Angular Margin-Mining Softmax Loss for Face Recognition

JWAJIN LEE, YOOSEUNG WANG, AND SUNYOUNG CHO
Defense Artificial Intelligence Center, Agency for Defense Development, Daejeon 34186, Republic of Korea
Corresponding author: Sunyoung Cho (sycho22@add.re.kr)

Abstract
Face recognition methods have been significantly improved in recent years owing to the advances made in loss functions. Typically, loss functions are designed to enhance the separability power by concentrating on hard samples in mining-based approaches or by increasing the feature margin between different classes in margin-based approaches. However, margin-based methods lack the utilization of informative hard sample, and mining-based methods also fail to learn the latent correlations between classes. Moreover, there are no methods that simultaneously consider the effects of hard samples and feature margin through the same shape of feature angular margin. Therefore, this paper introduces the Angular Margin-Mining Softmax (AMM-Softmax) loss function, which adaptively emphasizes hard samples while also increasing the decision margins. The proposed AMM-Softmax loss function introduces a linear angular margin for hard samples, enabling the direct optimization of the geodesic distance margin and maximization of class separability. Furthermore, the proposed AMM-Softmax loss function is computationally efficient and can be easily converged by rapidly switching from the hard samples to easy samples. The results of the extensive experimental analyses conducted on popular benchmarks demonstrate the superiority of the proposed AMM-Softmax loss function over the existing state-of-the-art methods.

Index Terms
Deep convolutional neural network, face recognition, metric-learning and softmax loss function.

I. INTRODUCTION
Existing deep face recognition methods are primarily focused on designing effective loss functions that enhance the separability of feature embedding. Adapting a triplet loss [14] for embedding learning is one of the pioneering methods in this field. This method first selects the representative triplet samples and then enforces a pair of samples from the same class, which are closer in distance than those from the different classes. Center loss [24] also improves the discriminability by learning centers for the features of each identity and using the centers to enforce the intra-class compactness. However, the triplet loss faces the drawback of a combinatorial explosion in the number of triplets, which results in a time-consuming learning process for large-scale datasets. Additionally, careful mining strategies are required to select the representative triplets within the informative mini-batch. For the center loss, updating the centers is difficult as the number of classes increases and embedding method for center loss produces a sub-optimal solution lacking in optimization of the inter-class margin. Several methods to improve the efficiency and capability of face recognition have been developed, which attempt to implement the margin-based softmax loss function [1], [10], [20], [22].

The margin-based softmax methods focus on increasing the feature margin between different classes to improve the power of feature discrimination by proposing their own margin function. SphereFace [10] introduces a multiplicative angular margin (A-Softmax) function between the features of different classes to impose a larger inter-class distance. CosFace [22] adds a cosine margin to the cosine similarity (logit) space and solves the instability problem of the training of SphereFace. ArcFace [1] proposes an additive angular margin to obtain the separability of features. Margin-based softmax methods present several advantages. However, they do not consider the significance of informative samples such as hard samples. Recently proposed hard sample mining has been acknowledged as a critical method to further improve the performance. MV-Softmax [23] attempts to integrate a margin-based softmax loss function with the feature mining technique. In this method, hard samples are defined as...
misclassified samples, and are emphasized by increasing their weights to loss. CurricularFace [7] also improves the performance by adaptively adjusting the relative importance of the samples during different training stages.

However, MV-Softmax and CurricularFace are vulnerable to the stability of the training due to their inconsistent decision margins of easy and hard samples. Their decision condition changes based on the hardness for each sample from the perspective of linearity for the decision boundary. The MV-Softmax presents a non-linear margin for hard samples, while its decision margin for easy samples becomes linear or non-linear depending on the type of the margin function. The CurricularFace presents both linear and non-linear decision boundaries for easy and hard samples, respectively, throughout all the stages. Their inconsistency in decision boundaries interrupts the process of changing from the hard sample to the easy sample because the models update their weight by loss for hard samples while mining the hard samples follows the loss for easy samples. They also result in the models being stuck at the local optimum with a considerable amount of samples, which fail to escape from hard samples even at the final stage of training.

This paper proposes an Angular Margin Mining Softmax loss (AMM-Softmax loss) function, which provides a margin and mines hard samples using only consistent linear angular margins for both easy and hard samples. The proposed AMM-Softmax loss corresponds perfectly to the geodesic distance on the hypersphere because of the linear margins, which results in the improvement of the discriminability of the network for face recognition by enabling the network to directly optimize the geodesic distance. The margin consistency presented in this paper for both easy and hard samples also accelerates the relocation from hard to easy samples, thus increasing the speed of convergence during the training procedure when compared to the existing state-of-the-art (SOTA) models.

The mechanism of the proposed AMM-Softmax loss function for hard samples is compared to that of the margin-mining softmax function, as shown in Figure 1. Our loss function presents consistent linear decision boundaries for both easy and hard samples, unlike earlier margin-mining softmax [7], [23]. Consider four samples, $x_1, x_2, x_3$, and $x_4$, which all belong to the same class. At first, all the samples are hard samples because they are all misclassified according to the decision boundary for the easy samples (represented by the green dashed line). The hard samples (sky-blue dot) then become easy samples (blue dot) except for one sample, i.e., $x_2$ of margin-mining softmax (red dot), as each method learns toward the decision boundary for hard sample (red dashed line). This is attributed to the fact that margin-mining softmax has a non-linear decision boundary for hard samples, which is different from those for easy samples. The consistency for decision boundaries presented by the proposed loss functions leads to the acceleration of the convergence by rapidly switching from hard samples to easy samples.

The main contributions of this paper can be summarized as follows:

- We propose a novel AMM-Softmax loss function to obtain more feature separability for face recognition. The proposed loss function presents a constant linear angular margin for hard samples, enabling the direct optimization of the geodesic distance margin, along with the maximization of the class separability. This is the first attempt to introduce a linear angular margin for hard samples to the best of our knowledge.
- The proposed AMM-Softmax loss function improves the speed and stability of training by rapidly switching from hard to easy samples, using consistent linear decision margins for both the easy and hard samples. Additionally, the AMM-Softmax loss function is computationally efficient as it provides a margin to the hard samples using the same parameter $m$, without using or calculating the weighting parameter $t$ in previous loss functions.
- Extensive experiments are conducted on LFW, CALFW, CPLFW, AgeDB, CFP, and IJB-C, and the superiority of the proposed AMM-Softmax loss function over the existing state-of-the-art methods is demonstrated.

The rest of the paper is organized as follows: Section II discusses the relevant works in the field of face recognition. Section III explains the underlying concept on loss functions and the proposed AMM-Softmax loss function. Section IV presents a detailed description of the experimental results and lastly, Section V concludes the paper.

II. RELATED WORKS

A. METRIC LEARNING

Metric learning is a study which aims to automatically construct a task-specific distance metric. It provides a mapping function from the sample to a latent vector space. Therefore, face recognition methods implement this metric to discriminate the identity of a person in an open-set environment. Some studies [14], [17]–[19], [24] have attempted to implement metric learning for the embedding features which
are extracted from the Deep Convolutional Neural Network (DCNN). The triplet loss [14] selects a triplet consisting of the anchor, positive, and negative features to separate the negative features from the anchor and positive features. Contrastive loss [17]–[19] utilizes the feature pairs to place the positive pair together and separates the negative pair. The center loss [24] learns center vectors to accurately represent the position of each class and reduces the distance between the sample features and the corresponding center vector. However, the aforementioned metric learning approaches face the drawbacks of high computational cost and sensitivity to the sample selecting strategy, which significantly affects the performance. NormFace [21] utilizes the normalized softmax loss function while eliminating the costs involved in selecting the samples. It was able to solve the two metric learning problems described earlier.

**B. MARGIN SOFTMAX FUNCTIONS**

Several studies [1], [10], [20], [22] have added a margin on the softmax loss function, based on NormFace [21], to enhance the feature separability power. They have added the margin between the embedded feature and the positive class weight vector, creating a decision boundary gap between the classes. AMSofmax [20] and CosFace [22] consider the margin in an additive manner for the cosine similarities, while the SphereFace [10] sets it in a multiplicative manner. The ArcFace [1] proposes an additive margin for the geodesic distance directly and unifies the previous margin softmax loss functions by formulating the loss function with three different margins: \( m_1 \), \( m_2 \), and \( m_3 \). The margin softmax losses improve the separability power of the face recognition models. However, they are vulnerable to hard samples such as a large-pose variant face. Furthermore, they lack the logit utilization for a non-ground truth class and ignore the significance of mining informative samples for training the separability of a feature extractor.

**C. MINING SOFTMAX FUNCTIONS**

The mining softmax approaches [8], [9], [16] are applied to solve the large-pose variant face problem and to improve the model performance for hard samples. They select the hard samples in the training stage and give them more weight to enhance the model discrimination power. Focal loss [8] focuses on the hard samples by reducing the weights for the easy samples, demonstrating a high inference probability. Shrivastava [16] proposed the online hard sample mining method which selects samples with high loss values from non-uniform and non-stationary distribution. Adaptive-Face [9] weights hard samples, by adaptively sampling the training data based on a sampling probability which depends on the model accuracy for each sample. However, they are vulnerable to over-fitting because of the repeated training of the same mini-batch only consisting of outliers. They also fail to learn the latent correlations between the classes since they use mining at the sample level.

**D. MARGIN-MINING SOFTMAX FUNCTIONS**

The major limitation in combining the features of the methods proposed in the previous two studies is that the margin softmax method adds the angular margin in the logit level while the mining softmax approaches select the margin on a sample level. The margin-mining softmax functions [7], [23] present the advantages of both the mining and margin softmax functions by weighting hard samples and adding margins on the same logit level. The MV-Softmax [23] combines two methods by proposing an indicator function on the previous margin softmax equation which selects misclassified logits for hard samples. They emphasize the importance of misclassified logits by weighting them with a constant parameter. CurricularFace [7] re-designs a weighting function which adopts curriculum learning and adaptively emphasizes the easy samples at earlier stages and the hard samples at the later stages during model training process.

The previously developed margin-mining approaches [7], [23] successfully overcome the limitations of the overfitting problem of mining softmax and success to learn a difficulty correlation between classes. However, they present a non-linear geometric margin for the hard samples which results in an instability and a low convergence rate in training. Conversely, to increase the convergence rate and improve the model stability, the proposed AMM-Softmax uses a linear weighting function for misclassified logits.

**III. METHOD**

**A. ANGULAR MARGIN-MINING SOFTMAX LOSS FUNCTION**

This section explains the novel AMM-Softmax loss function which adds a linear angular margin for misclassified cosine similarities for hard samples. Figure 2 illustrates the overall pipeline of our the proposed AMM-Softmax function.

We first consider the naive softmax loss function, which is represented as follows:

\[
L_{\text{naive}} = -\log \frac{e^{w_{y}^T x + b_y}}{\sum_{k=1}^{n} e^{w_{k}^T x + b_k}},
\]  

(1)

where \( x \in \mathbb{R}^d \) represents an output feature vector for a sample extracted from DCNNs and \( y \) denotes a ground truth label corresponding to \( x \), which is included in the interval of \( k \in 1, 2, \ldots, n \), where \( n \) denotes the number of classes. \( w_k \in \mathbb{R}^d \) represents a weight vector of the \( k \)-th class. \( d \) and \( w_y \) denote the embedding feature dimension and a weight vector of the ground truth class, respectively. \( b_y \) and \( b_k \) represent the biases. The feature vector \( x \) and class weight vector \( w_k \) are normalized to \( \frac{w_{y}}{|w_{y}|} \) and \( \frac{w_{k}}{|w_{k}|} \), respectively, using an \( l_2 \) normalization and the biases \( b_k \in \mathbb{R} \) are set to zero similar to [10], [21]. The cosine similarities are obtained in the form of \( \cos \theta_k = \frac{w_{y}^T x}{|w_{y}| |w_{k}|} \) by multiplying the two normalized values as shown in Figure 2, where \( \theta_k \) indicates the angle between the feature vector \( x \) and the \( k \)-th class weight vector \( w_k \). The cosine similarities are rescaled to \( s \) for modulating the logit magnitudes. Subsequently, the normalized and scaled
softmax loss function $L_{\text{norm}}$ can be represented as follows:

$$L_{\text{norm}} = - \log \sum_{k=1}^{n} e^{\text{scos}(\theta_k)}.$$

(2)

The cosine similarities (logits) are classified into four different groups: positive (blue), negative (red), misclassified (purple), and well-classified(green) logits, as shown in Figure 2. The positive logit $\text{cos}(\theta_k)$ represents a cosine similarity between feature vector $x$ and center vector for ground-truth class $y_k$. The negative logit $\text{cos}(\theta_k)$ represents a cosine similarity for the non-ground truth class, $k \in \{1, 2, \ldots, n\}$ and $k \neq y$, which can be separated into the well-classified logit and misclassified logit. The red box in Figure 2 represents the method used to extract the misclassified logit from the negative logit. The dashed line represents a decision boundary formulated in $\text{cos}(\theta_k + m) < \text{cos}(\theta_k)$ where $m$ denotes the parameter for the margin between the class weight vectors earlier. If the given sample $x$ satisfies the condition, then the weight vector $w_k$ for class $k$ is classified into the misclassified logit. Otherwise, the weight vector $w_k$ is classified into the well-classified logit.

The sample $x$ is categorized as a hard sample if it was able to identify at least one misclassified logit calculating the AMM-Softmax loss function. Correspondingly, the sample is categorized as an easy sample if no misclassified logit is identified while calculating the AMM-Softmax loss function. The face recognition task primarily aims to minimize the number of misclassified samples and logits.

An angular decision margin $m$ is added in the positive logit following the previous approaches using angular margin [1], [7], [23] to enhance the embedding feature discrimination (blue logit in Figure 2). Subsequently, $f(m, \theta_k)$ is used instead of $\text{cos} \theta_k$ in Eq. 2, where $f(m, \theta_k)$ denotes a margin function for the positive cosine similarity. Correspondingly, the decision boundary condition for the positive logit is set to $f(m, \theta_k) = \text{cos}(\theta_k)$ as shown in Table 1. The normalized softmax loss function with an angular margin is represented as:

$$L_{\text{margin}} = - \log \sum_{k=1}^{n, k \neq y} e^{f(m, \theta_k)} + \sum_{k=1}^{n} e^{\text{scos}(\theta_k)}.$$

(3)

In the general form, margin function is formulated as, $f(m, \theta_k) = \text{cos}(m_1 \theta_k + m_3) - m_2$. In our method, $m_1$, $m_2$, and $m_3$ are set to 1, 0, and $m$. An additional linear margin reweighting function is introduced for mis-classified logits, $v(\theta_k, m, I)$. The proposed linear margin reweighting function is represented as follows:

$$u(\theta_k, m, I) \begin{cases} \text{cos}(\theta_k), & I_k = 0, \\ \text{cos}(\theta_k - m), & I_k = 1. \end{cases}$$

(4)

$I_k$ denotes a binary mining indicator to categorize the misclassified logits among the negative logits,

$$I_k = \begin{cases} 0, & f(m, \theta_k) \geq \text{cos}(\theta_k), \\ 1, & f(m, \theta_k) < \text{cos}(\theta_k). \end{cases}$$

(5)

where $f(m, \theta_k) = \text{cos}(\theta_k + m)$ and $m > 0$. The misclassified logits categorized by the binary mining indicator $I$ are weighted by our linear margin function to increase the discrimination between the class weight vectors.

The overall AMM-Softmax loss function with the mining indicator and corresponding reweighting function is represented as follows:

$$L_{\text{amm}} = - \log \sum_{k=1}^{n, k \neq y} e^{f(m, \theta_k)} + \sum_{k=1}^{n} e^{\text{scos}(\theta_k + m)}.$$

(6)

The proposed AMM-Softmax has linear angular margins for both the positive and misclassified logits. The linear angular
FIGURE 3. Decision margins of different loss functions under binary classification case. \( \theta_i \) represents the angular distance between the feature vector and the \( i \)-th class weight vector, \( \theta_i = \arccos(|w_i|, x) \). Black and red dashed lines represent the decision margins for easy sample and hard sample, respectively.

TABLE 1. The decision boundaries of face recognition methods.

| Loss                | Decision Boundary                                                                 |
|---------------------|----------------------------------------------------------------------------------|
| Softmax             | \( \cos\theta_y = \cos\theta_k \)                                              |
| SphereFace          | \( \cos(m\theta_y) = \cos\theta_k \)                                           |
| CosFace             | \( \cos\theta_y - m = \cos\theta_k \)                                          |
| ArcFace             | \( \cos(\theta_y + m) = \cos\theta_k \)                                        |
| MV-Arc-Softmax      | \( \cos(\theta_y + m) = \cos\theta_k (\text{easy}) \)                          |
|                     | \( \cos(\theta_y + m) = (t + \cos\theta_k + t) (\text{hard}) \)               |
| CurricularFace      | \( \cos(\theta_y + m) = \cos\theta_k (\text{easy}) \)                          |
|                     | \( \cos(\theta_y + m) = \cos(\theta_k - m) (\text{hard}) \)                   |
| Ours (AMM)          | \( \cos(\theta_y + m) = \cos\theta_k (\text{easy}) \)                          |
|                     | \( \cos(\theta_y + m) = \cos(\theta_k - m) (\text{hard}) \)                   |

margin for misclassified logits enables the maximization of the class separability and optimizes the geodesic distance margin. Additionally, the AMM-Softmax loss function simplifies the training procedure and reduce the time required to relocate the hard samples to the easy samples.

B. COMPARISONS WITH DIFFERENT LOSS FUNCTIONS

This section presents a comparison between the proposed AMM-Softmax function and the margin, mining and margin-mining softmax functions, as shown in Figure 3 and Table 1.

The normalized naive softmax function without the angular margin is set as our baseline. The first subplot of Figure 3 and the first row of the Table 1 represents the boundary condition \( \cos\theta_y = \cos\theta_k \) and following decision boundary plot of the baseline.

1) MARGIN SOFTMAX FUNCTIONS

The margin softmax functions (CosFace [22] and ArcFace [1]) consider an angular margin based only on the positive logits. They are in the form of Eq. 3 and have different boundary conditions for positive logits as shown in Table 1. CosFace [22] introduced the method to add margins in logits to increase the inter-class compactness, and proposed the decision boundary condition as, \( \cos(\theta_y) - m = \cos(\theta_k) \). ArcFace [1] addressed the non-linearity problem of decision margins and defined the linear decision margin by adding a direct margin to \( \theta \) as \( \cos(\theta_y + m) = \cos(\theta_k) \).

The proposed AMM-Softmax function adopted the boundary condition for positive logit following ArcFace [1]. In addition to the margin softmax functions, the proposed AMM-Softmax introduces a mining indicator for the misclassified logits and exploits misclassified logits, by weighting them with a linear margin.

2) MINING SOFTMAX FUNCTIONS

The mining softmax approaches [8], [16] designed a mining function to select the informative samples and utilized them during training to emphasize the importance of difficult samples. The mining softmax function can be summarized as:

\[
L_{\text{mining}} = -g(x)\log \frac{e^{\cos\theta_y}}{\sum_{k=1}^{n} e^{\cos\theta_k}},
\]

where \( g(x) \) represents a mining function. In OHEM [16], the samples which have a higher loss are classified as hard samples with \( g(x) = 1 \), or \( g(x) = 0 \) for easy samples which have a lower loss. In focal loss [8], they designed the mining such that \( g(x) = (1 - p_y)^\gamma \), where a probability \( p_y = \frac{e^{\cos\theta_y}}{\sum_{k=1}^{n} e^{\cos\theta_k}} \), and \( \gamma \) is a constant modulating factor. However, previously proposed mining softmax functions utilize mining only to select informative samples, while the proposed AMM-Softmax utilizes mining at the logit level. Logit level mining presents the potential for a more consistent strategy to train a discriminative feature extractor, since the model can determine the difficulty of the samples class by class. Furthermore, there is an opacity to utilize loss [16] and probability [8] as mining strategies for informative samples to the training model, whereas the proposed AMM-Softmax function clearly defines the informative samples as misclassified samples.

3) MARGIN-MINING SOFTMAX FUNCTIONS

The margin-mining softmax functions extract the misclassified logits from the negative logits. They are added to insert the margin on positive logits similar to the margin softmax functions. The AMM-Softmax function is compared with two previous margin-mining softmax functions, MV-Softmax [23] and CurricularFace [7] in this subsection.
The general form of margin-mining softmax functions is defined as,

$$L_{mm} = -\log \frac{e^{f(m, \theta_k)}}{\sum_{k=1, j \neq 1}^{n} h(t, \theta_k, l_k) e^{f(m, \theta_k)}}$$  \hspace{1cm} (8)

where $h(t, \theta_k, l_k)$ represents a weighting function, and $t$ denotes a weighting parameter for misclassified logit. MVM-Softmax [23] defined the weighting function as,

$$h(t, \theta_k, l_k) = e^{t\cos(\theta_k) + 1/ \lambda} = e^{t\cos(\theta_k) + 1/ \lambda}$$.

The decision boundary condition of MVM-Softmax to the misclassified logits is in a non-linear form,

$$cos(\theta_k + m) = (t + 1)cos\theta_k + t$$,

as shown in Figure 3. CurricularFace [7] proposed the reweighting function,

$$h(t, \theta_k, l_k) = e^{t(1+\cos\theta_k)\cos\theta_k}$$, and $t$ as an Exponential Moving Average (EMA) of positive logits, $cos(\theta_k)$. It adaptively adjusts the leverage for misclassified logits and presents the varying boundary condition, $cos(\theta_k + m) = (t + \cos\theta_k)\cos\theta_k$.

The margin-mining softmax functions [7], [23] adopt the margin softmax to increase the decision margin of a positive logit with [1], implying that they use the same mining setting for their margin function. For margin-softmax, we follow the setting for their margin function. (i.e., 0.35 for MV-AM-Softmax, 0.5 for MV-ArcSoftmax, CurricularFace, and AMM-Softmax).

### IV. EXPERIMENT

#### A. EXPERIMENTAL SETTING

1) DATASETS

Table 2 presents the identity and the number of images of the datasets used in our experiment. All the models are trained on the same public datasets, MS1MV2, for a fair comparison. MS1MV2 presents an improvement on the previous MS-Celeb-1M [4] dataset for a semi-automatic refinement version. The models are tested for two tasks, a verification task and an identification task. The verification performance of the AMM-Softmax model is compared with other recent face recognition models [1], [7], [10], [22], [23] on the widely used LFW [6] and CFP-FF [15] datasets. Furthermore, we also measure the model verification-robustness in large-age and large-pose variance using the more challenging benchmarks, CFP-FP [15], AgeDB-30 [12], CPLFW [25], and CALFW [26]. The IJB-C [11] dataset was adopted to test verification and identification performances of the models in large-scale benchmarks using the 1:1 and 1:N protocol of IJB-C. The IJB-C dataset is an extension of IJB-B.

2) PREPROCESSING AND CNN SETUP

The input of the network is preprocessed using the following two steps: face detection and face alignment. Face detection is conducted with RetinaFace [2] to find the face in all the images along with the corresponding five facial landmarks (2 eyes, 1 nose, 2 tips of mouth) for each person. After finding the landmarks, affine warping is used for face alignment, based on landmarks. It stretches and rotates the faces and produces a preprocessed image with a size of $112 \times 112$ size in width and height. We employed SE-ResNet50-IR as our backbone network, which consists of a ResNet50 backbone network with a Squeeze-and-Excitation (SE) [5] module, and improved residual module (IR) [3] which is customized for the image recognition task. The latent vector with a 512 channel dimension is extracted from an $112 \times 112$ image by the backbone network.

3) TRAINING

We trained the model with Stochastic Gradient Descent (SGD) with momentum of 0.9 and weight decay of 0.0005 in all experiments. We used 0.05 initial learning rate and divided it by 0.1 at 7-th, 14-th, 21-th epoch. The total epoch is 25. The default scale size $s$ is set to 32, margin size $m$ is set to 1.35, 0.35, 0.5 for SphereFace [10], CosFace [22], and ArcFace [1], respectively. For margin-softmax, we follow $m$ setting for their margin function. The weight parameter for misclassified logit $t$ is set to 0.2 for MV-AM and 0.3 for MV-Arc following [23]. All experiment is implemented by PyTorch [13] framework and we used four NVIDIA RTX3090 in parallel with batch size 512. We train the backbone network from scratch by initializing fully connected (FC) layer at the back to calculate loss and to flow gradient to the backbone network.

4) TESTING

Preprocessed images are used as an input to test the models and we utilize only the trained backbone network without the FC layer to test the performance of the models. All the input images are converted to the feature vectors using the trained backbone network, and cosine similarities of the vectors are calculated for a 1:1 verification and 1:N identification. The verification and identification protocols utilize their own templates or lists to select the samples and quantify the performances of the models.

### TABLE 2. Face recognition datasets for training and test.

| Datasets      | #Identity | #Image/Video |
|---------------|-----------|--------------|
| MS1MV2 [4]    | 85K       | 5.8M         |
| LFW [6]       | 5,749     | 13,233       |
| CFP-FP [15]   | 500       | 7,000        |
| CFP-FF [15]   | 500       | 7,000        |
| AgeDB-30 [12] | 568       | 16,488       |
| CPLFW [25]    | 5,749     | 11,652       |
| CALFW [26]    | 5,749     | 12,174       |
| IJB-C [11]    | 3,531     | 138K         |

J. Lee et al.: Angular Margin-Mining Softmax Loss for Face Recognition
B. EVALUATION RESULTS

1) CONVERGENCE OF AMM-SOFTMAX

Figure 4 depicts the training loss curves of the margin-mining softmax models during training. The AMM-Softmax loss function reduces its training loss faster than any other margin-mining methods even though it starts with the second highest training loss. CurricularFace presents a superior training property and convergence of models since it is based on curriculum learning. It demonstrates a smaller dispersion and faster convergence when compared to that of the MV-Softmax loss function. However, the AMM-Softmax loss function presents faster and stable convergence performance than other methods after the early stage of training. It is difficult to mathematically demonstrate the superiority of training. However, considering the number of misclassified logits of margin-mining models shown in Figure 5, the convergence behavior directly corresponds to the speed with which the model exploits misclassified logits. This is crucial for the training of the separable feature extractor because it demonstrates the speed at which the model changes samples with high loss to samples with small loss. In this case, the proposed AMM-Softmax loss function, which uses a consistent margin method for the easy samples (linear) and hard samples (linear) achieves faster convergence behavior when compared to the model that has an inconsistent decision margin. Furthermore, the number of misclassified logits remaining at the last epoch of MV-AM is 200K, which is quantitatively large while AMM-Softmax remains in a few hundred, completely utilizing the misclassified logits during training process.

2) DISTRIBUTION OF \( \theta_y \)

To demonstrate the superiority in inter-class separability and class compactness of training with our AMM-Softmax loss function, we present the distribution of \( \theta_y \). Figure 6 represents the histogram of angle between ground-truth class weight vector and input feature that is extracted from different four losses (Softmax, Arc-Softmax [1], MV-Softmax [23] and our AMM-Softmax). The angles are measured three times at the start (1st epoch), middle (2nd epoch), and end (25th epoch) phase of training, and each mean value \( \mu \), and variance \( \sigma^2 \) of the angle is calculated at the end phase for a detailed comparison. At the start of the training phase, all the class weight vectors and weights of the backbone network are randomly initialized and lack any similarity between the feature vector and positive class weight vector, which are faces each other at an angle of 90°. As the training proceeds, the angle distribution decreases and becomes sharper because the model is trained to maximize the intra-class compactness and inter-class separability. Since the training a face recognition model aims to train a discriminative classifier, the distribution with the lower \( \mu \) and lower \( \sigma^2 \) at the end phase presents the better performance of the models. In Figure 6, the Softmax function shows the
TABLE 3. Verification performance (%) of different face recognition loss functions on the test sets LFW, CFP, AgeDB, CALFW, and CPLFW.

| Methods                | LFW  | CFP-FF | CFP-FF | AgeDB | CALFW | CPLFW |
|------------------------|------|--------|--------|-------|-------|-------|
| Softmax [1] (baseline) | 99.08| 98.80  | 94.39  | 92.33 | 94.66 | 87.76 |
| SphereFace [10]        | 99.33| 99.27  | 94.63  | 94.93 | 94.25 | 89.63 |
| CosFace [22]           | 99.47| 99.76  | 97.80  | 97.93 | 95.95 | 91.90 |
| ArcFace [1]            | 99.77| 99.79  | 97.91  | 97.97 | 96.07 | 92.03 |
| CurricularFace [7]     | 99.77| 99.79  | 97.94  | 97.87 | 95.90 | 92.10 |
| MV-AM [23]             | 99.78| 99.79  | 98.04  | 98.02 | 96.13 | 92.15 |
| AMM-Softmax (m=0.3, s=32) | 99.78| 99.79  | 98.17  | 98.02 | 96.17 | 92.47 |
| AMM-Softmax (m=0.35, s=32) | **99.82** | **99.80** | 98.11 | 97.95 | 96.16 | 92.31 |
| AMM-Softmax (m=0.3, s=48) | 99.77| 99.79  | 98.15  | 97.98 | 96.14 | 92.57 |

3) RESULTS ON VERIFICATION DATASETS
The proposed AMM-Softmax loss function is compared to the recent face recognition models (SphereFace [10], CosFace [22], ArcFace [1], CurricularFace [7], and MV-AM [23]) on famous public face verification benchmarks, LFW and CFP, as shown in Table 3. Furthermore, the performance of the AMM-Softmax loss function is demonstrated on relatively difficult datasets that consist of high variant samples in pose and age (CALFW, CPLFW, AgeDB). The model trained using AMM-Softmax loss function leads the baseline by a large margin, and surpasses the other models [1], [7], [10], [22], [23] exhibiting the highest performance for each benchmark, as shown in Table 3. The proposed AMM-Softmax loss function presents an improvement of 92.15%, which is 0.42% higher than that of the MV-AM and 4.81% higher than that of the baseline on CPLFW, which is the most difficult dataset among the considered benchmarks. This result demonstrates that the proposed AMM-Softmax method can considerably enhance the discriminability of the embedding network using consistent linear angular margin by simply changing the softmax loss function.

4) RESULTS ON IJB-C
Table 4 presents the verification and identification results of the model learned by the AMM-Softmax loss function when compared to the existing face recognition models. This study employs the 1:1 protocol of IJB-C as a verification benchmark and 1:N protocol as an identification benchmark to compare the performance of the models. To evaluate the robustness and reliability of the models in the face recognition task, the task positive rate (TPR) of the model at a very low false positive rate (FPR) is important. Therefore, a 1:1 verification TPR is reported when the FPR = 1e-6, and 1:N identification TPR when FPR = 1e-3 to estimate the AMM-Softmax performance. The proposed AMM-Softmax loss function outperforms the other state-of-the-art models [1], [7], [23] by a margin of 0.3% for verification and 3.6% for identification, as shown in Table 4. Figure 7 presents the receiver operating characteristic (ROC) curve of the proposed AMM-Softmax and other SOTA models on IJB-C.

TABLE 4. 1:1 Verification and 1:N Identification TPR at low FPR of different methods on IJB-C dataset.

| Methods   | 1:N Id(FPR=1e-3) | 1:1 Ver(FPR=1e-6) |
|-----------|-----------------|-------------------|
| Softmax   | 0.766           | 0.581             |
| ArcFace   | 0.845           | 0.713             |
| CurricularFace [7] | 0.851 | 0.667            |
| MV-Arc [23] | 0.856 | 0.707            |
| MV-AM [23] | 0.876 | 0.703            |
| (Ours)    | 0.879           | 0.739             |
1:1 protocol. The results of the SOTA methods on the test dataset present significantly high performance over 0.8 TPR@FPR = 1e-6. However, the proposed AMM-Softmax loss function reached the highest TPR@FPR = 1e-6 value and highest area under the ROC curve (AUC). The results demonstrate that the proposed AMM-Softmax loss function presents reliable performance through various FPR situations, and presents the best verification performance when compared to other SOTA methods.

5) EFFECT OF HYPER-PARAMETER, M AND S
Table 5 depicts the two ablation studies for different values of margin $m$ and scale size $s$. An ablation study for the margin $m$ is conducted in the scale of [0.2, 0.35] increasing by 0.05, while $s$ is fixed to 32 as shown in the first four rows in Table 5. In another ablation study for $s$ as shown in the last four rows, $m$ is set to a constant value of 0.3, and $s$ is increased from 16 to 64. The results demonstrate that small $s$ and small $m$ provide insufficient discriminability to the model. However, high margin and scale size do not necessarily ensure a high performance either. Instead, it was observed that the proposed AMM-Softmax loss function achieves the best result with the margin of 0.3 and 32 for the scale size. This result demonstrates that margin and scale are essential parameters for face recognition and the optimal parameters of AMM-Softmax are $m = 0.3$ and $s = 32$.

### Table 5. Verification performance (%) of our AMM loss function with different combination of hyper-parameter, $m$ and $s$.

| Method       | LFW  | CFP-FF | CFP-FP | AgeDB | CALFW | CPLFW |
|--------------|------|--------|--------|-------|-------|-------|
| AMM (m=0.2, s=32) | 99.77 | 99.70  | 98.00  | 97.73 | 96.10 | 92.10 |
| AMM (m=0.25, s=32) | 99.78 | 99.80  | 98.11  | 98.00 | 96.07 | 92.22 |
| AMM (m=0.3, s=32) | 99.78 | 99.79  | **98.17** | **98.02** | 96.17 | 92.47 |
| AMM (m=0.35, s=32) | **99.82** | **99.80** | 98.11  | 97.95 | 96.16 | 92.31 |
| AMM (m=0.3, s=16) | 99.75 | 99.71  | 97.53  | 97.50 | 95.82 | 91.28 |
| AMM (m=0.3, s=32) | 99.78 | 99.79  | 98.17  | 98.02 | 96.17 | 92.47 |
| AMM (m=0.3, s=48) | 99.77 | 99.79  | 98.15  | 97.98 | 96.14 | 92.57 |
| AMM (m=0.3, s=64) | 99.78 | 99.79  | 98.12  | 97.82 | 95.98 | 92.45 |

### REFERENCES

[1] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive angular margin loss for deep face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4690–4699.
[2] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, “RetinaFace: Single-stage dense face localisation in the wild,” 2019, arXiv:1905.06641.
[3] I. C. Duta, L. Liu, F. Zhu, and L. Shao, “Improved residual networks for image and video recognition,” in Proc. 25th Int. Conf. Pattern Recognit. (ICPR), Jan. 2021, pp. 9415–9422.
[4] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “MS-Celeb-1M: A dataset and benchmark for large-scale face recognition,” in Proc. Eur. Conf. Comput. Vis., vol. 9907, Oct. 2016, pp. 87–102.
[5] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 8, Jun. 2020, pp. 1–9.
[6] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database forstudiying face recognition in unconstrained environments,” in Proc. Workshop Facies ‘Real-Life’ Images, Detection, Alignment, Recognit., 2008, pp. 1–15.
[7] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang, “CurricularFace: Adaptive curriculum learning loss for deep face recogni- tion,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 5901–5920.
[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2980–2988.
[9] H. Liu, X. Zhu, Z. Lei, and S. Z. Li, “AdaptiveFace: Adaptive margin and sampling for face recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 11939–11948.
[10] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “SphereFace: Deep hypersphere embedding for face recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 212–220.
[11] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother, “IARPA Janus benchmark—C: Face dataset and protocol,” in Proc. Int. Conf. Biometrics (ICB), Feb. 2018, pp. 158–165.
[12] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, “AgeDB: The first manually collected, in-the-wild age database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 51–59.
[13] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, Dec. 2019, pp. 8026–8037.
[14] F. Schroff, D. Kalenichenko, and J. Philbin, “FaceNet: An unified embedding for face recognition and clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 815–823.
[15] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, “Frontal to profile face verification in the wild,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Mar. 2016, pp. 1–9.
[16] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[17] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 1988–1996.

[18] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[19] Y. Sun, X. Wang, and X. Tang, “Deeply learned face representations are sparse, selective, and robust,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2892–2900.

[20] F. Wang, W. Liu, H. Liu, and J. Cheng, “Additive margin softmax for face verification,” *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.

[21] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, “NormFace: L2 hypersphere embedding for face verification,” in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 1–9.

[22] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “CosFace: Large margin cosine loss for deep face recognition,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[23] X. Wang, S. Zhang, S. Wang, T. Fu, H. Shi, and T. Mei, “Mis-classified vector guided softmax loss for face recognition,” in *Proc. Conf. Artif. Intell. (AAAI)*, 2020, vol. 34, no. 7, pp. 12241–12248.

[24] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2016, pp. 499–515.

[25] T. Zheng and W. Deng, “Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments,” Beijing Univ. Posts Telecommun.. Beijing, China, Tech. Rep. 01, 2018.

[26] T. Zheng, W. Deng, and J. Hu, “Cross-age LFW: A database for studying Cross-age face recognition in unconstrained environments,” 2017, arXiv:1708.08197.

**JWAJIN LEE** received the B.S. degree in electrical and computer engineering (ECE) from the Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea, in 2020. Since 2020, he has been a Research Officer with the Defense Artificial Intelligence Center, Agency for Defense Development, Daejeon. His research interests include face recognition, facial expression recognition, and artificial intelligence.

**YOOSEUNG WANG** received the B.S. degree in electrical and computer engineering from the Ulsan National Institute of Science and Technology, in 2019. He is currently a Research Officer with the Defense Artificial Intelligence Center, Agency for Defense Development. His research interests include image processing, pattern recognition, and artificial intelligence.

**SUNYOUNG CHO** received the Ph.D. degree in computer science from Yonsei University, South Korea, in 2014. From 2015 to 2016, she was a Post-doctoral Fellow with the Human–Computer Interaction Institute, Carnegie Mellon University, PA, USA. She is currently a Senior Researcher with Agency for Defense Development, South Korea. Her research interests include facial expression recognition, emotion recognition, and object detection.