AnchorFace: Boosting TAR@FAR for Practical Face Recognition

Jiaheng Liu*[1], Haoyu Qin*[2], Yichao Wu [2], Ding Liang[2]

[1] State Key Lab of Software Development Environment, Beihang University
[2] SenseTime Group Limited
liujiaheng@buaa.edu.cn, qinhaoyu1@sensetime.com, wuyichao@sensetime.com, liangding@sensetime.com

Abstract

Within the field of face recognition (FR), it is widely accepted that the key objective is to optimize the entire feature space in the training process and acquire robust feature representations. However, most real-world FR systems tend to operate at a pre-defined False Accept Rate (FAR), and the corresponding True Accept Rate (TAR) represents the performance of the FR systems, which indicates that the optimization on the pre-defined FAR is more meaningful and important in the practical evaluation process. In this paper, we call the pre-defined FAR as Anchor FAR, and we argue that the existing FR loss functions cannot guarantee the optimal TAR under the Anchor FAR, which impedes further improvements of FR systems. To this end, we propose AnchorFace to bridge the aforementioned gap between the training and practical evaluation process for FR. Given the Anchor FAR, AnchorFace can boost the performance of FR systems by directly optimizing the non-differentiable FR evaluation metrics. Specifically, in AnchorFace, we first calculate the similarities of the positive and negative pairs based on both the features of the current batch and the stored features in the maintained online-updating set. Then, we generate the differentiable TAR loss and FAR loss using a soften strategy. Our AnchorFace can be readily integrated into most existing FR loss functions, and extensive experimental results on multiple benchmark datasets demonstrate the effectiveness of AnchorFace.

Introduction

Face recognition (FR) based on deep learning has been well investigated for many years (Sun et al. 2014; Sun, Wang, and Tang 2015). The mainstream of recent studies is to introduce new loss functions (Deng et al. 2019; Wang et al. 2018b; Huang et al. 2020) to maximize the inter-class discriminative ability and the intra-class compactness, which optimizes the classification accuracy for each identity in the training process, as shown in Fig. 1a. In other words, the current methods mainly focus on optimizing the entire feature space and generating robust and effective representation.

However, practically, current FR systems usually measure True Accept Rate (TAR) under a pre-defined False Accept Rate (FAR) as shown in Fig. 1b. Such measurement indicates that the similarity scores distribution of all positive and negative face image pairs under the realistic pre-defined FAR, rather than the entire feature distribution, actually determines the performance of FR systems. Specifically, as shown in Fig. 1c, from the ROC curves of three typical FR models, we observe that the TAR performance of MODEL1 outperforms all other models when FAR is greater than 1e-5, while the TAR performance of MODEL3 is the best under the FAR of 1e-6. In real-world scenarios, MODEL1 will be deployed when the FR systems fix the FAR as 1e-4, while MODEL3 would be an ideal option in the case of FAR as 1e-6. Furthermore, even though the TAR performance of MODEL2 does not rank the first in a long interval, MODEL2 preserves the best TAR performance when FAR is fairly small. Thus, optimization on the pre-defined specific FAR is essential to the training process of the FR model, which has not been well investigated before.

Motivated by the above analysis, in this work, we aim to investigate how to optimize the similarity scores distribution of positive and negative pairs under the pre-defined FAR for real-world FR systems. In other words, the objective is to boost the TAR performance under the pre-defined FAR. We call the pre-defined FAR as **Anchor FAR** and the whole optimization process under the Anchor FAR as **Anchor Optimization**. The biggest challenges of Anchor Optimization are how to construct the positive and negative pairs in the training process for the calculation of the evaluation metrics (i.e., TAR and FAR) and the non-differentiable property of these evaluation metrics. These two challenges make it difficult to optimize the TAR under the Anchor FAR directly.

In addition, (Liu et al. 2021b) proposed to search the loss functions automatically for different non-differentiable computer vision metrics. However, these methods rely on carefully designed search space and search strategy for different tasks, which are complex and time-consuming. Besides, the lack of Anchor Optimization in the training process can be considered as a gap between the training and evaluation for FR. There are also some works that analyze the gaps between the training and evaluation for FR. For example, for most softmax-based loss functions, sample-to-prototype similarities are optimized in the training process, while sample-to-sample similarities are used in practice (Deng et al. 2021). Furthermore, domain shift is another common gap (Sohn et al. 2017), where the FR models per-

*These authors contributed equally.

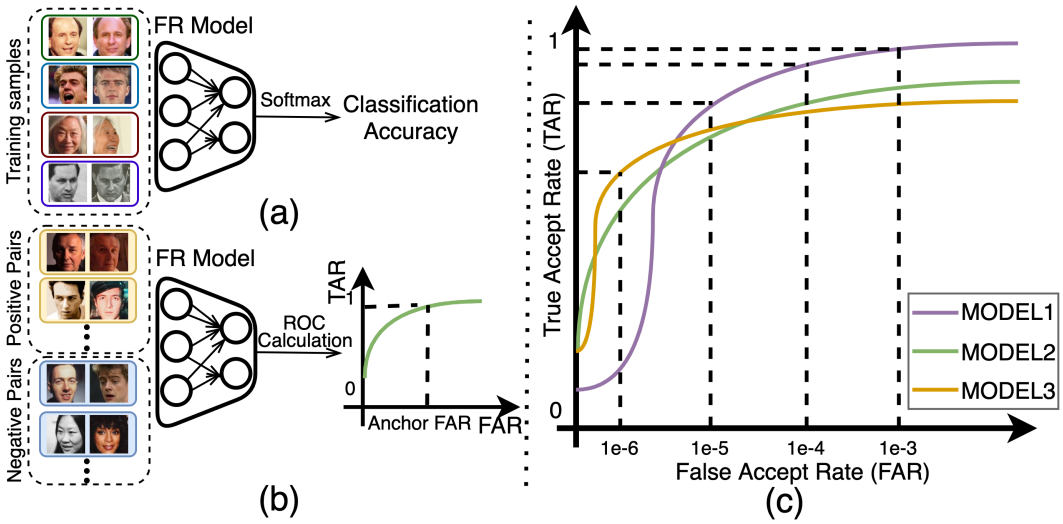

Figure 1: (a).The training process of FR. (b).The evaluation process of FR. (c).The ROC curves of different models.

form poorly on unseen ethnicity if the distribution of the training data is severely biased. When compared to these works, we mainly focus on Anchor Optimization for FR, which has not been discussed before.

In our AnchorFace, we introduce a pair of loss functions (i.e., TAR loss and FAR loss) as the supplement for the existing softmax-based loss functions in the training process, which aims to optimize TAR under the Anchor FAR directly (i.e., Anchor Optimization) on the training dataset. To address the aforementioned challenges of Anchor Optimization, we first construct an online-updating set to store the features of the samples for each identity in the training dataset, and update the features of the online-updating set in each iteration. Then, we construct the positive pairs and negative pairs using the features from the current batch and the online-updating set. Afterward, we calculate the similarity scores of the positive pairs and negative pairs and the Anchor Threshold under the Anchor FAR. Finally, based on the Anchor Threshold, we use a soften strategy to approximate the non-differentiable indicator function in the calculation process of the TAR and the FAR, and generate the differentiable TAR loss and FAR loss for FR.

The contributions of our work are summarized as follows:

- We first investigate the limitations of existing loss functions from a new perspective, and then propose a pair of loss functions (i.e., TAR loss and FAR loss) to directly optimize the evaluation metrics in the training process for practical face recognition, which is plug-and-play and can be easily integrated into most existing loss functions.
- In our AnchorFace, we introduce to construct the positive pairs and negative pairs by using the features from both the current batch and the maintained online-updating set, and utilize a soften strategy to produce the differentiable TAR loss and FAR loss.
- Extensive experiments on multiple benchmarks demonstrate the effectiveness of our proposed AnchorFace.

Related Works

Face Recognition. The success of deep face recognition (FR) can be mainly credited to the following three important reasons: effective neural networks (Taigman et al. 2014; Sun et al. 2014; Sun, Wang, and Tang 2015; Simonyan and Zisserman 2014; Szegedy et al. 2015; Sun et al. 2015), large-scale datasets (Guo et al. 2016; Yi et al. 2014; Kemelmacher-Shlizerman et al. 2016) and well-designed loss functions (Wang et al. 2017; Wen et al. 2016; Zhang et al. 2017; Liu et al. 2016, 2017; Sun et al. 2020; Wang et al. 2018a; Deng et al. 2019; Meng et al. 2021; Peng et al. 2019; Jin et al. 2019; Liu et al. 2021a; Li et al. 2020; Liu et al. 2022). First, with the development of neural network architecture, many effective networks (e.g., GoogleNet (Szegedy et al. 2015) and ResNet (He et al. 2016)) have obtained promising results for FR. Meanwhile, Neural Architecture Search was proposed to relieve the burden from the handcrafted network design process (Zoph and Le 2017; Liu et al. 2020). Second, many large-scale FR datasets (e.g., CASIA-WebFace (Yi et al. 2014), MS-Celeb-1M (Guo et al. 2016), WebFace260M (Zhu et al. 2021)) are proposed to improve the robustness and generalization ability for FR systems. Third, many well-designed loss functions are proposed to improve the generalization and discriminative ability of the learned feature representation for FR. For example, Triplet loss (Schroff, Kalenichenko, and Philbin 2015) aims to maximize the distances of negative pairs and minimize the distances of positive pairs, and Center loss (Wen et al. 2016) is proposed to reduce the intra-class variations by minimizing the distances within each class. Recently, many angular-margin based loss functions (Liu et al. 2016, 2017) are proposed by introducing the angular constraints into the cross-entropy loss function. To further increase the feature margin between different classes for enhanced discriminability, AM-softmax (Wang et al. 2018a), CosFace (Wang et al. 2018b), and ArcFace (Deng et al.

2019) introduce a margin item based on the aforementioned methods. Moreover, CurricularFace (Huang et al. 2020) and MV-Arc-Softmax (Wang et al. 2020) are used to introduce the mining-based strategies to emphasize the mis-classified samples. The recent work VPL (Deng et al. 2021) first analyzes the limitations of previous methods, which employ sample-to-prototype comparisons during training without considering sample-to-sample comparisons, and then introduces the sample-to-sample comparisons into the classification framework for FR. In contrast to existing works, our proposed AnchorFace discusses the necessity of the optimization under the Anchor FAR (i.e., Anchor Optimization) for practical FR from a new perspective, and introduces a pair of loss functions to and reduce the gap of the training and evaluation for FR.

Optimization on evaluation metrics. Owing to the non-differentiable property of most evaluation metrics, some loss functions have been proposed to simulate the evaluation metrics to improve the performance of different computer vision tasks (Berman, Triki, and Blaschko 2018; Eban et al. 2017; Brown et al. 2020; Zheng et al. 2020; Puthiya Parambath, Usunier, and Grandvalet 2014; Zheng et al. 2020). For example, the Lovasz-Softmax loss (Berman, Triki, and Blaschko 2018) for semantic segmentation and Distance-IoU loss (Zheng et al. 2020) for object detection. In contrast to previous works, our work aims at designing loss functions to directly optimize the evaluation metrics for face recognition, which has not been investigated before. Meanwhile, to remove the manual effort of designing these metric-approximating losses, (Liu et al. 2021b) proposed to search the loss functions automatically for object detection. However, these methods have the weaknesses that the elaborate design of search space and search strategy needs to be taken into account, which limits the application in real scenarios.

Method

The overall pipeline of our proposed AnchorFace is illustrated in Fig. 2. Specifically, for each iteration in the training process, we first extract the features of the current batch and update the online-updating set using the extracted features. Then, we construct the positive and negative pairs based on the features of the current batch and the stored features of the online-updating set. Afterward, we compute the cosine similarity scores of all pairs, and obtain the Anchor Threshold under the pre-defined Anchor FAR. Finally, we use the soften strategy to generate the TAR loss and FAR loss based on the Anchor Threshold.

Preliminary

True Accept Rate (TAR) and False Accept Rate (FAR) are the most commonly used evaluation metrics for practical FR systems, and we first describe the evaluation protocols of these two metrics.

Given N_n negative pairs, the FAR is computed as follows:

$$FAR = \frac{1}{N_n} \sum_{i=1}^{N_n} \mathbf{L}(s_n^i > t), \tag{1}$$

where t is the chosen similarity score threshold, s_n^i is the similarity score of the i-th negative pair, and $\mathbf{L}(x)$ is the indicator function, as shown in Fig. 3, which is not compatible with gradient based optimization.

Similarly, given N_p positive pairs, the TAR is defined as follows:

$$TAR = \frac{1}{N_p} \sum_{j=1}^{N_p} \mathbf{L}(s_p^j > t), \tag{2}$$

where s_p^j is the similarity score of the j-th positive pair.

In practice, we usually fix a pre-defined FAR (e.g., 1e-4), and the corresponding TAR represents the performance of the FR models. Specifically, the quantile of the similarity scores of all negative pairs determines the threshold of the specific FAR. When the threshold is obtained, we can calculate the TAR based on the similarities of all positive pairs. In our work, we call the pre-defined FAR as Anchor FAR and the corresponding threshold as Anchor Threshold.

AnchorFace

In this section, we first describe the necessity of sufficient positive and negative pairs, and introduce the construction scheme of the online-updating set. Then, we describe how to construct the positive pairs and negative pairs in the training process. Finally, we introduce the soften strategy to produce the TAR loss and FAR loss.

Necessity of sufficient positive and negative pairs. For Anchor Optimization, our proposed AnchorFace aims to directly optimize the TAR under the Anchor FAR (e.g., 1e-4). Therefore, to calculate the TAR and FAR in the training process, the construction of positive and negative pairs are needed. In addition, if the numbers of positive pairs and negative pairs are insufficient, the threshold estimation is not robust, which leads to an inaccurate TAR estimation and degrades the performance of our proposed AnchorFace. Thus, it is necessary to generate sufficient positive and negative pairs to ensure the effectiveness of our AnchorFace.

Construction scheme of the online-updating set. Inspired by MOCO (He et al. 2020) for unsupervised learning, which constructs a dynamic queue (i.e., memory bank) from the previous mini-batches to generate sufficient negative samples, we propose to maintain an online-updating set $\mathbf{S} \in \mathbb{R}^{N \times K \times d}$ in Fig. 2, where N is the number of identities of the training dataset, K is the maximum number of features for each identity, and d represents the dimension of the feature representation extracted by the neural network for each face image. In each iteration, we first update the online-updating set $\mathbf{S}$, and then utilize the stored features of $\mathbf{S}$ to construct the positive pairs and negative pairs with the features of the current batch.

Meanwhile, as discussed in VPL (Deng et al. 2021), features drift slowly for FR models, which indicates that features extracted previously can be considered as an approximation of the output of the current network within a certain number of training steps. Therefore, we also create a validness indicator $\mathbf{V} \in \mathbb{R}^{N \times K}$ to represent the validness of each feature in the online-updating set $\mathbf{S}$. Each item in

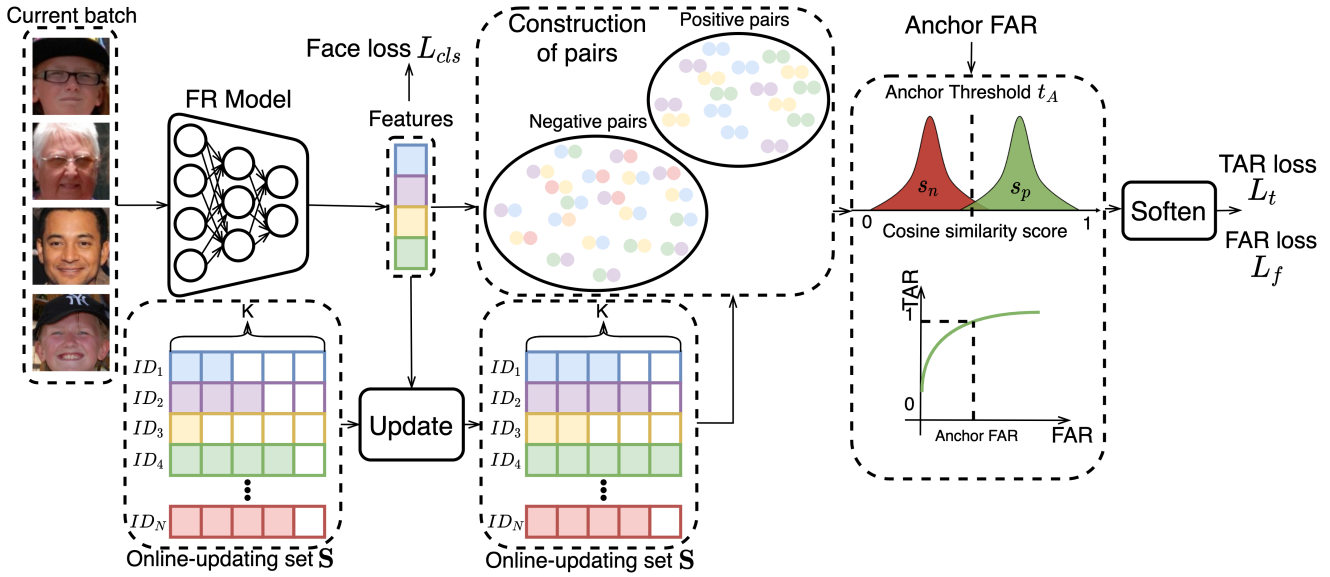

Figure 2: The overall framework of our proposed AnchorFace. N is the number of identities of the training dataset. K is the maximum number of features for each identity. s_n and s_p represent the similarity scores of the negative and positive pairs, respectively. t_A denotes the Anchor Threshold under the Anchor FAR.

$\mathbf{V}$ is a scalar value, which denotes the remaining valid steps for the corresponding feature in the online-updating set $\mathbf{S}$. The maximum number of valid steps for each feature is M, and we initialize all items of $\mathbf{V}$ as 0 at the beginning of the training process. In each iteration, in our AnchorFace, we first extract the features $\{f_i\}_{i=1}^m$ of the current batch, where m denotes the number of samples of this batch, and y_i is the corresponding label of the feature f_i. Then, we update the online-updating set $\mathbf{S}$ based on $\{f_i\}_{i=1}^m$. Specifically, for i-th feature f_i, when the number of the stored features for the corresponding identity y_i is smaller than K, we directly insert f_i into $\mathbf{S}$ based on the identity y_i. When the number of the stored features is equal to K for identity y_i, we first find out the index idx_i of the most oldest feature in $\mathbf{S}[y_i]$, which is also the index of the smallest value in $\mathbf{V}[y_i]$. Then, we replace the oldest feature with the newly extracted feature f_i based on the index idx_i, which means we set $\mathbf{S}[y_i][idx_i] = f_i$. After that, we set the number of valid step for f_i as M, which means we set $\mathbf{V}[y_i][idx_i] = M$. After each training step, $\mathbf{V}$ is updated by $\mathbf{V} = \mathbf{V} - 1$, which decreases the valid steps of all stored features in $\mathbf{S}$.

Construction of the positive and negative pairs. After the updating process for the online-updating set $\mathbf{S}$, the number of valid features in the online-updating set $\mathbf{S}$ is $\sum_{i=1}^{N}\sum_{j=1}^{K}\mathbf{L}(\mathbf{V}[i][j] > 0)$, where $\mathbf{L}(x)$ is the indicator function. We can easily construct the positive and negative pairs for TAR and FAR calculation. Specifically, for each feature f_i in the current batch, the positive pairs are constructed by using f_i and the stored valid features of the corresponding identity y_i in the online-updating set $\mathbf{S}$, while the negative pairs are constructed by using f_i and all other valid features with different identity labels in the online-updating set $\mathbf{S}$. Finally, we calculate the similarity scores for all positive and negative pairs, and obtain the TAR and the corresponding Anchor Threshold t_A under the Anchor FAR.

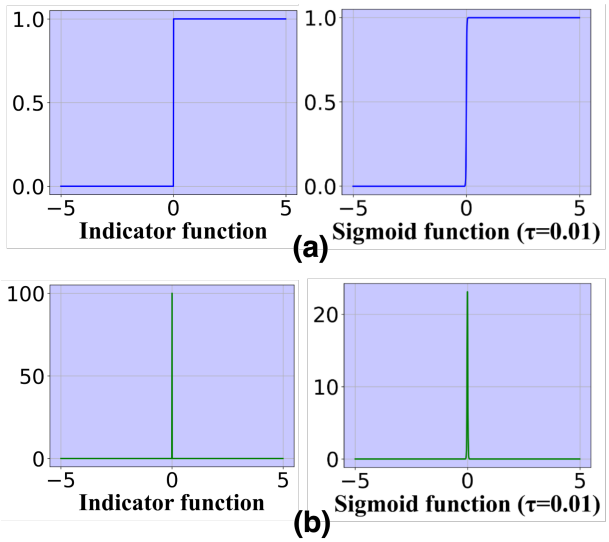

Figure 3: The illustrations of different functions in (a) and their derivations in (b).

TAR loss and FAR loss. As shown in Eq. 1 and Eq. 2, due to the non-differentiable property of the indicator function $\mathbf{L}(x)$, we cannot directly optimize the TAR under the Anchor FAR based on gradient methods.

To this end, in our work, we introduce a soften strategy by replacing the indicator function $\mathbf{1}(x)$ with the differentiable

sigmoid function, which is illustrated as follows:

$$\text{Sigmoid}(x;\tau) = \frac{1}{1+e^{-\frac{x}{\tau}}}, \tag{3}$$

where the temperature value τ is the hyper-parameter to control the approximation sharpness and the operating region with gradients. As shown in Fig. 3, we observe that the curves and derivatives of the sigmoid function and the indicator function are closer when the τ is smaller.

Thus, we can re-formulate Eq.1 and Eq.2 to generate the differentiable FAR loss L_f and TAR loss L_t as follows:

$$L_f = \frac{1}{N_n}\sum_{i=1}^{N_n}\frac{1}{1+e^{-\frac{s_n^i - t_A}{\tau}}}, \tag{4}$$

$$L_t = 1 - \frac{1}{N_p}\sum_{j=1}^{N_p}\frac{1}{1+e^{-\frac{s_p^j - t_A}{\tau}}}, \tag{5}$$

where t_A is the aforementioned Anchor Threshold for each iteration under the Anchor FAR, which is a scalar value. When minimizing the FAR loss L_f, we will reduce the similarities of negative pairs, and when minimizing the TAR loss, we will generate higher similarities of positive pairs. In this way, our proposed L_f and L_t simulate the evaluation metrics (i.e., FAR and TAR) well, which can be optimized with gradient methods. In order to maintain the discriminative ability of the learned representation, we also adopt the recent softmax-based FR loss function (e.g., ArcFace) in the training process, which is denoted as face loss L_{cls}. Therefore, the final loss function L of AnchorFace is as follows:

$$L = L_{cls} + \lambda_1 \cdot L_f + \lambda_2 \cdot L_t, \tag{6}$$

where λ_1, λ_2 are used to balance the losses.

As shown in Alg. 1, the algorithm pipeline of our AnchorFace is also provided for better clarification.

Discussion

Limitations of the existing loss functions. Softmax-based loss functions (e.g., ArcFace) have been widely adopted for FR model training, which aims to maximize the intra-class similarity and minimize the inter-class similarity. However, the gap between the training process and evaluation metric limits the further improvements of these methods for practical FR. The gap lies in two folds. First, the optimization target of the training process is not consistent with the evaluation metrics, as shown in Fig. 1. The softmax-based loss functions aim to improve the discriminative capability of the learned representation in the entire feature space and increase the classification accuracy in the training process. However, most FR systems are evaluated by TAR under predefined FAR (i.e., Anchor FAR), which reveals that the similarities distribution of all positive and negative pairs is more important in practice. Therefore, the optimization on the Anchor FAR (i.e., Anchor Optimization) is ignored for existing loss functions. Second, the sample-to-prototype similarities are used in the training process for most existing loss functions, while the sample-to-sample similarities are used in the evaluation process. This kind of gap is not considered in most existing loss functions, which is also mentioned in the recent work VPL (Deng et al. 2021). The loss functions in our AnchorFace consider the sample-to-sample comparisions in the training process.

The Strengths of AnchorFace. First, for face verification, the TAR performance will be drastically boosted if the similarities of the positive pairs slightly lower than the threshold can be shifted higher than the threshold and similarities of the negative pairs with slightly higher than the threshold can be shifted lower than the threshold. Our AnchorFace simulates the evaluation process by directly optimizing TAR under the Anchor FAR in each training iteration. Thus, we will pay more attention to those pairs that greatly affect the performance of FR systems. Besides, based on the property of sigmoid function, gradients usually vanish when the absolute value of the input is large, which indicates that AnchorFace mainly optimizes these pairs with similarities around the Anchor Threshold, instead of the hard pairs with similarities far from the Anchor Threshold. Second, in TAR loss and FAR loss, we optimize the similarities between the features of the current batch and the stored features of the online-updating set, which provides sample-to-sample comparisons in the optimization process. Therefore, our AnchorFace is able to directly optimize the evaluation metrics and introduce the sample-to-sample similarities optimization into the training process, which is an effective supplement for existing loss functions. Finally, in our AnchorFace, no extra costs (e.g., memory, time) are required at inference for FR verification, and the GPU memory usage in training are also acceptable.

Experiments

In this section, we first conduct extensive experiments on multiple benchmark datasets to demonstrate the effectiveness of our proposed AnchorFace. Then, we perform a detailed ablation study to further analyze the contributions of different components of AnchorFace.

Implementation Details

Datasets. For the training dataset, we follow many existing works to employ the refined version of MS-1M (Guo et al. 2016) dataset provided by (Deng et al. 2019), which consists of about 85k identities with 5.8M images. For the testing datasets, we use the following benchmark datasets, including IJB-B (Whitelam et al. 2017), IJB-C (Maze et al. 2018), and IFRT (InsightFace 2021).

Experimental setting. For the pre-processing of the training data, we follow the recent works (Deng et al. 2019; Kim, Park, and Shin 2020; Deng et al. 2020) to generate the normalized face crops (112×112). For the backbone network, we utilize the widely used neural networks(e.g., ResNet-50, ResNet-100 (He et al. 2016)), in which we follow (Deng et al. 2019) to produce 512-dim (i.e., d=512) feature embedding representation. For the training process of AnchorFace, the initial learning rate is 0.1 and divided by 10 at the 110k, 190k, 220k iterations. The batch size and the total iteration are set as 512 and 240k, respectively. For the online-updating set $\mathbf{S}$, by default, we set the maximum number of

Algorithm 1: AnchorFace

Input: FR model $\mathcal{E}$; Classifier $\mathcal{C}$; Current batch data B with m face images; The dimension of each feature d; The maximum number of features for each identity K; The number of identities N; The online-updating set $\mathbf{S} \in \mathbb{R}^{N \times K \times d}$; The validness indicator $\mathbf{V} \in \mathbb{R}^{N \times K}$ corresponding to the validness of the stored features in $\mathbf{S}$; The maximum number of valid steps M;

Randomly initialize $\mathcal{E}$, $\mathcal{C}$, and $\mathbf{S}$;
Zero initialize $\mathbf{V}$;
for each iteration in the training process **do**
    Get batch features $\{f_i\}_{i=1}^{m} = \mathcal{E}(B)$;
    for each feature f_i in $\{f_i\}_{i=1}^{m}$ **do**
        Select the index idx_i to insert f_i into $\mathbf{S}$ based on $\mathbf{V}$ and the corresponding label y_i;
        $\mathbf{S}[y_i][idx_i] = f_i$;
        $\mathbf{V}[y_i][idx_i] = M$;
    end for
    $\mathbf{V} = \mathbf{V} - 1$;
    Construct the positive pairs and negative pairs using $\{f_i\}_{i=1}^{m}$ and the valid features $\mathbf{S}[\mathbf{V} > 0]$;
    Calculate $\{s_p^j\}_{j=1}^{N_p}$ and $\{s_n^i\}_{i=1}^{N_n}$ of the positive pairs and negative pairs, respectively, and obtain t_A under the Anchor FAR;
    Calculate FAR loss L_f and TAR loss L_t based on Eq.4 and Eq.5;
    Calculate L_{cls} based on $\{f_i\}_{i=1}^{m}$ and $\mathcal{C}$;
    Update parameters in $\mathcal{E}$ and $\mathcal{C}$ based on the loss function $L = L_{cls} + \lambda_1 \cdot L_f + \lambda_2 \cdot L_t$;
end for

Output: The optimized FR model $\mathcal{E}$;

features of each identity (i.e., K) and the maximum number of valid steps for each feature (i.e., M) as 5 and 1000, respectively. We set τ as 0.01 in Eq. 4 and Eq. 5. Besides, the loss weights of FAR loss L_f (i.e., λ_1) are set as 1k for the Anchor FARs of 1e-4. The loss weight of TAR loss L_t (i.e., λ_2) is set as 10. To maintain the stability of the training process, we only use the face loss L_{cls} to train the FR model at the first 20k iterations. In the following experiments, we call our AnchorFace combined with ArcFace as **AF-ArcFace**.

Main Results

IJB-B (Whitelam et al. 2017) is composed of 67k face images, 7k face videos and 10k non-face images. Compared with IJB-B, IJB-C (Maze et al. 2018) includes new individuals with increased occlusion and diversity of geographic origin and is composed of 138k face images, 11k face videos and 10k non-face images. As shown in Table 1, we provide the results of AnchorFace using ResNet-100 trained on MS-1M dataset on the challenging IJB-B (Whitelam et al. 2017) and IJB-C (Whitelam et al. 2017) datasets. Since our method can be readily integrated into different existing loss functions, we conduct detailed experiments by combining AnchorFace with three popular functions (i.e., CosFace (Wang et al. 2018b), ArcFace (Deng et al. 2019) and CurricularFace (Ranjan, Castillo, and Chellappa 2017)), and optimize AnchorFace under the Anchor FAR of 1e-4. In Table 1, we name AnchorFace combined with three baseline methods as AF-ArcFace, AF-CosFace and AF-CurricularFace, respectively, and observe that our method achieves significant performance improvements to existing popular loss functions on IJB-B and IJB-C datasets, which shows that AnchorFace is robust and orthogonal for different loss functions. To evaluate the effectiveness of our proposed AnchorFace on face recognition across races, we also conduct experiments on more challenging and large-scale InsightFace Recognition Test (IFRT) (InsightFace 2021), which consists of 1.6M images of 242K identities (non-celebrity) covering four demographic groups: African, Caucasian, Indian and Asian. For each demographic group, all pairs between gallery and probe sets are used for the 1:1 face verification, which evaluates the TAR performance under the FAR of 1e-6. Based on the ArcFace baseline of ResNet-100 trained on MS-1M dataset, we combined our AnchorFace with ArcFace loss to optimize the TAR under Anchor FAR of 1e-6. As shown in Table 2, our AF-ArcFace achieves consistent improvements on all races, which further demonstrates the effectiveness of our proposed AnchorFace.

Methods	IJB-B	IJB-C
CosFace (Wang et al. 2018b)	94.20	95.85
AF-CosFace	**94.38**	**96.09**
ArcFace (Deng et al. 2019)	94.25	95.91
AF-ArcFace	**94.42**	**96.22**
CurricularFace (Huang et al. 2020)	94.85	96.13
AF-CurricularFace	**94.97**	**96.32**

Table 1: 1:1 verification TAR(@FAR=1e-4) on the IJB-B and IJB-C datasets with different loss functions.

Ablation Study

The effect of each component in AnchorFace. We first conduct the experiments based on ResNet-100 with MS-1M dataset to demonstrate the contributions of each loss in our AnchorFace, and the results on IJB-C dataset are reported in Table 3. Specifically, in AnchorFace (w/o TAR), we only use FAR loss without TAR loss. In AnchorFace (w/o FAR), we only use TAR loss without FAR loss. In Table 3, we observe that our AF-ArcFace is better than two alternative methods (i.e., AF-ArcFace (w/o TAR) and AF-ArcFace (w/o FAR)), which demonstrates that it is necessary to utilize both TAR loss and FAR loss.

The effect of the hyper-parameters. To investigate the performance variation of our method with respect to the hyper-parameters (i.e., the maximum number of features for each identity K and the maximum number of valid steps M), we evaluate AnchorFace on IJB-C dataset using different values of K and M. Specifically, to reduce the time and GPU costs of these experiments, we leverage a relatively small model (i.e., ResNet-50) as an example, which is trained on MS-1M dataset using ArcFace loss. In Table 4, we set M as 1000, and use different values of K. When the maximum number of features for each identity K increases from 1 to

Methods	African	Caucasian	Indian	Asian	All
ArcFace	79.228	86.718	85.405	58.341	81.235
AF-ArcFace	**79.314**	**87.001**	**85.593**	**59.702**	**82.062**

Table 2: 1:1 verification TAR(@FAR=1e-6) on the IFRT dataset.

5, our method achieves better performance. However, when we continue to increase the value of K, the improvement of performance becomes relatively stable. In Table 5, we set K as 5, and use different values of M. When M increases from 400 to 1000, our method achieves better performance, which indicates that it is effective to generate more pairs for our AnchorFace. However, when M continues to increase, the performance begins to gradually degrade. It is reasonable that the quality of feature representations begins to decrease when M is larger, which causes inaccurate threshold and TAR estimation. Therefore, to reduce the computation cost and maintain the performance of AnchorFace, by default, we set K as 5 and M as 1000, respectively.

Methods	IJB-C
ArcFace (Deng et al. 2019)	95.91
AF-ArcFace (w/o TAR)	96.02
AF-ArcFace (w/o FAR)	96.05
AF-ArcFace	96.22

Table 3: 1:1 verification TAR(@FAR=1e-4) on the IJB-C dataset of different methods.

K	1	3	5	7	9
TAR (%)	94.28	95.01	95.23	95.22	95.25

Table 4: 1:1 verification TAR(@FAR=1e-4) on the IJB-C dataset of AF-ArcFace when using different values of K.

M	400	800	1000	1200	1400
TAR (%)	95.09	95.16	95.23	95.20	95.18

Table 5: 1:1 verification TAR(@FAR=1e-4) on the IJB-C dataset of AF-ArcFace when using different values of M.

Further Analysis

Effectiveness of the online-updating set. We propose a variant of our proposed AnchorFace (i.e., AF-ArcFace-FC). For the AF-ArcFace-FC, we use the weights of the last FC layer and the features of the current batch to construct the positive pairs and negative pairs, which can also calculate the TAR loss and FAR loss. As shown in Table 6, we observe that AF-ArcFace-FC achieves marginal performance improvement when compared with ArcFace baseline. It is reasonable because the weights of the FC layer are updated slowly as discussed in VPL (Deng et al. 2021), which results in fewer variations of different iterations. Besides, in AF-ArcFace-FC, the number of positive and negative pairs are also insufficient, which makes the threshold and TAR estimations inaccurate.

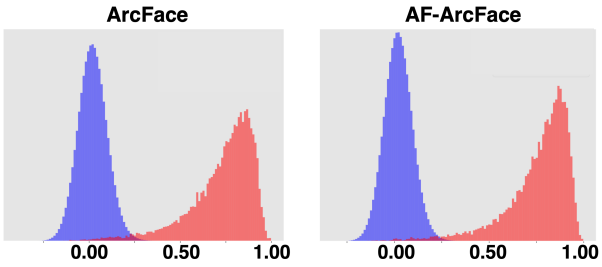

Figure 4: Cosine similarity distributions of the positive pairs and negative pairs between ArcFace and AF-ArcFace.

Methods	IJB-C
ArcFace (Deng et al. 2019)	95.91
AF-ArcFace-FC	95.98
AF-ArcFace	96.22

Table 6: 1:1 verification TAR(@FAR=1e-4) on the IJB-C dataset of different methods.

Visualization. We visualize the distributions of similarity scores on the IJB-C testing set of different methods (i.e., ArcFace, and AF-ArcFace) in Fig. 4. As shown in Fig. 4, when compared to ArcFace baseline, the similarity distributions of the positive pairs and the negative pairs in our AF-ArcFace are more compact, and the margin between the positive pairs and negative pairs of our proposed AF-ArcFace is more distinct, which further demonstrates the effectiveness of our proposed AF-ArcFace.

Conclusion

In this paper, we first investigate the limitations of existing loss functions for practical face recognition, where the optimization on the specific Anchor FAR (i.e., Anchor Optimization) is ignored. Besides, we propose the AnchorFace to directly optimize the non-differentiable evaluation metrics in the training process, where the online-updating set and soften strategy are introduced. Finally, we calculate a pair of loss functions (i.e., TAR loss and FAR loss). Extensive experiments on multiple face recognition benchmark demonstrate the effectiveness of our proposed AnchorFace.

Acknowledgments

This research was supported by National Natural Science Foundation of China under Grant 61932002.

References

Berman, M.; Triki, A. R.; and Blaschko, M. B. 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4413–4421.

Brown, A.; Xie, W.; Kalogeiton, V.; and Zisserman, A. 2020. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *European Conference on Computer Vision*, 677–694. Springer.

Deng, J.; Guo, J.; Liu, T.; Gong, M.; and Zafeiriou, S. 2020. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. In *Proceedings of the IEEE Conference on European Conference on Computer Vision*.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4690–4699.

Deng, J.; Guo, J.; Yang, J.; Lattas, A.; and Zafeiriou, S. 2021. Variational Prototype Learning for Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11906–11915.

Eban, E.; Schain, M.; Mackey, A.; Gordon, A.; Rifkin, R.; and Elidan, G. 2017. Scalable learning of non-decomposable objectives. In *Artificial intelligence and statistics*, 832–840. PMLR.

Guo, Y.; Zhang, L.; Hu, Y.; He, X.; and Gao, J. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, 87–102. Springer.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, Y.; Wang, Y.; Tai, Y.; Liu, X.; Shen, P.; Li, S.; Li, J.; and Huang, F. 2020. Curricularface: adaptive curriculum learning loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5901–5910.

InsightFace. 2021. InsightFace Recognition Test (IFRT). [Online]. https://github.com/deepinsight/insightface/tree/master/challenges/IFRT.

Jin, X.; Peng, B.; Wu, Y.; Liu, Y.; Liu, J.; Liang, D.; Yan, J.; and Hu, X. 2019. Knowledge Distillation via Route Constrained Optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Kemelmacher-Shlizerman, I.; Seitz, S. M.; Miller, D.; and Brossard, E. 2016. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4873–4882.

Kim, Y.; Park, W.; and Shin, J. 2020. BroadFace: Looking at Tens of Thousands of People at Once for Face Recognition. *ECCV*.

Li, Z.; Wu, Y.; Chen, K.; Wu, Y.; Zhou, S.; Liu, J.; and Yan, J. 2020. Learning to Auto Weight: Entirely Data-driven and Highly Efficient Weighting Framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4788–4795.

Liu, J.; Qin, H.; Wu, Y.; Guo, J.; Liang, D.; and Xu, K. 2022. CoupleFace: Relation Matters for Face Recognition Distillation.

Liu, J.; Wu, Y.; Wu, Y.; Li, C.; Hu, X.; Liang, D.; and Wang, M. 2021a. DAM: Discrepancy Alignment Metric for Face Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3814–3823.

Liu, J.; Zhou, S.; Wu, Y.; Chen, K.; Ouyang, W.; and Xu, D. 2020. Block proposal neural architecture search. *IEEE Transactions on Image Processing*, 30: 15–25.

Liu, P.; Zhang, G.; Wang, B.; Xu, H.; Liang, X.; Jiang, Y.; and Li, Z. 2021b. Loss Function Discovery for Object Detection via Convergence-Simulation Driven Search. *ICLR*.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, 7.

Maze, B.; Adams, J.; Duncan, J. A.; Kalka, N.; Miller, T.; Otto, C.; Jain, A. K.; Niggel, W. T.; Anderson, J.; Cheney, J.; et al. 2018. IARPA janus benchmark-c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, 158–165. IEEE.

Meng, Q.; Zhao, S.; Huang, Z.; and Zhou, F. 2021. Magface: A universal representation for face recognition and quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14225–14234.

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z. 2019. Correlation Congruence for Knowledge Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Puthiya Parambath, S.; Usunier, N.; and Grandvalet, Y. 2014. Optimizing F-measures by cost-sensitive classification. *Advances in Neural Information Processing Systems*, 27: 2123–2131.

Ranjan, R.; Castillo, C. D.; and Chellappa, R. 2017. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sohn, K.; Liu, S.; Zhong, G.; Yu, X.; Yang, M.-H.; and Chandraker, M. 2017. Unsupervised domain adaptation for face recognition in unlabeled videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 3210–3218.

Sun, Y.; Chen, Y.; Wang, X.; and Tang, X. 2014. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, 1988–1996.

Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; and Wei, Y. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6398–6407.

Sun, Y.; Liang, D.; Wang, X.; and Tang, X. 2015. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*.

Sun, Y.; Wang, X.; and Tang, X. 2015. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2892–2900.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going Deeper With Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018a. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7): 926–930.

Wang, F.; Xiang, X.; Cheng, J.; and Yuille, A. L. 2017. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, 1041–1049.

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018b. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5265–5274.

Wang, X.; Zhang, S.; Wang, S.; Fu, T.; Shi, H.; and Mei, T. 2020. Mis-classified vector guided softmax loss for face recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12241–12248.

Wen, Y.; Zhang, K.; Li, Z.; and Qiao, Y. 2016. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, 499–515. Springer.

Whitelam, C.; Taborsky, E.; Blanton, A.; Maze, B.; Adams, J.; Miller, T.; Kalka, N.; Jain, A. K.; Duncan, J. A.; Allen, K.; et al. 2017. Iarpa janus benchmark-b face dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 90–98.

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. 2014. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.

Zhang, X.; Fang, Z.; Wen, Y.; Li, Z.; and Qiao, Y. 2017. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, 5409–5418.

Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; and Ren, D. 2020. Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12993–13000.

Zhu, Z.; Huang, G.; Deng, J.; Ye, Y.; Huang, J.; Chen, X.; Zhu, J.; Yang, T.; Lu, J.; Du, D.; and Zhou, J. 2021. WebFace260M: A Benchmark Unveiling the Power of Million-Scale Deep Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10492–10502.

Zoph, B.; and Le, Q. V. 2017. Neural architecture search with reinforcement learning. *ICLR*.