and R. Li, "Noise-tolerant paradigm for training face recognition cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11887–11896.
- [38] X. Wang, S. Wang, J. Wang, H. Shi, and T. Mei, "Co-mining: Deep face recognition with noisy labels," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 9358–9367.
- [39] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.
- [40] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "Agedb: the first manually collected, in-the-wild age database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 51–59.
- [41] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *2014 IEEE international conference on image processing (ICIP)*. IEEE, 2014, pp. 343–347.
- [42] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv:1512.01274*, 2015.
- [43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.
- [45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [46] "Website of CASIA-WebFace Cleaned List," <https://github.com/happynear/FaceVerification>.
- [47] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.
- [48] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015.
- [49] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.
- [50] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song, "Learning towards minimum hyperspherical energy," in *Advances in neural information processing systems*, 2018, pp. 6222–6233.
- [51] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman, "Template adaptation for face verification and identification," *Image and Vision Computing*, vol. 79, pp. 35–48, 2018.
- [52] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua, "Neural aggregation network for video face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4362–4371.
- [53] Y. Duan, J. Lu, and J. Zhou, "Uniformface: Learning deep equidistributed representation for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3415–3424.
- [54] R. Ranjan, A. Bansal, H. Xu, S. Sankaranarayanan, J.-C. Chen, C. D. Castillo, and R. Chellappa, "Crystal loss and quality pooling for unconstrained face verification and recognition," *arXiv preprint arXiv:1804.01159*, 2018.
- [55] W. Xie and A. Zisserman, "Multicolumn networks for face recognition," in *British Machine Vision Conference*, 2018.
- [56] W. Xie, L. Shen, and A. Zisserman, "Comparator networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 782–797.