
Loss Function Search for Face Recognition

Xiaobo Wang^{*1} Shuo Wang^{*1} Cheng Chi² Shifeng Zhang² Tao Mei¹

Abstract

In face recognition, designing margin-based (*e.g.*, angular, additive, additive angular margins) softmax loss functions plays an important role in learning discriminative features. However, these hand-crafted heuristic methods are sub-optimal because they require much effort to explore the large design space. Recently, an AutoML for loss function search method AM-LFS has been derived, which leverages reinforcement learning to search loss functions during the training process. But its search space is complex and unstable that hindering its superiority. In this paper, we first analyze that the key to enhance the feature discrimination is actually **how to reduce the softmax probability**. We then design a unified formulation for the current margin-based softmax losses. Accordingly, we define a novel search space and develop a reward-guided search method to automatically obtain the best candidate. Experimental results on a variety of face recognition benchmarks have demonstrated the effectiveness of our method over the state-of-the-art alternatives.

1. Introduction

Face recognition is a fundamental and of great practice values task in the community of pattern recognition and machine learning. The task of face recognition contains two categories: face identification to classify a given face to a specific identity, and face verification to determine whether a pair of face images are of the same identity. In recent years, the advanced face recognition methods (Simonyan & Andrew, 2014; Guo et al., 2018; Wang et al., 2019b; Deng et al., 2019) are built upon convolutional neural networks (CNNs) and the learned high-level discriminative features are adopted for evaluation. To train CNNs with discriminative features, the loss function plays an important role.

^{*}Equal contribution ¹JD AI Research ²Institute of Automation, Chinese Academy of Science. Correspondence to: Shifeng Zhang <shifeng.zhang@nlpr.ia.ac.cn>.

Generally, the CNNs are equipped with classification loss functions (Liu et al., 2017; Wang et al., 2018f;e; 2019a; Yao et al., 2018; 2017; Guo et al., 2020), metric learning loss functions (Sun et al., 2014; Schroff et al., 2015) or both (Sun et al., 2015; Wen et al., 2016; Zheng et al., 2018b). Metric learning loss functions such as contrastive loss (Sun et al., 2014) or triplet loss (Schroff et al., 2015) usually suffer from high computational cost. To avoid this problem, they require well-designed sample mining strategies. So the performance is very sensitive to these strategies. Increasingly more researchers shift their attention to construct deep face recognition models by re-designing the classical classification loss functions.

Intuitively, face features are discriminative if their intra-class compactness and inter-class separability are well maximized. However, as pointed out by (Wen et al., 2016; Liu et al., 2017; Wang et al., 2018b; Deng et al., 2019), the classical softmax loss lacks the power of feature discrimination. To address this issue, Wen *et al.* (Wen et al., 2016) develop a center loss to learn centers for each identity to enhance the intra-class compactness. Wang *et al.* (Wang et al., 2017) and Ranjan *et al.* (Ranjan et al., 2017) propose to use a scale parameter to control the temperature of softmax loss, producing higher gradients to the well-separated samples to reduce the intra-class variance. Recently, several margin-based softmax loss functions (Liu et al., 2017; Chen et al., 2018; Wang et al., 2018c;b; Deng et al., 2019) to increase the feature margin between different classes have also been proposed. Chen *et al.* (Chen et al., 2018) insert virtual classes between different classes to enlarge the inter-class margins. Liu *et al.* (Liu et al., 2017) introduce an angular margin (A-Softmax) between the ground truth class and other classes to encourage larger inter-class variance. However, it is usually unstable and the optimal parameters need to be carefully adjusted for different settings. To enhance the stability of A-Softmax loss, Liang *et al.* (Liang et al., 2017) and Wang *et al.* (Wang et al., 2018b;c) propose an additive margin (AM-Softmax) loss to stabilize the optimization. Deng *et al.* (Deng et al., 2019) develop an additive angular margin (Arc-Softmax) loss, which has a clear geometric interpretation. However, despite great achievements have been made, all of them are hand-crafted heuristic methods that rely on great effort from experts to explore the large design space, which is usually sub-optimal in practice.

Recently, Li *et al.* (Li *et al.*, 2019) propose an AutoML for loss function search method (AM-LFS) from a hyper-parameter optimization perspective. Specifically, they formulate hyper-parameters of loss functions as a parameterized probability distribution sampling and achieve promising results on several different vision tasks. However, they attribute the success of margin-based softmax losses to the relative significance of intra-class distance to inter-class distance, which is not directly used to guide the design of search space. In consequence, the search space is complex and unstable, and is hard to obtain the best candidate.

To overcome the aforementioned shortcomings, including hand-crafted heuristic methods and the AutoML one AM-LFS, we try to analyze the success of margin-based softmax losses and conclude that the key to enhance the feature discrimination is to reduce the softmax probability. According to this analysis, we develop a unified formulation and define a novel search space. We also design a new reward-guided schedule to search the optimal solution. To sum up, the main contributions of this paper can be summarized as follows:

- We identify that for margin-based softmax losses, the key to enhance the feature discrimination is actually **how to reduce the softmax probability**. Based on this understanding, we develop a unified formulation for the prevalent margin-based softmax losses, which involves only one parameter to be determined.
- We define a simple but very effective search space, which can sufficiently guarantee the feature discrimination for face recognition. Accordingly, we design a random and a reward-guided method to search the best candidate. Moreover, for reward-guided one, we develop an efficient optimization framework to dynamically optimize the distribution for sampling of losses.
- We conduct extensive experiments on the face recognition benchmarks, including LFW, SLLFW, CALFW, CPLFW, AgeDB, CFP, RFW, MegaFace and Trillion-Pairs, which have verified the superiority of our new approach over the baseline Softmax loss, the hand-crafted heuristic margin-based Softmax losses, and the AutoML method AM-LFS. To allow more experimental verification, our code is available at <http://www.cbsr.ia.ac.cn/users/xiaobowang/>.

2. Preliminary Knowledge

Softmax. Softmax loss is defined as the pipeline combination of last fully connected layer, softmax function and cross-entropy loss. The detailed formulation is as follows:

$$\mathcal{L}_1 = -\log \frac{e^{\mathbf{w}_y^T \mathbf{x}}}{e^{\mathbf{w}_y^T \mathbf{x}} + \sum_{k \neq y}^K e^{\mathbf{w}_k^T \mathbf{x}}}, \quad (1)$$

where $\mathbf{w}_k \in \mathbb{R}^d$ is the k -th classier ($k \in \{1, 2, \dots, K\}$) and K is the number of classes. $\mathbf{x} \in \mathbb{R}^d$ denotes the feature belonging to the y -th class and d is the feature dimension. In face recognition, the weights $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_K\}$ and the feature \mathbf{x} of the last fully connected layer are usually normalized and their magnitudes are replaced as a scale parameter s (Wang *et al.*, 2017; Deng *et al.*, 2019; Wang *et al.*, 2019b). In consequence, given an input feature vector \mathbf{x} with its ground truth label y , the original softmax loss Eq. (1) can be re-formulated as follows (Wang *et al.*, 2017):

$$\mathcal{L}_2 = -\log \frac{e^{s \cos(\theta_{\mathbf{w}_y, \mathbf{x}})}}{e^{s \cos(\theta_{\mathbf{w}_y, \mathbf{x}})} + \sum_{k \neq y}^K e^{s \cos(\theta_{\mathbf{w}_k, \mathbf{x}})}}, \quad (2)$$

where $\cos(\theta_{\mathbf{w}_k, \mathbf{x}}) = \mathbf{w}_k^T \mathbf{x}$ is the cosine similarity and $\theta_{\mathbf{w}_k, \mathbf{x}}$ is the angle between \mathbf{w}_k and \mathbf{x} . As pointed out by a great many studies (Liu *et al.*, 2016; 2017; Wang *et al.*, 2018b; Deng *et al.*, 2019; Wang *et al.*, 2019b), the learned features with softmax loss are prone to be separable, rather than to be discriminative for face recognition.

Margin-based Softmax. To enhance the feature discrimination for face recognition, several margin-based softmax loss functions (Liu *et al.*, 2017; Wang *et al.*, 2018f;b; Deng *et al.*, 2019) have been proposed in recent years. In summary, they can be defined as follows:

$$\mathcal{L}_3 = -\log \frac{e^{sf(m, \theta_{\mathbf{w}_y, \mathbf{x}})}}{e^{sf(m, \theta_{\mathbf{w}_y, \mathbf{x}})} + \sum_{k \neq y}^K e^{s \cos(\theta_{\mathbf{w}_k, \mathbf{x}})}}, \quad (3)$$

where $f(m, \theta_{\mathbf{w}_y, \mathbf{x}}) \leq \cos(\theta_{\mathbf{w}_y, \mathbf{x}})$ is a carefully designed margin function. Basically, $f(m_1, \theta_{\mathbf{w}_y, \mathbf{x}}) = \cos(m_1 \theta_{\mathbf{w}_y, \mathbf{x}})$ is the motivation of A-Softmax loss (Liu *et al.*, 2017), where $m_1 \geq 1$ and is an integer. $f(m_2, \theta_{\mathbf{w}_y, \mathbf{x}}) = \cos(\theta_{\mathbf{w}_y, \mathbf{x}} + m_2)$ with $m_2 > 0$ is the Arc-Softmax loss (Deng *et al.*, 2019). $f(m_3, \theta_{\mathbf{w}_y, \mathbf{x}}) = \cos(\theta_{\mathbf{w}_y, \mathbf{x}}) - m_3$ with $m_3 > 0$ is the AM-Softmax loss (Wang *et al.*, 2018c;b). More generally, the margin function can be summarized into a combined version: $f(m, \theta_{\mathbf{w}_y, \mathbf{x}}) = \cos(m_1 \theta_{\mathbf{w}_y, \mathbf{x}} + m_2) - m_3$.

AM-LFS. Previous methods relay on hand-crafted heuristics that require much effort from experts to explore the large design space. To address this issue, Li *et al.* (Li *et al.*, 2019) propose a new AutoML for Loss Function Search (AM-LFS) to automatically determine the search space. Specifically, the formulation of AM-LFS is written as follows:

$$\mathcal{L}_4 = -\log \left(a_i \frac{e^{s \cos(\theta_{\mathbf{w}_y, \mathbf{x}})}}{e^{s \cos(\theta_{\mathbf{w}_y, \mathbf{x}})} + \sum_{k \neq y}^K e^{s \cos(\theta_{\mathbf{w}_k, \mathbf{x}})}} + b_i \right), \quad (4)$$

where a_i and b_i are the parameters of search space. $i \in [0, M - 1]$ is the i -th pre-divided bin of the softmax probability. M is the number of divided bins. Moreover, to consider different difficulty levels of examples, the parameters a_i and b_i may be different because they are randomly sampled for each bin. As a result, the search space can be viewed as a candidate set with piece-wise linear functions.

3. Problem Formulation

In this section, we first analyze the key to success of margin-based softmax losses from a new viewpoint and integrate them into a unified formulation. Based on this analysis, we define a novel search space and accordingly develop a random and a reward-guided loss function search method.

3.1. Analysis of Margin-based Softmax Loss

To begin with, let us retrospect the formulation of softmax loss Eq. (2) and margin-based softmax losses Eq. (3). The **softmax probability** p is defined as follows:

$$p = \frac{e^{s \cos(\theta_{w_y, \mathbf{x}})}}{e^{s \cos(\theta_{w_y, \mathbf{x}})} + \sum_{k \neq y}^K e^{s \cos(\theta_{w_k, \mathbf{x}})}}. \quad (5)$$

And the **margin-based softmax probability** p_m is formulated as follows:

$$p_m = \frac{e^{sf(m, \theta_{w_y, \mathbf{x}})}}{e^{sf(m, \theta_{w_y, \mathbf{x}})} + \sum_{k \neq y}^K e^{s \cos(\theta_{w_k, \mathbf{x}})}}. \quad (6)$$

According to the above formulations Eqs. (5) and (6), we can derive the following equation:

$$\begin{aligned} p_m &= \frac{e^{sf(m, \theta_{w_y, \mathbf{x}})}}{e^{sf(m, \theta_{w_y, \mathbf{x}})} + \sum_{k \neq y}^K e^{s \cos(\theta_{w_k, \mathbf{x}})}} \\ &= \frac{e^{sf(m, \theta_{w_y, \mathbf{x}})}}{e^{sf(m, \theta_{w_y, \mathbf{x}})} + e^{\cos(\theta_{w_y, \mathbf{x}})}(1-p)/p} \\ &= \frac{1}{p + e^{[\cos(\theta_{w_y, \mathbf{x}}) - f(m, \theta_{w_y, \mathbf{x}})]}(1-p)} * p \\ &= \frac{1}{ap + (1-a)} * p = h(a, p) * p, \end{aligned} \quad (7)$$

where

$$a = 1 - e^{[\cos(\theta_{w_y, \mathbf{x}}) - f(m, \theta_{w_y, \mathbf{x}})]}, \quad (8)$$

is a modulating factor with non-positive values ($a \leq 0$). Some existing choices are summarized in Table 1. Particularly, when $a = 0$, the margin-based softmax probability p_m becomes identical to the softmax probability p . $h(a, p) = \frac{1}{ap + (1-a)} \in (0, 1]$ is a modulating function to reduce the softmax probability. Therefore, we can claim that, no matter what kind of margin function $f(m, \theta_{w_y, \mathbf{x}})$ has been designed, the key to success of margin-based softmax losses is **how to reduce the softmax probability**.

Compared to the piece-wise linear functions $p_m = a_i p + b_i$ used in AM-LFS (Li et al., 2019), our $p_m = h(a, p) * p$ has several advantages: 1) Our $p_m = h(a, p) * p$ is always less than the softmax probability p while the piece-wise linear functions $p_m = a_i p + b_i$ are not. In other words, the discriminability of AM-LFS is not guaranteed; 2) There is only one parameter a to be searched in our formulation

Table 1. Some existing modulating factors including Softmax, A-Softmax, AM-Softmax and Arc-Softmax, respectively.

Method	Modulating Factor a
Softmax	$a = 0$
A-Softmax	$a = 1 - e^{s[\cos(\theta_{w_y, \mathbf{x}}) - \cos(m\theta_{w_y, \mathbf{x}})]}$
AM-Softmax	$a = 1 - e^{sm}$
Arc-Softmax	$a = 1 - e^{s[\cos(\theta_{w_y, \mathbf{x}}) - \cos(\theta_{w_y, \mathbf{x}} + m)]}$

while the AM-LFS needs search $2M$ parameters. The search space of AM-LFS is complex and unstable; 3) Our method has a reasonable range of the parameter (*i.e.*, $a \leq 0$) hence facilitating the searching procedure, while the parameters of AM-LFS a_i and b_i are without any constraints.

3.2. Random Search

Based on the above analysis, we can insert a simple modulating function $h(a, p) = \frac{1}{ap + (1-a)}$ into the original softmax loss Eq. (2) to generate a unified formulation, which encourages the feature margin between different classes and has the capability of feature discrimination. In consequence, we define our search space as the choices of $h(a, p)$, whose impacts on the training procedure are decided by the modulating factor a . The unified formulation is re-written as:

$$\mathcal{L}_5 = -\log(h(a, p) * p), \quad (9)$$

where the modulating function $h(a, p)$ has a bounded range $(0, 1]$ and the modulating factor is $a \leq 0$. To validate our formulation Eq. (9), we first randomly set the modulating factor $a \leq 0$ at each training epoch and denote this simple manner as **Random-Softmax** in this paper.

3.3. Reward-guided Search

The Random-Softmax can validate that the key to enhance the feature discrimination is to reduce the softmax probability. But it may not be optimal because it is without any guidance for training. To solve this problem, we propose a hyper-parameter optimization method which samples B hyper-parameters $\{a_1, a_2, \dots, a_B\}$ from a distribution at each training epoch and use them to train the current model. Specifically, we model the hyper-parameter a as the Gaussian distribution, described by:

$$a \sim \mathcal{N}(\mu, \sigma^2), \quad (10)$$

where μ is the mean or expectation of the distribution and σ is its standard deviation. After training for one epoch, B models are generated and the rewards $\mathcal{R}(a_i)$, $i \in [1, B]$ of these models are used to update the distribution of hyper-parameter μ by REINFORCE (Williams, 1992) as follows:

$$\mu_{e+1} = \mu_e + \eta \frac{1}{B} \sum_{i=1}^B \mathcal{R}(a_i) \nabla_a \log(g(a_i; \mu, \sigma)), \quad (11)$$

Algorithm 1 Search-Softmax

Input: Training set $\mathcal{S}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$; Validation set \mathcal{S}_v ; Initialized model \mathcal{M}_{w_0} ; Initialized distribution μ_0 ; Distribution learning rate η ; Training epochs E .

for $e = 1$ **to** E **do**

1. Shuffle the training set \mathcal{S}_t and sample B hyper-parameters $\{a_1, a_2, \dots, a_B\}$ via Eq. (10);
2. Train the model \mathcal{M}_{w_e} for one epoch separately with the sampled hyper-parameters and get B candidate models $\{\mathcal{M}_{w_e^1}, \mathcal{M}_{w_e^2}, \dots, \mathcal{M}_{w_e^B}\}$;
3. Calculate the reward for each candidate and get the corresponding scores $\{\mathcal{R}(a_1), \mathcal{R}(a_2), \dots, \mathcal{R}(a_B)\}$;
4. Update the mean μ_{e+1} by using Eq. (11);
5. Decide the index of model with the highest score $i = \operatorname{argmax}_{j \in [1, B]} \mathcal{R}(a_j)$;
6. Broadcast the model $\mathcal{M}_{w_{e+1}} = \mathcal{M}_{w_e^i}$;

end for

Output: The model \mathcal{M}_{w_E}

where $g(a_i; \mu, \sigma)$ is PDF of Gaussian distribution. We update the distribution of a by Eq. (11) and search the best model from these B candidates for the next epoch. We denote this manner as **Search-Softmax** in this paper.

3.4. Optimization

In this part, we give the training procedure of our Search-Softmax loss. Suppose we have a network model \mathcal{M}_w parameterized by w . The training set and validation set are denoted as $\mathcal{S}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and \mathcal{S}_v , respectively. The target of our loss function search is to maximize the model \mathcal{M}_w 's rewards $r(\mathcal{M}_{w^*(a)}, \mathcal{S}_v)$ (e.g., accuracy) on the validation set \mathcal{S}_v with respect to the modulating factor a , and the model \mathcal{M}_w is obtained by minimizing the following search loss:

$$\begin{aligned} \max_a \mathcal{R}(a) &= r(\mathcal{M}_{w^*(a)}, \mathcal{S}_v) \\ \text{s. t. } w^*(a) &= \operatorname{argmin}_w \sum_{(\mathbf{x}, y) \in \mathcal{S}_t} \mathcal{L}^a(\mathcal{M}_w(\mathbf{x}), y). \end{aligned} \quad (12)$$

According to the works (Colson et al., 2007; Li et al., 2019), the Eq. (12) refers to a standard bi-level optimization problem, where the modulating factor a is regarded as a hyper-parameter. We train model parameters w which minimize the training loss \mathcal{L}^a (i.e., Eq. (9)) at the inner level, while seeking a good loss function hyper-parameter a which results in a model parameter w^* that maximizes the reward on the validation set \mathcal{S}_v at the outer level. The model with the highest score is used in next epoch. At last, when the training converges, we directly take the model with the highest score as the final model without any retraining. To simplify the problem, we fix σ as a constant and optimize over μ . For clarity, the whole scheme of our Search-Softmax is summarized in Algorithm 1.

Table 2. Face datasets for training and test. (P) and (G) refer to the probe and gallery set, respectively.

	Datasets	#Identities	Images
Training	CASIA-WebFace-R	9,879	0.43M
	MS-Celeb-1M-v1c-R	72,690	3.28M
Test	LFW	5,749	13,233
	SLLFW	5,749	13,233
	CALFW	5,749	12,174
	CPLFW	5,749	11,652
	AgeDB	568	16,488
	CFP	500	7,000
	RFW	11,430	40,607
	MegaFace	530 (P)	1M (G)
	Trillion-Pairs	5,749 (P)	1.58M (G)

4. Experiments

4.1. Datasets

Training Data. This paper involves two popular training datasets, including CASIA-WebFace (Yi et al., 2014) and MS-Celeb-1M (Guo et al., 2016). Unfortunately, the original CASIA-WebFace and MS-Celeb-1M datasets consist of a great many face images with noisy labels. To be fair, here we use the clean version of CASIA-WebFace (Zhao et al., 2019; 2018) and MS-Celeb-1M-v1c (Deepglint, 2018) for training.

Test Data. We use nine popular face recognition benchmarks, including LFW (Huang et al., 2007), SLLFW (Deng et al., 2017), CALFW (Zheng et al., 2017), CPLFW (Zheng et al., 2018a), AgeDB (Moschoglou et al., 2017), CFP (Sengupta et al., 2016), RFW (Wang et al., 2018d), MegaFace (Kemelmacher-Shlizerman et al., 2016; Nech & Kemelmacher-Shlizerman, 2017) and Trillion-Pairs (Deepglint, 2018), as the test data. For more details about these test datasets, please see their references.

Dataset Overlap Removal. In face recognition, it is very important to perform open-set evaluation, i.e., there should be no overlapping identities between training set and test set. To this end, we need to carefully remove the overlapped identities between the employed training datasets and the test datasets. For the overlap identities removal tool, we use the publicly available script provided by (Wang et al., 2018b) to check whether if two names (one of which is from training set and the other comes from test set) are of the same person. In consequence, we remove 696 identities from the training set CASIA-WebFace and 14,718 identities from MS-Celeb-1M-v1c. For clarity, we denote the refined training datasets as CASIA-WebFace-R and MS-Celeb-1M-v1c-R, respectively. Important statistics of all the involved datasets are summarized in Table 2. To be rigorous, all the experiments are based on the refined training sets.

4.2. Experimental Settings

Data Processing. We detect the faces by adopting the FaceBoxes detector (Zhang et al., 2017; 2019a) and localize five landmarks (two eyes, nose tip and two mouth corners) through a simple 6-layer CNN (Feng et al., 2018; Liu et al., 2019). The detected faces are cropped and resized to 144×144 , and each pixel (ranged between $[0, 255]$) in RGB images is normalized by subtracting 127.5 and divided by 128. For all the training faces, they are horizontally flipped with probability 0.5 for data augmentation.

CNN Architecture. In face recognition, there are many kinds of network architectures (Liu et al., 2017; Wang et al., 2018b;a; Deng et al., 2019). To be fair, the CNN architecture should be same to test different loss functions. To achieve a good balance between computation and accuracy, we use the SEResNet50-IR (Deng et al., 2019) as the backbone, which is also publicly available at the website¹. The output of SEResNet50-IR gets a 512-dimension feature.

Training. Since our Search-Softmax loss is a bi-level optimization problem, our implementation settings can be divided into inner level and outer level. In the inner level, the model parameter w is optimized by stochastic gradient descent (SGD) algorithm and is trained from scratch. The total batch size is 128. The weight decay is set to 0.0005 and the momentum is 0.9. The learning rate is initially 0.1. For the CASIA-WebFace-R, we empirically divide the learning rate by 10 at 9, 18, 26 epochs and finish the training process at 30 epoch. For the MS-Celeb-1M-v1c-R, we divide the learning rate by 10 at 4, 8, 10 epochs, and finish the training process at 12 epoch. For all the compared methods, we run their source codes and keep the same experimental settings. In the outer level, we optimize the modulating factor a by REINFORCE (Williams, 1992) with rewards (*i.e.*, accuracy on LFW) from a fixed number of sampled models. We normalized the rewards returned by each sample to zero mean and unit variance, which is set as the reward of each sample. We use Adam optimizer with a learning rate of $\eta = 0.05$ and set $\sigma = 0.2$ for updating the distribution parameter μ . After that, we broadcast the model parameter w^* with the highest reward for synchronization. All experiments in this paper are implemented by Pytorch (Paszke et al., 2019).

Test. At test stage, only the original image features are employed to compose the face representations. All the reported results in this paper are evaluated by a single model, without model ensemble or other fusion strategies.

For the evaluation metric, the cosine similarity is utilized. We follow the unrestricted with labelled outside data protocol (Huang et al., 2007) to report the performance on LFW, SLLFW, CALFW, CPLFW, AgeDB, CFP and RFW. On Megaface and Trillion-Pairs Challenge, face identification

and verification are conducted by ranking and thresholding the similarity scores. Specifically, for face identification, the Cumulative Match Characteristics (CMC) curves are adopted to evaluate the Rank-1 accuracy. For face verification, the Receiver Operating Characteristic (ROC) curves are adopted. The true positive rate (TPR) at low false acceptance rate (FAR) is emphasized since in real applications false acceptance gives higher risks than false rejection.

For the compared methods, we compare our method with the baseline Softmax loss (Softmax) and the hand-crafted heuristic methods (including A-Softmax (Liu et al., 2017), V-Softmax (Chen et al., 2018), AM-Softmax (Wang et al., 2018c;b) and Arc-Softmax (Deng et al., 2019)) and one AutoML for loss function search method (AM-LFS (Li et al., 2019)). For all the hand-crafted heuristic competitors, their source codes can be downloaded from the github or authors' webpages. While for AM-LFS, we try our best to re-implement it since its source code is not publicly available yet. The corresponding parameter settings of each competitor are mainly determined according to their paper's suggestions. Specifically, for V-Softmax, the number of virtual classes is set as the batch size. For A-Softmax, the margin parameter is set as $m_1 = 3$. While for Arc-Softmax and AM-Softmax, the margin parameters are set as $m_2 = 0.5$ and $m_3 = 0.35$, respectively. The scale parameter s has already been discussed sufficiently in previous works (Wang et al., 2018b;c; Zhang et al., 2019b). In this paper, we empirically fixed it to 32 for all the methods.

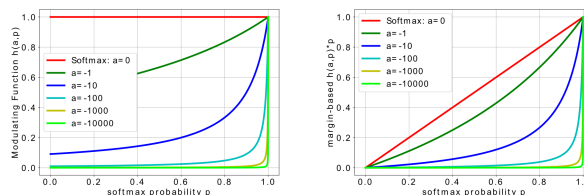


Figure 1. **From Left to Right:** The derived modulating function $h(a, p) = \frac{1}{ap + (1-a)}$ and the corresponding margin-based softmax probability $p_m = h(a, p) * p$ with different modulating factors a .

4.3. Ablation Study

Effect of reducing softmax probability. We study the effect of reducing softmax probability by setting different modulating factors $a \leq 0$. Specifically, we manually sample several values $a = \{0, -1, -10, -100, -1000, -10000\}$. The corresponding modulating functions $h(a, p)$ are shown in the left sub-figure of Figure 1. From the curves, we can see that $h(a, p)$ is a monotonically increasing function with the defined domain $a \leq 0$ (*i.e.*, the value of $h(a, p)$ decreases as the value of a decreases). The function $h(a, p)$ is in the range $(0, 1]$ hence makes $p_m = h(a, p) * p$ always less

¹https://github.com/wujiyang/Face_Pytorch

Table 3. Effect of reducing softmax probability by setting the modulating factor $a \leq 0$. The training set is **MS-Celeb-1M-v1c-R**.

	0	-1	-10	-100	-1000	-10000
LFW	99.53	99.56	99.66	99.71	99.61	99.71
SLLFW	98.78	98.91	99.20	99.28	99.36	99.36

Table 4. Effect of the number of sampled models by setting B . The training set is **MS-Celeb-1M-v1c-R**.

	$B = 2$	$B = 4$	$B = 8$	$B = 16$
LFW	99.79	99.78	99.79	99.78
SLLFW	99.31	99.56	99.53	99.58

than the softmax probability p . The corresponding margin-based softmax probabilities $p_m = h(a, p) * p$ are displayed in the right sub-figure of Figure 3. Moreover, we also report the performance on LFW and SLLFW in Table 3. From the values, it can be concluded that reducing the softmax probability (*i.e.*, $a < 0$) achieves better performance than the softmax probability (*i.e.*, $a = 0$). Eventually, the experiments have indicated that the key to enhance the feature discrimination is to reduce the softmax probability and give us a cue to design the search space for our Search-Softmax.

Effect of the number of sampled models. We investigate the effect of the number of sampled models in the optimization procedure by changing the parameter B in our Search-Softmax loss. Note that it costs more computation resources (GPUs) as B increases. We report the performance results of different B values selected from $\{2, 4, 8, 16\}$ in Table 4 in terms of accuracy on the LFW and SLLFW test sets. The results show that when B is small (*e.g.*, $B = 2$), the performance is not satisfactory because the best candidate cannot be obtained without enough samples. We also observe that the performance exhibits saturation when we keep enlarging B (*e.g.*, $B \geq 4$). For a trade-off between the performance and the training efficiency, we choose to fix B as 4 during training. For all the datasets, each sampled model is trained with 2 P40 GPUs, so a total of 8 GPUs are used.

Convergence. Although the convergence of our method is not easy to be theoretically analyzed, it would be intuitive to see its empirical behavior. Here, we give the loss changes as the number of epochs increases. From the curves in Figure 2, it can be observed that the loss changes of our Random-Softmax is fluctuated because the modulating factor $a \leq 0$ is randomly selected at each epoch. Nevertheless, the overall trend is converged. For our Search-Softmax, we can see that it has a good behavior of convergence. The loss values obviously decrease as the number of epochs increases and the curve is much more smooth than the Random-Softmax.

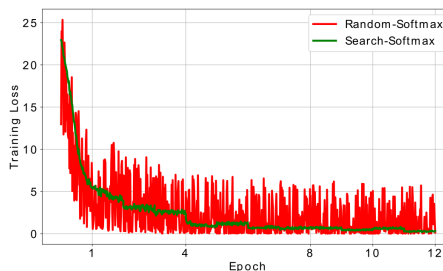


Figure 2. Convergence of the proposed Random-Softmax and Search-Softmax losses. From the curves, we can see that our methods have a good behavior of convergence.

The reason behind this is that our Search-Softmax updates the distribution parameter μ by the rewards of the sampled models. The parameter μ is towards optimal distribution thus the sampled a for each epoch is towards decreasing the loss values to achieve better performance.

4.4. Results on LFW, SLLFW, CALFW, CPLFW, AgeDB, CFP

Tables 5 and 6 provide the quantitative results of the compared methods and our method on the LFW, SLLFW, CALFW, CPLFW, AgeDB and CFP sets. The bold number in each column represents the best result. For the accuracy on LFW, it is well-known that the protocol is typical and easy and almost all the competitors can achieve saturated performance. So the improvement of our Search-Softmax loss is not quite large. On the test sets SLLFW, CALFW, CPLFW, AgeDB and CFP, we can observe that our Random-Softmax loss is better than the baseline Softmax loss and is comparable to most of the margin-based softmax losses. Our Search-Softmax loss further boost the performance and is better than the state-of-the-art alternatives. Specifically, when training by the CASIA-WebFace-R dataset, our Search-Softmax achieves about 0.72% average improvement over the best competitor AM-Softmax. When training by the MS-Celeb-1M-v1c-R dataset, our Search-Softmax still outperforms the best competitor AM-Softmax with 0.31% average improvement. The main reason is that the candidates sampled from our proposed search space can well approximate the margin-based loss functions, which means their good properties can be sufficiently explored and utilized during the training phase. Meanwhile, our optimization strategy enables that the dynamic loss can guide the model training of different epochs, which helps further boost the discrimination power. Nevertheless, we can see that the improvements of our method on these test sets are not by a large margin. The reason is that the test protocol is relatively easy and the performance of all the methods on these test sets are near saturation. So there is an urgent need

Table 5. Verification performance (%) of different methods on the test sets LFW, SLLFW, CALFW, CPLFW, AgeDB and CFP. The training set is **CASIA-WebFace-R**.

Method	LFW	SLLFW	CALFW	CPLFW	AgeDB	CFP	Avg.
Softmax	97.85	92.98	86.05	78.58	89.91	90.58	89.32
A-Softmax (Liu et al., 2017)	98.30	93.40	86.36	78.13	89.43	90.11	89.28
V-Softmax (Chen et al., 2018)	98.60	93.11	85.36	78.10	88.86	91.08	89.18
AM-Softmax (Wang et al., 2018b)	99.23	97.01	90.38	82.65	93.65	93.11	92.67
Arc-Softmax (Deng et al., 2019)	99.00	96.29	89.93	81.66	93.70	92.88	92.24
AM-LFS (Li et al., 2019)	98.88	95.23	88.14	80.63	91.41	92.67	91.16
Random-Softmax (Ours)	99.26	97.03	90.71	83.38	93.88	93.32	92.93
Search-Softmax (Ours)	99.15	97.68	90.98	84.21	94.15	94.21	93.39

Table 6. Verification performance (%) of different methods on the test sets LFW, SLLFW, CALFW, CPLFW, AgeDB and CFP. The training set is **MS-Celeb-1M-v1c-R**.

Method	LFW	SLLFW	CALFW	CPLFW	AgeDB	CFP	Avg.
Softmax	99.53	98.78	93.38	86.25	96.66	93.00	94.60
A-Softmax (Liu et al., 2017)	99.56	98.63	93.86	86.40	96.31	93.57	94.72
V-Softmax (Chen et al., 2018)	99.65	99.23	94.66	87.51	97.06	93.67	95.29
AM-Softmax (Wang et al., 2018b)	99.68	99.40	95.26	88.63	97.60	95.22	95.96
Arc-Softmax (Deng et al., 2019)	99.69	99.26	95.21	88.33	97.35	95.00	95.80
AM-LFS (Li et al., 2019)	99.68	99.01	94.18	86.85	96.70	93.70	95.02
Random-Softmax (Ours)	99.65	99.39	95.10	89.03	97.63	95.01	95.97
Search-Softmax (Ours)	99.78	99.56	95.40	89.50	97.75	95.64	96.27

Table 7. Verification performance (%) of different methods on the test set RFW. The training set is **CASIA-WebFace-R**.

Method	Caucasian	Indian	Asian	African
Softmax	89.16	77.50	78.16	74.16
A-Softmax	88.16	79.33	79.33	77.50
V-Softmax	87.00	78.00	79.49	73.83
AM-Softmax	92.33	83.83	82.50	82.33
Arc-Softmax	91.49	83.66	81.00	80.66
AM-LFS	91.49	78.99	78.50	79.83
Random-Softmax	91.99	84.83	83.66	82.33
Search-Softmax	90.99	86.50	85.00	85.16

Table 8. Verification performance (%) of different methods on the test set RFW. The training set is **MS-Celeb-1M-v1c-R**.

Method	Caucasian	Indian	Asian	African
Softmax	97.50	90.49	91.49	87.33
A-Softmax	97.50	91.49	90.66	87.66
V-Softmax	96.33	93.16	93.50	91.49
Arc-Softmax	98.99	93.49	93.49	94.00
AM-Softmax	99.00	95.16	94.66	94.16
AM-LFS	91.49	93.49	92.33	89.99
Random-Softmax	98.83	96.16	93.66	93.33
Search-Softmax	99.00	96.17	94.67	95.33

to test the performance of all the competitors on new test sets or test with more complicated protocols.

4.5. Results on RFW

Firstly, we evaluate all the competitors on the recent proposed new test set RFW (Wang et al., 2018d). RFW is a face recognition benchmark for measuring racial bias, which consists of four test subsets, namely Caucasian, Indian, Asian and African. Tables 7 and 8 display the performance comparison of all the involved methods. From the values, we can conclude that the results on the four subsets exhibit the same trends, *i.e.*, our method is better than the baseline Softmax loss, the hand-crafted margin-based losses and the recent AM-LFS in most of cases. Concretely, our Random-

Softmax obviously outperforms the Softmax loss by a large margin, which reveals that reducing the softmax probability will enhance the feature discrimination for face recognition. Our reward-guided one Search-Softmax, which defines an effective search space to well approximate the margin-based loss functions and uses rewards to explicitly search the best candidate at each epoch, is more likely to enhance the discriminative feature learning. Therefore, our Search-Softmax loss usually learns more discriminative face features and achieves higher performance than previous alternatives.

4.6. Results on MegaFace and Trillion-Pairs

We then test all the competitors with more complicated protocols. Specifically, the identification (Id.) Rank-1 and the

Table 9. Performance (%) of different loss functions on the test sets MegaFace and Trillion-Pairs. The training set is CASIA-WebFace-R.

Method	MegaFace		Trillion-Pairs	
	Id.	Veri.	Id.	Veri.
Softmax	65.17	71.29	12.34	11.35
A-Softmax	64.48	71.98	11.83	11.11
V-Softmax	60.09	65.40	9.08	8.65
Arc-Softmax	79.91	84.57	21.32	20.97
AM-Softmax	82.86	87.33	25.26	24.66
AM-LFS	71.30	77.74	16.16	15.06
Random-Softmax	82.51	86.13	27.70	27.28
Search-Softmax	84.38	88.34	29.23	28.49

Table 10. Performance (%) of different loss functions on the test sets MegaFace and Trillion-Pairs. The training set is MS-Celeb-1M-v1c-R.

Method	MegaFace		Trillion-Pairs	
	Id.	Veri.	Id.	Veri.
Softmax	91.10	92.30	50.34	46.63
A-Softmax	90.81	93.49	49.99	45.59
V-Softmax	94.45	95.25	63.85	61.17
Arc-Softmax	96.39	96.86	67.60	66.46
AM-Softmax	96.77	97.20	69.02	67.94
AM-LFS	92.51	93.80	54.85	52.76
Random-Softmax	96.15	96.81	68.73	68.03
Search-Softmax	96.97	97.84	70.41	68.67

verification (Veri.) TPR@FAR=1e-6 on MegaFace, the identification (Id.) TPR@FAR=1e-3 and the verification (Veri.) TPR@FAR=1e-9 on Trillion-Pairs are reported in Tables 9 and 10, respectively. From the numbers, we observe that our Search-Softmax achieves the best performance over the baseline Softmax loss, the margin-based softmax losses, the AutoML one AM-LFS and our naive Random-Softmax, on both MegaFace and Trillion-Pairs Challenge. Specifically, on MegaFace, for our proposed Search-Softmax, it obviously beats the best margin-based competitor AM-Softmax loss by a large margin (about 1.5% on identification and 1.0% on verification when training by CASIA-WebFace-R, and 0.2% and 0.6% when training by MS-Celeb-1M-v1c-R). Compared to the AM-LFS, our Search-Softmax loss is also better due to our new designed search space. In Figures 3 and 4, we draw both of the CMC curves to evaluate the performance of face identification and the ROC curves to evaluate the performance of face verification on MegaFace Set 1. From the curves, we can see the similar trends at other measures. On Trillion-Pairs Challenge, we can observe that the results exhibit the same trends that emerged on MegaFace test set. Besides, the trends are more obvious. In particular, we achieve about 4% improvements by CASIA-WebFace-R and 1% improvements by MS-Celeb-1M-v1c-R at both the identification and the verification. In these experiments, we

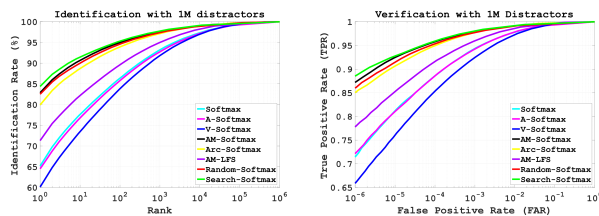


Figure 3. From Left to Right: CMC curves and ROC curves of different loss functions with 1M distractors on MegaFace Set 1. The training set is CASIA-WebFace-R.

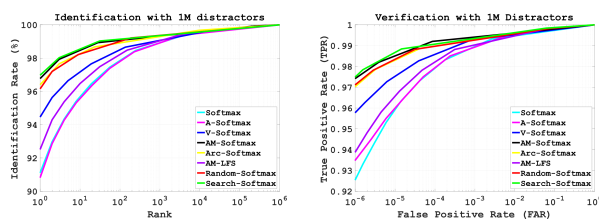


Figure 4. From Left to Right: CMC curves and ROC curves of different loss functions with 1M distractors on MegaFace Set 1. The training set is MS-Celeb-1M-v1c-R.

have clearly demonstrated that our Search-Softmax loss is superior for both the identification and verification tasks, especially when the false positive rate is very low. To sum up, by designing a simple but very effective search space and using rewards to guide the discriminative learning, our new developed Search-Softmax loss has shown its strong generalization ability for face recognition.

5. Conclusion

This paper has summarized that the key to enhance the feature discrimination for face recognition is how to reduce the softmax probability. Based on this knowledge, we design a unified formulation for the prevalent margin-based softmax losses. Moreover, we define a new search space to guarantee the feature discrimination. Accordingly, we develop a random and a reward-guided loss function search method to obtain the best candidate. An efficient optimization framework for optimizing the distribution of search space is also proposed. Extensive experiments on a variety of face recognition benchmarks have validated the effectiveness of our new approach over the baseline softmax loss, the hand-crafted heuristic methods, *i.e.*, margin-based softmax losses, and the recent AutoML one AM-LFS.

References

- Chen, B., Deng, W., and Shen, H. Virtual class enhanced discriminative embedding learning. In *Advances in Neural Information Processing Systems*, pp. 1942–1952, 2018.
- Colson, B., Marcotte, P., and Savard, G. An overview of bilevel optimization. *Annals of operations research*, 153 (1):235–256, 2007.
- Deepglint. <http://trillionpairs.deepglint.com/overview>, 2018.
- Deng, J., Guo, J., Xue, N., and Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4690–4699, 2019.
- Deng, W., Hu, J., Zhang, N., Chen, B., and Guo, J. Fine-grained face verification: Fglfw database, baselines, and human-dcmn partnership. *Pattern Recognition*, 66:63–73, 2017.
- Feng, Z.-H., Kittler, J., Awais, M., Huber, P., and Wu, X.-J. Wing loss for robust facial landmark localisation with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2235–2245, 2018.
- Guo, J., Zhu, X., Lei, Z., and Li, S. Z. Face synthesis for eyeglass-robust face recognition. In *Chinese Conference on Biometric Recognition*, pp. 275–284. Springer, 2018.
- Guo, J., Zhu, X., Zhao, C., Cao, D., Lei, Z., and Li, S. Z. Learning meta face recognition in unseen domains. *arXiv preprint arXiv:2003.07733*, 2020.
- Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pp. 87–102. Springer, 2016.
- Huang, G., Ramesh, M., and Miller, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *Technical Report*, 2007.
- Kemelmacher-Shlizerman, I., Seitz, S. M., Miller, D., and Brossard, E. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4873–4882, 2016.
- Li, C., Yuan, X., Lin, C., Guo, M., Wu, W., Yan, J., and Ouyang, W. Am-lfs: Automl for loss function search. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8410–8419, 2019.
- Liang, X., Wang, X., Lei, Z., Liao, S., and Li, S. Z. Soft-margin softmax for deep classification. In *International Conference on Neural Information Processing*, pp. 413–421. Springer, 2017.
- Liu, W., Wen, Y., Yu, Z., and Yang, M. Large-margin softmax loss for convolutional neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, volume 2, pp. 7, 2016.
- Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. Sphreface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 212–220, 2017.
- Liu, Y., Shi, H., Si, Y., Shen, H., Wang, X., and Mei, T. A high-efficiency framework for constructing large-scale face parsing benchmark. *arXiv preprint arXiv:1905.04830*, 2019.
- Moschoglou, S., Papaioannou, A., Sagonas, C., Deng, J., Kotsia, I., and Zafeiriou, S. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 51–59, 2017.
- Nech, A. and Kemelmacher-Shlizerman, I. Level playing field for million scale face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7044–7053, 2017.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.
- Ranjan, R., Castillo, C., and Chellappa, R. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.
- Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.
- Sengupta, S., Chen, J.-C., Castillo, C., Patel, V. M., Chellappa, R., and Jacobs, D. W. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–9. IEEE, 2016.
- Simonyan, K. and Andrew, Z. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

- Sun, Y., Chen, Y., Wang, X., and Tang, X. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pp. 1988–1996, 2014.
- Sun, Y., Wang, X., and Tang, X. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2892–2900, 2015.
- Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1041–1049, 2017.
- Wang, F., Chen, L., Li, C., Huang, S., Chen, Y., Qian, C., and Change Loy, C. The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 765–780, 2018a.
- Wang, F., Cheng, J., Liu, W., and Liu, H. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018b.
- Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018c.
- Wang, M., Deng, W., Hu, J., Peng, J., Tao, X., and Huang, Y. Racial faces in-the-wild: Reducing racial bias by deep unsupervised domain adaptation. *arXiv:1812.00194*, 2018d.
- Wang, X., Wang, S., Zhang, S., Fu, T., and Mei, T. Support vector guided softmax loss for face recognition. *arXiv preprint arXiv:1812.11317*, 2018e.
- Wang, X., Zhang, S., Lei, Z., Liu, S., Guo, X., and Li, S. Z. Ensemble soft-margin softmax loss for image classification. *arXiv preprint arXiv:1805.03922*, 2018f.
- Wang, X., Wang, S., Wang, J., Shi, H., and Mei, T. Co-mining: Deep face recognition with noisy labels. In *Proceedings of the IEEE international conference on computer vision*, pp. 9358–9367, 2019a.
- Wang, X., Zhang, S., Wang, S., Fu, T., Shi, H., and Mei, T. Mis-classified vector guided softmax loss for face recognition. *arXiv preprint arXiv:1912.00833*, 2019b.
- Wen, Y., Zhang, K., Li, Z., and Qiao, Y. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pp. 499–515. Springer, 2016.
- Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
- Yao, T., Pan, Y., Li, Y., and Mei, T. Incorporating copying mechanism in image captioning for learning novel objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6580–6588, 2017.
- Yao, T., Pan, Y., Li, Y., and Mei, T. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 684–699, 2018.
- Yi, D., Lei, Z., Liao, S., and Li, S. Z. Learning face representation from scratch. *arXiv:1411.7923*, 2014.
- Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., and Li, S. Z. Faceboxes: A cpu real-time face detector with high accuracy. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–9. IEEE, 2017.
- Zhang, S., Wang, X., Lei, Z., and Li, S. Z. Faceboxes: A cpu real-time and accurate unconstrained face detector. *Neurocomputing*, 2019a.
- Zhang, X., Zhao, R., Qiao, Y., Wang, X., and Li, H. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10823–10832, 2019b.
- Zhao, J., Cheng, Y., Xu, Y., Xiong, L., Li, J., Zhao, F., Jayashree, K., Pranata, S., Shen, S., Xing, J., et al. Towards pose invariant face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2207–2216, 2018.
- Zhao, J., Li, J., Tu, X., Zhao, F., Xin, Y., Xing, J., Liu, H., Yan, S., and Feng, J. Multi-prototype networks for unconstrained set-based face recognition. *arXiv preprint arXiv:1902.04755*, 2019.
- Zheng, T., Deng, W., Hu, J., and Hu, J. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv:1708.08197*, 2017.
- Zheng, T., Deng, W., Zheng, T., and Deng, W. Cross-pose lfw: A database for studying crosspose face recognition in unconstrained environments. *Tech. Rep.*, 2018a.
- Zheng, Y., Pal, D. K., and Savvides, M. Ring loss: Convex feature normalization for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5089–5097, 2018b.