2020 International Conference on Machine Vision and Image Processing (MVIP)
Faculty of Engineering, College of Farabi, University of Tehran, Iran, 19 & 20 Feb. 2020

Gaussian Soft Margin Angular Loss for Face Recognition

Bahman Rouhani
Computer Engineering Faculty
Amirkabir University of Technology
Tehran, Iran
brouhani@aut.ac.ir

Ali Samadzadeh
Computer Engineering Faculty
Amirkabir University of Technology
Tehran, Iran
a_samad@aut.ac.ir

Mohammad Rahmati
Computer Engineering Faculty
Amirkabir University of Technology
Tehran, Iran
rahmati@aut.ac.ir

Ahmad Nickabadi
Computer Engineering Faculty
Amirkabir University of Technology
Tehran, Iran
nickabadi@aut.ac.ir

***Abstract*—Advances in deep learning has lead to drastic improvements in face recognition. A key part of these deep models is their loss function. Consequently developing an efficient and suitable loss function has been an important topic in face recognition in the recent years. Angular-margin-based losses achieve an acceptable performance and inter-class separability. However they are held back by their enforcement of hard margins on all the samples of the training dataset, regardless of whether these samples actually differ from all the other classes enough to enforce a margin. It can be argued that in a large enough dataset with many different settings and age gaps, some faces will look similar to the faces of other classes. In an intuitive and expressive embedding, we expect some faces to be embedded near similar classes with a small margin. Thus we propose a loss function that while maximizing the inter-class distance and intra-class compactness, allows for the samples which naturally reside further from class center to have a smaller margin. We implement an extremely light and fast to train model using MobileNets and achieve accuracy comparable to state of the art method.**

***Index Terms*—Computer Vision, Face Recognition, Angular-Margin-Based Loss, Loss Function, Deep Learning.**

I. Introduction

Face recognition (FR) is one of the oldest, most important and widely used applications of computer vision. It is used in a variety of different fields such as a security measure for access management, criminal identification [1], surveillance [2] [3] [4] and even payment methods [5]. Despite the widespread use and evergrowing demand for reliable FR methods, there are still number of challenges. Namely pose, illumination and general setting of a photo which can affect the accuracy of FR models [6]. Additionally there might be age differences among images belonging to a certain identity. In some applications there might be occlusions or facial expressions that pose challenges for the FR methods.

In the recent years and after the impressive success of deep learning, many deep models have been introduced for FR [7] [8] [9] [10] [11]. Generally in FR the faces which we want to recognized in the testing phase are unknown at the time of training. Therefore it is simply not practical to train a network that differentiates between a set of known faces, rather these models usually employ a network that is very efficient in extracting discriminative features of faces. In computer vision and machine learning most improvements are achieved by developing new architectures. However it was discovered early on that employing a suitable loss function plays a key role in face recognition. This is perhaps because face samples have a great deal of similarities compared to other tasks (e.g object recognition where samples of two different classes are usually a lot more different). As a result, a good loss function is needed in order to help the model learn to extract optimal features.

Recently angular-margin-based losses have been introduced which try to enforce a margin between samples of different classes [12] [13] [14] [15]. While these methods have been very successful, we hypothesize that they can be improved greatly by designing a smarter, more natural margin. In particular, the current angular-margin models enforce a constant margin onto all the samples regardless of whether these samples can be intrinsically mapped into distinct regions of the feature space with large margins or not. Here we propose a novel loss function that considers this natural limitations while enforcing a margin on the samples.

Our main contributions in this paper are a new loss function called Gaussian Soft Margin Angular Loss (GSMA) and analyzing this loss function. We also implement our model with a MobileNet [16] architecture which results in a fast and light network that is usable even on mobile devices, where face recognition is a much needed task. We test our model on AgeDB-30 [17] which is considered a challenging dataset.

The remaining parts of this paper are organized as follows: in the section II we discuss prior models and previous methods for FR. In the sections III and IV we explain our approach and present our results. Finally we explore future works in section V.

978-1-7281-6832-6/20/$31.00 ©2020 IEEE

II. Related works

The earliest methods for face recognition used manually and hand-designed representations of face in low dimensions such as the eigenface method introduced in 1990's [18]. Other early approaches also included finding low dimensional representations by assuming distributions about face images. Namely some of the most popular approaches were finding sparse representations [19] [20] [21] [22] and linear subspace [23] [24] [25]. These methods were mainly abandoned after the introduction of deep learning in 2012 [26]. Most of the recent state of the art FR methods have used deep learning as the basis of their work. These methods mainly vary in face processing, architecture and loss functions and can be categorized based on these criteria [6] [27].

A. Face Processing

Deep learning methods are drastically more robust to light or pose changes than the previous hand-designed approaches [27] [6]. However, they are still prone to errors when presented with pose, age and setting variations specially in cases where there hasn't been a form of face processing in order to prepare the model for these variations. These processes are denoted as face processing. In [6] FR methods are categorized based on their face processing approach into two main categories of "one-to-many augmentation" and "many-to-one normalization". The first are methods that by having a single image, try to generate enough patches in order for the model to learn setting and pose-invariant features. The latter take the different path of reconstructing a standard and canonical representation or view of a face from a number of (or perhaps only one) images provided to the model.

B. Architecture

The architecture of deep FR methods vary in many ways including depth, the types of layers, and the learning mechanisms. Perhaps an early example of using deep learning in FR was DeepFace [7] which was one of the first to use several CNN layers. One year later at 2015 FaceNet [28] was inspired by GoogleNet [29] which achieved a remarkable accuracy for the time (99.63% on LFW) exceeding DeepFaces accuracy by around 2%. At the same year, VGGFace [30] was introduced which adopted the architecture of VGGNet [31], a neural network known for its superb ability in extracting good features.

C. Loss Function

Previously, the cross-entropy based softmax loss was widely used in classification problems. However, softmax loss was not successful and sufficient for FR task as the resulting classifier dos not enforce large margins between different classes. As a result, part of research in FR was devoted to finding better and more expressive loss functions, tailored and well-suited for FR task. In general these new loss functions can be categorized as below:

1) Softmax-based Losses: These methods concentrate on improving softmax loss. One way to do so is by normalizing features or weights (or both, as done in the angular loss function:) as:

$$\hat{W} = \frac{W}{\|W\|} \text{ and } \hat{x} = \alpha \frac{x}{\|x\|}, \tag{1}$$

where W is the weight vector for a class and x is a feature vector extracted from a sample. After calculating $\hat{W}$ and $\hat{x}$, W and x are replaced by them. The $\|.\|$ denotes a Euclidean norm. It has been shown that the constant α is necessary in order to prevent the model from being trapped while training [32].

There are few other methods of improving softmax beside normalization. For instance noisy softmax was introduced in [33] in order to even out the early saturation in softmax.

2) Euclidean-Distance-based Losses: As the name suggests, Euclidean distance loss measures the distance of two samples in the Euclidean space, that is, the network learns to map the samples into a Euclidean space, and the loss function tries to minimize distance between similar samples while maximizing the distance between samples of different classes. There are a number of important Euclidean distance losses. Perhaps the most important of them are contrastive, triplet and center loss.

Contrastive loss [34] [35] [36] is defined as follows:

$$\begin{aligned}\mathbb{L} = y_{ij} max(0, \|f(x_i) - f(x_j)\|_2 - \epsilon^+) \\ + (1 - y_{ij}) max(0, \epsilon^- - \|f(x_i) - f(x_j)\|_2),\end{aligned} \tag{2}$$

where for two samples of the same class $y_{ij} = 1$ and for two samples of different classes $y_{ij} = -1$. The f function denotes feature extraction or embedding and ϵ^+ and ϵ^- are margin terms for intra-class and inter-class cases respectively. In other words, this loss takes two samples and pulls them close if they are from the same class and pushes them far from each other in case they are of different classes.

We should note that the contrastive loss considers the absolute distance between two samples. This is where the triplet loss differs which calculates the relative distance between two samples. This loss was published by google along with [28]. Given 3 face samples, the triplet loss minimizes the distance between an anchor and another sample of class and maximizes the distance between the anchor and a sample of a different class.

Both mentioned losses can be somewhat unstable while training. Center loss [37] was introduced as an stable Euclidean distance loss after observing these difficulties. The center loss tries to find a center for each class and penalize the distance of the samples of each class from the center of that class and is defined as:

$$\mathbb{L}_C = \frac{1}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 , \tag{3}$$

in which x_i is the feature vector of the ith sample, y_i is its class and c_j denotes center of class j.

TABLE I
MOBILENET IN COMPARISON TO OTHER STATE OF THE ART NETWORKS

model name	accuracy on ImageNet	Million parameters
MobileNet	60.2 %	1.32
SqueezeNet	57.5 %	1.25
AlexNet	57.2 %	60

3) Angular-margin-based Loss: In 2016, [13] introduced L-Softmax which marginalizes the angle between the class center (or the column corresponding to the class in the weight matrix) and a sample. This loss requires that:

$$\|W_1\| \, \|x_i\| cos(m \times \theta_{1i}) > \|W_2\| \, \|x_i\| cos(m \times \theta_{2i}), \quad (4)$$

where W_i is the column in the weight matrix of softmax loss for the ith class (which is also the center for the ith class) and θ_{ij} is the angle between the ith class center and features extracted from the jth sample. And is defined as:

$$\mathbb{L}_i = -log\Big(\frac{e^{\|W_{y_i}\|\|x_i\|\psi(\theta_{y_i})}}{e^{\|W_{y_i}\|\|x_i\|\psi(\theta_{y_i})} + \sum_{j \neq y_i} e^{\|W_{y_i}\|\|x_i\| cos(\theta_{y_i})}}\Big), \quad (5)$$

in which its required that:

$$\psi(\theta) = \begin{cases} cos(m\theta), & 0 \leq \theta \leq \frac{\pi}{m} \\ D(m\theta), & \frac{\pi}{m} < \theta \leq \pi \end{cases}. \quad (6)$$

In above formula, m is an integer related to the margin, $D(\theta)$ needs to be a monotonically decreasing function and $D(\pi/m)$ should equal $cos(\pi/m)$.

L-Softmax is good at marginalizing features but makes convergence hard. ArcFace [12] and CosFace [14] were introduced as result. The ArcFace (Additive Angular Margin Loss) is defined as:

$$\mathbb{L}_{ArcFace} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(s\, cos(\theta_{y_i} + m))}{exp(s\, cos(\theta_{y_i} + m)) + \sum_{j=1, j\neq y_i}^{n} exp(s\, cos\theta_j)}, \quad (7)$$

while Large Margin Cosine Loss (introduced in CosFace) is defined as:

$$\mathbb{L}_{lcmc} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(s\, cos(\theta_{y_i} - m))}{exp(s\, cos(\theta_{y_i} - m)) + \sum_{j=1, j\neq y_i}^{n} exp(s\, cos\theta_j)}. \quad (8)$$

In both of these equations s is a constant, m is the margin term and θ, x_i and y_i are denoted as before. Our model builds on the ArcFace loss which has shown the best performance among all three of these loss functions.

III. OUR APPROACH

Typically face recognition in the training phase consists of face detection, feature extraction and finally choosing the best match based on the extracted features by a fully connected layer. Consequently a loss function is applied to the output of this last layer to train the network. The goal is not to learn the faces presented in the dataset, but for the network to learn to extract expressive and discriminative features. This means the loss function will have a drastic effect on what features are learned.

In the testing phase after the face detection, we use the network (except for the last layer) to extract a feature vector from a given sample, this feature vector is then compared to known vectors in order to identify the class of sample.

Our method uses MTCNN [38] for face detection. For feature extraction network we use a lighter version of architecture proposed by [12]. To make our model lighter, a class of artificial neural networks called MobileNets [16] are used. This makes our model to consist of less than 1 million parameters which occupies less than 5 megabytes of disk space, which is much lighter than the original model. Table I compares a MobileNet to other state of the art networks in accuracy and parameter size on ImageNet dataset [39]. A visual demonstration of our model and its modules is shown at Fig. 1.

In the following parts of this section we describe our proposed loss function.

A. Softmax Loss as inner product

After extracting features from a face image, the features are usually used for a classification task via one or multiple final fully connected layers. It is mostly the case that these layers try to minimize a Softmax Loss function, which can be written as:

$$\mathbb{L}_1 = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(W_{y_i}^T x_i + b_{y_i})}{\sum_{j=1}^{n} exp(W_{y_j}^T x_i + b_{y_j})}, \quad (9)$$

where $x_i \in \mathbb{R}^d$ is the features extracted from ith sample the class of which we denote by y_i. Here, d denotes the embedded feature dimension which is set to 512 in [37] and [15]. $W \in \mathbb{R}^{d\times n}$ is the weight matrix and each column $W_j \in \mathbb{R}^d$ denotes the column corresponding to the jth class. $b \in \mathbb{R}^n$ is the bias, n is the number of classes and finally N is the batch size.

If we eliminate the bias term (by setting it to 0) we can rewrite the $W_j^T x_i + b_j$ in the formula (9) as:

$$W_{y_i}^T x_i + b_{y_i} = W_{y_i}^T x_i = \|W_j\|\|x_i\| cos\theta_j, \quad (10)$$

where θ_j denotes the angle between jth class center (W_j) and the features from ith sample (x_i). Set the W_j and x_i to 1 and a scale s respectively for simplicity. This means we will be able to write the (10) as:

$$W_{y_i}^T x_i + b_{y_i} = s\, cos\theta_j. \quad (11)$$

We will then write the (9) as:

$$\mathbb{L}_2 = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(s\, cos\theta_{y_i})}{\sum_{j=1}^{n} exp(s\, cos\theta_j)}. \quad (12)$$

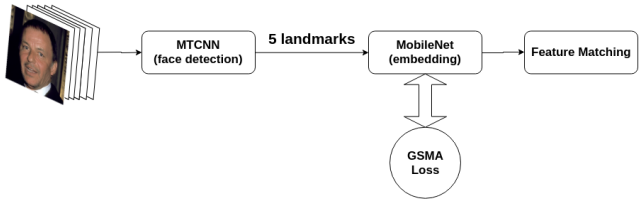

Fig. 1. The pipeline for our model. Faces are detected using MTCNN network, MobileNet is trained using our novel loss function and finally feature matching is applied for FR.

B. Angular-margin losses

Occasionally, Softmax loss is used in FR methods. However, a big downside of using this loss function is that it does not enforce the intra-class features to be similar and have small distances or the inter-class features to have a margin from each other. This stops deep FR methods from fulfilling their full potential.

As shown in (11), there is a geometric interpretation to definition of this specific case of Softmax loss, that is, the loss function is penalizing the angle between the feature vector extracted from a sample belonging to the sample i and the column in weight vector corresponding to the class y_i. As mentioned earlier, a weakness of Softmax loss is its inability to concentrate features of same classes around the class center and to maintain a margin between feature vectors belonging to different classes. To tackle this issue, several methods have been proposed recently [15] [14] [13]. Most importantly in [12] a loss function is proposed as:

$$\mathbb{L}_3 = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(s\ cos(\theta_{y_i} + m))}{exp(s\ cos(\theta_{y_i} + m)) + \sum_{j=1,j\neq y_i}^{n} exp(s\ cos\theta_j)}, \tag{13}$$

where the added term m penalizes not only the angle between the feature vector and class center but also the angle within a margin of feature vector and the class center.

C. GSMA

The above method has proved highly efficient as demonstrated in [12]. However, we hypothesize that in a large-scale dataset with a great variety of poses and settings, it will be very hard for the network to learn a feature space that discriminates between different classes with a hard margin as proposed in [12]. We suspect that if the dataset contains a diverse enough collection of poses and settings of each face, it is likely that a few samples will have features close to that of other classes. Therefore we believe it would be beneficiary to propose a loss function that acknowledges the intrinsic characteristics of the problem. By doing so, we hope that the learning procedure for face recognition will speed up while moere robust and intuitive features are learned by the network.

To this end, we propose **Gaussian Soft Margin Angular Loss** (**GSMA**). We assume that a value for m would be samples drawn from a Gaussian distribution centered at the distance M_{old} of the class center for features of each class where M_{old} is the fixed margin in previous methods. We want the margin for most of the features to be less than or equal to the hard margin in previous methods so we set the standard variation of this distribution to $\frac{M}{2}$. We set $M = 0.5$ in our implementation. Our loss function is defined as:

$$\mathbb{L}_{GSMA} = \mathbb{E}_{m\sim\mathcal{N}(0,m/2)}\Big[-\frac{1}{N}\sum_{i=1}^{N} log\big(f(i,m)\big)\Big], \tag{14}$$

in which:

$$f(i,m) = \frac{e^{s\ cos(\theta_{y_i}+(M_{old}-\|m\|))}}{e^{s\ cos(\theta_{y_i}+(M_{old}-\|m\|))} + \sum_{j=1,j\neq y_i}^{n} e^{s\ cos\theta_j}}. \tag{15}$$

In above equation the term $M_{old} - \|m\|$ denotes a random variable which is concentrated around M_{old} but might be close to 0 at times. Thus allowing the less expressive samples to reside further from the class center.

IV. Experimentation and Results

A. Implementation

The implementation of our model can be divided into 3 main modules: Face Detection, MobileNet and the Loss Function. These 3 modules together alongside an arbitrary feature matching algorithm make up the pipeline for our FR model.

1) Face Detection: We use a pretrained version of MTCNN [38] for detecting the faces. This network has shown exemplary accuracy and cutting edge performance in face detection. The detected faces are then aligned based their 5 key landmarks (provided by MTCNN) and cropped into a fixed size of 112×112 and are normalized.

2) MobileNet: MobileNets [16] were introduced in 2017 by Howared et al. This network employ a depthwise separable convolution which drastically reduces the parameter count in the model and greatly contributes to the models light weight. As mentioned above, our model is able to perform with an accuracy comparable to the state of the art models with under 1 million parameters and 5 megabytes in model size. Coupled with our proposed new loss function, this architecture makes our model considerably fast compared to other models. As mentioned before, this model is trained on the AgeDB-30 [17] dataset.

TABLE II
Our proposed model in comparison to the ArcFace method

model name	accuracy on ImageNet
GSMA (proposed method)	**92.1** %
ArcFace (our implementation on MoblienNet)	91.3%
ArcFace (original implementation)	95.15%

3) Loss Function: In practice, it would be somewhat time consuming to calculate the expected value in (14); instead we sample the margin m at from a Gaussian distribution for each sample. Since every sample will be presented to the network a number of times before the training finishes, this will act similar to an approximation of expected value.

B. Results

We train and test our model on AgeDB-30 [17]. We implement the previous state of the art models and our model with MobileNet architecture. Our implementation of ArcFace model has the accuracy of 91.3. Our proposed method is able to achieve the accuracy of 92.1.

This improvement of accuracy is accompanied with slightly better time performance in our model in comparison to previous models.

It is worth noting that implementing the model in a MobileNet architecture affects the accuracy. While this architecture employs a very smaller set of parameters and is highly faster to train, it results to a somewhat lower accuracy than that of the more complex models. Our results along with the accuracy for our implementation of Arcface network and its original accuracy are presented in table II. Note that our model performs better than the ArcFace model when both are implemented by MobileNets. However the accuracy reported in the original paper [12] is obtained by a much bigger network than the MobileNet and so it is only natural that it should have higher accuracy.

V. Concluding Remarks and Future Work

We implemented our loss function by sampling the margin from a Gaussian distribution. We suspect that this approximation will reduce the models accuracy. As a next step, it would be intuitive to try more accurate ways to approximate our loss function. One way would be to set out to calculate the expected value. However it is also possible to develop a heuristic which will be able to determine how *well represented* a sample is i.e. approximate how close each sample is to the class center (or in other words how many useful features are present in the sample). We would then be able to penalize each sample with a proportional margin.

We hypothesized that our model will develop more robust features since our loss function models the problem in a more natural and intuitive way. We plan to test this theory in future work.

References

[1] X. Chen, L. Qing, X. He, J. Su, and Y. Peng, "From eyes to face synthesis: a new approach for human-centered smart surveillance," *IEEE Access*, vol. 6, pp. 14 567–14 575, 2018.

[2] T. Min and F.-q. ZHONG, "Deep learning-based new methods of images processing for criminal investigation," *DEStech Transactions on Computer Science and Engineering*, no. iece, 2018.

[3] U. Zafar, M. Ghafoor, T. Zia, G. Ahmed, A. Latif, K. R. Malik, and A. M. Sharif, "Face recognition with bayesian convolutional networks for robust surveillance systems," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, p. 10, 2019.

[4] J. Lin, L. Xiao, and T. Wu, "Face recognition for video surveillance with aligned facial landmarks learning," *Technology and Health Care*, vol. 26, no. S1, pp. 169–178, 2018.

[5] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[6] M. Wang and W. Deng, "Deep face recognition: A survey," 2018.

[7] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[8] X. Wu, R. He, and Z. Sun, "A lightened cnn for deep face representation," *arXiv preprint arXiv:1511.02683*, vol. 4, no. 8, 2015.

[9] C. Xiong, X. Zhao, D. Tang, K. Jayashree, S. Yan, and T.-K. Kim, "Conditional convolutional neural network for modality-aware face recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3667–3675.

[10] Y. Sun, X. Wang, and X. Tang, "Sparsifying neural network connections for face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4856–4864.

[11] A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller, "One-to-many face recognition with bilinear cnns," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[12] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[13] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *ICML*, vol. 2, no. 3, 2016, p. 7.

[14] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[15] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[17] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "Agedb: the first manually collected, in-the-wild age database," vol. 2, no. 3, p. 5, 2017.

[18] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[19] W. Deng, J. Hu, and J. Guo, "Extended src: Undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1864–1870, 2012.

[20] ——, "Face recognition via collaborative representation: Its discriminant nature and superposed representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 10, pp. 2513–2521, 2017.

[21] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *2011 International conference on computer vision*. IEEE, 2011, pp. 471–478.

[22] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2008.

[23] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 711–720, 1997.

[24] B. Moghaddam, W. Wahid, and A. Pentland, "Beyond eigenfaces: Probabilistic matching for face recognition," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE, 1998, pp. 30–35.

[25] W. Deng, J. Hu, J. Lu, and J. Guo, "Transform-invariant pca: A unified approach to fully automatic facealignment, representation, and recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1275–1284, 2013.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA:

Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[27] G. Guo and N. Zhang, "A survey on deep learning based face recognition," *Computer Vision and Image Understanding*, vol. 189, p. 102805, 2019.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[30] O. M. Parkhi, A. Vedaldi, A. Zisserman *et al.*, "Deep face recognition." in *bmvc*, vol. 1, no. 3, 2015, p. 6.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[32] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: l 2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*. ACM, 2017, pp. 1041–1049.

[33] B. Chen, W. Deng, and J. Du, "Noisy softmax: Improving the generalization ability of dcnn via postponing the early softmax saturation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5372–5381.

[34] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2892–2900.

[35] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in neural information processing systems*, 2014, pp. 1988–1996.

[36] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[37] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[38] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.