Scalable Similarity-Consistent Deep Metric Learning for Face Recognition

BOWEN WU[1,3], (Member, IEEE), AND HUAMING WU[2], (Member, IEEE)
[1]JD Intelligent Cities Business Unit, Beijing 100176, China
[2]Center for Applied Mathematics, Tianjin University, Tianjin 300072, China
[3]Center for Combinatorics, Nankai University, Tianjin 300071, China

Corresponding author: Huaming Wu (whming@tju.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0809800, in part by the National Natural Science Foundation of China under Grant 61801325, and in part by the Huawei Innovation Research Program under Grant HO2018085347 and HO2018085138.

ABSTRACT With the development of deep learning, deep metric learning (DML) has achieved great improvements in face recognition. Specifically, the widely used softmax losses in the training process often bring large intra-class variations, and feature normalization is only exploited in the testing process to compute the pair similarities. To bridge the gap, we impose the intra-class cosine similarity between the features and weight vectors in softmax loss larger than a margin in the training step and extend it from four aspects. First, we explore the effect of a hard sample mining strategy. To alleviate the human labor of adjusting the margin hyper-parameter, a self-adaptive margin updating strategy is proposed. Then, a normalized version is given to take full advantage of the cosine similarity constraint. Furthermore, we enhance the former constraints to consider the intra-class and inter-class constraints simultaneously in the exponential feature projection space. The extensive experiments on the labeled face in the wild (LFW), youtube faces (YTF), and IARPA Janus benchmark A (IJB-A) datasets demonstrate that the proposed methods outperform the mainstream DML methods and approach the state-of-the-art performance.

INDEX TERMS Deep metric learning, face recognition, convolutional neural network, intra-class similarity, inter-class similarity, cosine similarity.

I. INTRODUCTION

Face recognition has been one of the most challenging and attractive areas in computer vision, due to its close relationship with some actual applications, such as biometrics and surveillance. However, face recognition problem is far from solved, since it is closely related to face detection, face alignment, feature extraction (or face representation) and classification, which influence the final performance from different aspects. Especially, feature extraction plays a paramount role. Conventional feature extraction methods (such as LBP, Gabor and SIFT) always work with suitable metric distances (such as Euclidean distance and cosine distance). However, the features extracted by these methods are not discriminative enough to meet the demands for more complex face recognition scenarios. And the situation may be worse when accompanied by inappropriate metric distances.

Convolutional Neural Network (CNN), which emerges as a powerful feature extraction method, has drawn much attention due to its excellent performance in computer vision community. Several Deep Metric Learning (DML) methods, which unify deep learning and metric learning into a joint learning framework, have been proposed and successfully employed in various visual tasks, such as objection classification [1], [2], image retrieval [3], [4], person reidentification [5], and so on. Specifically, DML has surpassed the humans'abilities on some benchmark datasets in the field of face recognition [6]–[12].

Face recognition can be classified into two tasks, namely face identification and face verification (Fig. 1). The former aims to classify an input image to a specific identity, while the latter is to determine whether a pair of face images are from the same identity or not. In general end-to-end CNN based face recognition, Euclidean distance is used to measure the similarities between features in the training process. Whereas, the cosine similarity or normalized inner product

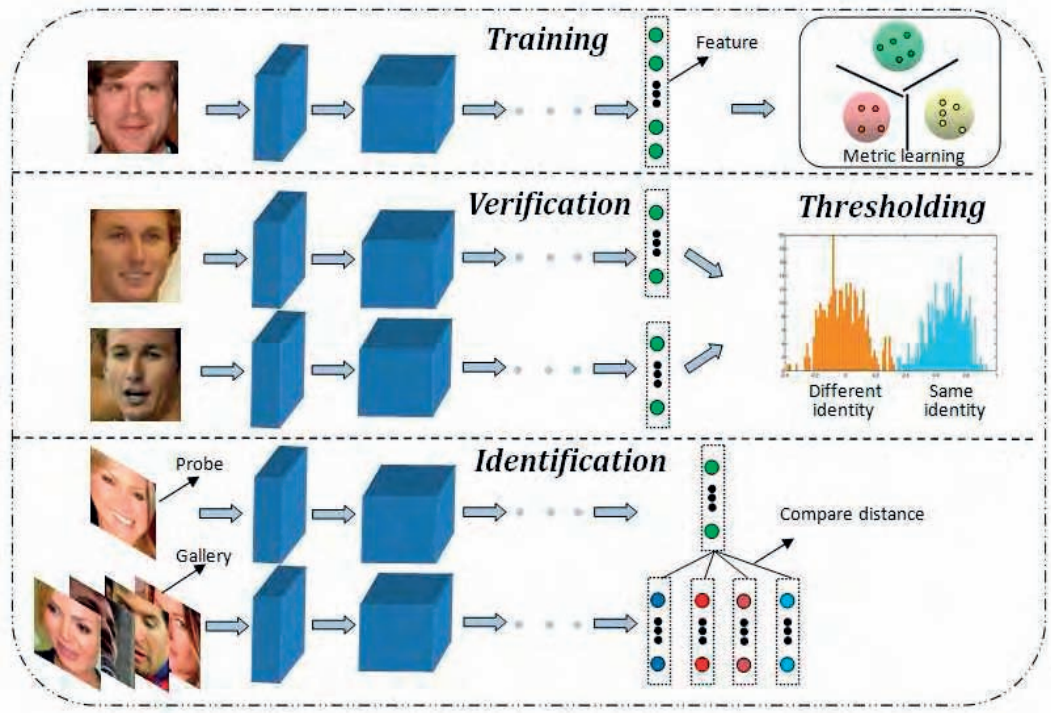


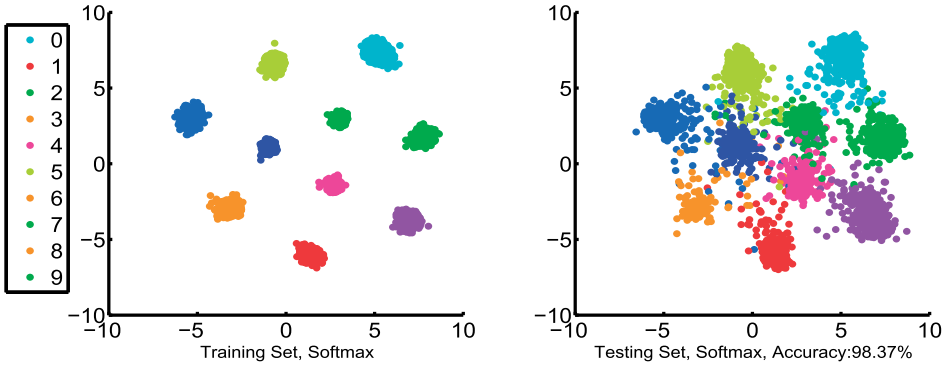
FIGURE 1. The face recognition pipeline in this paper.

is widely used in the testing process. As illustrated in [9], Euclidean distance or Euclidean margin-based loss is not always suitable for learning discriminative features, and using normalized features to compute the pair similarities for testing can boost the performance. These properties motivate some work [13], [14] to incorporate the cosine similarity constraint into training stage to keep the consistency with testing. However, they have not explored the effect of important samples which violate the margin constraint between intra-class and inter-class variations. Inspired by the large intra-class variation of softmax loss, which is not expected in classification task, we first force the intra-class cosine similarity larger than a given margin in this paper. Combined with the separability of softmax loss, our original method achieves $0.6\% \sim 0.8\%$ accuracy improvement on Labeled Face in the Wild (LFW) dataset and $1\% \sim 1.5\%$ accuracy improvement on Youtube Faces (YTF) dataset. Some previous work [7], [15] have clarified the importance of hard sample mining procedure in training CNN, but they haven't exhibited the specific comparative experiment results about whether to use it or not. Here, we compare the effect of cosine similarity constraint on the original training set and the misclassified hard samples of softmax loss. For the diversity of data and the ubiquitously heterogeneous distribution, the global cosine similarity is insufficient to faithfully characterize the true feature distance, as stated in [16]. A self-adaptive margin updating technique is exploited afterwards, so that the local uniqueness of each identity is considered and the human labor of adjusting the margin is largely saved. To acquire more discriminative features, only imposing the intra-class cosine similarity larger than a margin doesn't seem to be the best choice. So we improve the former constraints to a more powerful case, which enforces the intra-class cosine similarity larger than the mean of the nearest neighboring inter-class cosine similarities in the normalized exponential feature projection space.

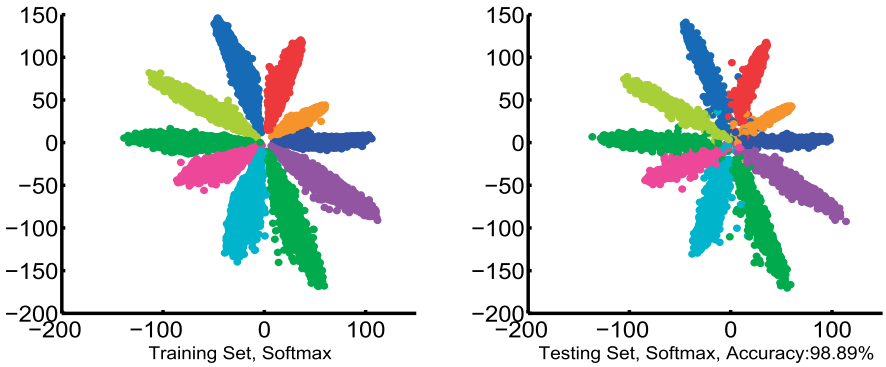
In conclusion, to improve the performance on face recognition, we have explored some different strategies from the view of loss function. The major contributions can be summarized as follows: 1) We first propose a novel metric loss function to directly force the intra-class cosine similarity larger than a fixed margin, so that the training process coincides with the normalized testing criterion. 2) We conduct a contrastive experiment to show the effect of a hard sample mining strategy on the proposed loss function. 3) A self-adaptive margin strategy about the updated feature space is incorporated to strengthen the supervision in training. 4) To avoid the side effects of infinitely growing norm of features, we further normalize the features and weight vectors of softmax loss to the same value in each mini-batch. 5) A more progressive metric loss function to consider the intra-class and inter-class variations simultaneously is proposed to achieve the discriminative features. Finally, we conduct extensive experiments on three face recognition benchmark datasets, namely LFW [17], YTF [18] and IARPA Janus Benchmark A (IJB-A) [19], to verify the excellent performance of our approaches.

II. THE PROPOSED APPROACHES

In this section, we reveal the existing phenomenon of large intra-class variation in deeply learned features trained by



(a)



(b)

FIGURE 2. **Visualization of the deeply learned 2-D features on MNIST with (a) MNIST network and (b) LeNet++ network.**

softmax loss, and propose several novel metric loss functions to alleviate this problem.

A. RECALLING SOFTMAX LOSS

From the viewpoint of probability, softmax function aims to convert a vector of real weights to a probability distribution. The original softmax loss is the cross entropy of softmax function, which can be written as

$$\mathcal{L}_{\mathcal{S}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{N} e^{W_j^T x_i + b_j}}, \tag{1}$$

where M is the number of training samples, N is the number of classes, x_i is the feature of the i-th sample, y_i is the corresponding class label in range $[1, N]$, W and b are the weight matrix and bias vector of the last inner-product layer before softmax loss, W_j is the j-th column of W and b_j is the corresponding bias term. For simplicity, we omit the bias term in the following experiments, as in [13]. Understandingly, if all classes are well-separated, W_j will roughly correspond to the mean of features in j-th class, and it can also be recognized as the center of j-th class in general cases.

To visualize the effects of softmax loss on diverse cases, we conduct a contrastive experiment on the MNIST dataset [20] with two different CNNs, namely MNIST network [21] and LeNet++ network [8]. We reduce the number of the last feature dimension to 2, so the features can be plotted directly on the 2-D surface. The resulting 2-D features of training and testing sets with the above two different networks are shown in Fig. 2. We can see that the deeply learned features are separable under the supervision of softmax loss, but not discriminative enough. Especially, there exist significant intra-class variations in the feature space of LeNet++ network, which coincides with the phenomenon elaborated in [13] that softmax loss encourages the features to have larger magnitudes.

B. LMC LOSS AND HLMC LOSS

To remove the large intra-class variation of softmax loss and to keep the consistency between training and testing, we first propose the Large Margin Cosine (LMC) loss function, which enforces the intra-class cosine similarity between a sample x_i and the corresponding weight vector W_{y_i} in the last inner-product layer before softmax loss larger than a given margin α. The LMC loss function is formulated as follows:

$$\mathcal{L}_{\mathcal{C}} = \frac{1}{M}\sum_{i=1}^{M}\left\{\alpha - \widetilde{W}_{y_i}^T\widetilde{x}_i\right\}_+, \tag{2}$$

where $\{\cdot\}$ is the hinge loss used in SVM, $\alpha \in [0, 1]$, $\widetilde{W}_{y_i} = \frac{W_{y_i}}{\|W_{y_i}\|_2}$, $\widetilde{x}_i = \frac{x_i}{\|x_i\|_2}$, and $\widetilde{W}_{y_i}^T\widetilde{x}_i$ corresponds to the cosine similarity of W_{y_i} and x_i.

Specifically, the joint supervision of softmax loss and LMC loss is necessary to train the CNN for learning discriminative features. The final LMC loss function for training is

$$\begin{aligned}\mathcal{L}_{\mathcal{LMC}} &= \mathcal{L}_{\mathcal{S}} + \lambda\mathcal{L}_{\mathcal{C}} \\ &= -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{N} e^{W_j^T x_i}} + \frac{\lambda}{M}\sum_{i=1}^{M}\left\{\alpha - \widetilde{W}_{y_i}^T\widetilde{x}_i\right\}_+,\end{aligned} \tag{3}$$

where λ is a weighting parameter that is used for balancing the two loss functions.

It is widely observed that there are often many more easy examples than the meaningful hard ones, an effective data sampling strategy is thus crucial to ensure the learning efficiency of deep features. Therefore, the Hard Large Margin Cosine (HLMC) loss function is proposed to impose the previous intra-class cosine similarity constraint on the hard samples. Here, we refer to the hard samples as the ones misclassified by softmax loss, which alleviates the costly computational complexity of pair/triple samples mining strategy adopted in the contrastive/triplet loss [7], [15].

$$\begin{aligned}\mathcal{L}_{\mathcal{HLMC}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{N} e^{W_j^T x_i}} \\ + \frac{\lambda}{M}\sum_{i=1}^{M}\gamma_i\left\{\alpha - \widetilde{W}_{y_i}^T\widetilde{x}_i\right\}_+,\end{aligned} \tag{4}$$

where $\gamma_i \in \{0, 1\}$ is a misclassified sample indicator, and $\gamma_i = 1$ if x_i is the misclassified sample of softmax loss.

C. MALMC LOSS

The existing metric loss functions like contrastive loss, triplet loss and L-Softmax loss, often bring in additional hyper-parameters as their fixed margins throughout the training process. An intractable hyper-parameter searching process is thus crucial to the successful training. Following the work [22], we have to suspend training and search for a new margin for every several epochs. In this part, we provide a Margin-Adaptive Large Margin Cosine (MALMC) method, which gives each class an independently updated margin and set it as the maximum of an initially given value and the mean of $p \times Intra(j)$ largest intra-class cosine similarities in the mini-batch.

$$\begin{aligned}\mathcal{L}_{\mathcal{MALMC}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{W_{y_i}^T x_i}}{\sum_{j=1}^{N} e^{W_j^T x_i}} \\ + \frac{\lambda}{M}\sum_{i=1}^{M}\left\{\alpha_{y_i} - \widetilde{W}_{y_i}^T\widetilde{x}_i\right\}_+,\end{aligned} \tag{5}$$

where $\alpha_j = max\left(\alpha_0, \frac{\sum_{i=1}^{M}\delta(j=y_i, i\in S_{p\times Intra(j)})\widetilde{W}_j^T\widetilde{x}_i}{1+\sum_{i=1}^{M}\delta(j=y_i, i\in S_{p\times Intra(j)})}\right)$, α_0 is an initially given margin, $\delta(\cdot)$ is the indicator function where $\delta(\cdot) = 1$ if the condition is satisfied and $\delta(\cdot) = 0$ for else, $Intra(j)$ is the number of intra-class cosine similarities between x_i and W_j in class j and these similarities are sorted in descending order, p is a predefined percentage to control the valid number of intra-class similarities. We refer to $S_{p\times Intra(j)}$ as the set including the indices of the largest $p \times Intra(j)$ similarities. Analytically, this self-adaptive margin strategy is more suitable for the realistic data distribution, relating the margin to the dynamic feature space and largely alleviating the multifarious human labor of adjusting the margin.

D. NLMC LOSS

Accompanying the cosine similarity constraints in previous parts is the changing norm of features and weight vectors in a mini-batch. As illustrated in Fig. 2, softmax loss is prone to amplifying the norm. The trade-off between dynamic norm and intra-class cosine similarity constraint seems to be harmful to the final testing accuracy computed by the pair cosine similarities. To better exert the power of proposed constraints in the training process without sacrificing most of the time on amplifying the norm, we normalize both the features and weight vectors of the last inner-product layer before softmax loss to a same value s, which is automatically learned as in [13]. The Normalized Large Margin Cosine (NLMC) loss function is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\mathcal{NLMC}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{s^2\widetilde{W}_{y_i}^T\widetilde{x}_i}}{\sum_{j=1}^{N} e^{s^2\widetilde{W}_j^T\widetilde{x}_i}} \\ + \frac{\lambda}{M}\sum_{i=1}^{M}\left\{\alpha - \widetilde{W}_{y_i}^T\widetilde{x}_i\right\}_+,\end{aligned} \tag{6}$$

where we substitute $s\widetilde{W}_j$ for W_j and $s\widetilde{x}_i$ for x_i in original softmax loss.

In this case, the training process will pay more attention to the intra-class cosine similarity constraint, because all the deeply learned features are distributed on a circle with the same radius in each iteration and the angular between them is an appropriate distance metric.

E. DLMC LOSS

It seems that the intra-class constraint alone is not enough to obtain discriminative features. Inspired by the form of softmax loss, we extend the NLMC loss to Discriminative Large Margin Cosine (DLMC) loss, which aims to enforce the intra-class cosine similarity larger than the mean of $p \times Inter(j)$ nearest neighboring inter-class cosine similarities with a fixed margin in the normalized exponential feature space.

$$\begin{aligned}\mathcal{L}_{\mathcal{DLMC}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{s^2\widetilde{W}_{y_i}^T\widetilde{x}_i}}{\sum_{j=1}^{N} e^{s^2\widetilde{W}_j^T\widetilde{x}_i}} \\ + \frac{\lambda}{M}\sum_{i=1}^{M}\left\{-\log\frac{e^{\widetilde{W}_{y_i}^T\widetilde{x}_i-\alpha}}{\sum_{j=1}^{p\times Inter(y_i)} e^{\frac{\widetilde{W}_j^T\widetilde{x}_i}{p\times Inter(y_i)}}}\right\}_+,\end{aligned} \tag{7}$$

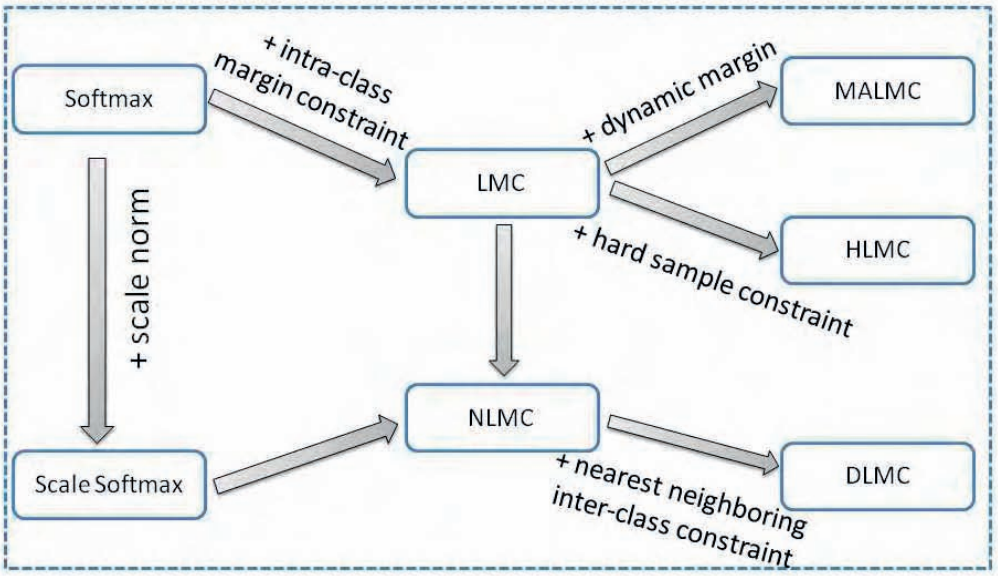


FIGURE 3. Flow chart of the relationships among the proposed approaches.

where $Inter(j)$ is the number of different inter-class cosine similarities between a sample of class j and the weight vectors of other classes in a mini-batch, and these cosine similarities are sorted in descending order, α is a predefined margin to discriminate the intra-class and inter-class similarities, p is a predefined percentage to control the valid number of inter-class similarities.

For datasets with many classes, most inter-class similarities are too small that they are useless to the final calculation of Eq. 7. While, the proposed neighborhood sampling strategy can incorporate the most meaningful classes to acquire the reliable mean inter-class similarity. Specifically, when $p \times Inter(j) = 1$, the DLMC loss immediately reduces to a variant of triplet loss.

$$\mathcal{L}'_{\mathcal{DLMC}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{e^{s^2\widetilde{W}_{y_i}^T\widetilde{x}_i}}{\sum_{j=1}^{N}e^{s^2\widetilde{W}_j^T\widetilde{x}_i}} + \frac{\lambda}{M}\sum_{i=1}^{M}\left\{\widetilde{W}_j^T\widetilde{x}_i - \widetilde{W}_{y_i}^T\widetilde{x}_i + \alpha\right\}_+. \quad (8)$$

Compared to the Euclidean distance constraint in the original feature space of triplet loss, this variant loss function imposes the cosine similarity constraint between a sample and the weight vectors in the normalized feature space, analytically strengthening the robustness in the training process. The relationships of these proposed approaches are shown in Fig. 3.

III. EXPERIMENTS

The implementation details are given in Section III.A. In Section III.B, some exploratory experiments are conducted to find the best settings of hyper-parameters in each method. Finally, we evaluate our approaches on three face recognition benchmark datasets in Section III.C and III.D.

A. IMPLEMENTATION DETAILS

1) BASIC TRAINING SETTINGS

To test the sensitivity of face recognition results regarding different face detectors, we preprocess the face images by MTCNN [23] and SeetaFace [24] detectors, respectively. We use the publicly available CASIA-WebFace [22] as the training set, which originally has 494,414 labeled face images from 10,575 individuals. After removing the undetected images, the resulting datasets have 490,869 images for MTCNN and 437,633 images for SeetaFace. The obvious difference between these two detectors is that there is a high false negative rate of SeetaFace, such that the resulting training set has fewer false positive samples. We use the Caffe library [25] to implement the CNN model [8] in this part, which is a reduced version of ResNet with only 27 convolutional layers. The input faces are cropped to 112×96 RGB images, followed by subtracting 127.5 and dividing by 128. The batch size is set to 256 in all the experiments, and the images are horizontally flipped for data augmentation. For LMC, HLMC and MALMC, we train the models from scratch. The initial learning rate is set to 0.1, then divided by 10 at 16K, 24K iterations. The complete training terminates at 28K iterations. While, we fine-tune the networks of NLMC, NLMC+MALMC and DLMC from the softmax baseline model and a relatively small learning rate of 0.001 is applied. For other compared metric loss functions, we train them to achieve their best performance. The classical back-propagation algorithm and mini-batch based Stochastic Gradient Descent (SGD) work well for the training, and the momentum and weight decay are set to 0.9 and 0.0005.

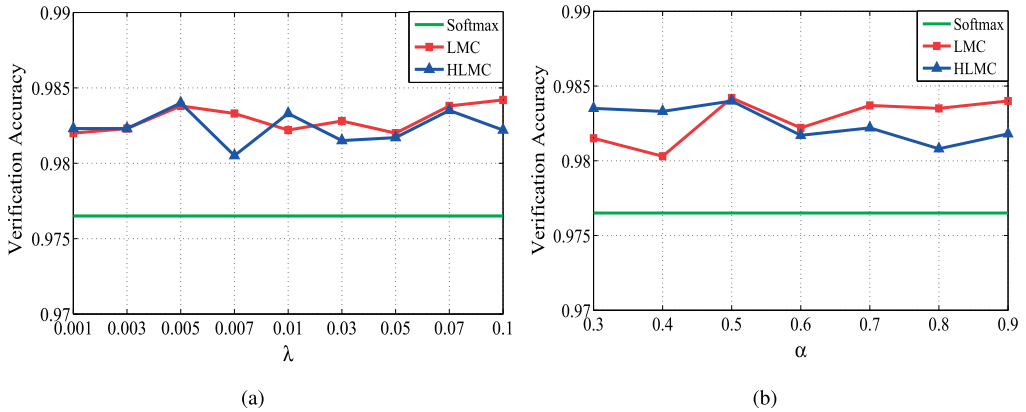


FIGURE 4. Verification accuracies on LFW of LMC and HLMC (a) with different λ and fixed $\alpha = 0.5$. (b) With different α, $\lambda = 0.1$ for LMC and $\lambda = 0.005$ for HLMC.

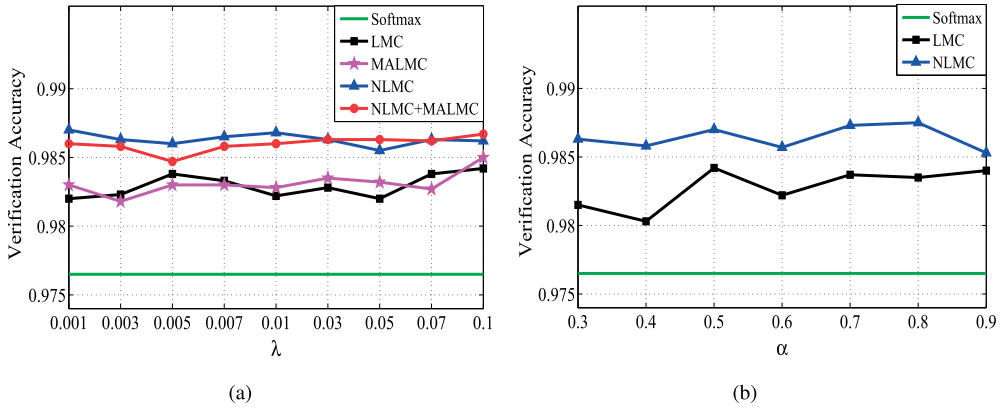


FIGURE 5. Verification accuracies on LFW of some proposed methods (a) with different λ and fixed $\alpha_0 = 0.2, \alpha = 0.5$. (b) With different α and the best setting of λ in each method according to (a).

2) EVALUATION

The proposed methods are evaluated on three face recognition datasets, namely LFW, YTF and IJB-A datasets. 10-fold validation is used to acquire the final performance. We extract the features from both the frontal face and its mirror one, and merge the two features by element-wise summation. PCA dimension reduction is applied to the final representations. Nearest neighbor and threshold comparison are used for both identification and verification tasks. Note that we only use single model for all the testings.

B. EXPLORATORY EXPERIMENTS

All the experiments in this section are conducted on the resulting datasets by MTCNN detector, if not specified.

1) EFFECT OF THE HARD SAMPLE MINING STRATEGY

To ensure the learning efficiency in the training process, we explore a new hard sample mining strategy, where the hard samples refer to the ones misclassified by softmax loss. The hyper-parameters λ and α dominate the balance between intra-class and inter-class variations. Properly selected values of them can improve the performance of the proposed methods. So we conduct a pair of contrastive experiments on LMC and HLMC to investigate the sensitivity of these two parameters (Fig. 4).

In this experiment, we can see that both LMC and HLMC perform much better than the softmax loss. The accuracies fluctuate with different λ and α. The best settings are $\lambda = 0.1, \alpha = 0.5$ (98.42% on LFW) for LMC and $\lambda = 0.005, \alpha = 0.5$ (98.45% on LFW) for HLMC. Though the accuracies are almost the same, HLMC simplifies the training process by discarding the easy samples, and we will not deeply explore it in the following experiments.

2) EFFECT OF λ AND α

We can find the importance of choosing the appropriate values of λ and α in Fig. 5. Here, we explore the best settings of these two hyper-parameters only in some of the proposed methods.

In the first experiment, we fix α to 0.5 and vary λ from 0.001 to 0.1. In the second experiment, we fix λ as their respective best settings in the first experiment (0.005 for LMC and 0.001 for NLMC) and vary α from 0.3 to 0.9. Specifically, we set $\alpha_0 = 0.2$ and $p = 0.6$ in MALMC. We can observe that

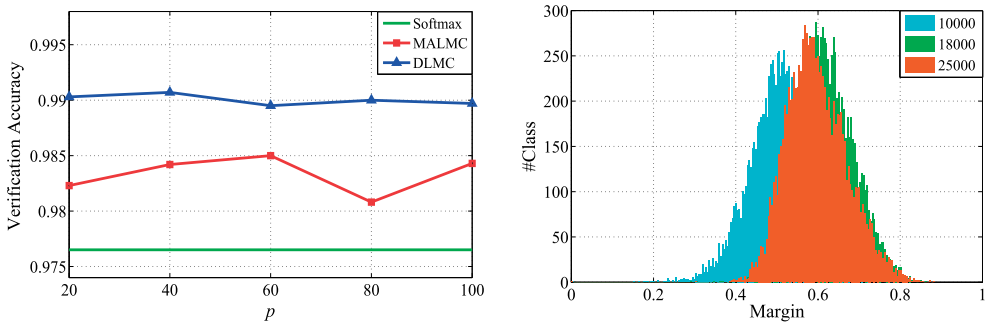


FIGURE 6. **(a) Verification accuracies on LFW with different p, fixed $\lambda = 0.1$ for MALMC by MTCNN and $\lambda = 0.03$, $\alpha = 0.01$ for DLMC by SeetaFace. (b) The margin distribution of each class with different training iteration steps in MALMC, where $\lambda = 0.1$, $\alpha_0 = 0.2$ and $p = 0.6$.**

the performance of our models is always stable with different λ and α at most of the time, and simply using the softmax loss is not a good choice.

3) EFFECT OF p IN MALMC AND DLMC

In this part, we explore the effect of different neighbors on the performance of MALMC and DLMC, namely the verification accuracies on LFW with different p in MALMC and DLMC, while keeping other parameters fixed as their best settings in the previous experiments (Fig. 6a).

It is obvious that MALMC is sensitive to p, and its best setting is 0.6, which controls the valid number of intra-class similarities in a mini-batch. While, DLMC is robust to p across a wide range. The reason is that there exists an inconsistent distribution in each mini-batch, a fixed p doesn't seem suitable for measuring the updated feature subspace of each class in MALMC. However, the robustness of DLMC stems from its similarity to softmax loss which is accompanied by significant inter-class separability, so that the first largest inter-class similarities play the most important role in the training process.

4) SELF-ADAPTIVE MARGIN STRATEGY IN MALMC

To make clear the margin updating process in MALMC, we perform a toy example of the margin statistics of each class with different iteration steps (10000, 18000 and 25000) during training (Fig. 6b). One can find that the margin is prone to be larger, and eventually fluctuates around the best value of 0.6. The distribution illustrates the necessity to assign a respective margin to 10,575 classes in CASIA-WebFace.

C. EXPERIMENTS ON LFW AND YTF DATASETS

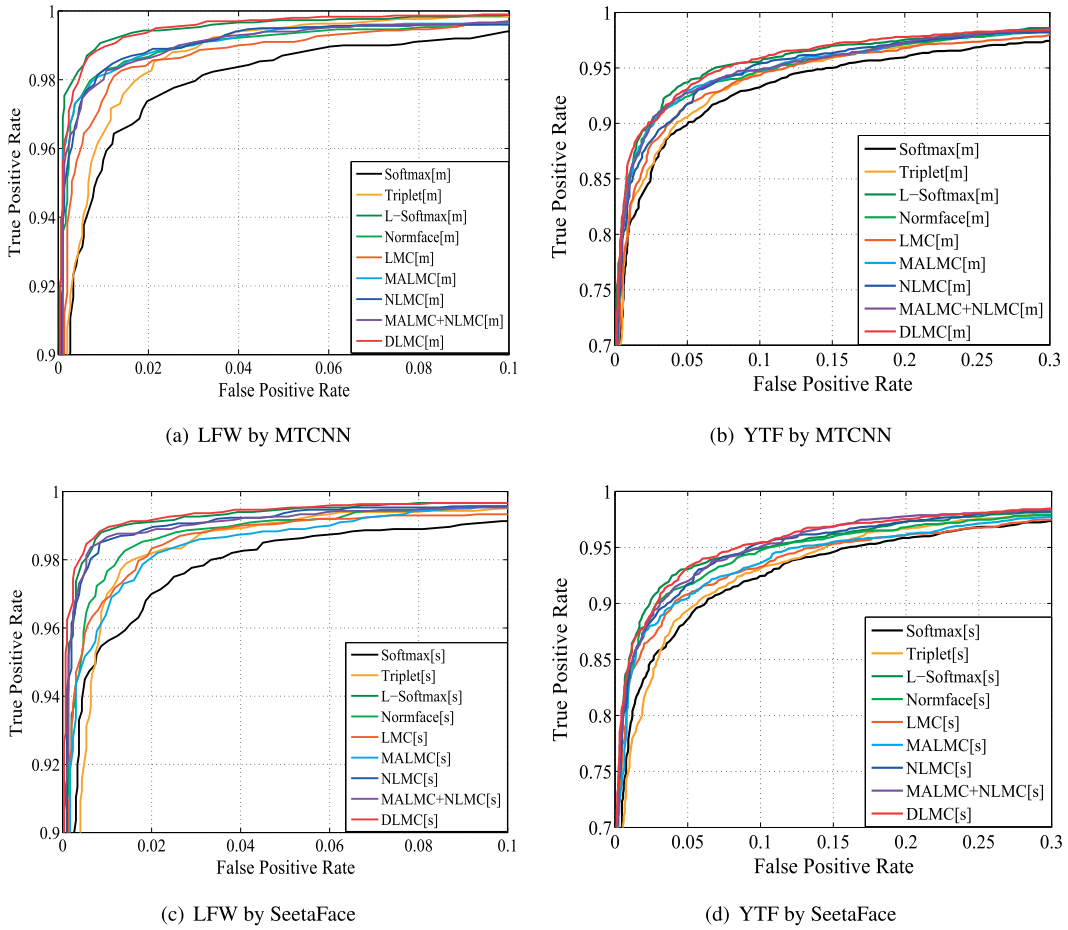
LFW This dataset contains 13,233 face images of 5,749 different identities from the Internet, with large variations in pose, expression and illumination. For comparison, algorithms typically report the mean face verification accuracies and the ROC curves on 6,000 given face pairs, following the standard protocol of unrestricted with labeled outside data [17].

TABLE 1. **Face verification performance on LFW and YTF datasets, where [m] refers to the result by MTCNN detector and [s] refers to the result by SeetaFace detector.**

Method	#Alig.	#Train	#Net	Acc. on LFW (%)	Acc. on YTF (%)
High-dim LBP	27	100K	-	95.17	-
DeepFace	73	4M	3	97.35	91.40
Gaussian Face	-	20K	1	98.52	-
DeepID	5	200K	1	97.45	-
DeepID-2+	18	300K	25	99.47	93.20
Center Loss	5	700K	1	99.28	94.90
FaceNet	-	200M	1	**99.63**	**95.10**
CASIA-WebFace	2	490K	1	97.73	90.60
Softmax[m]	5	490K	1	97.65	92.26
Triplet[m]	5	490K	1	98.12	92.96
L-Softmax[m]	5	490K	1	98.98	93.94
NormFace[m]	5	490K	1	98.55	94.04
SphereFace[m]	5	430K	1	**99.04**	94.14
LMC[m]	5	490K	1	98.42	93.30
MALMC[m]	5	490K	1	98.50	93.68
NLMC[m]	5	490K	1	98.75	94.06
NLMC+MALMC[m]	5	490K	1	98.67	94.14
DLMC[m]	5	490K	1	98.80	**94.16**
Softmax[s]	5	430K	1	97.42	91.52
Triplet[s]	5	430K	1	98.20	92.16
L-Softmax[s]	5	430K	1	98.86	94.14
NormFace[s]	5	430K	1	98.57	93.74
SphereFace[s]	5	430K	1	99.02	93.89
LMC[s]	5	430K	1	98.05	93.04
MALMC[s]	5	430K	1	98.07	92.90
NLMC[s]	5	430K	1	98.88	93.60
NLMC+MALMC[s]	5	430K	1	98.93	93.78
DLMC[s]	5	430K	1	**99.07**	**94.16**

YTF This dataset consists of 3,425 videos from 1,595 different people, with an average of 2.15 videos for each identity. Just as the experiments on LFW, we follow the standard protocol of unrestricted with labeled outside data [18], and report the results on 5,000 video pairs. The final similarity score of each video pair is computed by the average of the cosine similarities from 100 frame pairs.

In this experiment, we test the methods presented in Section II on datasets preprocessed by two different face detectors, namely MTCNN and SeetaFace. Some state-of-the-art methods (High-dim [26], DeepFace [27], Gaussian Face [28], DeepID [6], DeepID-2+ [29], Center Loss [8], FaceNet [7], CASIA-WebFace [22]) are incorporated as a contrast, even though most of their high performance is achieved by huge training data or model ensemble. As can be observed in Table 1, while using a single model trained on the



(a) LFW by MTCNN

(b) YTF by MTCNN

(c) LFW by SeetaFace

(d) YTF by SeetaFace

FIGURE 7. ROC curves of compared metric loss functions on LFW and YTF datasets by two different face detectors.

publicly available small dataset, our methods are still competitive with other models using high-quality private datasets, such as DeepFace (4M) and FaceNet (200M).

For a fair comparison, some typical metric loss functions (Triplet [7], L-Softmax [21], NormFace [13], SphereFace [9]) are also tested in our own settings. Among these compared loss functions, the proposed methods consistently outperform softmax loss by a significant margin. Specifically, the DLMC loss performs superior by MTCNN (98.80% accuracy on LFW and 94.16% accuracy on YTF) and SeetaFace (99.07% accuracy on LFW and 94.16% accuracy on YTF). Compared with NormFace, the NLMC, NLMC+MALMC and DLMC methods obviously show the advantages of margin based cosine similarity constraint in the training process. Similarly, the performance of triplet loss is also not satisfactory. As illustrated in Section II, the DLMC loss immediately reduces to a variant of triplet loss when $p \times Inter(j) = 1$. And the hard triplet mining strategy in triplet loss is avoided here, largely reduces the exponentially increased computational complexity of training dataset. The results in Table 1 convincingly demonstrate that the DLMC loss can alleviates the difficult convergence and big data dependence of triplet loss. The Receiver Operating Characteristic (ROC) curves of them are shown in Fig. 7. One should notice that there exists a discrepancy between the results of the two different face detectors, and the trends vary from one loss to another. Though, our DLMC method always among the top performance.

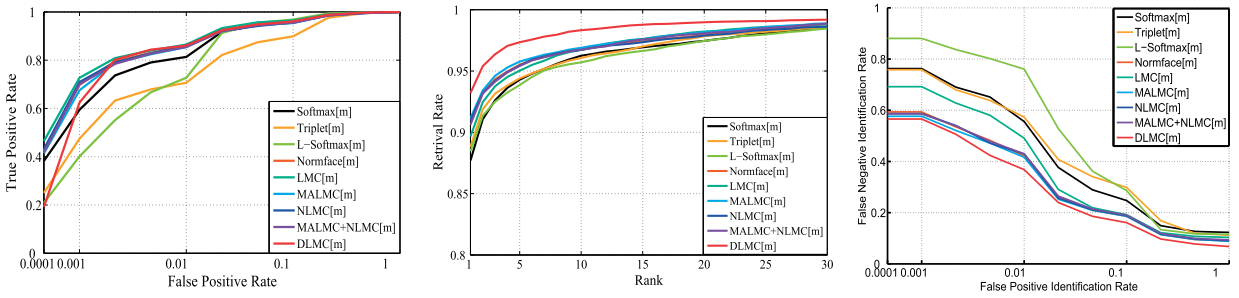
D. EXPERIMENTS ON IJB-A DATASET

IJB-A This dataset contains 5,712 images and 2,085 videos of 500 subjects, with an average of 11.4 images and 4.2 videos per subject. The IJB-A evaluation protocol consists of open-set verification (1:1 comparison) and identification (1:N search) over 10 random training and testing splits. Unlike the LFW and YTF datasets, the IJB-A dataset divides the testing images/video frames into gallery and probe sets, and the subjects are described by templates. Moreover, the images in the IJB-A dataset contain extreme pose, illumination and expression variations without being filtered by a commercial face detector. These factors essentially make IJB-A a challenging unconstrained face recognition dataset [19]. We use the Softmax operator [30] to compute the similarity score of two sets described by templates.

For simplicity, we only present the results by MTCNN detector here. As in the experiments of LFW and YTF, we compare our methods with some state-of-the-art approaches (GOTS [19], Deep Multi-Pose [31],

TABLE 2. Results on the IJB-A dataset. The True Accept Rate (TAR) at False Accept Rate (FAR)=0.01 and 0.001 for the ROC curves. The Rank-1, Rank-5, and Rank-10 retrieval accuracies for the CMC curves. The True Negative Identification Rate (TNIR) at False Positive Identification Rate (FPIR)=0.1 and 0.01 for the DET curves.

Method	IJB-A Ver.(TAR) (%)		IJB-A Id.(Rec. Rate) (%)			IJB-A Id.(TNIP) (%)	
Metric	FAR 0.01	FAR 0.001	Rank-1	Rank-5	Rank-10	FPIR 0.1	FPIR 0.01
GOTS	40.6	19.8	44.3	59.5	-	23.5	4.7
OpenBR	23.6	10.4	24.6	37.5	-	14.9	6.6
VGG-Face	80.5	60.4	91.3	-	98.1	67.0	46.1
Deep Multi-Pose	87.6	-	84.6	**92.7**	94.7	75.0	52.0
Triplet Embedding	90.0	81.3	93.2	-	97.7	86.3	75.3
Template Adaptation	**93.9**	**83.6**	92.8	-	98.6	88.2	77.4
All-In-One Face	92.2	82.3	**94.7**	-	**98.8**	**88.7**	**79.2**
Softmax[m]	81.36	59.46	87.70	94.29	96.25	75.26	44.40
Triplet [m]	70.66	47.48	88.82	94.40	96.06	70.14	42.67
L-Softmax [m]	72.71	40.10	88.45	93.85	95.71	71.39	23.91
NormFace [m]	85.86	69.77	90.98	95.51	96.76	81.36	57.79
SphereFace [m]	86.23	70.41	**93.42**	97.26	97.54	82.49	60.57
LMC[m]	**86.47**	**72.71**	89.66	95.02	96.73	80.80	50.91
NLMC[m]	86.19	70.94	90.96	95.45	96.72	81.23	57.09
MALMC[m]	85.41	67.55	91.15	95.77	96.90	81.42	58.36
NLMC + MALMC[m]	85.52	70.05	90.63	95.41	96.73	81.00	57.00
DLMC[m]	86.02	62.45	93.21	**97.34**	**98.33**	**83.89**	**63.20**



(a) (b) (c)

FIGURE 8. Recognition accuracies on IJB-A dataset. (a) ROC curves for the compare protocol. (b) CMC curves for the search protocol. (c) DET curves for the search protocol.

Template Adaptation [32], All-In-One Face [33]) using larger training datasets or model ensemble. Whereas, simply comparing our methods to those state-of-the-art results is unfair, because their system designs and implementation details are different from ours, and the difficult access to their codes and data makes it hard to say exactly how much improvement our proposed methods acquire. So several mainstream DML approaches (Triplet [7], L-Softmax [21], NormFace [13], SphereFace [9]) under the same settings as ours are compared. From the results in Table 2, we can find that our proposed methods significantly improve over the off the shelf commercial systems GOTS. Compared to some deep learning based methods, our approaches still achieve satisfactory performance. To better show the comparison results with some typical DML methods under our own settings, the ROC curves for face verification and the Cumulative Match Characteristic (CMC) curves for face identification are plotted in Fig. 8. Obviously, our methods exhibit prominent advantages consistently over other DML methods, and always among the top performance. However, the performance of triplet loss and L-Softmax loss on the IJB-A dataset is not as good as that on the LFW and YTF datasets, due to the large variations of IJB-A.

IV. CONCLUSION AND FUTURE WORK

In this paper, we introduce the margin based intra-class cosine similarity constraint into the training process, to alleviate the large intra-class variation of softmax loss and keep consistency with the testing process. Accompanied by the inter-class separability of softmax loss, the original LMC loss achieves a significant improvement. Based on this, the MALMC loss is proposed to mitigate the fussy human labor of adjusting the margin hyper-parameter. Furthermore, the NLMC loss is given to take full advantage of the intra-class cosine similarity constraint with all the features and weight vectors in a mini-batch fixed to the same norm. To acquire more discriminative features, a profound idea of considering the intra-class and inter-class constraints

simultaneously is proposed to form the DLMC loss. Extensive experiments on several public face recognition benchmark datasets convincingly demonstrate the effectiveness and robustness of these proposed methods, even on a small training dataset.

Noticeably, these loss functions are not differentiable everywhere, and some smoothed versions seem to be a meaningful research direction. We will apply the proposed methods on other metric leaning tasks in the future, such as person re-identification or image retrieval. Furthermore, how to develop robust DML methods regarding different face detectors is an interesting future direction for research.

REFERENCES

[1] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. CVPR*, Jun. 2016, pp. 4004–4012.

[2] M. Sun, Y. Yuan, F. Zhou, and E. Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 805–821.

[3] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. CVPR*, Jun. 2015, pp. 1556–1564.

[4] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5022–5030.

[5] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. ICPR*, Aug. 2014, pp. 34–39.

[6] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. CVPR*, Jun. 2014, pp. 1891–1898.

[7] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, Jun. 2015, pp. 815–823.

[8] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. ECCV*. Cham, Switzerland: Springer, 2016, pp. 499–515.

[9] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. CVPR*, Jul. 2017, pp. 212–220.

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

[11] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4690–4699.

[12] 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 Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[13] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L_2 hypersphere embedding for face verification," in *Proc. ACM MM*, 2017, pp. 1041–1049.

[14] C. Luo, J. Zhan, L. Wang, and Q. Yang, "Cosine normalization: Using cosine similarity instead of dot product in neural networks," Feb. 2017, *arXiv:1702.05870*. [Online]. Available: https://arxiv.org/abs/1702.05870

[15] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. NIPS*, 2014, pp. 1988–1996.

[16] C. Huang, C. C. Loy, and X. Tang, "Local similarity-aware deep feature embedding," in *Proc. NIPS*, 2016, pp. 1262–1270.

[17] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. 14–003, 2014.

[18] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. CVPR*, Jun. 2011, pp. 529–534.

[19] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark A," in *Proc. CVPR*, Jun. 2015, pp. 1931–1939.

[20] Y. LeCun, C. Cortes, and C. J. Burges. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[21] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, pp. 507–516.

[22] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," Nov. 2014, *arXiv:1411.7923*. [Online]. Available: https://arxiv.org/abs/1411.7923

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

[24] S. Wu, M. Kan, Z. He, S. Shan, and X. Chen, "Funnel-structured cascade for multi-view face detection with alignment-awareness," *Neurocomputing*, vol. 221, pp. 138–145, Jan. 2017.

[25] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrel, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM MM*, 2014, pp. 675–678.

[26] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proc. CVPR*, Jun. 2013, pp. 3025–3032.

[27] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. CVPR*, Jun. 2014, pp. 1701–1708.

[28] C. Lu and X. Tang, "Surpassing human-level face verification performance on lfw with gaussianface," in *Proc. AAAI*, 2014, pp. 1–9.

[29] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proc. CVPR*, Jun. 2015, pp. 2892–2900.

[30] I. Masi, A. T. Tr an, T. Hassner, J. T. Leksut, and G. Medioni, "Do we really need to collect millions of faces for effective face recognition?" in *Proc. ECCV*, 2016, pp. 579–596.

[31] W. AbdAlmageed, Y. Wu, S. Rawls, S. Harel, T. Hassner, I. Masi, J. Choi, J. Lekust, J. Kim, P. Natarajan, R. Nevatia, and G. Medioni, "Face recognition using deep multi-pose representations," in *Proc. WACV*, Mar. 2016, pp. 1–9.

[32] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman, "Template adaptation for face verification and identification," in *Proc. FG*, May/Jun. 2017, pp. 1–8.

[33] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," in *Proc. FG*, May/Jun. 2017, pp. 17–24.

BOWEN WU received the Ph.D. degree from Nankai University, China, in 2018. He is currently a Researcher with the JD Intelligent Cities Business Unit, Beijing, China. His research interests include deep learning, machine learning, big data, smart city, face recognition, and graph theory and combinatorics.

HUAMING WU received the B.E. and M.S. degrees in electrical engineering from the Harbin Institute of Technology, China, in 2009 and 2011, respectively, and the Ph.D. degree (Hons.) in computer science from the Free University of Berlin, Germany, in 2015. He is currently an Associate Professor with the Center for Applied Mathematics, Tianjin University. His current research interests include mobile and cloud computing, edge computing, wireless and mobile network systems, the Internet of Things (IoTs), and deep learning.