EOGFACE: DEEP FACE RECOGNITION VIA EXTENSIONAL LOGITS

Xingying Zhao, Hao Jiang, Dong Shen

School of Mathematics, Renmin University of China

ABSTRACT

The core of face recognition task is to learn the discriminative feature representation, which has intra-class compactness and inter-class separability. In recent years, some margin-based softmax loss functions were designed to encourage the intra-class compactness, but they neglect the inter-class separability. RegularFace were proposed to increase the inter-class separability. However, RegularFace is inefficient and memory-consumptive on large datasets with large numbers of identities. In this paper, we propose a novel method, named EogFace. It can encourage both the intra-compactness and the inter-class separability. EogFace has intuitive geometric interpretation and theoretical proof, which is easy to implement and only adds negligible computational overhead. Extensive experiments on popular benchmarks of face recognition showed the effectiveness of method over existing state-of-the-art(SOTA) algorithms. Our codes will be released soon.

Index Terms— Face Recognition, Loss, Separability, Compactness

1. INTRODUCTION

The broad prospect of face recognition has attracted significant attention in the field of computer vision and biometrics. In the past few decades, there are many studies[1, 2, 3, 4, 5] on face recognition. Traditional machine learning techniques[6, 7] rely on artificially designed features, which have been replaced by deep learning methods[8, 9, 10, 11, 12] due to the significant success of deep convolutional neural networks (DCNNs).

Face recognition is usually trained as a classification task, which maps the face image into the embeding space using DCNNs[13][14]. Differing from image classification, face recognition is usually an open-set recognition problem that includes unseen identities. It thus expects the intra-class compactness and inter-class separability properties. Due to inability of the traditional softmax loss to obtain highly discriminative embedding features, a series of novel methods have been suggested, such as SphereFace[12][9], CosFace[11][15] and ArcFace[8]. However, the existing methods like SphereFace[9], CosFace[11][15] and ArcFace[8] merely pay attention to the angle relationship between features and class centres and neglected the angles between different class centres and not optimize them explicitly. RegularFace[16] considered the importance of the inter-class separability. However, RegularFace is calculated from a big cosine similarity matrix. For a dataset with large number of identities, the computation is memory-consumptive and inefficient.

To solve these problems, we propose a new loss function called EogFace. In order to optimize the cosine similarity between different class centres, we introduce extensional logits term. In each mini-batch learning, we separately extract the class centre vectors corresponding to each sample, and then calculate the logits between the extracted class centres and all class centres. We take out the negative logits and concatenate them with the feature logits. Lastly, we calculate softmax. The contributions of this study are as follows:

- The proposed method can explicitly widen the angle distance between different class centres to promote inter-class separability.
- The proposed method can be seamlessly plugged into existing approaches to improve the performance without much GPU memory consumption and additional hyper-parameters.
- The superiority of the proposed method is verified on IJB[17][18], LFW[19], MegaFace Challege[20].

2. RELATED WORK

The mainstream approach to train face recognition model is to treat it as a fine-grained classification task. As the most common loss function used in the classification task, Softmax's general form is as follows:

$$\mathcal{L}_{softmax} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{C} e^{w_j^T x_i + b_j}} \tag{1}$$

Here, $x_i \in R^D$ denotes the deep feature of the i-th sample in the D-dimensional space, which belongs to y_i class. The weight parameter w is a matrix of $D \times C$, where $w_j \in R^D$ denotes the j-th column of the w, and $b_j \in R^D$ is the bias term. N is the batch size and C is the class number.

SphereFace[9] and NormFace[21] set the weight $\|w\| = 1$

and the feature $\|x\| = s$ by L_2 normalization and fix the bias $\|b\| = 0$, which makes the training process pay more attention to the optimization of angles. Thus, the learned feature is deeply distinguishable in the hypersphere manifold space.

Based on NormFace[21] and SphereFace[9], the loss function was further improved in [11][15] by introducing an additive cosine margin to the softmax loss. ArcFace[8] modified the approach of margin, introducing an additive angular margin, and directly maximized decision boundary in angular space. The above several variants of softmax can be developed in a united framework:

$$\mathcal{L}_u = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\mathcal{F}(\alpha_{y_i})}}{e^{s\mathcal{F}(\alpha_{y_i})}+\sum_{j=1,j\neq y_i}^{C} e^{s\cos\alpha_j}} \quad (2)$$

Where $\mathcal{F}(\alpha_{y_i}) = \cos(m_1\alpha_{y_i} + m_2) - m_3$.

3. PROPOSED APPROACH

3.1. EogFace

In each mini-batch learning, we extract the class centre vectors which are corresponding to the samples, and then calculate the logits between the extracted centre and all centres. Due to the positive logits are 1 and need not to be optimized, we take out the negative logits and concatenate them with the sample feature logits. Lastly, we calculate softmax. We call it EogFace. EogFace, as a plug-in, can be easily inserted into any softmax-based function. For the sake of clarity, we plug EogFace into the above united loss function $\mathcal{L}_u$. The formula is given as follows:

$$\mathcal{L}_{Eog} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\mathcal{F}(\alpha_{y_i})}}{e^{s\mathcal{F}(\alpha_{y_i})}+\sum_{j=1,j\neq y_i}^{C}(e^{s\cos\alpha_j}+e^{s\cos\beta_j})} \quad (3)$$

Where $\cos\alpha_j = w_j^T x_i$ and $\cos\beta_j = w_j^T w_{y_i}$.

Here, N is the number of samples in a mini-batch, the feature x and weight vector w are normalized by L_2 normalization, x_i is the i-th feature belonging to the y_i-th class, and w_{y_i} is the class centre vector corresponding to the class of y_i. α_j is the angle between w_j and x_i, and β_j is the angle between w_j and w_{y_i}. s and m are hyper-parameters.

3.2. Inter-class Separability

Inter-class separability and intra-class compactness are two key indicators of distinguishing features. Existing methods[9, 11, 15, 8] only focused on intra-class compactness. Let's give a three-classification task example as shown in Figure 1. In order to facilitate comparison, we make some formal transformation of loss function.

*Property*1. $\max\{\cos\alpha_j, j \neq y_i\} < LSE(s, \cos\alpha_j) <= \max\{\cos\alpha_j, j \neq y_i\} + constant$.

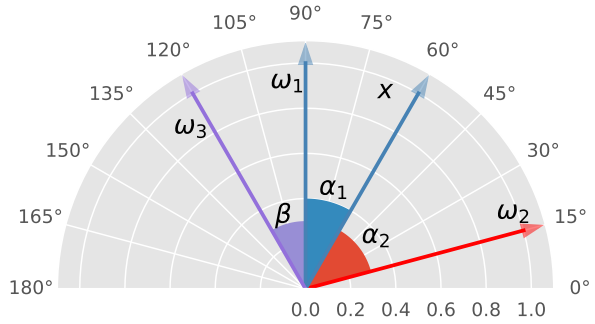


Fig. 1. Simple illustration of face embedding features. w_1, w_2, and w_3 are centre vectors of Class 1, Class 2, and Class 3, respectively. x represents the embedding feature of Class 1. α_1 is the angle between w_1 and x. α_2 is the angle between w_2 and x. β is the angle between w_1 and w_3.

Proof. Here, $LSE(s, \cos\alpha_j) = \frac{1}{s}\log\sum_{j=1,j\neq y_i}^{C} e^{s\cos\alpha_j}$. First, we prove the left part of the inequality:

$$\begin{aligned} LSE(s,\cos\alpha_j) &= \frac{1}{s}\log\sum\nolimits_{j=1,j\neq y_i}^{C} e^{s\cos\alpha_j} \\ &> \frac{1}{s}\log(e^{s(\max\{\cos\alpha_j, j\neq y_i\})}) \\ &= \max\{\cos\alpha_j, j\neq y_i\} \end{aligned} \quad (4)$$

For the right part of the inequality:

$$\begin{aligned} LSE(s,\cos\alpha_j) &= \frac{1}{s}\log\sum_{j=1,j\neq y_i}^{C} e^{s\cos\alpha_j} \\ &= \frac{1}{s}\log(\sum_{j=1,j\neq y_i}^{C} e^{s(\cos\alpha_j - \max\{\cos\alpha_j, j\neq y_i\})}) \\ &+ \frac{1}{s}\log(e^{s(\max\{\cos\alpha_j, j\neq y_i\})}) \\ &<= \frac{1}{s}\log(\sum_{j=1,j\neq y_i}^{C} e^{0}) + \max\{\cos\alpha_j, j\neq y_i\} \\ &= \frac{1}{s}\log(C-1) + \max\{\cos\alpha_j, j\neq y_i\} \end{aligned} \quad (5)$$

So we usually use the LSE function as an approximation of the max function:

$$LSE(s,\cos\alpha_j) \approx \max\{\cos\alpha_j, j\neq y_i\} \quad (6)$$

Here we make a transformation to the form of existing margin-based loss function:

$$\begin{aligned} \mathcal{L}_u &= \log\frac{e^{s\mathcal{F}(\alpha_{y_i})} + \sum_{j=1,j\neq y_i}^{C} e^{s\cos\alpha_j}}{e^{s\mathcal{F}(\alpha_{y_i})}} \\ &= \log\left(1 + e^{s\left(\frac{1}{s}\log\sum_{j=1,j\neq y_i}^{C} e^{s\cos\alpha_j} - \mathcal{F}(\alpha_{y_i})\right)}\right) \end{aligned} \quad (7)$$

Finally, according to $Property1$, we can get the transformation form as follows:

$$\mathcal{L}_u = s * Softplus\left(\frac{1}{s}\log\sum_{j=1,j\neq y_i}^{C} e^{s\cos\alpha_j} - \mathcal{F}(\alpha_{y_i})\right)$$
$$\approx s * Softplus\Big(\max\{\cos\alpha_j, j\neq y_i\} - \mathcal{F}(\alpha_{y_i})\Big) \tag{8}$$

For (8), the optimization goal is to make $\mathcal{F}(\alpha_{y_i}) > \max\{\cos\alpha_j, j\neq y_i\}$. For the example in Figure 1, the goal is to make $\mathcal{F}(\alpha_1) > \cos\alpha_2$. Thus, it focuses on narrowing the intra-class angle between x and w_1 as well as widening the inter-class angle between x and w_2. Figure 1 shows that, rather than only consider the angle between x and w_2, we should also pay attention to widening the angle between w_1 and w_3. The existing methods fail to address this problem. Consider our loss function again:

$$\mathcal{L}_{Eog} = -\log\frac{e^{s\mathcal{F}(\alpha_{y_i})}}{e^{s\mathcal{F}(\alpha_{y_i})}+\sum_{j=1,j\neq y_i}^{C}(e^{s\cos\alpha_j}+e^{s\cos\beta_j})} \tag{9}$$

According to the transformation process of $\mathcal{L}_u$, The transformed form of (9) is as follows:

$$\mathcal{L}_{Eog} = s * Softplus\Big(\max\{\cos\alpha_j, \cos\beta_j, j\neq y_i\} - \mathcal{F}(\alpha_{y_i})\Big) \tag{10}$$

The aim of (10) is to make $\mathcal{F}(\alpha_{y_i}) > \max\{\cos\alpha_j, \cos\beta_j, j\neq y_i\}$. In Figure 1, considering the case of $\cos\alpha_2 < \cos\beta$, the goal of EogFace is to make $\mathcal{F}(\alpha_1) > \cos\beta$. Thus, EogFace can explicitly widen the distance between $w1$ and $w3$ to achieve the aim of inter-class separability.

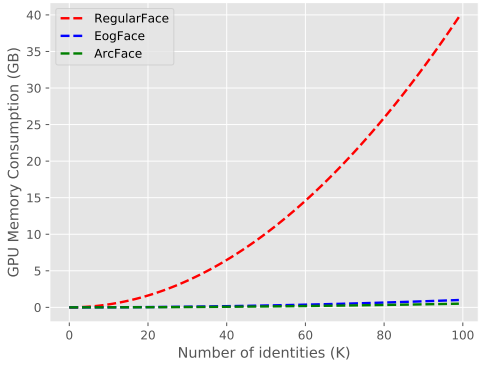


Fig. 2. GPU memory consumption of RegularFace, EogFace and ArcFace.

3.3. GPU Memory Consumption

RegularFace[16] has to calculate all the angular distance between different class centres to penalize the angle between an identity and its nearest neighbor. For a dataset with large number of identities, the computation is memory-consumptive and inefficient.

In order to quantitatively calculate the gpu memory consumption of different loss function in the classified layer, we draw the curve of the change of GPU memory consumption as the number of identities increases. In Figure 2, we can see that with the increase of identites, the GPU memory consumption of RegularFace increases rapidly compared with ArcFace. So this expensive GPU memory consumption limits its widespread use. EogFace, by contrast, doesn't add much gpu memory consumption.

4. EXPERIMENTS

4.1. Implementation Details

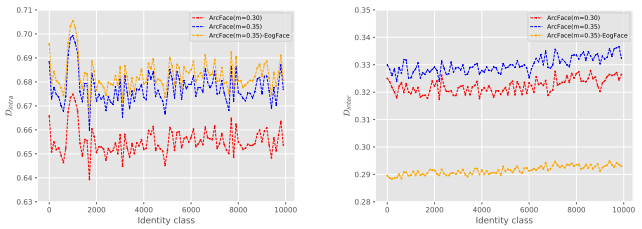
Datasets. We trained the proposed models on the publicly available CASIA-WebFace[22] dataset containing about 0.5M images of 10K identities. For fair comparisons with other methods, we extensively test the proposed method on various popular face datasets, such as LFW[19], IJB-B[17], IJB-C[18] and MegaFace[20].

Training Setting. We follow ArcFace[8] to normalize the images to 112 × 112 with five landmarks. For the embedding network, we employ ResNet18[8]. The proposed method is implemented by Pytorch[23]. We train models with batch size 256, using SGD algorithm, with momentum 0.9 and weight decay $5e-4$, and the learning rate starts from 0.1, it is divided by 10 at 20, 28 and 32 epochs, and stops the training process at 34 epochs.

4.2. Ablation Study

Table 1. The 1:1 verification accuracy on the LFW. Identification and verification evaluation on MegaFace Challenge1 using FaceScrub as the probe set. "Id" refers to the rank-1 face identification accuracy with 1M distractors, and "Ver" refers to the face verification TAR@FPR=1e-6.

Method	LFW	Id	Ver
Softmax	97.88	52.86	65.93
Center Loss [12]	98.91	65.23	76.52
L-Softmax[24]	99.01	67.13	80.42
SphereFace[9]	99.26	69.62	83.16
RegularFace[16]+[9]	**99.33**	70.23	84.07
ArcFace(m=0.25)	99.06	69.55	84.29
ArcFace(m=0.30)	99.15	71.52	87.49
ArcFace(m=0.35)	99.13	72.34	87.57
ArcFace(m=0.35)+ RegularFace	99.16	72.41	87.54
ArcFace(m=0.35)+ EogFace	99.23	**72.61**	**88.02**



(a) (b)

Fig. 3. Intra-class and inter-class distance curve trained on CASIA-WebFace with 10K identities.

Results on LFW and MegaFace. ArcFace is one of the most popular loss function. In order to prove the inter-class separability effect of EogFace, we plug it into ArcFace. we trained ArcFace with different parameters m. The results are listed in table 1, where we can see that EogFace can significantly improve the performance of ArcFace.

Compactness and Separability Analysis. In order to prove the effectiveness of EogFace more intuitively, we quantitatively evaluate the intra-class compactness and inter-class separability based on the weight parameter W in the classified layer. The weight parameter W is a matrix of $D \times C$, where $w_{y_j} \in R^D$ is the y_j-th column denotes the weight vector for the y_j-th identity class. We measure the intra-class compactness of y_j-th identity class by

$$\mathcal{D}_{Intra} = \frac{1}{K}\sum_{i=1}^{K}\cos(\theta_{y_j,i}) = \frac{1}{K}\sum_{i=1}^{K}\frac{w_{y_j}\cdot x_i}{\|w_{y_j}\|\cdot\|x_i\|} \quad (11)$$

Where K denotes the sample numbers of the identity class, x_i is the feature of i-th sample. For the intra-class compactness, $\mathcal{D}_{Intra}$ is expected to be as large as possible. We measure the inter-class separability of the y_j-th class centre by

$$\mathcal{D}_{Inter} = \max_{y_j \neq i}\cos(\phi_{y_j,i}) = \max_{y_j \neq i}\frac{w_{y_j}\cdot w_i}{\|w_{y_j}\|\cdot\|w_i\|} \quad (12)$$

Where w_i is the i-th column of the W. For the inter-class separability, $\mathcal{D}_{Inter}$ is expected to be as small as possible. As can be seen from Figure 3, when parameter m changes from 0.3 to 0.35, The intra-class distance $\mathcal{D}_{Intra}$ of ArcFace increases obviously, but at the same time, the inter-class distance $\mathcal{D}_{Inter}$ also increases, which may be the reason why the performance improvement is no longer obvious. However, when EogFace is inserted, the inter-class distance can be significantly reduced while maintaining increasing the intra-class distance. Therefore, the performance in Table 1 is still significantly improved.

Training Convergence. We have drawn the accuracy On LFW during the training in Figure 4. After an epoch, the accuracy of ArcFace reaches 86.08%, the RegularFace reaches 94.66%, and the EogFace reaches 95.33%. Therefore, EogFace has faster convergence.

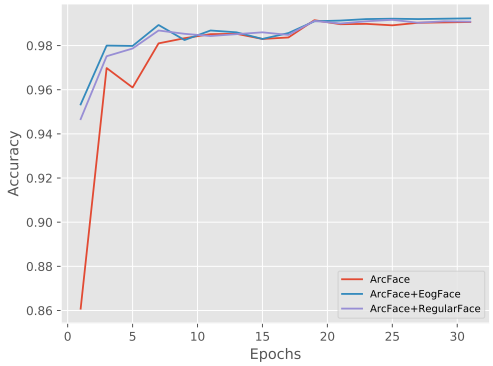


Fig. 4. The accuracy of LFW during training.

Table 2. 1:1 verification TAR@FAR=1e-4 on the IJB-B and IJB-C datasets.

Method	IJB-B	IJB-C
Softmax	68.15	73.83
Softmax+EogFace	**69.03**	**73.92**
CosFace[11]	79.09	83.84
CosFace+EogFace	**80.00**	**84.01**
ArcFace[8]	78.45	82.67
ArcFace+EogFace	**79.02**	**83.16**
CurricularFace[25]	79.15	83.87
CurricularFace+EogFace	**80.28**	**84.15**

Results on IJB-B and IJB-C. Finally, we plug EogFace into the most several popular loss functions, and evaluate the performance on the large-scale template face datasets IJB-B and IJB-C, which are closer to real-world surveillance scenarios and environments. The IJB-B[17] dataset contains about 1.8K subjects, including 21.8K still images and 7K videos. Based on IJB-B, the IJB-C[18] dataset is extended to about 3.5K subjects, including 31.3K still images and 11,779 videos. The verification accuracy on IJB-B and IJB-C are presented in Table 2, where we can see that EogFace has a consistent performance improvement on these popular loss functions.

5. CONCLUSION

In this paper, we proposed a novel loss function, which considers both the intra-class compactness and inter-class separability. The proposed method is easy to implement without much GPU memory consumption and need not to introduce additional hyper-parameters. The comparison with some existing state-of-the-art competitors on popular open face benchmarks showed that the proposed method achieved consistent improvement, demonstrating the superiority of our algorithm.

6. REFERENCES

[1] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen, "Face recognition with local binary patterns," in *ECCV*. Springer, 2004, pp. 469–481.

[2] Dong Chen, Xudong Cao, Fang Wen, and Jian Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *CVPR*, 2013, pp. 3025–3032.

[3] Karen Simonyan, Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman, "Fisher vector faces in the wild.," in *BMVC*, 2013, vol. 2, p. 4.

[4] Matthew A Turk and Alex P Pentland, "Face recognition using eigenfaces," in *CVPR*. IEEE Computer Society, 1991, pp. 586–587.

[5] Lior Wolf, Tal Hassner, and Yaniv Taigman, "Descriptor based methods in the wild," in *Workshop on faces in'real-life'images: Detection, alignment, and recognition*, 2008.

[6] Vytautas Perlibakas, "Distance measures for pca-based face recognition," *PRL*, vol. 25, no. 6, pp. 711–724, 2004.

[7] Chengjun Liu and Harry Wechsler, "Enhanced fisher linear discriminant models for face recognition," in *PR*. IEEE, 1998, vol. 2, pp. 1368–1372.

[8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *CVPR*, 2019, pp. 4690–4699.

[9] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," in *CVPR*, 2017, pp. 212–220.

[10] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015, pp. 815–823.

[11] 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 *CVPR*, 2018, pp. 5265–5274.

[12] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*. Springer, 2016, pp. 499–515.

[13] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *FG*. IEEE, 2018, pp. 67–74.

[14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014, pp. 1701–1708.

[15] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[16] Kai Zhao, Jingyi Xu, and Ming-Ming Cheng, "Regularface: Deep face recognition via exclusive regularization," in *CVPR*, 2019, pp. 1136–1144.

[17] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al., "Iarpa janus benchmark-b face dataset," in *CVPR workshops*, 2017, pp. 90–98.

[18] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al., "Iarpa janus benchmark-c: Face dataset and protocol," in *ICB*. IEEE, 2018, pp. 158–165.

[19] 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.

[20] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *CVPR*, 2016, pp. 4873–4882.

[21] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille, "Normface: L2 hypersphere embedding for face verification," in *ACM MM*, 2017, pp. 1041–1049.

[22] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[23] 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.

[24] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang, "Large-margin softmax loss for convolutional neural networks," *JMLR.org*, 2016.

[25] 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 recognition," *IEEE*, 2020.