CurricularFace: Adaptive Curriculum Learning Loss for Deep Face Recognition

Yuge Huang[†] Yuhan Wang[§] Ying Tai[†*] Xiaoming Liu[‡]
Pengcheng Shen[†] Shaoxin Li[†*] Jilin Li[†] Feiyue Huang[†]
[†]Youtu Lab, Tencent [§]Zhejiang University [‡]Michigan State University
[†]{yugehuang, yingtai, quantshen, darwinli, jerolinli, garyhuang}@tencent.com
[§]wang_yuhan@zju.edu.cn, [‡]liuxm@cse.msu.edu
https://github.com/HuangYG123/CurricularFace

Abstract

As an emerging topic in face recognition, designing margin-based loss functions can increase the feature margin between different classes for enhanced discriminability. More recently, the idea of mining-based strategies is adopted to emphasize the misclassified samples, achieving promising results. However, during the entire training process, the prior methods either do not explicitly emphasize the sample based on its importance that renders the hard samples not fully exploited; or explicitly emphasize the effects of semi-hard/hard samples even at the early training stage that may lead to convergence issue. In this work, we propose a novel Adaptive Curriculum Learning loss (CurricularFace) that embeds the idea of curriculum learning into the loss function to achieve a novel training strategy for deep face recognition, which mainly addresses easy samples in the early training stage and hard ones in the later stage. Specifically, our CurricularFace adaptively adjusts the relative importance of easy and hard samples during different training stages. In each stage, different samples are assigned with different importance according to their corresponding difficultness. Extensive experimental results on popular benchmarks demonstrate the superiority of our CurricularFace over the state-of-the-art competitors.

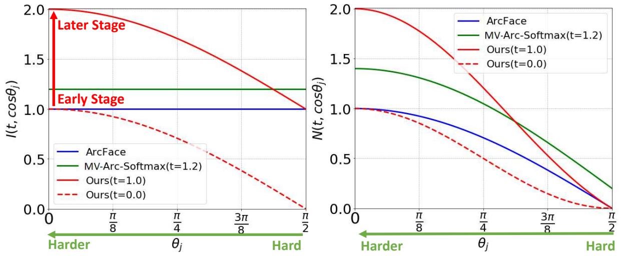


Figure 1. **Different training strategies** for modulating negative cosine similarities of hard samples (*i.e.*, the mis-classified samples) in ArcFace [8], MV-Arc-Softmax [31] and our CurricularFace. **Left**: The modulation coefficients $I(t, \cos\theta_j)$ for negative cosine similarities of hard samples in different methods, where t is an adaptively estimated parameter and θ_j denotes the angle between the hard sample and the non-ground truth j-class center. **Right**: The corresponding hard samples' negative cosine similarities $N(t, \cos\theta_j) = I(t, \cos\theta_j)\cos\theta_j + c$ after modulation, where c indicates a constant. On one hand, during early training stage (*e.g.*, t is close to 0), hard sample's negative cosine similarities are usually reduced, and thus leads to smaller hard sample loss than the original one. Therefore, easier samples are relatively emphasized; during later training stage (*e.g.*, t is close to 1), the hard sample's negative cosine similarities are enhanced, and thus leads to larger hard sample loss. On the other hand, in the same training stage, we modulate the hard samples' negative cosine similarities with $\cos\theta_j$. Specifically, **the smaller the angle θ_j is, the larger the modulation coefficient should be**.

1. Introduction

The success of Convolutional Neural Networks (CNNs) on face recognition can be mainly credited to: enormous training data, network architectures, and loss functions. Recently, designing effective loss functions that enhance discriminative power is pivotal for training deep face CNNs.

Current state-of-the-art (SOTA) face recognition methods mainly adopt softmax-based classification loss. Since the learned features with the original softmax is not sufficiently discriminative for the practical face recognition problem [14], which means that the testing identities are usually disjoint from the training set, several margin-based variants have been proposed to enhance features' discriminative power. For example, explicit margin, *i.e.*, CosFace [30], Sphereface [14], ArcFace [8], and implicit margin, *i.e.*, Adacos [38], supplement the original softmax function to enforce greater intra-class compactness and inter-class discrepancy, which result in more discriminate features. However, these margin-based loss functions *do not explicitly emphasize each sample according to its im-*

[*] denotes Ying Tai and Shaoxin Li are corresponding authors.

portance.

As demonstrated in [5, 10], hard sample mining is also a critical step to further improve the final accuracy. As a commonly-used hard sample mining method, OHEM [26] focuses on the large-loss samples in one mini-batch, in which the percentage of hard samples is empirically decided and easy samples are completely discarded. Focal loss [16] is a soft mining variant that rectifies the loss function to a elaborately designed form, in which two hyper-parameters should be tuned with a lot of efforts to decide the weights of each sample and hard samples are emphasized by reducing the weights of easy samples. Recently, Triplet loss [23] and MV-Arc-Softmax [31] are motivated by integrating both margin and mining into one framework. Triplet loss adopts a semi-hard mining strategy to obtain semi-hard triplets and enlarges the margin between triplet samples. MV-Arc-Softmax [31] clearly defines hard samples as mis-classified samples and emphasizes them by increasing the weights of their negative cosine similarities with a preset constant. In a nutshell, mining-based loss functions *explicitly emphasize the effects of semi-hard or hard samples [23]*.

However, there are drawbacks in training strategies of both margin- and mining-based loss functions. The general softmax-based loss function can be formulated as follows:

$$\mathcal{L} = -\log \frac{e^{sT(\cos\theta_{y_i})}}{e^{sT(\cos\theta_{y_i}) + \sum_{j=1, j\neq y_i}^{n} e^{sN(t,\cos\theta_j)}}}, \tag{1}$$

where $T(\cos\theta_{y_i})$ and $N(t,\cos\theta_j) = I(t,\cos\theta_j)\cos\theta_j + c$ are the functions to define the positive and negative cosine similarities, respectively. $I(t,\cos\theta_j)$ denotes the modulation coefficients of negative cosine similarities and c is a constant. For margin-based methods, mining strategy is ignored and thus the difficultness of each sample is not exploited, which may lead to convergence issues when using a large margin on small backbones, *e.g.*, MobileFaceNet [6]. As shown in Fig. 1, the modulation coefficients $I(\cdot)$ for the negative cosine similarities are fixed as a constant of 1 in ArcFace for all samples during the entire training process. For mining-based methods, over-emphasizing hard samples in early training stage may hinder the model to converge. MV-Arc-Softmax emphasizes hard samples by modulating the negative cosine similarity as $N(t,\cos\theta_j) = t\cos\theta_j + t - 1$, *i.e.*, $I(t,\cos\theta_j) = t$, where t is a manually defined constant. As MV-Arc-Softmax claimed, t plays a key role in the model convergence property and a slight larger value (*e.g.*, >1.4) may cause the model difficult to converge. Thus t needs to be carefully tuned.

In this work, we propose a novel adaptive curriculum learning loss, termed CurricularFace, to achieve a novel training strategy for deep face recognition. Motivated by the nature of human learning that easy cases are learned first and then come the hard ones [2], our CurricularFace incorporates the idea of Curriculum Learning (CL) into face recognition in an *adaptive* manner, which differs from the traditional CL in two aspects. First, **the curriculum construction is adaptive**. In traditional CL, the samples are *ordered* by the corresponding difficultness, which are often defined by a prior and then fixed to establish the curriculum. In CurricularFace, the samples are *randomly* selected in each mini-batch, while the curriculum is established adaptively via mining the hard samples online, which shows the *diversity* in samples with different importance. Second, **the importance of hard samples are adaptive**. On one hand, the relative importance between easy and hard samples is dynamic and could be adjusted in different training stages. On the other hand, the importance of each hard sample in current mini-batch depends on its own difficultness.

Specifically, the mis-classified samples in mini-batch are chosen as hard samples and weighted by adjusting the modulation coefficients $I(t, cos\theta_j)$ of cosine similarities between the sample and the non-ground truth class center vectors, *i.e.*, negative cosine similarity $cos\theta_j$. To achieve the goal of adaptive curricular learning in the entire training, we design a novel coefficient function $I(\cdot)$ that is determined by two factors: 1) the adaptively estimated parameter t that utilizes moving average of positive cosine similarities between samples and the corresponding ground-truth class center to unleash the burden of manually tuning; and 2) the angle θ_j that defines the difficultness of hard samples to achieve adaptive assignment. To sum up, the contributions of this work are:

- We propose an adaptive curriculum learning loss for face recognition, which automatically emphasizes easy samples first and hard samples later. To the best of our knowledge, it is the first work to introduce the idea of adaptive curriculum learning for face recognition.

- We design a novel modulation coefficient function $I(\cdot)$ to achieve adaptive curriculum learning during training, which connects positive and negative cosine similarity simultaneously without the need of manually tuning any additional hyper-parameter.

- We conduct extensive experiments on popular facial benchmarks, which demonstrate the superiority of our CurricularFace over the SOTA competitors.

2. Related Work

Margin-based loss function. Loss design is pivotal for large-scale face recognition. Current SOTA deep face recognition methods mostly adopt softmax-based classification loss [28]. Since the learned features with the original softmax loss are not guaranteed to be discriminative enough for practical face recognition problem [14], margin-based losses [18, 14, 8] are proposed. Though the margin-based loss functions are verified to obtain good performance, they do not take the difficultness of each sample into consideration, while our CurricularFace emphasizes easy samples first and hard samples later, which is more reasonable and effective.

Mining-based loss function. Though some mining-based loss function such as Focal loss [16], Online Hard Sample Mining (OHEM) [26] are prevalent in the field of object detection, they are rarely used in face recognition. OHEM focuses on the large-loss samples in one mini-batch, in which the percentage of the hard samples is empirically determined and easy samples are completely discarded. Focal loss emphasizes hard samples by reducing the weights of easy samples, in which two hyper-parameters should be manually tuned. The recent work, MV-Arc-Softmax [31] fuses the motivations of both margin and mining into one framework for deep face recognition. They define hard samples as misclassified samples and enlarge the weights of hard samples with a preset constant. Our method differs from MV-Arc-Softmax in three aspects: 1) We do not always emphasize hard samples, especially in the early training stages. 2) We assign different weights for hard samples according to their corresponding difficultness. 3) We adaptively estimate the additional hyper-parameter t without manual tuning.

Curriculum Learning. Learning from easier samples first and harder samples later is a common strategy in Curriculum Learning (CL) [2, 42]. The key problem in CL is to define the difficultness of each sample. For example, [1] takes the negative distance to the boundary as the indicator for easiness in classification. However, the ad-hoc curriculum design in CL turns out to be difficult to implement in different problems. To alleviate this issue, [12] designs a new formulation, called Self-Paced Learning (SPL), where examples with lower losses are considered to be easier and emphasized during training. The key differences between our CurricularFace with SPL are: 1) Our method focuses on easier samples in the early training stage and emphasizes hard samples in the later stage. 2) Our method proposes a novel function $N(\cdot)$ for negative cosine similarities, which achieves not only adaptive assignment on modulation coefficients $I(\cdot)$ for different samples in the same training stage, but also adaptive curriculum learning strategy in different stages.

3. The Proposed CurricularFace

3.1. Preliminary Knowledge on Loss Function

The original softmax loss is formulated as follows:

$$\mathcal{L} = -\log \frac{e^{W_{y_i} x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j x_i + b_j}}, \tag{2}$$

where $x_i \in R^d$ denotes the deep feature of i-th sample which belongs to the y_i class, $W_j \in R^d$ denotes the j-th column of the weight $W \in R^{d \times n}$ and b_j is the bias term. The class number and the embedding feature size are n and d, respectively. In practice, the bias is usually set to $b_j = 0$ and the individual weight is set to $||W_j|| = 1$ by l_2 normalization. The deep feature is also normalized and re-scaled to s. Thus, the original softmax can be modified as follows:

$$\mathcal{L} = -\log \frac{e^{s(\cos\theta_{y_i})}}{e^{s(\cos\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{s(\cos\theta_j)}}. \tag{3}$$

Since the learned features with original softmax loss may not be discriminative enough for practical face recognition problem, several variants are proposed and can be formulated in a general form:

$$\mathcal{L} = -G(p(x_i)) \log \frac{e^{sT(\cos\theta_{y_i})}}{e^{sT(\cos\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{sN(t, \cos\theta_j)}}, \tag{4}$$

where $p(x_i) = \frac{e^{sT(\cos\theta_{y_i})}}{e^{sT(\cos\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{sN(t, \cos\theta_j)}}$ is the predicted ground truth probability and $G(p(x_i))$ is an indicator function. $T(\cos\theta_{y_i})$ and $N(t, \cos\theta_j) = I(t, \cos\theta_j)\cos\theta_j + c$ are the functions to modulate the positive and negative cosine similarities, respectively, where c is a constant, and $I(t, \cos\theta_j)$ denotes the modulation coefficients of negative cosine similarities. In margin-based loss function, *e.g.*, ArcFace, $G(p(x_i)) = 1$, $T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$, and $N(t, \cos\theta_j) = \cos\theta_j$. It only modifies the positive cosine similarity of each sample to enhance the feature discrimination. As shown in Fig. 1, the modulation coefficients $I(\cdot)$ of each sample's negative cosine similarities are fixed as 1. The recent work, MV-Arc-Softmax emphasizes hard samples by increasing $I(t, \cos\theta_j)$ for hard samples. That is, $G(p(x_i)) = 1$ and $N(t, \cos\theta_j)$ is formulated as follows:

$$N(t, cos_{\theta_j}) = \begin{cases} \cos\theta_j, & T(\cos\theta_{y_i}) - \cos\theta_j \geq 0 \\ t\cos\theta_j + t - 1, & T(\cos\theta_{y_i}) - \cos\theta_j < 0. \end{cases} \tag{5}$$

If a sample is defined to be easy, its negative cosine similarity is kept the same as the original one, $\cos\theta_j$; if as a hard sample, its negative cosine similarity becomes $t\cos\theta_j + t - 1$. That is, as shown in Fig. 1, $I(\cdot)$ is a constant and determined by a preset hyper-parameter t. Meanwhile, since t is always larger than 1, $t\cos\theta_j + t - 1 > \cos\theta_j$ always holds true, which means the model always focuses on hard samples, even in the early training stage. However, the parameter t is sensitive that a large pre-defined value (*e.g.*, > 1.4) may lead to convergence issue.

3.2. Adaptive Curricular Learning Loss

Next, we present the details of our proposed adaptive curriculum learning loss, which is the first attempt to intro-

Algorithm 1: CurricularFace

```
Input: The deep feature of $i$-th sample $x_i$ with its label $y_i$,
       last fully-connected layer parameters $W$, cosine
       similarity $\cos\theta_j$ of two vectors, embedding network
       parameters $\Theta$, learning rate $\lambda$, and margin $m$
iteration number $k \leftarrow 0$, parameter $t \leftarrow 0$, $m \leftarrow 0.5$;
while not converged do
    if $\cos(\theta_{y_i} + m) \geq \cos\theta_j$ then
        $N(t, \cos\theta_j) = \cos\theta_j$;
    else
        $N(t, \cos\theta_j) = (t^{(k)} + \cos\theta_j)\cos\theta_j$ ;
    end
    $T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$;
    Compute the loss $\mathcal{L}$ by Eq. 10;
    Compute the gradients of $x_i$ and $W_j$ by Eq. 8;
    Update the parameters $W$ and $\Theta$ by:
      $W^{(k+1)} = W^{(k)} - \lambda^{(k)} \frac{\partial L}{\partial W}$,
      $\Theta^{(k+1)} = \Theta^{(k)} - \lambda^{(k)} \frac{\partial L}{\partial x_i} \frac{\partial x_i}{\partial \Theta^{(k)}}$;
    $k \leftarrow k + 1$;
    Update the parameter $t$ by Eq. 9;
end
Output: $W$, $\Theta$.
```

duce adaptive curriculum learning into deep face recognition. The formulation of our loss function is also contained in the general form, where $G(p(x_i)) = 1$, positive and negative cosine similarity functions are defined as follows:

$$T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m), \tag{6}$$

$$N(t, \cos_{\theta_j}) = \begin{cases} \cos\theta_j, & T(\cos\theta_{y_i}) - \cos\theta_j \geq 0 \\ \cos\theta_j(t + \cos\theta_j), & T(\cos\theta_{y_i}) - \cos\theta_j < 0. \end{cases} \tag{7}$$

It should be noted that the positive cosine similarity can adopt any margin-based loss functions and here we adopt ArcFace as an example. As shown in Fig. 1, the modulation coefficient $I(t, \theta_j)$ of hard sample negative cosine similarity depends on both the value of t and θ_j. In the early training stage, learning from easy samples is beneficial to model convergence. Thus, t should be close to zero and $I(\cdot) = t + \cos\theta_j$ is smaller than 1. Therefore, the weights of hard samples are reduced and easy samples are emphasized relatively. As training goes on, the model gradually focuses on the hard samples, *i.e.*, the value of t shall increase and $I(\cdot)$ is larger than 1. Thus, the hard samples are emphasized with larger weights. Moreover, within the same training stage, $I(\cdot)$ is monotonically decreasing with θ_j so that harder sample can be assigned with larger coefficient according to its difficultness. The value of the parameter t is *automatically* estimated in our CurricularFace, otherwise it may require lots of efforts for manual tuning.

Optimization. Next, we show our CurricularFace can be easily optimized by the conventional stochastic gradient descent. Assuming x_i denotes the deep feature of i-th sample which belongs to the y_i class, the input of the proposed function is the logit f_j, where j denotes the j-th class.

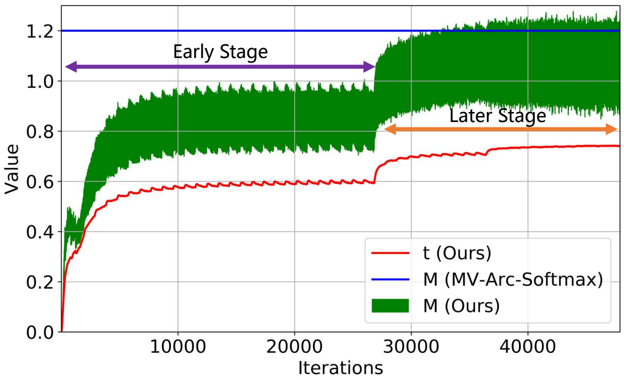


Figure 2. **The adaptive parameter t (red line) and gradient modulation coefficients M of ours (green area) and MV-Arc-Softmax (blue line) in training**. Since the number of mined hard samples reduces as training progresses, the green area, *i.e.*, the range of M values, is relatively smooth in early stage and exhibits burrs in later stage.

In the forwarding process, when $j = y_i$, it is the same as the ArcFace, *i.e.*, $f_j = sT(\cos\theta_{y_i})$, $T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$. When $j \neq y_i$, it has two cases, if x_i is an easy sample, it is the the same as the original softmax, *i.e.*, $f_j = s\cos\theta_j$. Otherwise, it will be modulated as $f_j = sN(t, \cos\theta_j)$, where $N(t, \cos\theta_j) = (t + \cos\theta_j)\cos\theta_j$. In the backward propagation process, the gradients w.r.t. x_i and W_j can also be divided into three cases and computed as follows:

$$\begin{aligned} \frac{\partial L}{\partial x_i} &= \begin{cases} \frac{\partial L}{\partial f_{y_i}}(s\frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}})W_{y_i}, & j = y_i \\ \frac{\partial L}{\partial f_j} sW_j, & j \neq y_i, \text{easy} \\ \frac{\partial L}{\partial f_j} s(2\cos\theta_j + t)W_j & j \neq y_i, \text{hard} \end{cases} \\ \frac{\partial L}{\partial W_j} &= \begin{cases} \frac{\partial L}{\partial f_{y_i}}(s\frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}})x_i, & j = y_i \\ \frac{\partial L}{\partial f_j} sx_i, & j \neq y_i, \text{easy} \\ \frac{\partial L}{\partial f_j} s(2\cos\theta_j + t)x_i & j \neq y_i, \text{hard} \end{cases} \end{aligned} \tag{8}$$

Based on the above formulations, we can find the gradient modulation coefficients of hard samples are determined by $M(\cdot) = 2\cos\theta_j + t$, which consists of two parts, the negative cosine similarity $\cos\theta_j$ and the value of t. As shown in Fig. 2, on the one hand, the coefficients increase with the adaptive estimation of t (described in the next subsection) to emphasize hard samples. On the other hand, these coefficients are assigned with different importance according to their corresponding difficultness ($\cos\theta_j$). Therefore, the values of M in Fig. 2 are plotted as a range at each training iteration. However, the coefficients are fixed to be 1 and a constant t in ArcFace and MV-Arc-Softmax, respectively.

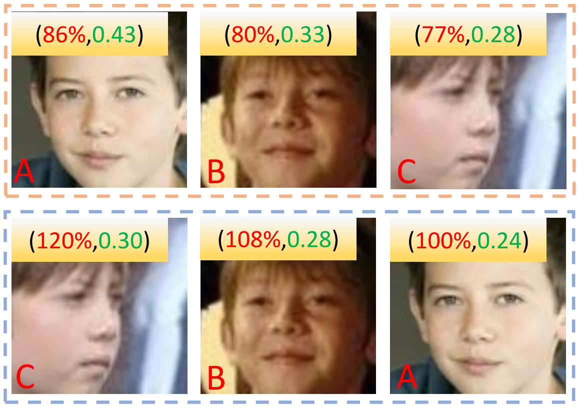


Figure 3. Illustrations on (**ratio between our loss and ArcFace**, **maximum** $cos\theta_j$) in different training stages. **Top**: Early training stage. **Bottom**: Later training stage.

Adaptive Estimation of t. It is critical to determine appropriate values of t in different training stages. Ideally the value of t can indicate the model training stages. We empirically find the *average* of positive cosine similarities is a good indicator. However, mini-batch statistic-based methods usually face an issue: when many extreme data are sampled in one mini-batch, the statistics can be vastly noisy and the estimation will be unstable. Exponential Moving Average (EMA) is a common solution to address this issue [13]. Specifically, let $r^{(k)}$ be the *average of the positive cosine similarities* of the k-th batch and be formulated as $r^{(k)} = \sum_i \cos\theta_{y_i}$, we have:

$$t^{(k)} = \alpha r^{(k)} + (1-\alpha)t^{(k-1)}, \tag{9}$$

where $t^0 = 0$, α is the momentum parameter and set to 0.99. With the EMA, we avoid the hyper-parameter tuning and make the modulation coefficients of hard sample negative cosine similarities $I(\cdot)$ adaptive to the current training stage. To sum up, the loss function of our CurricularFace is formulated as follows:

$$\mathcal{L} = -\log \frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{j=1,j\neq y_i}^{n} e^{sN(t^{(k)},\cos\theta_j)}}, \tag{10}$$

where $N(t^{(k)}, \cos\theta_j)$ is defined in Eq. 7. The entire training process is summarized in Algorithm 1.

Fig. 3 illustrates how the loss changes from ArcFace to our CurricularFace during training. Here are some observations: 1) As we excepted, hard samples (B and C) are suppressed in early training stage but emphasized later. 2) The ratio is monotonically increasing with $cos\theta_j$, since the larger $cos\theta_j$ is, the harder the sample is. 3) The positive cosine similarity of a perceptual-well image is often large. However, during the early training stage, the negative cosine similarities of the perceptual-well image (A) may also be large so that it could be classified as the hard one.

Table 1. **The decision boundaries of popular loss functions**.

Loss	Decision Boundary
Softmax	$\cos\theta_{y_i} = \cos\theta_j$
SphereFace	$\cos(m\theta_{y_i}) = \cos\theta_j$
CosFace	$\cos\theta_{y_i} - m = \cos\theta_j$
ArcFace	$\cos(\theta_{y_i} + m) = \cos\theta_j$
MV-Arc-Softmax	$\cos(\theta_{y_i} + m) = \cos\theta_j$ (easy) $\cos(\theta_{y_i} + m) = t\cos\theta_j + t - 1$ (hard)
CurricularFace (Ours)	$\cos(\theta_{y_i} + m) = \cos\theta_j$ (easy) $\cos(\theta_{y_i} + m) = (t + \cos\theta_j)\cos\theta_j$(hard)

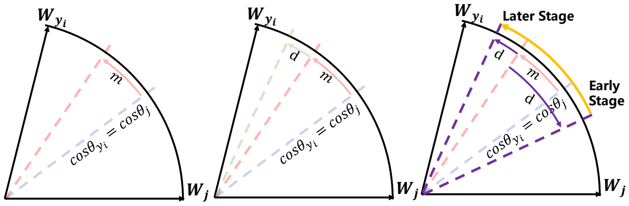

W_{y_i}
m
$cos\theta_{y_i} = cos\theta_j$
W_j
W_{y_i}
d
m
$cos\theta_{y_i} = cos\theta_j$
W_j
W_{y_i}
Later Stage
d
m
d
Early Stage
$cos\theta_{y_i} = cos\theta_j$
W_j

Figure 4. **Blue line**, **red line**, **green line** and **purple line** denote the decision boundary of Softmax, ArcFace, MV-Arc-Softmax, and ours, respectively. m denotes the angular margin added by ArcFace. d denotes the additional margin of MV-Arc-Softmax and ours. In MV-Arc-Softmax, $d = (t-1)\cos\theta_j + t - 1$. In ours, $d = (t + \cos\theta_j - 1)\cos\theta_j$.

3.3. Discussions with SOTA Loss Functions

Comparison with ArcFace and MV-Arc-Softmax. We first discuss the difference between our CurricularFace and the two competitors, ArcFace and MV-Arc-Softmax, from the perspective of the decision boundary in Tab. 1. ArcFace introduces a margin function $T(\cos\theta_{y_i}) = \cos(\theta_{y_i} + m)$ from the perspective of the positive cosine similarity. As shown in Fig. 4, its decision condition changes from $\cos\theta_{y_i} = \cos\theta_j$ (*i.e.*, blue line) to $\cos(\theta_{y_i} + m) = \cos\theta_j$ (red line) for each sample. MV-Arc-Softmax introduces additional margin from the perspective of negative cosine similarity for hard samples, and the decision boundary becomes $\cos(\theta_{y_i} + m) = t\cos\theta_j + t - 1$ (green line). Conversely, we adaptively adjust the weights of hard samples in different training stages. The decision condition becomes $\cos(\theta_{y_i} + m) = (t + \cos\theta_j)\cos\theta_j$ (purple line). During training, the decision boundary for hard samples changes from one purple line (**early stage**) to another (**later stage**), which emphasizes easy samples first and hard samples later.

Comparison with Focal Loss. Focal loss is formulated as: $G(p(x)) = \alpha(1 - p(x_i))^\beta$, where α and β are modulating factors to be tuned manually. The definition of hard samples in Focal loss is ambiguous, since it focuses on *relatively* hard samples by reducing the weight of easier samples during entire training process. In contrast, the definition of hard samples in our CurricularFace is more clear, *i.e.*, misclassified samples. Meanwhile, the weights of hard samples are *adaptively* determined in different training stages.

Table 2. **Verification performance (%) of different values of t.**

Methods (%)	LFW	CFP-FP
$t = 0$	99.32	95.90
$t = 0.3$	99.37	96.47
$t = 0.7$	99.42	96.66
$t = 1$	99.45	93.94
Adaptive t	**99.47**	**96.96**

Table 3. **Verification performance (%) of different strategies for setting t.**

Methods (%)	LFW	CFP-FP
Mode($\cos\theta_{y_i}$)	99.42	96.49
Mean(p_{x_i})	99.42	95.39
Mean($\cos\theta_{y_i}$)	**99.47**	**96.96**

4. Experiments

4.1. Implementation Details

Datasets. We separately employ CASIA-WebFace [36] and refined MS1MV2 [8] as our training data for fair comparisons with other methods. CASIA-WebFace contains about 0.5M of 10 individuals, and MS1MV2 contains about 5.8M images of 85K individuals. We extensively test our method on several popular benchmarks, including LFW [9], CFP-FP [24], CPLFW [41], AgeDB [20], CALFW [40], IJB-B [33], IJB-C [19], and MegaFace [11].

Training Setting. We follow [8] to crop the 112×112 faces with five landmarks [37, 27]. For the embedding network, we adopt ResNet50 and ResNet100 as in [8]. Our framework is implemented in Pytorch [21]. We train models on 4 NVIDIA Tesla P40 GPU with batch size 512. The models are trained with SGD algorithm, with momentum 0.9 and weight decay $5e-4$. On CASIA-WebFace, the learning rate starts from 0.1 and is divided by 10 at 28, 38, 46 epochs. The training process is finished at 50 epochs. On MS1MV2, we divide the learning rate at 10, 18, 22 epochs and finish at 24 epochs. We follow the common setting as [8] to set scale $s = 64$ and margin $m = 0.5$.

4.2. Ablation study

Effects on Fixed vs. Adaptive Parameter t. We first investigate the effect of adaptive estimation of t. We choose four fixed values between 0 and 1 for comparison. Specifically, 0 means the modulation coefficient $I(\cdot)$ of each hard sample's negative cosine similarity is always reduced based on its difficultness. In contrast, 1 means the hard samples are always emphasized. 0.3 and 0.7 are between the two cases. Tab. 2 shows that it is more effective to learn from easier samples first and hard samples later based on our adaptively estimated parameter t.

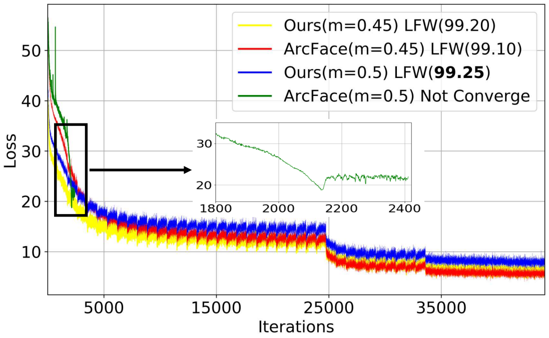


Figure 5. **Illustrations on loss curves** of our CurricularFace and ArcFace with the small backbone MobileFaceNet.

Effects on Different Statistics for Estimating t. We now investigate the effects of several other statistics, *i.e.*, mode of positive cosine similarities in a mini-batch, or mean of the predicted ground truth probability for estimating t in our loss. As Tab. 3 shows: 1) The mean of positive cosine similarities is better than mode. 2) The positive cosine similarity is more accurate than the predicted ground truth probability to indicate the training stages.

Robustness on Training Convergence. As claimed in [15], ArcFace exhibits the divergence issue when using small backbones like MobileFaceNet. As a result, softmax loss must be incorporated for pre-training. To illustrate the robustness of our loss function on convergence issue with small backbones, we use the MobileFaceNet as the network architecture and train it on CASIA-WebFace. As shown in Fig. 5, when the margin m is set to 0.5, the model trained with our loss achieves 99.25% accuracy on LFW, while the model trained with ArcFace does not converge and the loss is **NAN** at about 2,400-th step. When the margin m is set to 0.45, both losses can converge, but our loss achieves better performance (99.20% vs. 99.10%). Comparing the yellow and red curves, since the losses of hard samples are reduced in early training stages, our loss converges much faster in the beginning, leading to lower loss than ArcFace. Later on, the value of our loss is slightly larger than ArcFace, because we emphasize the hard samples in later stages. The results illustrate that learning from easy samples first and hard samples later is beneficial to model convergence.

4.3. Comparisons with SOTA Methods

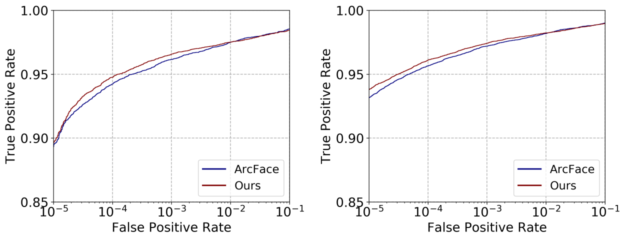
Results on LFW, CFP-FP, CPLFW, AgeDB and CALFW. Next, we train our CurricularFace on dataset MS1MV2 with ResNet100, and compare with the SOTA competitors on various benchmarks, including LFW for unconstrained face verification, CFP-FP and CPLFW for large pose variations, AgeDB and CALFW for age variations. As reported in Tab. 4, our CurricularFace achieves comparable result (*i.e.*, 99.80%) with the competitors on LFW where

Table 4. **Verification comparison with SOTA methods** on LFW, two pose benchmarks: CFP-FP and CPLFW, and two age benchmarks: AgeDB and CALFW. $*$ denotes our re-implemented results with the backbone ResNet100 [8].

Methods (%)	LFW	CFP-FP	CPLFW	AgeDB	CALFW
Center Loss (ECCV'16)	98.75	–	77.48	–	85.48
SphereFace (CVPR'17)	99.27	–	81.40	–	90.30
DRGAN (CVPR'17)	–	93.41	–	–	–
Peng *et al.* (ICCV'17)	–	93.76	–	–	–
VGGFace2 (FG'18)	99.43	–	84.00	–	90.57
Dream (CVPR'18)	–	93.98	–	–	–
Deng *et al.* (CVPR'18)	99.60	94.05	–	–	–
ArcFace (CVPR'19)	99.77	98.27	92.08	98.15	95.45
MV-Arc-Softmax (AAAI'20)	99.78	-	-	–	–
MV-Arc-Softmax$*$	**99.80**	98.28	92.83	97.95	96.10
CurricularFace (Ours)	**99.80**	**98.37**	**93.13**	**98.32**	**96.20**

Table 5. **1:1 verification TAR (@FAR=$1e{-}4$)** on the IJB-B and IJB-C datasets. $*$ denotes our re-implemented results with the backbone ResNet100 [8].

Methods (%)	IJB-B	IJB-C
ResNet50+SENet50 (FG'18)	80.0	84.1
Multicolumn (BMVC'18)	83.1	86.2
DCN (ECCV'18)	84.9	88.5
ArcFace-VGG2-R50 (CVPR'19)	89.8	92.1
ArcFace-MS1MV2-R100 (CVPR'19)	94.2	95.6
Adocos (CVPR'19)	–	92.4
P2SGrad (CVPR'19)	–	92.3
PFE (ICCV'19)	–	93.3
MV-Arc-Softmax$*$ (AAAI'20)	93.6	95.2
Ours-MS1MV2-R100	**94.8**	**96.1**



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

Figure 6. **ROC of 1:1 verification protocol** on IJB-B and IJB-C.

Table 6. **Verification comparison with SOTA methods** on MegaFace Challenge 1 using FaceScrub as the probe set. Id refers to the rank-1 face identification accuracy with 1M distractors, and Ver refers to the face verification TAR at $1e^{-6}$ FAR. The column R refers to data refinement on both probe set and 1M distractors. $*$ denotes our re-implemented results with the backbone ResNet100 [8].

Methods (%)	Protocol	R	Id	Ver
Triplet (CVPR'15)	Small		64.79	78.32
Center Loss (ECCV'16)	Small		65.49	80.14
SphereFace (CVPR'17)	Small		72.73	85.56
CosFace (CVRP'18)	Small		77.11	89.88
AM-Softmax (SPL'18)	Small		72.47	84.44
ArcFace-R50 (CVPR'19)	Small		77.50	92.34
ArcFace-R50	Small	✓	91.75	93.69
Ours-R50	Small		**77.65**	**92.91**
Ours-R50	Small	✓	**92.48**	**94.55**
CosFace-R100	Large		80.56	96.56
CosFace-R100	Large	✓	97.91	97.91
ArcFace-R100	Large		81.03	96.98
ArcFace-R100	Large	✓	98.35	98.48
PFE (ICCV'19)	Large		78.95	92.51
Adacos (CVPR'19)	Large	✓	97.41	–
P2SGrad (CVPR'19)	Large	✓	97.25	–
MV-Arc-Softmax (AAAI'20)	Large	✓	97.14	97.57
MV-Arc-Softmax*	Large		80.59	96.22
MV-Arc-Softmax*	Large	✓	97.76	97.80
Ours-R100	Large		**81.26**	**97.26**
Ours-R100	Large	✓	**98.71**	**98.64**
AdaptiveFace-R50 (CVPR19)	Large	✓	95.02	95.61
Ours-R50	Large	✓	**98.25**	**98.44**

the performance is near saturated. While for both CFP-FP and CPLFW, our method shows superiority over the baselines including general methods, *e.g.*, [32], [4], and cross-pose methods, *e.g.*, [29], [22], [3] and [7]. As a recent face recognition method, MV-Arc-Softmax achieves better performance than ArcFace, but still worse than Our CurricularFace. Finally, for AgeDB and CALFW, as Tab. 4 shows, our CurricularFace again achieves the best performance than all of the other SOTA methods.

Results on IJB-B and IJB-C. The IJB-B dataset contains $1,845$ subjects with 21.8K still images and 55K frames from $7,011$ videos. In the 1:1 verification, there are $10,270$ positive matches and 8M negative matches. The IJB-C dataset is a further extension of IJB-B, which contains about $3,500$ identities with a total of $31,334$ images and $117,542$ unconstrained video frames. In the 1:1 verification, there are $19,557$ positive matches and $15,638,932$ negative matches. On IJB-B and IJB-C datasets, we employ MS1MV2 and the ResNet100 for a fair comparison with recent methods. We follow the testing protocol in ArcFace and take the *average of the image features* as the corresponding template representation without bells and whistles. Note that our method is not proposed for set-based face recognition task, and DOES not adopt any specific strategies for set-based face recognition. The experiments on these two datasets are just to prove that our loss can obtain more discriminate features than the baselines like ArcFace, which are also generic methods for face recognition. Tab. 5 exhibits the performance of different methods, *e.g.*, Multicolumn [35], DCN [34], Adacos [38], P2SGrad [39], PFE [25] and MV-Arc-Softmax [31] on IJB-B and IJB-C 1:1 verification, our method again achieves the best performance. Fig. 6 shows the ROC curves of CurricularFace and ArcFace on IJB-B/C with the backbone ResNet100, our method achieves better performance.

Results on MegaFace. Finally, we evaluate the performance on the MegaFace Challenge. The gallery set of MegaFace includes 1M images of 690K subjects, and the probe set includes 100K photos of 530 unique individuals from FaceScrub. We report the two testing results under two protocols (large or small training set). Here, we use CASIA-WebFace and MS1MV2 under the small protocol and large protocol, respectively. In Tab. 6, our method

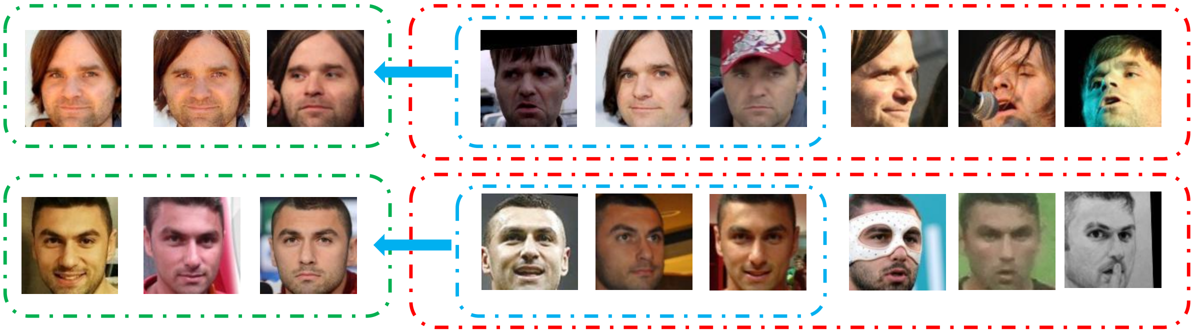


Figure 7. **Easy and hard examples from two subjects classified by our CurricularFace on early and later training stage, respectively.** Green box indicates easy samples. Red box indicates hard samples. Blue box means samples are classified as hard in early stage but relabeled as easy in later stage, which indicates samples' transformation from hard to easy during the training procedure.

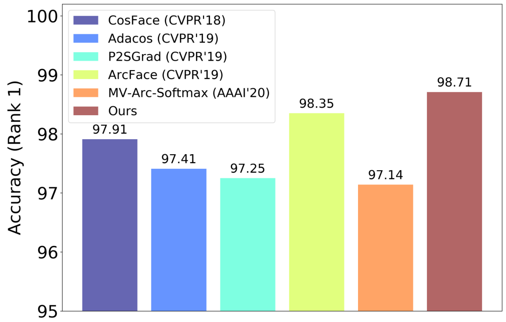
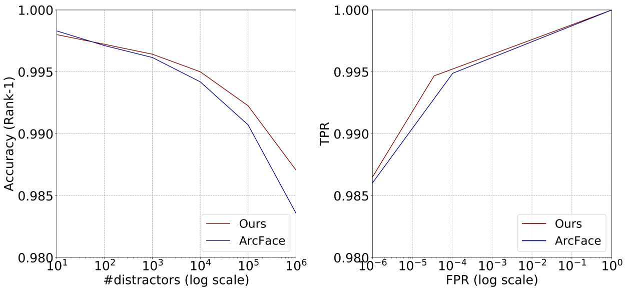


Figure 8. **The rank-**1 **face identification accuracy on MegaFace Challenge** 1 with both the 1M distractors and the probe set refined by ArcFace.



(a) TOP 1

(b) ROC

Figure 9. **Illustrations on Top** 1 **of different distractors and ROC on Megaface**. Results are evaluated on refined MegaFace dataset. The results of ArcFace are from the official ResNet100 pre-trained with MS1M.

achieves the best single-model identification and verification performance under both protocols, surpassing the recent strong competitors, *e.g.*, CosFace, ArcFace, Adacos, P2SGrad and PFE. We also report the results following the ArcFace testing protocol, which refines both the probe set and the gallery set. As shown in Fig. 8, our method still clearly outperforms the competitors and achieves the best performance on identification. Compared with ArcFace, our loss shows better performance under both identification and verification scenarios as shown in Fig. 9. Adapitve-Face [17] is another recent margin-based loss function for face recognition. We train our model with the same training data MS1MV2 and the same backbone ResNet50 [8] as AdaptiveFace for a fair comparison. The results in Tab. 6 demonstrate the superiority of our method.

Time Complexity. The proposed method only brings small burden on training complexity, but has the same cost as the backbone model during inference. Specifically, compared with the conventional margin-based loss functions, our loss only additionally adjusts the negative cosine similarity of hard samples. Under the same environment and batchsize, ArcFace [8] costs 0.370s for each iteration on NVIDIA P40 GPUs, while ours costs 0.378s.

Discussion on Easy and Hard Samples During Training. Finally, Fig. 7 shows the easy and hard samples classified by our method in different training stages. As we can see, the front and clear faces are usually considered as easy samples in early training stage, and our model mainly learns the identity information from these samples. With the model continues training, slightly harder samples (*i.e.*, Blue box) are gradually focused and corrected as the easy ones.

5. Conclusions

In this paper, we propose a novel Adaptive Curriculum Learning Loss that embeds the idea of adaptive curriculum learning into deep face recognition. Our key idea is to address easy samples in the early training stage and hard ones in the later stage. Our method is easy to implement and robust to converge. Extensive experiments on popular facial benchmarks demonstrate the effectiveness of our method compared to the SOTA competitors. Following the main idea of this work, future research can be expanded in various aspects, including designing a better function $N(\cdot)$ for negative cosine similarity that shares similar *adaptive* characteristic during training, and investigating the effects of *noise* samples that might be optimized as hard samples.

References

[1] Sumit Basu and Janara Christensen. Teaching classification boundaries to humans. In *AAAI*, 2013. 3

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 2, 3

[3] Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-robust face recognition via deep residual equivariant mapping. In *CVPR*, 2018. 7

[4] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018. 7

[5] Beidi Chen, Weiyang Liu, Animesh Garg, Zhiding Yu, Anshumali Shrivastava, and Anima Anandkumar. Angular visual hardness. In *ICML Workshop on Deep Phenomena*, 2019. 2

[6] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices. In *CCBR*, 2018. 2

[7] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *CVPR*, 2018. 7

[8] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 1, 2, 6, 7, 8

[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. 6

[10] Yuge Huang, Pengcheng Shen, Ying Tai, Shaoxin Li, Xiaoming Liu, Jilin Li, Feiyue Huang, and Rongrong Ji. Distribution distillation loss: Generic approach for improving face recognition from hard samples. *arXiv preprint arXiv:2002.03662*, 2020. 2

[11] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016. 6

[12] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *NIPS*, 2010. 3

[13] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *AAAI*, 2019. 5

[14] Weiyang Li, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017. 1, 2

[15] Xianyang Li. Airface: Lightweight and efficient model for face recognition. *arXiv:1907.12256*, 2019. 6

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2, 3

[17] Hao Liu, Xiangyu Zhu, Zhen Lei, and Stan Z Li. Adaptiveface: Adaptive margin and sampling for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11947–11956, 2019. 8

[18] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016. 2

[19] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *ICB*, 2018. 6

[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 *CVPR Workshops*, 2017. 6

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 6

[22] Xi Peng, Xiang Yu, Kihyuk Sohn, Dimitris Metaxas, and Manmohan Chandraker. Reconstruction-based disentanglement for poseinvariant face recognition. In *ICCV*, 2017. 7

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 2

[24] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D.W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016. 6

[25] Yichun Shi, Anil K Jain, and Nathan D Kalka. Probabilistic face embeddings. In *ICCV*, 2019. 7

[26] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 2, 3

[27] Ying Tai, Yicong Liang, Xiaoming Liu, Lei Duan, Jilin Li, Chengjie Wang, Feiyue Huang, and Yu Chen. Towards highly accurate and stable face alignment for high-resolution videos. In *AAAI*, 2019. 6

[28] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 2

[29] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *CVPR*, 2017. 7

[30] 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 *CVPR*, 2018. 1

[31] Xiaobo Wang, Shifeng Zhang, Shuo Wang, Tianyu Fu, Hailin Shi, and Tao Mei. Mis-classified vector guided softmax loss for face recognition. In *AAAI*, 2020. 1, 2, 3, 7

[32] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016. 7

[33] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *CVPR Workshops*, 2017. 6

[34] Weidi Xie, Li Shen, and Andrew Zisserman. Comparator networks. In *ECCV*, 2018. 7

[35] Weidi Xie and Andrew Zisserman. Multicolumn networks for face recognition. In *BMVC*, 2018. 7

[36] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch. *arXiv:1411.7923*, 2014. 6

[37] 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. 6

[38] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *CVPR*, 2019. 1, 7

[39] Xiao Zhang, Rui Zhao, Junjie Yan, Mengya Gao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. P2sgrad: Refined gradients for optimizing deep face models. In *CVPR*, 2019. 7

[40] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv:1708.08197*, 2017. 6

[41] Tianyue Zheng, Weihong Deng, and Jiani Hu. 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. 6

[42] Tianyi Zhou and Jeff Bilmes. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *ICLR*, 2018. 3