Support Vector Guided Softmax Loss for Face Recognition

Xiaobo Wang[1], Shuo Wang[1], Shifeng Zhang[2], Tianyu Fu[1], Hailin Shi[1], Tao Mei[1]
[1]JD AI Research [2] Institute of Automation, Chinese Academy of Science.
{wangxiaobo8,wangshuo30,futianyu,shihailin,tmei}@jd.com, shifeng.zhang@nlpr.ia.ac.cn

Abstract

Face recognition has witnessed significant progresses due to the advances of deep convolutional neural networks (CNNs), the central challenge of which, is feature discrimination. To address it, one group tries to exploit mining-based strategies (e.g., hard example mining and focal loss) to focus on the informative examples. The other group devotes to designing margin-based loss functions (e.g., angular, additive and additive angular margins) to increase the feature margin from the perspective of ground truth class. Both of them have been well-verified to learn discriminative features. However, they suffer from either the ambiguity of hard examples or the lack of discriminative power of other classes. In this paper, we design a novel loss function, namely support vector guided softmax loss (SV-Softmax), which adaptively emphasizes the mis-classified points (support vectors) to guide the discriminative features learning. So the developed SV-Softmax loss is able to eliminate the ambiguity of hard examples as well as absorb the discriminative power of other classes, and thus results in more discrimiantive features. To the best of our knowledge, this is the first attempt to inherit the advantages of mining-based and margin-based losses into one framework. Experimental results on several benchmarks have demonstrated the effectiveness of our approach over state-of-the-arts.

1. Introduction

Face recognition is a fundamental and of great practice values task in the community of computer vision and patter recognition. 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. Though it has been extensively studied for decades [38, 2, 35, 25, 27, 21], there still exist a great many challenges for accurate face recognition, especially on large-scale test datasets, such as MegaFace Challenge [9] or Trillion Pairs Challenge[1].

[1]http://trillionpairs.deepglint.com/overview

In recent years, the advanced face recognition models are usually built upon deep convolutional neural networks [31, 7, 23] and the learned discriminative features play a significant role. To train deep models, the CNNs are generally equipped with classification loss functions [28, 32, 37, 10, 14, 36], metric learning loss functions [26, 20, 34] or both [18, 27, 37, 41]. Metric learning loss functions such as contrastive loss [26] or triplet loss [20] usually suffer from high computational cost. To avoid this problem, they require carefully designed sample mining strategies and the performance is very sensitive to these strategies. So increasingly more researchers shift their attentions to construct deep face recognition models by redesigning the 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 many recent studies [37, 32, 14, 30, 36, 4], the current prevailing classification loss function (*i.e.*, Softmax loss) usually lacks the power of feature discrimination for deep face recognition. To address this issue, one group proposes to explore the mining-based loss functions [22, 12, 24, 39]. Shrivastava *et al.* [22] develop a hard mining softmax (HM-Softmax) to improve the feature discrimination by constructing mini-batches using high-loss examples. Among which, the percentage of hard examples is empirically decided and the easy examples are completely discarded. In contrast, Lin *et al.* [12] design a relatively soft mining softmax, namely Focal loss (F-Softmax), to focus training on a sparse set of hard examples. It usually achieves more promising results than the simple hard mining softmax. Yuan *et al.* [39] select the hard examples based on model complexity and train an ensemble to model examples of different hard levels. The other group prefers to design margin-based loss functions [14, 30, 4]. This group does not focus on optimizing hard examples but directly increasing the feature margin between different classes. Wen *et al.* [37] develop a center loss to learn centers for each identity to enhance the intra-class compactness. Wang *et al.* [32] and Ranjan *et al.* [19] propose to use a scale parameter to control the temperature of softmax loss, producing higher gradients to the well-separated samples to

shrink the intra-class variance. Liu *et al.* [13, 14] introduce an angular margin (A-Softmax) between the ground truth class and other classes to encourage the larger inter-class variance. However, it is usually unstable and the optimal parameters are hard to determinate. To enhance the stability of A-Softmax loss, several alternative approaches [30, 36, 15, 4] have been proposed. Wang *et al.* [30] design an additive margin (AM-Softmax) loss to stabilize the optimization and have achieved promising performance. Deng *et al.* [4] develop an additive angular margin (Arc-Softmax) loss, which has a more clear geometric interpretation.

Although these two groups have been well-verified to learn discriminative features for face recognition. The motivation of mining-based losses is to focus on hard examples while margin-based losses are to enlarge the feature margin between different classes. Currently, they develop independently and both of them have their own intrinsic drawbacks. To the mining-based losses, the definition of hard examples is ambiguous and they are often empirically selected. How to semantically decide the hard examples is still an open problem. To the margin-based losses, most of them learn discriminative features by enlarging the feature margin, only from the perspective of ground truth class (*self-motivation*). They usually ignore the discriminative power from the perspective of other non-ground truth classes (*other-motivation*). Moreover, the relation between mining-based and margin-based losses remains unclear.

To overcome the above shortcomings, this paper tries to design a new loss function, which adaptively emphasizes on the informative support vectors to bridge the gap between mining-based and margin-based losses and semantically integrate them into one framework. To sum up, the main contributions of this paper can be summarized as follows:

- We propose a novel SV-Softmax loss, which eliminates the ambiguity of hard examples as well as absorbs the discriminative power of other classes by focusing on support vectors. To the best of our knowledge, this is the first attempt to semantically fuse the mining-based and margin-based losses into one framework.

- We deeply analyze the relations of our SV-Softmax loss to the current mining-based and margin-based losses, and further develop an improved version SV-X-Softmax loss to enhance the feature discrimiantion. Our code will be available at `https://github.com/xiaoboCASIA/SV-X-Softmax`.

- We conduct extensive experiments on the benchmarks of LFW [8], MegaFace Challenge [9, 16] and Trillion Pairs Challenge, which have verified the superiority of our new approach over the baseline Softmax loss, the mining-based Softmax losses, the margin-based Softmax losses, and their naive fusions.

2. Preliminary Knowledge

Softmax. Softmax loss is defined as the pipeline combination of the last fully connected layer, the softmax function and the cross-entropy loss. In face recognition, the weights $\boldsymbol{w}_k$, (where $k \in \{1, 2, \ldots, K\}$ and K is the number of classes) and the feature $\boldsymbol{x}$ of the last fully connected layer are usually normalized and the magnitude is replaced as a scale parameter s [32, 30, 4]. In consequence, given an input feature vector $\boldsymbol{x}$ with its corresponding ground truth label y, the softmax loss can be formulated as follows:

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

where $\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}}) = \boldsymbol{w}_k^T\boldsymbol{x}$ is the cosine similarity and $\theta_{\boldsymbol{w}_k,\boldsymbol{x}}$ is the angle between $\boldsymbol{w}_k$ and $\boldsymbol{x}$. As pointed out by a great many studies [13, 14, 30, 4], the learned features with softmax loss are prone to be separable, rather than to be discriminative for face recognition.

Mining-based Softmax. Hard example mining is becoming a common practice to effectively train deep CNNs. Its idea is to focus training on the informative examples, thus it usually results in more discriminative features. There are recent works that select hard examples based on loss value [22, 12] or model complexity [39] to learn discriminative features. Generally, they can be summarized as:

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

where $p_y = \frac{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}})}}{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}})} + \sum_{k\neq y}^{K} e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}}$ is the predicted ground truth probability and $g(p_y)$ is an indicator function. Basically, to the soft mining method Focal loss [12] (F-Softmax), $g(p_y) = (1-p_y)^\gamma$, γ is a modulating factor. To the hard mining method HM-Softmax [22], $g(p_y) = 0$ when the sample is indicated as easy while $g(p_y) = 1$ when the sample is hard. However, the definition of hardness is ambiguous and they usually lead to sensitive performance.

Margin-based Softmax. To directly enhance the feature discrimination, several margin-based softmax loss functions [14, 36, 30, 4] have been proposed in recent years. In summary, they can be defined as follows:

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

where $f(m, \theta_{\boldsymbol{w}_y,\boldsymbol{x}})$ is a carefully designed margin function. Basically, $f(m_1, \theta_{\boldsymbol{w}_y,\boldsymbol{x}}) = \cos(m_1\theta_{\boldsymbol{w}_y,\boldsymbol{x}})$ is the motivation of A-Softmax loss [14], where $m_1 \geq 1$ and is an integer. $f(m_2, \theta_{\boldsymbol{w}_y,\boldsymbol{x}}) = \cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}) - m_2$ with $m_2 > 0$ is the AM-Softmax loss [30]. $f(m_3, \theta_{\boldsymbol{w}_y,\boldsymbol{x}}) = \cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}} + m_3)$ with $m_3 > 0$ is the Arc-Softmax loss [4]. More generally, the

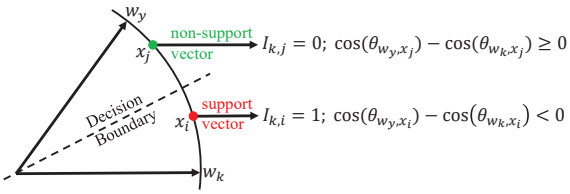

Figure 1. A geometrical interpretation of SV-Softmax loss from feature perspective. The support vectors (red circle points) are those who are mis-classified by the current classifiers. SV-Softmax loss semantically focuses on optimizing such support vectors.

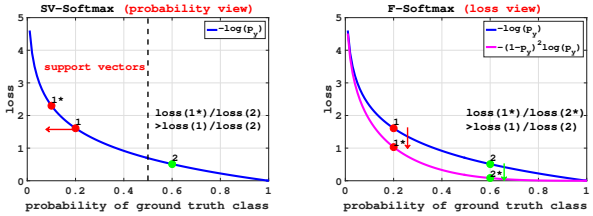

Figure 2. **From left to right**: SV-Softmax loss vs. Mining-based softmax loss (*e.g.*, Focal loss [12]). SV-Softmax loss semantically defines the hard examples (support vectors) and emphasizes them from the probability view, while the hard examples of Focal loss are ambiguous and are concerned from the loss view.

margin function can be summarized into a combined version: $f(m, \theta_{\boldsymbol{w}_y,\boldsymbol{x}}) = \cos(m_1\theta_{\boldsymbol{w}_y,\boldsymbol{x}} + m_3) - m_2$. However, all these methods achieve the feature margin only from the perspective of ground truth class y. They are not aware of the importance of other non-ground truth classes.

3. Problem Formulation

3.1. Naive Mining-Margin Softmax Loss

The mining-based loss functions aim to focus on the hard examples while the margin-based loss functions are to enlarge the feature margin between different classes. Therefore, these two branches can seamlessly incorporate into each other. The naive motivation to directly integrate them can be formulated as:

$$\mathcal{L}_4 = -g(p_y)\log\frac{e^{sf(m,\theta_{\boldsymbol{w}_y,\boldsymbol{x}})}}{e^{sf(m,\theta_{\boldsymbol{w}_y,\boldsymbol{x}})} + \sum_{k\neq y}^{K} e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}}. \tag{4}$$

However, this formulation Eq. (4) only absorbs their own merits. It can not solve their respective shortcomings. Detailedly, it only encourages the feature margin from the perspective of the ground truth class by $f(m, \theta_{\boldsymbol{w}_y,\boldsymbol{x}})$ (*self-motivation*), ignoring the feature discriminative power of other non-ground truth classes (*other-motivation*). Moreover, the hard examples are still empirically selected by the indicator function $g(p_y)$, without semantic guidance. In other words, the definition of hard examples is ambiguous.

3.2. Support Vector Guided Softmax Loss

Intuition says that considering the well-separated feature vectors has little effect on the learning problem. That means the mis-classified feature vectors are more crucial to enhance the feature discriminability. Motivated by this, the hard example mining [22] and the recent Focal loss [12] techniques are proposed to focus training on a sparse set of hard examples and ignore the vast number of easy ones during training. However, they either empirically sample hard examples according to loss values or empirically down-weight the easy examples by a modulating factor. In other words, the definition of hard examples is ambiguous, and without intuitive interpretation.

To address it, we alternatively introduce a more elegant way to focus training on the informative features (*i.e.*, support vectors). Specifically, we define a binary mask to adaptively indicate whether a sample is selected as the support vector by a specific classifier in the current stage. To the end, the binary mask is defined as follows:

$$I_k = \begin{cases} 0, & \cos(\theta_{\boldsymbol{w}_y,x}) - \cos(\theta_{\boldsymbol{w}_k,x}) \geq 0 \\ 1, & \cos(\theta_{\boldsymbol{w}_y,x}) - \cos(\theta_{\boldsymbol{w}_k,x}) < 0 \end{cases}. \tag{5}$$

From the definition, we can see that if a sample is mis-classified, *i.e.*, $\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}}) - \cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}}) < 0$, it will be emphasized temporarily. In this way, the concept of hard examples is clearly defined and we mainly focus on such a sparse set of support vectors. Consequently, our Support Vector Guided Softmax (**SV-Softmax**) loss is formulated:

$$\mathcal{L}_5 = -\log\frac{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}})}}{e^{s\cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}})} + \sum_{k\neq y}^{K} h(t,\theta_{\boldsymbol{w}_k,\boldsymbol{x}},I_k)\, e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}}, \tag{6}$$

where t is a preset hyperparameter and the indicator function $h(t, \theta_{\boldsymbol{w}_k,\boldsymbol{x}}, I_k)$ is defined as:

$$h(t,\theta_{\boldsymbol{w}_k,\boldsymbol{x}},I_k) = e^{s(t-1)(\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})+1)I_k}. \tag{7}$$

Obviously, when $t = 1$, the designed SV-Softmax loss becomes identical to the original softmax loss. Figure 1 gives the geometrical interpretation of our SV-Softmax loss.

3.2.1 Relation to Mining-based Softmax Losses

To illustrate the advantages of our SV-Softmax loss over the traditional mining-based loss functions (*e.g.*, Focal loss [12]), we use the binary classification case as an example. Assume that we have two samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, both of them are from class 1. Figure 2 gives a diagram, where $\boldsymbol{x}_1$ is relatively hard while $\boldsymbol{x}_2$ is relatively easy. The traditional mining-based Focal loss is to differentially re-weight the

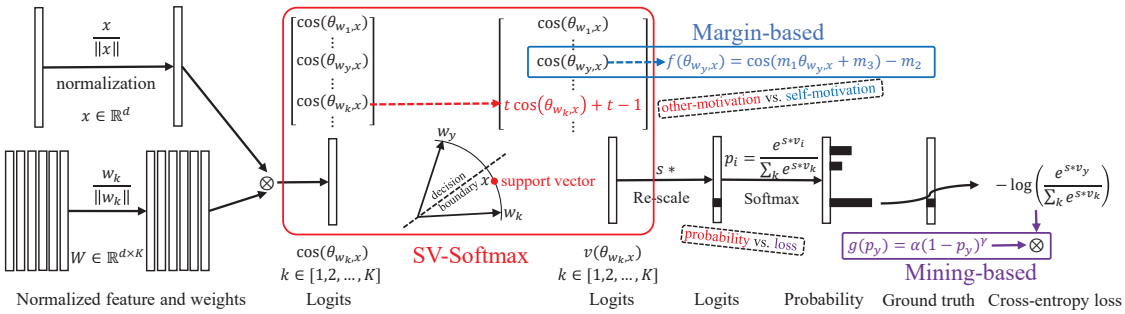

Figure 3. Pipeline of our SV-Softmax loss and its relations to the existing mining-based and margin-based losses. Our SV-Softmax loss semantically integrates the motivation of mining-based and margin-based losses into one framework, but from different viewpoints.

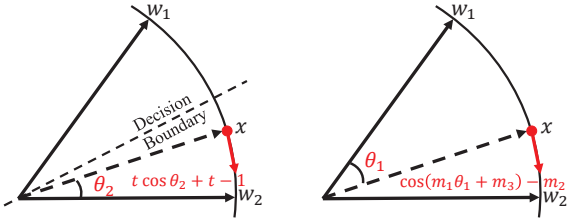

Figure 4. **From left to right**: SV-Softmax loss vs. Margin-based softmax loss. SV-Softmax loss enlarges the feature margin from other classes (*other-motivation*) while current margin-based losses are directly from the ground truth class (*self-motivation*).

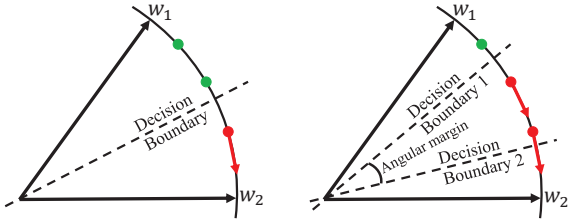

Figure 5. **From left to right**: SV-Softmax loss vs. SV-X-Softmax loss. To increase the mining range, we adopt the margin-based decision boundaries to select support vectors. Thus the non-support vectors in SV-Softmax may be support vectors in SV-X-Softmax.

losses of hard and easy examples, such that:

$$\frac{\text{loss}_{1^*}}{\text{loss}_{2^*}} > \frac{\text{loss}_1}{\text{loss}_2}. \tag{8}$$

In that way, the importance of hard examples is emphasized. This strategy is directly from the loss perspective and the definition of hard examples is ambiguous. While our SV-Softmax loss is from a different way. Firstly, we semantically define the hard examples (support vectors) according to the decision boundary. Then, to the support vector $\boldsymbol{x}_1$, we reduce its probability, such that:

$$\frac{\text{loss}_{1^*}}{\text{loss}_2} > \frac{\text{loss}_1}{\text{loss}_2}. \tag{9}$$

In summary, the differences between SV-Softmax loss and mining-based Focal loss [12] are displayed in Figure 2.

3.2.2 Relation to Margin-based Softmax Losses

Similarly, assume that we have a sample $\boldsymbol{x}$ from class 1, and it is a little far way from its ground truth class, (e.g., the red circle point in Figure 4). The original softmax loss aims to make $\boldsymbol{w}_1^T\boldsymbol{x} > \boldsymbol{w}_2^T\boldsymbol{x} \Longleftrightarrow \cos(\theta_1) > \cos(\theta_2)$. To make the objective more rigorous, margin-based losses usually introduce a margin function $f(m, \theta_1) = \cos(m_1\theta_1 + m_3) - m_2$ from the perspective of ground truth class [14, 30, 4]:

$$\cos(\theta_1) \geq f(m, \theta_1) > \cos(\theta_2). \tag{10}$$

In contrast, our SV-Softmax loss enlarge the feature margin from the perspective of other non-ground truth classes. Specifically, we have introduced a margin function $h^*(t, \theta_2)$ to these mis-classified features:

$$\cos(\theta_1) > h^*(t, \theta_2) \geq \cos(\theta_2), \tag{11}$$

where $h^*(t, \theta_2) = \log[h(t, \theta_2)e^{\cos(\theta_2)}] = t\cos(\theta_2) + t - 1$. Our SV-Softmax loss semantically enlarges the feature margin from other non-ground truth classes while margin-based losses make theirs efforts from the ground truth class. For multi-class case, Our SV-Softmax loss is class-specific margins. Figure 4 gives their geometrical comparison. To sum up, Figure 3 shows the pipeline of our SV-Softmax loss and its relations to the mining-based and margin-based losses.

3.2.3 SV-X-Softmax

According to the above discussions, our SV-Softmax loss semantically fuses the motivation of mining-based and margin-based losses into one framework, but from different viewpoints. Therefore, we can also absorb their strengths into our SV-Softmax loss. Specifically, to increase the mining range, we adopt the margin-based decision boundaries

to indicate the support vectors. Consequently, the improved SV-X-Softmax loss can be formulated as:

$$\mathcal{L}_6 = -\log \frac{e^{s f(m,\theta_{\boldsymbol{w}_y,\boldsymbol{x}})}}{e^{s f(m,\theta_{\boldsymbol{w}_y,\boldsymbol{x}})} + \sum_{k\neq y}^{K} h(t,\theta_{\boldsymbol{w}_k,\boldsymbol{x}}, I_k) e^{s\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}}, \tag{12}$$

where X is the margin-based losses. It can be A-Softmax [14], AM-Softmax [30] and Arc-Softmax [4] *etc*. The indicator mask I_k is re-computed according to margin-based decision boundaries[2]. Specifically,

$$I_k = \begin{cases} 0, & f(m,\theta_{\boldsymbol{w}_y,x}) - \cos(\theta_{\boldsymbol{w}_k,x}) \geq 0 \\ 1, & f(m,\theta_{\boldsymbol{w}_y,x}) - \cos(\theta_{\boldsymbol{w}_k,x}) < 0 \end{cases}. \tag{13}$$

Figure 5 gives the geometrical illustration of our SV-X-Softmax loss. It is best because from the motivation of margin-based losses, SV-X-Softmax loss enlarges the feature margin by integrating the self-motivation of ground truth class and the other-motivation of other classes into one framework. While from the motivation of mining-based losses, it semantically enlarges the mining range.

4. Optimazation

In this section, we show that the proposed SV-Softmax loss (6) is trainable and can be easily optimized by the typical stochastic gradient descent. The difference between the original softmax loss and the proposed SV-Softmax loss lies in the last fully connected layer $\boldsymbol{v} = [v_1, v_2, \ldots, v_K]^T = [\cos(\theta_{\boldsymbol{w}_1,\boldsymbol{x}}), \cos(\theta_{\boldsymbol{w}_2,\boldsymbol{x}}), \ldots, \cos(\theta_{\boldsymbol{w}_K,\boldsymbol{x}})]^T$.

To the forward, when $k = y$, it is the same as the original softmax loss (*i.e.*, $v_y = \cos(\theta_{\boldsymbol{w}_y,\boldsymbol{x}})$). When $k \neq y$, it has two cases, if the feature vector is easy for an specific class, it is the same as the original softmax (*i.e.*, $v_k = \cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})$). Otherwise, it will be recomputed as $\log[h(t,\theta_{\boldsymbol{w}_k,\boldsymbol{x}})e^{\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}})}] = t\cos(\theta_{\boldsymbol{w}_k,\boldsymbol{x}}) + t - 1$. To the backward propagation, we use the chain rule to compute the partial derivative. The derivative of $\boldsymbol{W}$ and the CNN feature $\boldsymbol{x}$ of the last fully connected layer should be re-emphasized:

$$\frac{\partial \mathcal{L}_5}{\partial \boldsymbol{W}} = \begin{cases} \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{w}_y} = \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\boldsymbol{x}, & k = y \\ \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{w}_k} = \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\boldsymbol{x}, & k \neq y; v_y \geq v_k \\ \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{w}_k} = t\frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\boldsymbol{x}, & k \neq y; v_y < v_k \end{cases} \tag{14}$$

$$\frac{\partial \mathcal{L}_5}{\partial \boldsymbol{x}} = \begin{cases} \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} = \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\boldsymbol{w}_y, & k = y \\ \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} = \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\boldsymbol{w}_k, & k \neq y; v_y \geq v_k \\ \frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} = t\frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}\boldsymbol{w}_k, & k \neq y; v_y < v_k \end{cases} \tag{15}$$

[2]That why we uniformity call the hard examples as "support vectors", because it is similar to the definition in [3].

Algorithm 1: SV-Softmax

Input: A CNN feature $\boldsymbol{x}$ with its corresponding label y. Initialized parameters $\boldsymbol{\Theta}$ in convolution layers. Parameter $\boldsymbol{W}$ in the last fully connected layer. The learning rate λ and the indicator parameter t. The number of iteration $\alpha \leftarrow 0$.

while *not converged* **do**

1: $\alpha \leftarrow \alpha + 1$;

2: According to the definition of hard examples (5), we compute the SV-Softmax loss by (6);

3: Compute the back-propagation error of each CNN feature $\boldsymbol{x}$ by (15) and the weight $\boldsymbol{W}$ by (14);

4: Update the parameters $\boldsymbol{W}$ and $\boldsymbol{\Theta}$ by

$\boldsymbol{W}^{(\alpha+1)} = \boldsymbol{W}^{(\alpha)} - \lambda^{(\alpha)}\frac{\partial \mathcal{L}_5}{\partial \boldsymbol{W}^{(\alpha)}}$;

$\boldsymbol{\Theta}^{(\alpha+1)} = \boldsymbol{\Theta}^{(\alpha)} - \lambda^{(\alpha)}\frac{\partial \mathcal{L}_5}{\partial \boldsymbol{x}^{(\alpha)}}\frac{\partial \boldsymbol{x}^{(\alpha)}}{\partial \boldsymbol{\Theta}^{(\alpha)}}$;

end

Output: Parameters $\boldsymbol{\Theta}$ and $\boldsymbol{W}$.

where the computation form of $\frac{\partial \mathcal{L}_5}{\partial \boldsymbol{v}}$ is the same as the original softmax loss. The whole scheme for a single image is summarized in Algorithm 1. It is trivial to perform derivation with mini-batch input. Moreover, it is also straightforward to the SV-X-Softmax loss case.

5. Experiments

5.1. Datasets

Training Data. The MS-Celeb-1M dataset [6] contains about 100k identities with 10 million images. However, it consists of a great many noisy face images. Fortunately, the trillionpairs consortium has made their efforts to get a high-quality version MS-Celeb-1M-v1c, which is well-cleaned with 86,876 identities and 3,923,399 aligned images.

Validation Data. We employ Labelled Faces in the Wild (LFW) [8] as the validation data. LFW contains 13,233 web-collected images from 5,749 different identities, with large variations in pose, expression and illuminations.

Test Data. We use two datasets, MegaFace [9] and Trillion Pairs[3], as the test data. MegaFace datasets aim at evaluating the performance of face recognition algorithms at the million scale of distractors, which include gallery set and probe set. The gallery set, a subset of Flickr photos from Yahoo, consists of more than one million images from 690,000 different individuals. The probe set has two existing databases: Facescrub [17] and FGNET [1]. In this study, we use the Facescrub as the probe set, which contains 100,000 photos of 530 unique individuals, wherein 55,742 images are males, and 52,076 images are females. Trillion Pairs datasets are recently released as a public available testing benchmark, which are consisted of the following two parts, ELFW and DELFW. ELFW is the face images of

[3]http://trillionpairs.deepglint.com/overview

	Method	LFW 6000 Pairs Accuracy	LFW BLUFR TPR@FAR=1e-3	LFW BLUFR TPR@FAR=1e-4	LFW BLUFR TPR@FAR=1e-5
Baseline	Softmax	99.26	99.46	98.44	95.24
Mining-based	F-Softmax [12]	99.46	99.62	98.76	95.97
	HM-Softmax [22]	99.26	99.48	98.48	95.11
Margin-based	A-Softmax [14]	99.36	99.68	99.09	97.20
	Arc-Softmax[4]	99.63	99.86	99.68	98.18
	AM-Softmax [30]	99.61	99.86	99.75	98.18
Naive-fused	F-Arc-Softmax	99.66	99.87	99.73	98.32
	F-AM-Softmax	99.66	99.87	99.76	98.39
	HM-Arc-Softmax	99.51	99.86	99.70	98.74
	HM-AM-Softmax	99.63	99.87	99.75	98.90
Ours	SV-Softmax	99.48	99.78	99.39	98.14
	SV-Arc-Softmax	**99.78**	99.85	99.77	98.52
	SV-AM-Softmax	99.76	**99.87**	**99.81**	**99.22**

Table 1. Verification performance (%) of different loss functions on LFW test data.

celebrities in LFW name list. There are 274,000 images from 5,700 identities. DELFW is the distractors for ELFW. There are in total 1.58 million face images from Flickr.

5.2. Experimental Settings

Data Processing. We detect the faces by adopting the FaceBoxes detector [40] and localize five landmarks (two eyes, nose tip and two mouth corners) through a simple 6-layer CNN [5]. The detected faces are cropped and resized to 120×120, and each pixel (ranged between [0,255]) in RGB images is normalized by subtracting 127.5 and then being 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 network architectures [14, 30, 29]. To be fair, the CNN architecture should be the same to test different loss functions. As suggested by the work [29], we use Attention-56 [31] as our baseline architecture to achieve a good balance between computation and accuracy. The output of Attention-56 has and finally gets a 512-dimension feature by the operation of averaging pooling. The scale parameter s has already been discussed sufficiently in previous works [30, 33]. In this paper, we directly fixed it to 30. For details, the adopted Attention-56 architecture is provided in supplementary materials.

Training. All the CNN models are trained with stochastic gradient descent (SGD) algorithm and trained from scratch, with the batch size of 32 on 4 P40 GPUs parallelly, total batch size 128. The weight decay is set to 0.0005 and the momentum is 0.9. The learning rate is initially 0.1 and divided by 10 at the 100k, 160k, 220k iterations, and we finish the training process at 240k iterations.

Test. At the testing stage, only the features of original image are employed (512-dimension) to compose the face representation. All the reported results in this paper are evaluated by a single model, without model ensemble or other fusion strategies.

To the evaluation metrics, the cosine distance of features is computed as the similarity score. Face identification and verification are conducted by ranking and thresholding the scores. Specifically, for face identification, the Cumulative Match Characteristics (CMC) curves are adopted to evaluate the Rank-1 face identification 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. We test our models on several popular public face datasets, including LFW [8], MegaFace Challenge [9, 16] and the recent Trillion Pairs Challenge. Specifically, for LFW, the unrestricted with labeled outside data on 6000 pairs accuracy [8] and the BLUFR [11] protocols are reported. For Megaface Challenge, the identification Rank-1 accuracy and the verification rate TPR@FAR =1e-6 are reported. For Trillion Pairs Challenge, every pair between ELFW and DELFW is used. There are in total 0.4 trillion pairs. To the face identification task, they provide a 1.58 million-size gallery and a 270k-size query for top-1 identification and the metric TPR@FAR=1e-3 is reported. While to the face verification task, the verification rate TPR@FAR=1e-9 is reported. For more details about the protocols, please refer to the works [8, 11, 9].

To the compared methods, we compare our method with the baseline Softmax loss (**Softmax**) and the recently proposed state-of-the-arts, including 2 mining-based softmax losses (*i.e.*, hard example mining (**HM-Softmax** [22]) and Focal loss (**F-Softmax** [12])), 3 margin-based softmax losses (the angular Softmax loss (**A-Softmax**[34]), the

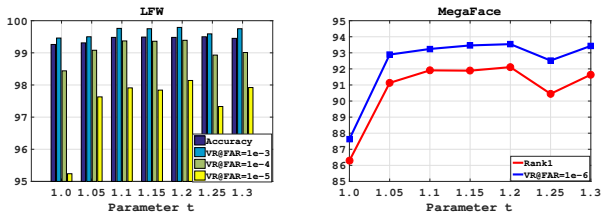

Figure 6. **From left to right**: Identification and Verification performance (%) of SV-Softmax loss with different indicator parameter t on LFW and MegaFace, respectively.

additive margin Softmax loss (**AM-Softmax**[30]), and the additive angular margin Softmax loss (**Arc-Softmax**[4])) and their 4 naive fusions (**F-AM-Softmax**, **F-Arc-Softmax**, **HM-AM-Softmax** and **HM-Arc-Softmax**). For all the compared methods, their source codes can be downloaded from the github or from authors' webpages. The corresponding parameters are determined according to their suggestions (*e.g.*, the feature margin parameter m is 0.35 for AM-Softmax and is 0.5 for Arc-Softmax). For more details, please refer to the supplementary materials.

5.3. Effects of indicator parameter t

Since the indicator parameter t plays an important role in the developed SV-Softmax loss, we first conduct experiments to search its possible best value. By varying t from 1.0 to 1.3 (If t is larger than 1.4, the model may fail to converge), we use the Attention-56 network and the SV-Softmax loss to train models on the MS-Celeb-1M-v1c dataset and evaluate its performance on the validation set LFW. As illustrated in the left sub-figure of Figure 6, with t being increased, the 6000 pairs accuracy and the BLUFR of LFW are improved consistently, and get saturated at $t = 1.2$. This demonstrates the effectiveness of our SV-Softmax loss (compared $t \neq 1.0$ with $t = 1.0$). To validate the sensitivity of our indicator parameter t, we directly use the trained models to test them on MegaFace, the effects are reported in the right sub-figure of Figure 6. From the curves, we can see that our SV-Softmax loss is insensitive to the indicator parameter t in a certain range. According to this study, t is set to fixed 1.2 in the subsequent experiments.

5.4. Experiments on LFW

Table 4 provides the quantitative results of all the competitors on LFW dataset. The bold number in each column represents the best performance. To the 6000 pairs accuracy protocol, it is well-known that this protocol is typical and easy for deep face recognition, and all the competitors can achieve over 99% accuracy rate. So the improvement of our SV-Softmax loss is not quite large. From the numbers, we observe that the naive fusions of mining-based and margin-based losses, *e.g.*, HM-AM-Softmax and F-AM-Softmax,

Method	Identification Rank1@1e6	Verification TPR@FAR=1e-6
Softmax	86.29	87.63
F-Softmax [12]	88.29	89.83
HM-Softmax [22]	86.58	88.39
A-Softmax [14]	88.54	89.40
Arc-Softmax [4]	93.67	94.47
AM-Softmax [30]	94.77	95.44
F-Arc-Softmax	93.98	95.10
F-AM-Softmax	94.47	94.84
HM-Arc-Softmax	94.05	95.26
HM-AM-Softmax	94.78	95.57
SV-Softmax	92.11	93.54
SV-Arc-Softmax	97.14	**97.57**
SV-AM-Softmax	**97.20**	97.38

Table 2. Results (%) of different losses on MegaFace Challenge.

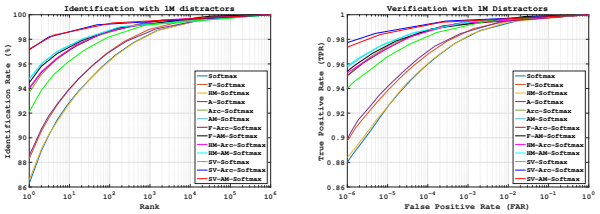

Figure 7. **Left**: CMC curves of different loss functions with 1M distractors on MegaFace [9] Set 1. **Right**: ROC curves of different loss functions with 1M distractors on MegaFace [9] Set 1.

outperform the simple mining-based or margin-based ones. Despite this, our imporved SV-AM-Softmax still achieves about 0.3% improvements. To the BLUFR protocol, the similar trends as the 6000 pairs accuracy, our improved SV-AM-Softmax loss achieves the best performance among all the competitors. Due to the evaluation protocols on LFW are nearly to be saturated, it would be better to test our models on MegaFace and Trillion Pair Challenges.

5.5. Experiments on MegaFace Challenge

Table 5 shows the identification and verification results on MegaFace dataset. In particular, compared with the baseline Softmax loss and the mining-based Softmax losses, our SV-Softmax loss achieves at least 3% improvements at both the Rank-1 identification rate and the verification TPR@FAR=1e-6 rate. The reason is that our SV-Softmax loss has clearly defined the hard examples (*i.e.*, support vectors), thus it is better than existing mining-based losses. While compared with the margin-based Softmax losses, the performance of our SV-Softmax loss is slightly lower than them. This is reasonable because the support vectors decided by the Softmax decision boundary in SV-Softmax loss may not be enough for learning discriminative fea-

Method	Identification TPR@FAR=1e-3	Verification TPR@FAR=1e-9
Softmax	36.61	33.87
F-Softmax [12]	39.80	37.14
HM-Softmax [22]	36.75	34.46
A-Softmax [14]	43.89	43.76
Arc-Softmax [4]	57.48	57.45
AM-Softmax [30]	61.80	61.61
F-Arc-Softmax	56.80	56.87
F-AM-Softmax	61.85	61.79
HM-Arc-Softmax	55.93	56.63
HM-AM-Softmax	61.42	61.33
SV-Softmax	51.18	46.78
SV-Arc-Softmax	71.19	70.33
SV-AM-Softmax	**73.56**	**72.71**

Table 3. Performance (%) of different loss functions on Trillion Pairs Challenge.

tures. Our improved versions SV-Arc-Softmax and SV-AM-Softmax losses, wherein the support vectors are determined by the margin-based decision boundaries, can further boost the performance because they absorb the complementary merits of margin-based losses. Specifically, to our SV-AM-Softmax loss, it beats the best margin-based competitor AM-Softmax loss by a large margin (about 2.4% at Rank-1 identification rate and 1.9% verification rate). Compared with the naive fusions of mining-based and margin-based losses, our improved SV-AM-Softmax loss is also better than them. It is about 2.4% higher at Rank-1 identification rate and 1.8% higher at verification rate than the second best competitor HM-AM-Softmax loss. To sum up, our imporved SV-X-Softmax losses, which eliminate the ambiguity of hard examples as well as absorb the discriminative power of other classes by focusing on support vectors, are inherently the best in the current stage. In Figure 7, 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. In this experiment, our SV-Softmax loss with its improved version SV-AM-Softmax approach have shown their superiority for both the identification and verification tasks.

5.6. Experiments on Trillion Pairs Challenge

Table 3 displays the performance comparison on the recent Trillion Pairs Challenge, from which, we can conclude that the results exhibit the same trends that emerged on LFW and MegaFace datasets. Besides, the trends are more obvious. Concretely, both of the current mining-based and margin-based losses are better than the simple softmax loss for face recognition. However, the margin-based losses usually achieve higher performance than the mining-based losses, because the motivation of margin-based losses is to enhance the feature discrimination while the motivation of mining-based losses is to focus training on hard examples. Their naive fusions can slightly improve the performance further. However, the naive fusions are still suffering from the ambiguity of hard examples and the lack of discriminative power of other classes. Therefore, they are limited for face recognition. Our SV-X-Softmax (*e.g.*, SV-AM-Softmax) losses absorb the strengths and discard the drawbacks of the current ming-based and margin-based loss functions, thus they achieve the highest performance.

LFW 6000 Accuracy	LFW BLUFR TPR @FAR=1e-3	LFW BLUFR TPR @FAR=1e-4	LFW BLUFR TPR @FAR=1e-5
99.85 (our)	99.92	99.89	99.13
99.87 (1st)	- -	- -	- -

Table 4. Performance (%) of SV-AM-Softmax loss on LFW.

MegaFace Identification Rank-1@1e6	MegaFace Verification TPR@FAR=1e-6
98.82 (our)	99.03 (our)
99.93 (1st)	**99.93** (1st)

Table 5. Performance (%) of SV-AM-Softmax loss on MegaFace.

Trillion Pairs Identification TPR@FAR=1e-3	Trillion Pairs Verification TPR@FAR=1e-9
82.25 (our)	78.49 (our)
85.67 (1st)	**82.29** (1st)

Table 6. Performance (%) of SV-AM-Softmax loss on Trillion Pairs.

6. Improvement by Designing Architectures

To further boost the performance, we try to make the adopted Attention-56 [31] architecture deeper. Specifically, we change the stages of [1,1,1] used in Attention-56 into [3,6,2]. Moreover, inspired by [4], we incorporate the IRSE module into the architecture. The results are displayed in Tables 4-6. Note that all current results are training based on the simple MS-Celeb-1Mv1c dataset and only the single model performance is reported. From the numbers, we can see that our SV-AM-Softmax loss has achieved the competitive absolute performance. In the future, it would be better to fuse the MS1M-ArcFace [4] and Asian datasets[4] and design model ensemble methods (*e.g.*, feature concatenation).

7. Conclusion

This paper has proposed a simple but very effective loss function, namely support vector guided softmax loss

[4] http://trillionpairs.deepglint.com/data

(*i.e.*, SV-Softmax), for face recognition. In specific, SV-Softmax loss explicitly concentrates on optimizing the support vectors. Thus it semantically integrates the motivation of mining-based and margin-based loss functions into one framework. Consequently, it is intrinsically better than the current mining-based losses, margin-based losses and their naive fusions. Extensive experiments on several benchmark datasets have clearly demonstrated the advantages of our new approach over the state-of-the-art alternatives.

References

[1] Fg-net aging database. `http://www.fgnet.rsunit.com/`. 2010.

[2] S. Cai, W. Zuo, L. Zhang, X. Feng, and P. Wang. Support vector guided dictionary learning. In *ECCV*, 2014.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3), 1995.

[4] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *arXiv preprint arXiv:1801.07698*, 2018.

[5] Z.-H. Feng, J. Kittler, M. Awais, P. Huber, and X.-J. Wu. Wing loss for robust facial landmark localisation with convolutional neural networks. *arXiv preprint arXiv:1711.06753*, 2017.

[6] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *ECCV*, 2016.

[7] K. He, X. Zhang, and S. Ren. Deep residual learning for image recognition. In *CVPR*, 2016.

[8] G. Huang, M. Ramesh, T. Berg, and E. Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained enviroments. In *Technical Report*, 2007.

[9] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *CVPR*, 2016.

[10] X. Liang, X. Wang, Z. Lei, S. Liao, and S. Li. Soft-margin softmax for deep classification. In *ICONIP*, 2017.

[11] S. Liao, Z. Lei, D. Yi, and S. Z. Li. A benchmark study of large-scale unconstrained face recognition. In *ICB*, 2014.

[12] Y. Lin, P. Goyal, and R. Girshick. Focal loss for dense object detection. In *ICCV*, 2017.

[13] W. Liu, Y. Wen, and Z. Yu. Large-margin softmax loss for convolutional neural networks. In *ICML*, 2016.

[14] W. Liu, Y. Wen, Z. Yu, M. Li, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[15] Y. Liu, H. Li, and X. Wang. Learning deep features via congenerous cosine loss for person recognition. In *ICCV*, 2017.

[16] A. Nech and I. Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In *CVPR*, 2017.

[17] H.-W. Ng and S. Winkler. A data-driven approach to cleaning large face datasets. In *ICIP*, 2014.

[18] O. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015.

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

[20] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[21] H. Shi, X. Wang, D. Yi, Z. Lei, X. Zhu, and S. Z. Li. Cross-modality face recognition via heterogeneous joint bayesian. *IEEE Signal Processing Letters*, 24(1):81–85, 2017.

[22] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016.

[23] K. Simonyan and Z. Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[24] H. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[25] Y. Sun, Y. Chen, and X. Wang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014.

[26] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014.

[27] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *CVPR*, 2015.

[28] Y. Taigman, M. Yang, and M. Ranzato. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[29] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy. The devil of face recognition is in the noise. In *ECCV*, 2018.

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

[31] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. *arXiv preprint arXiv:1704.06904*, 2017.

[32] F. Wang, X. Xiang, J. Chen, and A. Yuille. Normface: l_2 hypersphere embedding for face verification.. In *ACM MM*, 2017.

[33] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. *arXiv preprint arXiv:1801.09414*, 2018.

[34] J. Wang, F. Zhou, and S. Wen. Deep metric learning with angular loss. In *ICCV*, 2017.

[35] X. Wang, X. Guo, and S. Z. Li. Adaptively unified semi-supervised dictionary learning with active points. In *ICCV*, 2015.

[36] X. Wang, S. Zhang, Z. Lei, S. Liu, X. Guo, and S. Z. Li. Ensemble soft-margin softmax loss for image classification. *arXiv preprint arXiv:1805.03922*, 2018.

[37] Y. Wen, K. Zhang, and Z. Li. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[38] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 2009.

[39] Y. Yuan, K. Yang, and C. Zhang. Hard-aware deeply cascaded embedding. In *ICCV*, 2017.

[40] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. Faceboxes: A cpu real-time face detector with high accuracy. In *IJCB*, 2017.

[41] Y. Zheng, D. K. Pal, and M. Savvides. Ring loss: Convex feature normalization for face recognition. In *CVPR*, 2018.