# BoundaryFace: A mining framework with noise label self-correction for Face Recognition

Shijie Wu and Xun Gong✉

School of Computing and Artificial Intelligence, Southwest Jiaotong University,
Chengdu, Sichuan, China
xgong@swjtu.edu.cn

**Abstract.** Face recognition has made tremendous progress in recent years due to the advances in loss functions and the explosive growth in training sets size. A properly designed loss is seen as key to extract discriminative features for classification. Several margin-based losses have been proposed as alternatives of softmax loss in face recognition. However, two issues remain to consider: 1) They overlook the importance of hard sample mining for discriminative learning. 2) Label noise ubiquitously exists in large-scale datasets, which can seriously damage the model's performance. In this paper, starting from the perspective of decision boundary, we propose a novel mining framework that focuses on the relationship between a sample's ground truth class center and its nearest negative class center. Specifically, a closed-set noise label self-correction module is put forward, making this framework work well on datasets containing a lot of label noise. The proposed method consistently outperforms SOTA methods in various face recognition benchmarks. Training code has been released at `https://gitee.com/swjtugx/classmate/tree/master/OurGroup/BoundaryFace`.

**Keywords:** Face Recognition, Noise Label, Hard Sample Mining, Decision Boundary

## 1 Introduction

Face recognition is one of the most widely studied topics in the computer vision community. Large-scale datasets, network architectures, and loss functions have fueled the success of Deep Convolutional Neural Networks (DCNNs) on face recognition. Particularly, with an aim to extract discriminative features, the latest works have proposed some intuitively reasonable loss functions.

For face recognition, the current existing losses can be divided into two approaches: one deems the face recognition task to be a general classification problem, and networks are therefore trained using softmax [12,25,26,24,3,1,5,16,31]; the other approaches the problem using metric learning and directly learns an embedding, such as [21,17,20]. Since metric learning loss usually suffers from sample batch combination explosion and semi-hard sample mining, the second problem needs to be addressed by more sophisticated sampling strategies. Loss functions have therefore attracted increased attention.

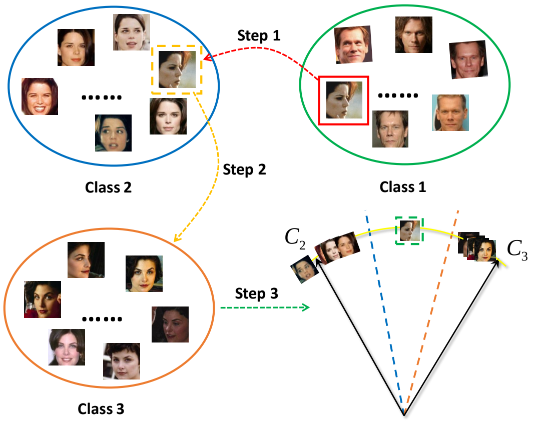

**Fig. 1.** The motivation of BoundaryFace. Step 1 denotes closed-set noise label self-correction. Step 2 denotes nearest negative class match. Step 3 denotes hard sample handle. For a noisy hard sample, we first correct its label, then match the nearest negative class based on the correct label, and finally emphasize it using the decision boundary consisting of this sample's ground truth class center and the nearest negative class center.

It has been pointed out that the classical classification loss function (*i.e.*, Softmax loss) cannot obtain discriminative features. Based on current testing protocols, the probe commonly has no overlap with the training images, so it is particularly crucial to extract features with high discriminative ability. To this end, Center loss [29] and NormFace [25] have been successively proposed to obtain discriminative features. Wen *et al*. [29] developed a center loss that learns each subject's center. To ensure the training process is consistent with testing, Wang *et al*. [25] made the features extracted by the network and the weight vectors of the last fully connected layer lay on the unit hypersphere. Recently, some margin-based softmax loss functions [12,26,24,3,13] have also been proposed to enhance intra-class compactness while enlarging inter-class discrepancy, resulting in more discriminative features.

The above approaches have achieved relatively satisfactory results. However, there are two very significant issues that must still be addressed: 1) Previous research has ignored the importance of hard sample mining for discriminative learning. As illustrated in [8,2], hard sample mining is a crucial step in improving performance. Therefore, some mining-based softmax losses have emerged. Very recently, MV-Arc-Softmax [28], and CurricularFace [9] were proposed. They were inspired by integrating both margin and mining into one framework. However, both consider the relationship between the sample ground truth class and all negative classes, which may complicate the optimization of the decision boundary. 2) Both margin-based softmax loss and mining-based softmax loss ignore the influence of label noise. Noise in face recognition datasets is composed of two types: *closed-set noise*, in which some samples are falsely given the labels of other identities within the same dataset, and *open-set noise*, in which a subset of samples that do not belong to any of the classes, are mistakenly assigned one of their labels, or contain some non-faces. Wang *et al.* [23] noted that noise, especially closed-set noise, can seriously impact model's performance. Unfortunately, removing noise is expensive and, in many cases, impracticable. Intuitively, the mining-based softmax loss functions can negatively impact the model if the training set is noisy. That is, mining-based softmax is likely to perform less well than baseline methods on datasets with severe noise problems. Designing a loss function that can perform hard sample mining and tolerate noise simultaneously is still an open problem.

In this paper, starting from the perspective of decision boundary, we propose a novel mining framework with tolerating closed-set noise. Fig. 1 illustrates our motivation using a noisy hard sample processing. Specifically, based on the premise of closed-set noise label correction, the framework directly emphasizes hard sample features that are between the ground truth class center and the nearest negative class center. We find out that if a sample is a closed-set noise, there is a high probability that the sample is distributed within the nearest negative class's decision boundary, and the nearest negative class is likely to be the ground truth class of the noisy sample. Based on this finding, we propose a module that automatically discovers closed-set noise during training and dynamically corrects its labels. Based on this module, the mining framework can work well on large-scale datasets under the impact of severe noise. To sum up, the contributions of this work are:

- We propose a novel mining framework with noise label self-correction, named BoundaryFace, to explicitly perform hard sample mining as a guidance of the discriminative feature learning.
- The closed-set noise module can be used in any of the existing margin-based softmax losses with negligible computational overhead. To the best of our knowledge, this is the first solution for closed-set noise from the perspective of the decision boundary.
- We have conducted extensive experiments on popular benchmarks, which have verified the superiority of our BoundaryFace over the baseline softmax and the mining-based softmax losses.

## 2 Related Work

### 2.1 Margin-based softmax

Most recently, researchers have mainly focused on designing loss functions in the field of face recognition. Since basic softmax loss cannot guarantee facial features that are sufficiently discriminative, some margin-based softmax losses [13,12,26,24,3,33], aiming at enhancing intra-class compactness while enlarging inter-class discrepancy, have been proposed. Liu *et al.* [13] brought in multiplicative margin to face recognition in order to produce discriminative feature. Liu *et al.* [12] introduced an angular margin (A-Softmax) between ground truth class and other classes to encourage larger inter-class discrepancy. Since multiplicative margin could encounter optimization problems, Wang *et al.* [26] proposed an additive margin to stabilize optimization procedure. Deng *et al.* [3] changed the form of the additive margin, which generated a loss with clear geometric significance. Zhang *et al.* [33] studied on the effect of two crucial hyper-parameters of traditional margin-based softmax losses, and proposed the AdaCos, by analyzing how they modulated the predicted clasification probability. Even these margin-based softmax losses have achieved relatively good performance, none of them takes into account the effects of hard sample mining and label noise.

### 2.2 Mining-based softmax

There are two well-known hard sample mining methods, *i.e.*, Focal loss [11], Online Hard Sample Mining (OHEM) [19]. Wang *et al.* [28] has shown that naive combining them to current popular face recognition methods has limited improvement. Some recent work, MV-Arc-Softmax [28], and CurricularFace [9] are inspired by integrating both margin and mining into one framework. MV-Arc-Softmax explicitly defines mis-classified samples as hard samples and adaptively strengthens them by increasing the weights of corresponding negative cosine similarities, eventually producing a larger feature margin between the ground truth class and the corresponding negative target class. CurricularFace applies curriculum learning to face recognition, focusing on easy samples in the early stage and hard samples in the later stage. However, on the one hand, both take the relationship between the sample ground truth class and all negative classes into consideration, which may complicate the optimization of the decision boundary; on the other hand, label noise poses some adverse effect on mining. It is well known that the success of face recognition nowadays benefits from large-scale training data. Noise is inevitably in these million-scale datasets. Unfortunately, Building a “clean enough” face dataset, however, is both costly and difficult. Both MV-Arc-Softmax and CurricularFace assume that the dataset is clean (*i.e.*, almost noiseless), but this assumption is not true in many cases. Intuitively, the more noise the dataset contains, the worse performance of the mining-based softmax loss will be. Unlike open-set noise, closed-set noise can be part of the clean data as soon as we correct their labels. Overall, our method differs from the currently popular mining-based softmax in that our method can conduct hard sample mining along with the closed-set noise well being handled, while the current methods cannot do so.

## 3 The Proposed Approach

### 3.1 Preliminary Knowledge

**Margin-based softmax.** The original softmax loss formula is as follows:

$$L = -\log \frac{e^{W_{y_i} x_i + b_{y_i}}}{\sum\limits_{j=1}^{n} e^{W_j x_i + b_j}} \tag{1}$$

where $x_i$ denotes the feature of the $i$-th sample belonging to $y_i$ class in the min-batch, $W_j$ denotes the $j$-th column of the weight matrix $W$ of the last fully connected layer, and $b_j$ and $n$ denote the bias term and the number of identities, respectively.

To make the training process of the face recognition consistent with the testing, Wang *et al.* [25] let the weight vector $W_j$ and the sample features $x_i$ lie on a hypersphere by $l_2$ normalization. And to make the networks converge better, the sample features are re-scaled to $s$. Thus, Eq. 1 can be modified as follows:

$$L = -\log \frac{e^{s(\cos \theta_{y_i})}}{\sum\limits_{j=1}^{n} e^{s(\cos \theta_j)}} \tag{2}$$

With the above modification, $W_j$ has a clear geometric meaning which is the class center of $j$-th class and we can even consider it as a feature of the central sample of $j$-th class. $\theta_{\mathrm{y}_i}$ can be seen as the angle between the sample and its class center, in particular, it is also the geodesic distance between the sample and its class center from the unit hypersphere perspective. However, as mentioned before, the original softmax does not yield discriminative features, and the aforementioned corrections (*i.e.*, Eq. 2) to softmax do not fundamentally fix this problem, which has been addressed by some variants of softmax based on margin. They can be formulated in a uniform way:

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

E.g, in baseline softmax (*e.g.*, ArcFace), $f(\cos \theta_{y_i}) = \cos(\theta_{y_i} + m)$. As can be seen, the currently popular margin-based softmax losses all achieve intra-class compactness and inter-class discrepancy by squeezing the distance between a sample and its ground truth class center.

**Mining-based softmax.** Hard sample mining is to get the network to extra focus valuable, hard-to-learn samples. There are two main categories in the existing mining methods that are suitable for face recognition: 1) focusing on samples with large loss values from the perspective of loss. 2) focusing on samples misclassified by the network from the relationship between sample ground truth class and negative classes. They can be formed by a unified formula as below:

$$L = -I(p(x_i)) \log \frac{e^{sf(\cos \theta_{y_i})}}{e^{sf(\cos \theta_{y_i})} + \sum\limits_{j=1, j \neq y_i}^{n} e^{sg(t, \cos \theta_j)}} \tag{4}$$

where $p(x_i) = \frac{e^{sf(\cos \theta_{y_i})}}{e^{sf(\cos \theta_{y_i})} + \sum\limits_{j=1, j \neq y_i}^{n} e^{sg(t, \cos \theta_j)}}$ is the predicted ground truth probability and $I(p(x_i))$ is an indicator function. For type 1, such as Focal loss,

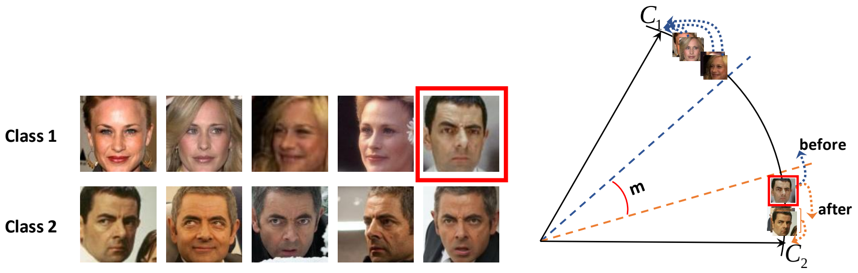

**Fig. 2. Left:** Each row represents one person. The red box includes a closed-set noise sample. **Right:** The distribution of the samples in the left figure is shown from the perspective of the decision boundary. The dashed arrows represent the optimized direction of the samples.

$I(p(x_i)) = (1 - p(x_i))^\lambda$, $f(\cos\theta_{y_i}) = \cos\theta_{y_i}$ and $g(t, \cos\theta_j) = \cos\theta_j$, $\lambda$ is a modulating factor. For type 2, MV-Arc-Softmax and CurricularFace handle hard samples with varying $g(t, \cos\theta_j)$. Let $N = f(\cos\theta_{y_i}) - \cos\theta_j$, thus, MV-Arc-Softmax is described as:

$$g(t, \cos\theta_j) = \begin{cases} \cos\theta_j, & N \geq 0 \\ \cos\theta_j + t, & N < 0 \end{cases} \tag{5}$$

and CurricularFace formula is defined as follows:

$$g(t, \cos\theta_j) = \begin{cases} \cos\theta_j, & N \geq 0 \\ \cos\theta_j(t + \cos\theta_j), & N < 0 \end{cases} \tag{6}$$

From the above formula, we can see that if a sample is a easy sample, then its negative cosine similarity will not change. Otherwise, its negative cosine similarity will be amplified. Specially, in MV-Arc-Softmax, $\cos\theta_j + t > \cos\theta_j$ always holds true since $t$ is a fixed hyper parameter and is always greater than 0. That is, the model always focuses on hard samples. In contrast, $t$ is calculated based on the Exponential Moving Average (EMA) in CurricularFace, which is gradually changing along with iterations. Moreover, the $\cos\theta_j$ can reflect the difficulty of the samples, and these two changes allow the network to learn easy samples in the early stage and hard samples in the later stage.

### 3.2 Label self-correction

In this section, we discuss the mining framework's noise tolerance module. Unlike open-set noise, a closed-set noise sample is transformed into a clean sample if its label can be corrected appropriately. The existing mining-based softmax losses, as their prerequisite, assume that the training set is a clean dataset. Suppose the labels of most closed-set noise in a real noisy dataset are corrected; in that case, poor results from hard sample mining methods on noisy datasets can be

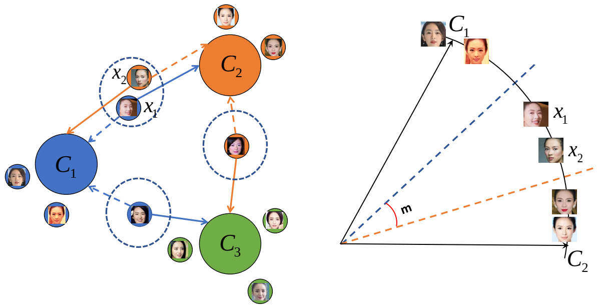

**Fig. 3. Left:** Blue, orange, and green represent three individuals. The samples in the ellipse are the hard samples. The solid arrows represent distance maximization, and the dashed arrows represent distance minimization. **Right:** The sample distribution of two persons in the left from the perspective of decision boundary.

adequately mitigated. More specifically, we find that when trained moderately, networks have an essential ability for classification; and that closed-set noise is likely to be distributed within the nearest negative class's decision boundary. Additionally, the negative class has a high probability of being the ground truth class of this sample. As shown in Fig. 2, the red box includes a closed-set noise sample, which is labeled as class 1, but the ground truth label is class 2. The closed-set noise will be distributed within the decision boundary of class 2. At the same time, we dynamically change that sample's label so that the sample is optimized in the correct direction. That is, before the label of this closed-set noise is corrected, it is optimized in the direction of $C_1$ (*i.e.*, the ”before” arrow); after correction, it is optimized in the direction of $C_2$ (*i.e.*, the ”after” arrow). The label self-correction formula, named BoundaryF1, is defined as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)}+\sum\limits_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}} \tag{7}$$

where $if\ \max\{\cos(\theta_k+m)\ for\ all\ k\neq y_i\}-\cos\theta_{y_i}>0 : y_i=k$. It means that, before each computing of Eq. 7, we decide whether to correct the label based on whether the sample is distributed within the decision boundary of the nearest negative class. Noted that, as a demonstration, apply this method to ArcFace. It can be applied to other margin-based losses could also be used.

### 3.3 BoundaryFace

Unlike the mining-based softmax's semantic for assigning hard samples, we only consider samples located in the margin region between the ground truth class and the nearest negative class. In other words, as each sample in the high-dimensional feature space has a nearest negative class center, if the sample feature is in the margin region between its ground truth class center and the nearest negative class center, then we label it as a hard sample. As shown in Fig. 3 (left), the nearest negative class for each class's sample may be different (*e.g.*, the nearest negative classes of two samples belonging to the $C_1$ class are $C_2$ and $C_3$, respectively). In Fig. 3, the right image presents the two classes of the left subfigure from the perspective of the decision boundary. Since samples $x_1$ and $x_2$ are in the margin region between their ground truth class and the nearest negative class, we treat them as hard samples. An additional regularization term $f$ is added to allow the network to strengthen them directly. Additionally, to ensure its effectiveness on noisy datasets, we embed the closed-set noise label correction module into the mining framework. As shown in Fig. 4, after the network has obtained discriminative power, for each forward propagation; and based on the normalization of feature $x_i$ and the weight matrix $W$, we obtain the cosine similarity $\cos\theta_j$ of sample feature $x_i$ to each class center $W_j$. Next, we calculate the position of feature $x_i$ at the decision boundary based on $\cos\theta_j$. Assuming that the nearest negative class of the sample is $y_n$. If it is distributed within the nearest negative class's decision boundary, we dynamically correct its label $y_i$ to $y_n$; otherwise proceed to the next step. We then calculate $\cos(\theta_{y_i}+m)$. After that, we simultaneously calculate two lines: one is the traditional pipeline; and the other primarily determines whether the sample is hard or not. These two lines contribute to each of the final loss function's two parts. Since our idea is based on the perspective of the decision boundary, we named our approach BoundaryFace. The final loss function is defined as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(\log\frac{e^{sT(\cos\theta_{y_i})}}{e^{sT(\cos(\theta_{y_i}))} + \sum\limits_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} - \lambda f) \tag{8}$$

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

$$f = \max\{0, \max\{\cos\theta_j|\ for\ \text{all}\ j\neq y_i\} - T(\cos\theta_{y_i})\}$$

where $\ if\ \ \max\{\cos(\theta_k+m)\ for\ all\ k\neq y_i\} - \cos\theta_{y_i} > 0 : y_i = k$. $\lambda$ is a balance factor. As with BoundaryF1, before each computing of final loss, we decide whether to correct the label based on whether the sample is distributed within the decision boundary of the nearest negative class.

**Optimization** In this part, we show that out BoundaryFace is trainable and can be easily optimized by the classical stochastic gradient descent (SGD). Assuming $x_i$ denotes the deep feature of $i$-th sample which belongs to the $y_i$ class, $L_1 = -\log(\frac{e^{f_{y_i}}}{\sum\limits_k e^{f_k}})$, $L_2 = \lambda\max\{0,\max\{\cos\theta_k|\ for\ \text{all}\ k\neq y_i\} - \cos(\theta_{y_i}+m)\}$, the input of the $L_1$ is the logit $f_k$, where $k$ denotes the $k$-th class.

In the forward propagation, when $k = y_i$, $f_k = s\cos(\theta_{y_i}+m)$, when $k\neq y_i$, $f_k = s\cos(\theta_k)$. Regardless of the relationship of $k$ and $y_i$, there are two cases for $L_2$, if $x_i$ is a easy sample, $L_2 = 0$. Otherwise, it will be constituted as $L_2 = \lambda(\max\{\cos\theta_k|\ for\ \text{all}\ k\neq y_i\} - \cos(\theta_{y_i}+m))$. In the backward propagation process, the gradients w.r.t. $x_i$ and $w_k$ can be computed as follows:

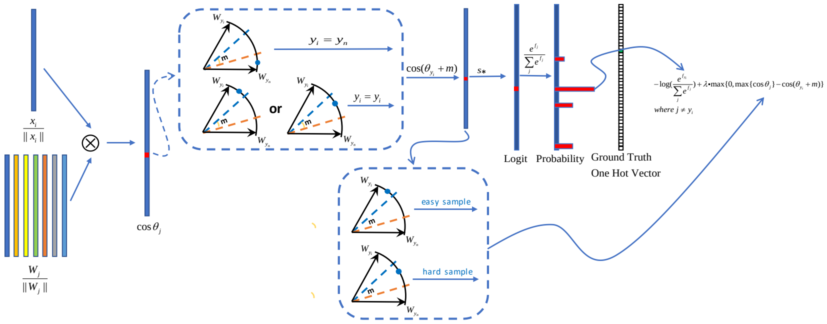

**Fig. 4.** Overview of BoundaryFace. The part included in the upper dashed box represents the closed-set noise processing. The part included in the lower dashed box represents the judgment of whether it is a hard sample or not.

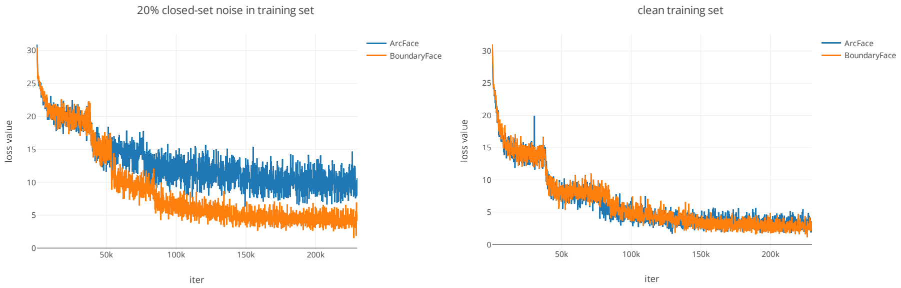

**Fig. 5.** Convergence process of BoundaryFace.

$when\ k = y_i$

$$\frac{\partial L}{\partial x_i} = \begin{cases} \frac{\partial L_1}{\partial f_{y_i}} (s \frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}}) \frac{w_{y_i}}{||w_{y_i}||} X, & easy \\ (\frac{\partial L_1}{\partial f_{y_i}} s + \frac{\partial L_2}{\partial \cos(\theta_{y_i}+m)}) \frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}} \frac{w_{y_i}}{||w_{y_i}||} X, & hard \end{cases} \tag{9}$$

$$\frac{\partial L}{\partial w_k} = \begin{cases} \frac{\partial L_1}{\partial f_{y_i}} (s \frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}}) \frac{x_i}{||x_i||} W, & easy \\ (\frac{\partial L_1}{\partial f_{y_i}} s + \frac{\partial L_2}{\partial \cos(\theta_{y_i}+m)}) \frac{\sin(\theta_{y_i}+m)}{\sin\theta_{y_i}} \frac{x_i}{||x_i||} W, & hard \end{cases} \tag{10}$$

$when\ k \neq y_i$

$$\frac{\partial L}{\partial x_i} = \begin{cases} \frac{\partial L_1}{\partial f_k} s \frac{w_k}{||w_k||} X, & easy \\ (\frac{\partial L_1}{\partial f_k} s + \frac{\partial L_2}{\partial \cos\theta_k}) \frac{w_k}{||w_k||} X, & hard \end{cases} \tag{11}$$

$$\frac{\partial L}{\partial w_k} = \begin{cases} \frac{\partial L_1}{\partial f_k} s \frac{x_i}{||x_i||} W, & easy \\ (\frac{\partial L_1}{\partial f_k} s + \frac{\partial L_2}{\partial \cos\theta_k}) \frac{x_i}{||x_i||} W, & hard \end{cases} \tag{12}$$

**Algorithm 1:** BoundaryFace

```
Input  : The feature of $i$-th sample $x_i$ with its label $y_i$, last fully-connected
         layer parameters $W$, cosine similarity $\cos\theta_j$ of two vectors,
         embedding network parameters $\Theta$, and margin $m$
iteration number $k \leftarrow 0$, parameter $m \leftarrow 0.5$, $\lambda \leftarrow \pi$;
while not converged do
    for all $j \neq y_i$,
    if $\max\{\cos(\theta_j + m)\} > \cos\theta_{y_i}$ then
        $y_i = j$;
    else
        $y_i = y_i$;
    end
    if $\cos(\theta_{y_i} + m) > \max\{\cos\theta_j\}$ then
        $f = 0$;
    else
        $f = \max\{\cos\theta_j\} - \cos(\theta_{y_i} + m)$;
    end
    Compute the loss $L$ by Eq. 3.3;
    Compute the gradients of $x_i$ and $W_j$ by Eq. 9,10,11,12;
    Update the parameters $W$ and $\Theta$;
    $k \leftarrow k + 1$;
end
Output: $W$, $\Theta$
```

where, both $X$ and $W$ are symmetric matrices.

Further, in Fig. 5, we give the loss curves of baseline and BoundaryFace on the clean dataset and the dataset containing 20% closed-set noise, respectively. It can be seen that our method converges faster than baseline. The training procedure is summarized in Algorithm 1.

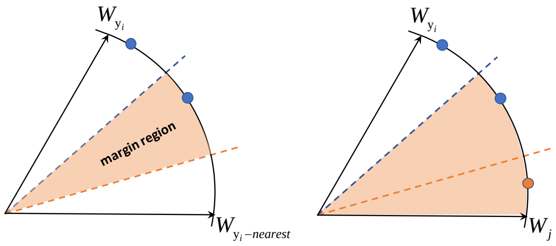

**Fig. 6.** The difference between BoundaryFace and SOTA methods in emphasizing the region where the hard samples are located.

### 3.4 Discussions with SOTA Loss Functions

**Comparison with baseline softmax.** The baseline softmax (*i.e.*, ArcFace, CosFace) introduces a margin from the perspective of positive cosine similarity, and they treat all samples equally. Our approach mining hard sample by introducing a regularization term to make the network pay extra attention to the hard samples.

**Comparison with MV-Arc-Softmax and CurricularFace.** MV-Arc-Softmax and CurricularFace assign the same semantics to hard samples, only differing in the following handling stage. As shown in Fig. 6 (right), both treat mis-classified samples as hard samples and focus on the relationship between the ground truth class of the sample and all negative classes. Instead, in Fig. 6 (left), our approach regards these samples located in the margin region as hard samples and focuses only on the relationship between the ground truth class of the sample and the nearest negative class. Moreover, the labels of both blue and orange samples are $y_i$, but the orange sample ground truth class is $j$-th class. Obviously, if there is closed-set noise in the dataset, SOTA methods not only emphasize hard samples but also reinforce closed-set noise.

## 4 Experiments

### 4.1 Implementation Details

**Datasets.** CASIA-WebFace [30] which contains about 0.5M of 10K individuals, is the training set that is widely used for face recognition, and since it has been cleaned very well, we take it as a clean dataset. In order to simulate the situation where the dataset contains much of noise, based on the CASIA-WebFace, we artificially synthesize noisy datasets which contain a different ratio of noise. In detail, for closed-set noise, we randomly flip the sample labels of CASIA-WebFace; for open-set noise, we choose MegaFace [10] as our open-set noise source and randomly replace the samples of CASIA-WebFace. Finally, we use the clean CASIA-WebFace and noisy synthetic datasets as our training set, respectively. We extensively test our method on several popular benchmarks, including LFW [7], AgeDB [14], CFP-FP [18], CALFW [35], CPLFW [34], SLLFW [4], RFW [27]. RFW consists of four subsets: Asian, Caucasian, Indian, and African. Note that in the tables that follow, CA denotes CALFW, CP denotes CPLFW, and Cau denotes Caucasian.

**Training Setting.** We follow [3] to crop the 112×112 faces with five landmarks [32] [22]. For a fair comparison, all methods should be the same to test different loss functions. To achieve a good balance between computation and accuracy, we use the ResNet50 [6] as the backbone. The output of backbone gets a 512-dimension feature. Our framework is implemented in Pytorch [15]. We train modules on 1 NVIDIA TitanX GPU with batch size of 64. The models are trained with SGD algorithm, with momentum 0.9 and weight decay $5e$ -4. The

**Table 1.** Verification performance (%) of our BoundaryFace with different hyper-parameter $\lambda$.

| Method | SLLFW | CFP-FP |
|---|---|---|
| $\lambda = 2$ | 98.05 | 94.8 |
| $\lambda = 2.5$ | 97.9 | 94.71 |
| $\lambda = \pi$ | **98.12** | **95.03** |
| $\lambda = 3.5$ | 97.8 | 94.9 |

**Table 2.** Verification performance (%) of ArcFace with different hyper-parameter $m$ on datasets which contain different noise mixing ratios (%).

| $m$ | closed-set ratio | open-set ratio | LFW | AgeDB |
|---|---|---|---|---|
| 0.3 | 10% | 30% | **99.07** | **91.82** |
| 0.5 | 10% | 30% | 98.22 | 89.02 |
| 0.3 | 30% | 10% | 98.42 | 88.5 |
| 0.5 | 30% | 10% | **98.73** | **89.93** |

learning rate starts from 0.1 and is divided by 10 at 6, 12, 19 epochs. The training process is finished at 30 epochs. We set scale $s = 32$ and margin $m = 0.3$ or $m = 0.5$. Moreover, to make the network with sufficient discrimination ability, we first pre-train the network for 7 epochs using margin-based loss. The margin-based loss can also be seen as a degenerate version of our BoundaryFace.

### 4.2 Hyper-parameters

**Parameter $\lambda$.** Since the hyper-parameter $\lambda$ plays an essential role in the proposed BoundaryFace, we mainly explore its possible best value in this section. In Tab. 1, we list the performance of our proposed BoundaryFace with $\lambda$ varies in the range [2, 3.5]. We can see that our BoundaryFace is insensitive to the hyper-parameter $\lambda$. And, according to this study, we empirically set $\lambda = \pi$.

**Parameter $m$.** Margin $m$ is essential in both margin-based softmax loss and mining-based softmax loss. For clean datasets, we follow [3] to set margin $m = 0.5$. In Tab. 2, we list the performance of different $m$ for ArcFace on datasets with different noise ratios. It can be concluded that if most of the noise in the training set are open-set noise, we set $m = 0.3$; otherwise, we set $m = 0.5$.

### 4.3 Comparisons with SOTA Methods

**Results on a dataset that is clean or contains only closed-set noise.** In this section, we first train our BoundaryFace on the clean dataset as well as datasets containing only closed-set noise. We use BoundaryF1 (Eq. 7) as a reference to illustrate the effects of hard sample mining. Tab. 3 provides the quantitative results. It can be seen that our BoundaryFace outperforms the baseline and achieves comparable results when compared to the SOTA competitors

**Table 3.** Verification performance (%) of different loss functions when the training set contains different ratios of closed-set noise. Ratio 0% means that the training set is the original CASIA-WebFace (*i.e.*, clean dataset).

| ratio | Method | LFW | AgeDB | CFP | CA | CP | SLLFW | Asian | Cau | Indian | African |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0% | ArcFace | 99.38 | 94.05 | 94.61 | 93.43 | 89.45 | 97.78 | 86.5 | 93.38 | 89.9 | 86.72 |
| | MV-Arc-Softmax | 99.4 | 94.17 | 94.96 | 93.38 | 89.48 | 97.88 | 86.23 | 93.27 | 90.12 | 87.03 |
| | CurricularFace | 99.42 | 94.37 | 94.94 | **93.52** | 89.7 | 98.08 | 86.43 | **94.05** | **90.55** | **88.07** |
| | BoundaryF1 | 99.41 | 94.05 | 95.01 | 93.27 | **89.8** | 97.75 | 85.72 | 92.98 | 89.98 | 86.43 |
| | **BoundaryFace** | **99.42** | **94.4** | **95.03** | 93.28 | 89.4 | **98.12** | **86.5** | 93.75 | 90.5 | 87.3 |
| 10% | ArcFace | 99.33 | 93.81 | 94.34 | 93.11 | 89.1 | 97.67 | 85.87 | 92.98 | 90.15 | 86.52 |
| | MV-Arc-Softmax | **99.43** | 93.9 | 94.27 | 93.15 | **89.47** | 97.82 | 85.7 | 93.17 | 90.45 | 87.28 |
| | CurricularFace | 99.33 | 93.92 | 93.97 | 93.12 | 88.78 | 97.52 | 85.43 | 92.98 | 89.53 | 86.53 |
| | BoundaryF1 | 99.4 | 94.02 | 94.3 | 93.18 | 89.32 | 97.85 | 86.5 | 93.32 | 90.33 | 86.95 |
| | **BoundaryFace** | 99.35 | **94.28** | **94.79** | **93.5** | 89.43 | **98.15** | **86.55** | **93.53** | **90.33** | **87.37** |
| 20% | ArcFace | 99.3 | 93 | 93.49 | 92.78 | 88.12 | 97.57 | 84.85 | 91.92 | 88.82 | 85.08 |
| | MV-Arc-Softmax | 99.12 | 93.12 | 93.26 | 93.12 | 88.3 | 97.37 | 85.15 | 92.18 | 89.08 | 85.32 |
| | CurricularFace | 99.13 | 91.88 | 92.56 | 92.28 | 87.17 | 96.62 | 84.13 | 91.13 | 87.7 | 83.6 |
| | BoundaryF1 | 99.32 | 94.02 | **94.5** | 93.18 | **89.03** | 97.63 | 86 | **93.28** | 89.88 | 86.48 |
| | **BoundaryFace** | **99.38** | **94.22** | 93.89 | **93.4** | 88.45 | **97.9** | **86.23** | 93.22 | **90** | **87.27** |

on the clean dataset; our method demonstrates excellent superiority over baseline and SOTA methods on closed-set noise datasets. Furthermore, we can easily draw the following conclusions: 1) As the closed-set noise ratio increases, the performance of every compared baseline method drops quickly; this phenomenon did not occur when using our method. 2) Mining-based softmax has the opposite effect when encountering closed-set noisy data, and the better the method performs on the clean dataset, the worse the results tend to be. In addition, in Fig. 7, given 20% closed-set noise, we present the detection of closed-set noise by BoundaryFace during training and compare it with ArcFace. After closed-set noise is detected, our BoundaryFace dynamically corrects its labels. Correct labels result in a shift in the direction of the closed-set noise being optimized from wrong to right, and also lead to more accurate class centers. Furthermore, more accurate class centers in turn allow our method to detect more closed-set noise at each iteration, eventually reaching saturation.

**Results on noisy synthetic datasets.** The real training set contains not only closed-set noise but also open-set noise. As described in this section, we train our BoundaryFace on noisy datasets with different mixing ratios and compare it with the SOTA competitors. In particular, we set the margin $m = 0.5$ for the training set containing closed-set noise ratio of 30% and open-set noise ratio of 10%, and we set the margin $m = 0.3$ for the other two mixing ratios. As reported in Tab. 4, our method outperforms the SOTA methods on all synthetic datasets. Even on the dataset containing 30% open-set noise, our method still performs better than baseline and SOTA competitors.

## 5 Conclusions

In this paper, we propose a novel mining framework (*i.e.*, BoundaryFace) with tolerating closed-set noise for face recognition. BoundaryFace largely alleviates

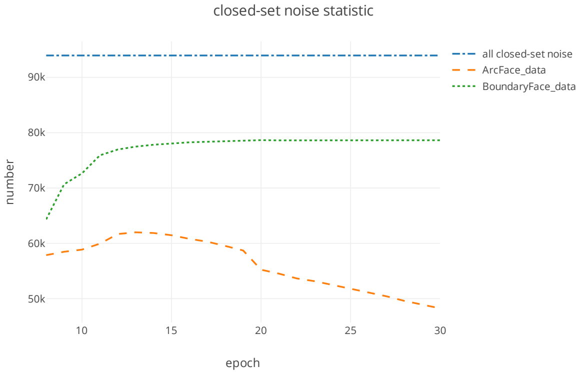

**Fig. 7.** Comparison of closed-set noise detected by BoundaryFace and ArcFace, respectively. The dash dot line indicates the total number of closed-set noise in the training set.

**Table 4.** Verification performance (%) of different loss functions when the training set contains different mixing noise ratios. C denotes closed-set noise ratio(%). O denotes open-set noise ratio(%).

| C | O | Method | LFW | AgeDB | CFP | CA | CP | SLLFW | Asian | Cau | Indian | African |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 20 | ArcFace | 98.87 | 89.55 | 89.56 | 90.43 | 84.4 | 94.87 | 80.97 | 88.37 | 85.83 | 79.72 |
| | | MV-Arc-Softmax | 98.93 | 89.95 | 89.61 | 91.67 | 84.9 | 95.67 | 82.25 | 88.98 | 86.22 | 80.6 |
| | | CurricularFace | 98.33 | 88.07 | 88.29 | 90.18 | 83.65 | 93.87 | 80.3 | 87.78 | 84.43 | 77.35 |
| | | BoundaryF1 | 99.12 | 92.5 | 89.66 | 92.27 | 84.43 | 96.75 | 84.42 | 91.4 | 88.33 | **84.55** |
| | | **BoundaryFace** | **99.2** | **92.68** | **93.28** | **92.32** | **87.2** | **96.97** | **84.78** | **91.88** | **88.8** | 84.48 |
| 10 | 30 | ArcFace | 99.07 | 91.82 | 91.34 | 91.7 | 85.6 | 96.45 | 83.35 | 90 | 87.33 | 81.83 |
| | | MV-Arc-Softmax | 98.92 | 91.23 | 91.27 | 91.9 | 85.63 | 96.17 | 83.98 | 90.15 | 87.57 | 82.33 |
| | | CurricularFace | 98.88 | 91.58 | 91.71 | 91.83 | 85.97 | 96.17 | 82.98 | 89.97 | 86.97 | 82.13 |
| | | BoundaryF1 | 99 | 92.23 | 92.07 | **92.05** | **86.62** | 96.42 | 83.97 | 90.47 | 87.83 | 82.98 |
| | | **BoundaryFace** | **99.17** | **92.32** | **92.4** | 91.95 | 86.22 | **96.55** | **84.15** | **91.05** | **88.17** | **83.28** |
| 30 | 10 | ArcFace | 98.73 | 89.93 | 89.21 | 91.02 | 82.93 | 95.15 | 81.25 | 88.33 | 85.87 | 80.03 |
| | | MV-Arc-Softmax | 98.78 | 89.73 | 88.54 | 91.22 | 82.57 | 95.22 | 81.52 | 88.47 | 85.83 | 80.12 |
| | | CurricularFace | 98.18 | 87.65 | 88.1 | 90.12 | 82.82 | 93.13 | 79.7 | 86.65 | 84.1 | 77.22 |
| | | BoundaryF1 | 99.1 | 92.3 | **90.34** | 92.28 | **85.28** | 96.7 | 83.52 | 90.9 | 87.53 | 83.12 |
| | | **BoundaryFace** | **99.1** | **93.38** | 88.24 | **92.5** | 82.22 | **96.88** | **83.77** | **91.18** | **88.18** | **83.67** |

the poor performance of mining-based softmax on datasets with severe noise problem. BoundaryFace is easy to implement and converges robustly. Moreover, we investigate the effects of noise samples that might be optimized as hard samples. Extensive experiments on popular benchmarks have demonstrated the generalization and effectiveness of our method when compared to the SOTA.

## Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (61876158), Fundamental Research Funds for the Central Universities (2682021ZTPY030).

## References

1. Dong Cao, Xiangyu Zhu, Xingyu Huang, Jianzhu Guo, and Zhen Lei. Domain balancing: Face recognition on long-tailed domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5671–5679, 2020.
2. Beidi Chen, Weiyang Liu, Zhiding Yu, Jan Kautz, Anshumali Shrivastava, Animesh Garg, and Animashree Anandkumar. Angular visual hardness. In *International Conference on Machine Learning*, pages 1637–1648. PMLR, 2020.
3. Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.
4. Weihong Deng, Jiani Hu, Nanhai Zhang, Binghui Chen, and Jun Guo. Fine-grained face verification: Fglfw database, baselines, and human-dcmn partnership. *Pattern Recognition*, 66:63–73, 2017.
5. Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6172, 2020.
6. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
7. Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.
8. Yuge Huang, Pengcheng Shen, Ying Tai, Shaoxin Li, Xiaoming Liu, Jilin Li, Feiyue Huang, and Rongrong Ji. Improving face recognition from hard samples via distribution distillation loss. In *European Conference on Computer Vision*, pages 138–154. Springer, 2020.
9. Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. Curricularface: adaptive curriculum learning loss for deep face recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5901–5910, 2020.
10. Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4873–4882, 2016.
11. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
12. Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.
13. Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7, 2016.
14. Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 51–59, 2017.

15. Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
16. Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2017.
17. Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
18. Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.
19. Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.
20. Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016.
21. Yi Sun. *Deep learning face representation by joint identification-verification*. The Chinese University of Hong Kong (Hong Kong), 2015.
22. Ying Tai, Yicong Liang, Xiaoming Liu, Lei Duan, Jilin Li, Chengjie Wang, Feiyue Huang, and Yu Chen. Towards highly accurate and stable face alignment for high-resolution videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8893–8900, 2019.
23. Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 765–780, 2018.
24. Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.
25. Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017.
26. Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.
27. Mei Wang, Weihong Deng, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Racial faces in-the-wild: Reducing racial bias by deep unsupervised domain adaptation. *arXiv preprint arXiv:1812.00194*, 5, 2018.
28. Xiaobo Wang, Shifeng Zhang, Shuo Wang, Tianyu Fu, Hailin Shi, and Tao Mei. Mis-classified vector guided softmax loss for face recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12241–12248, 2020.
29. Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.
30. Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.
31. Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Feature incay for representation regularization. *arXiv preprint arXiv:1705.10284*, 2017.
32. Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.

33. Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019.
34. Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5:7, 2018.
35. Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments. *arXiv preprint arXiv:1708.08197*, 2017.