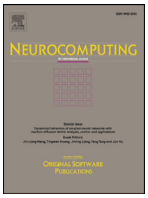
Neurocomputing 290 (2018) 108–120

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Exponential discriminative metric embedding in deep learning

Bowen Wu[a,*], Zhangling Chen[b], Jun Wang[c], Huaming Wu[b]

[a] *Center for Combinatorics, Nankai University, Tianjin 300071, China*
[b] *Center for Applied Mathematics, Tianjin University, Tianjin 300072, China*
[c] *School of Mathematics, Tianjin University, Tianjin 300072, China*

ARTICLE INFO

Article history:
Received 9 March 2017
Revised 3 February 2018
Accepted 6 February 2018
Available online 14 February 2018

Communicated by Dr. Nianyin Zeng

Keywords:
Deep metric learning
Object recognition
Face verification
Intra-class compactness
Inter-class separability

ABSTRACT

With the remarkable success achieved by the Convolutional Neural Networks (CNNs) in object recognition recently, deep learning is being widely used in the computer vision community. Deep Metric Learning (DML), integrating deep learning with conventional metric learning, has set new records in many fields, especially in classification task. In this paper, we propose a replicable DML method, called Include and Exclude (IE) loss, to force the distance between a sample and its designated class center away from the mean distance of this sample to other class centers with a large margin in the exponential feature projection space. With the supervision of IE loss, we can train CNNs to enhance the intra-class compactness and inter-class separability, leading to great improvements on several public datasets ranging from object recognition to face verification. We conduct a comparative study of our algorithm with several typical DML methods on three kinds of networks with different capacity. Extensive experiments on three object recognition datasets and two face recognition datasets demonstrate that IE loss is always superior to other mainstream DML methods and approach the state-of-the-art results.

1. Introduction

Recently, Convolutional Neural Networks (CNNs) are continuously setting new records in classification aspect, such as object recognition [1–4], scene recognition [5,6], face recognition [7–12], age estimation [13,14] and so on. Facing the more and more complex data, the deeper and wider CNNs tend to obtain better accuracies. Meanwhile, many troubles will show up, such as gradient saturating, model overfitting, parameter augmentation, etc. To solve the first problem, some non-linear activations [15–17] have been proposed. Considerable efforts have been made to reduce model overfitting, such as data augmentation [1,18], dropout [1,19], regularization [15,20]. Besides, some model compressing methods [21,22] have largely reduced the computing complexity of original models, with the performance improved simultaneously.

In general object recognition, scene recognition and age estimation, the identities of the possible testing samples are within the training set. So the training and testing sets have the same object classes but not the same images. In this case, softmax classifier is often used to designate a label to the input.

For face recognition, the deeply learned features need to be not only separable but also discriminative. It can be roughly divided into two aspects, namely face identification and face verification. The former is the same as object recognition, the training and testing sets have the same face identities, aims at classifying an input image into a large number of identity classes. Face verification is to classify a pair of images as belonging to the same identity or not (i.e. binary classification). Since it is impractical to pre-collect enough number of all the possible testing identities for training, face verification is becoming the mainstream in this field. As clarified by DeepID series [9,10,23]: classifying all the identities simultaneously instead of binary classifiers for training can make the learned features more discriminative between different classes. So we decide to use the joint supervision of softmax classifier and metric loss function to train and the verification signal of feature similarity discriminant to test as shown in Section 4.3. Fig. 1 illustrates the general face recognition pipeline, which maps the input images to the discriminative deep features progressively, then to the predicted labels.

A recent trend towards deep learning with more discriminative features is to reinforce CNNs with better metric loss functions, namely Deep Metric Learning (DML), such that the intra-class compactness and inter-class separability are simultaneously maximized. Inspired by this idea, many metric learning methods have been proposed. It can be traced back to early subspace face recognition methods such as Linear Discriminant Analysis (LDA) [24], Bayesian face [25], and unified subspace [26]. For example, LDA aims at maximizing the ratio between inter-class and

[*] Corresponding author.
E-mail addresses: wbw@mail.nankai.edu.cn, 986381313@qq.com (B. Wu).

https://doi.org/10.1016/j.neucom.2018.02.040
0925-2312/© 2018 Elsevier B.V. All rights reserved.

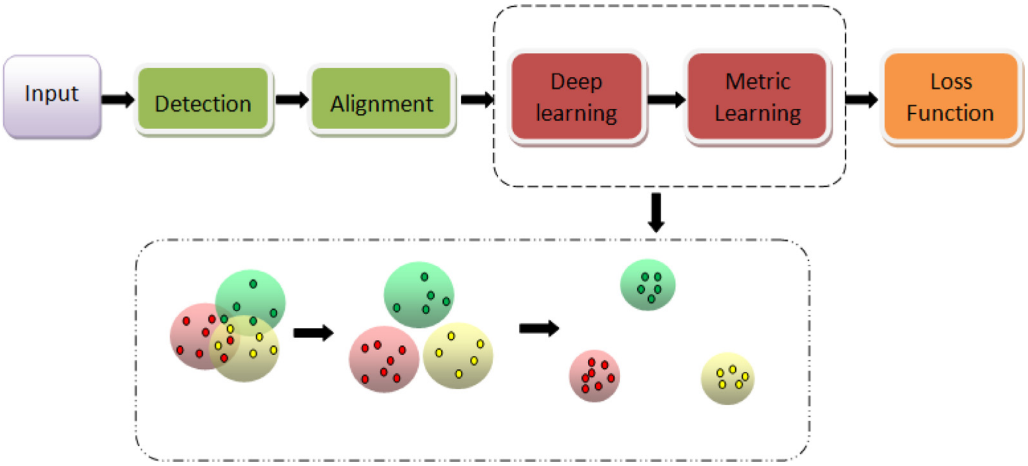


Fig. 1. The typical framework of face recognition. The process of deep feature learning and metric learning is shown in the second row.

intra-class variations by finding the optimal projection direction. Some metric learning methods [27–29] have been proposed to project the original feature space into another metric space, such that the features of the same identity are close and those of different identities stay apart. Subsequent contrastive loss [23] and triplet loss [11] have witnessed their success in face recognition.

Interestingly, closely related to DML is the Learning to Hash, which is one of the major solutions to nearest neighbor search problem. Given the high dimensionality and high complexity of multimedia data, the cost of finding the exact nearest neighbor is prohibitively high. Learning to Hash, a data-dependent hashing approach, aims to learn hash functions from a specific dataset so that the nearest neighbor search result in the hash coding space is as close as possible to the search result in the original space, significantly improving the search efficiency and space cost. The main methodology of Learning to Hash is similarity preserving, i.e., minimizing the gap between the similarities computed in the original space and the similarities in the hash coding space in various forms. [30] utilizes linear LDA with trace ratio criterion to learn hash functions, where the pseudo labels and the hash codes are jointly learned. [31] proposes a semi-supervised deep learning hashing method for fast multimedia retrieval, to simultaneously learn a good multimedia representation and hash function. More comprehensive survey about dimension reduction and using different similarity preserving algorithms to hashing can be found in [32,33]. Surprisingly, most of the similarity metric loss functions could be used for Learning to Hash.

Because of the large scale of training set, it is unreasonable to address all of them in each iteration. Mini-batch based Stochastic Gradient Descent (SGD) algorithm [34] does not reflect the real distribution of the total training set, so a superior sampling strategy becomes very important to the training process. Besides, selecting appropriate pairs or triplets like previous may dramatically increase the number of training samples. As a result, it is inevitably hard to converge to an optimum steadily. In this paper, we propose a novel well-generalized metric loss function, named Include and Exclude (IE) loss, to make the deeply learned features more discriminative between different classes and closer to each other between images of the same class. This idea is verified by Fig. 2 in Section 3.1. Obviously, the inter-class distance is away from the intra-class distance with a large margin. When training, we learn a center for each class like center loss [12] does. Subsequently, we show that center loss is a variant of the special case of our method. There is another parameter σ^2 to regularize the distance between the features and their corresponding class centers. Furthermore, we use a hyperparameter Q to control the number of valuable inter-class distances to accelerate the convergence of our model. We simultaneously use the supervision signals of softmax loss and IE loss to train the network. Extensive experiments on object recognition and face verification validate the effectiveness of IE loss. Our method significantly improves the performance compared to the original softmax method, and competitive with other nowadays mainstream DML algorithms. The main contributions are summarized as follows:

- To the best of our knowledge, we are the first to practice the idea of enforcing the mean inter-class distance larger than the intra-class distance with a margin in the exponential feature projection space, as opposed to the distance between a sample and its nearest cluster centers in magnet loss [35], avoiding the large intra-class distances.
- Instead of some off-line complicated sampling strategies, our DML method can achieve a satisfactory result only using the mini-batch based SGD, greatly simplifying the training process.
- To achieve a better performance rapidly, we introduce a hyperparameter Q to restrict the number of nearest inter-class distances in each mini-batch to accelerate the convergence of our model.
- We do extensive experiments on several common datasets, including MNIST, CIFAR10, CIFAR100, Labeled Faces in the Wild (LFW) and YouTube Faces (YTF), to verify the effectiveness, robustness and generalization of IE loss.

2. Related work

In recent years, deep learning has been successfully applied in computer vision and other AI domains, such as object recognition [3], face recognition [11], image retrieval [36,37], speech recognition [38] and natural language processing [39]. Most of the time, deep learning models are prone to be deeper and wider. But more complicated deep networks are accompanied by larger training set, model overfitting and costly computational overhead. Considering these, there produce some new DML methods, which concatenate the conventional metric learning losses to the end of the deeply learned features. In classification aspect, DML generally aims at mapping the originally learned features into a more discriminative feature space by maximizing the inter-class variations and minimizing the intra-class variations. To some degree, a properly chosen metric loss function would make the training easy to converge to an optimal model without too much training data. We will briefly discuss some typical DML methods below.

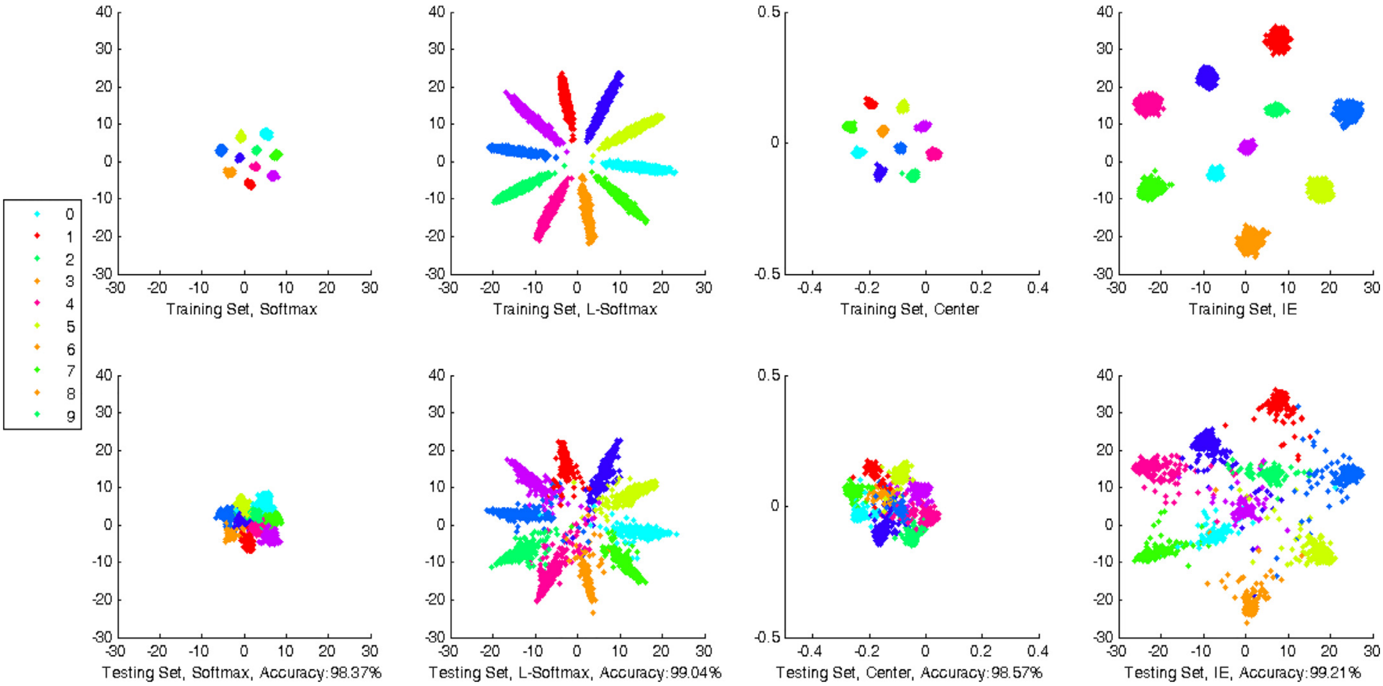


Fig. 2. Visualization of the deeply learned 2D features on training and testing sets of MNIST, regarding softmax loss, L-Softmax loss, center loss and IE loss, respectively. The points with different colors correspond to the features from different classes.

Sun et al. [23] encourage all faces of one identity to be projected onto a single point in the embedding space. They use an ensemble of 25 networks on different face patches to get the final concatenated features. Both PCA and Joint Bayesian classifier [27] are used to achieve the final performance of 99.47% on LFW. The loss function is mainly based on the idea of contrastive loss, which minimizes the intra-class distance and enforces the inter-class distance larger than a fixed margin.

Schroff et al. [11] employ the triplet loss, which stems from LMNN [28], to encourage a distance constraint similar to the contrastive loss. Differently, the triplet loss requires a triple of training samples as input at a time, not a pair. The triplet loss minimizes the distance between an anchor sample and a positive sample, and maximizes the distance between the anchor sample and a negative sample, in order to make the inter-class distance larger than the intra-class distance by a margin relatively. They also use the so far largest training database about 200M face images, and set an insurmountable record on LFW of 99.63%.

Rippel et al. [35] propose a novel magnet loss, which is explicitly designed to maintain the distribution of different classes in feature space. In terms of computational performance, it alleviates the training inefficiency of the traditional triplet loss, which is verified from classification task to attribute concentration. But, the complicated off-line sampling strategy makes it too difficult to reproduce. In addition, the intra-class distribution maintaining by local clusters would impair the inter-class separability in general classification tasks, especially in face recognition.

3. The proposed approaches

We first clarify the notations which will be used in subsequential sections. Let us assume the training set consists of M input-label pairs $\mathcal{D} = \{x_n, y_n\}_{n=1}^{M}$ belonging to C classes. We consider a parameterized map $f(x_n, \Theta), n = 1, \dots, M,$ and Θ are the model parameters. In this work, the transformation is selected as some complex CNN architectures. We further define $C(f_n)$ as the class label of feature f_n, and $\mu_{C(f_n)}$ as the corresponding class center.

3.1. Some existing methods

In this section, some existing superior DML methods are first presented.

Triplet Loss Schroff et al. [11] have verified the effectiveness of triplet loss with a large training set. But the exponentially increased computational complexity of training examples and the difficulty of convergence impede its general application. The formula is as follows:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{M}\left\{\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right\}_+. \tag{1}$$

Here, x_i^a, x_i^p and x_i^n refer to the anchor, positive and negative images in a triplet, respectively. α is the predefined margin.

$L - Softmax$ Loss Liu et al. [40] achieve a flexible learning objective with adjustable difficulty, by altering the classification angle margin between classes. Although the relatively rigorous learning objective with adjustable angle margin can avoid overfitting, the difficult convergence hinders its generalization to many other deep networks. It is crucial to continuously adjust the component weight between softmax and L-Softmax to guarantee the progressing of training.

$$\mathcal{L}(\Theta) = -\frac{1}{M}\sum_{i=1}^{M}\log \times\left(\frac{exp(\|W_{y_i}\|\|x_i\|\psi(\theta_{y_i}))}{exp(\|W_{y_i}\|\|x_i\|\psi(\theta_{y_i})) + \sum_{j\neq y_i} exp(\|W_j\|\|x_i\|cos(\theta_j))}\right). \tag{2}$$

It generally requires that

$$\psi(\theta) = \begin{cases}\cos(m\theta), & 0 \le \theta \le \frac{\pi}{m}\\ \mathcal{D}(\theta), & \frac{\pi}{m} < \theta \le \pi\end{cases} \tag{3}$$

where W is the weight matrix of the fully connected layer before softmax layer, and W_{y_i} is the y_ith column of W. θ_{y_i} is the angle between x_i and its corresponding weight vector W_{y_i}, and m is an integer to control the learning objective. Meanwhile, $\mathcal{D}(\theta)$ must be monotonically decreased to satisfy the requirement for any θ.

Center Loss Wen et al. [12] propose a new loss function, which regards the distance of a sample away from its corresponding class center as the objective penalization. The joint supervision of center loss and softmax loss makes this approach outperform most existing best results on some face recognition benchmark databases.

$$\mathcal{L}(\Theta) = \frac{1}{2M}\sum_{i=1}^{M}\|f(x_i) - \mu(f(x_i))\|_2^2, \tag{4}$$

where $\mu(f(x_i))$ is the class center of $f(x_i)$.

3.2. IE loss

As clarified in [35], magnet loss liberates us from the unreasonable prior target neighborhood assignments, and divides each class into several clusters, aims at maintaining the distributions of different classes in the representation space. As a result, the similar samples in different classes may be closer than that in the same classes. Specifically, intra-class variations may be larger than inter-class variations in object recognition and face recognition. Thus some local distribution maintaining loss functions like magnet loss will not bring so many benefits to the practical classification tasks. Despite the great performance on LFW by triplet loss on GoogLeNet [3], its training ineffectiveness and the exponentially increased training samples hinder the widespread application to generic classification tasks.

Considering the difficulty of magnet loss to reproduce and the disadvantages mentioned above, we propose a replicable DML method, called IE loss, to learn the discriminative features. We calculate all the distances between a sample and other class centers in a mini-batch to take of advantage of batch information, as compared to the pair/triplet samples like previous. The objective is initially defined as follows:

$$\mathcal{L}(\Theta) = \frac{1}{M}\sum_{n=1}^{M}\left\{-log\frac{exp(-\frac{1}{2\sigma^2}\|f_n - \mu_{C(f_n)}\|_2^2 - \alpha)}{\sum_{c\neq C(f_n)} exp(-\frac{1}{2\sigma^2}\|f_n - \mu_c\|_2^2)}\right\}_+, \tag{5}$$

where $\{\cdot\}_+$ is the hinge loss function, α is a predefined margin hyperparameter, $\sigma^2 = \frac{1}{M-1}\sum_{n\in\mathcal{D}}\|f_n - \mu_{C(f_n)}\|_2^2$ is the variance of examples away from their respective class centers in the feature space. When training, the class center $\mu_{C(f_n)}$ and variance σ^2 should update together with the deep feature f_n. This means we should use the entire training set in each iteration. Obviously, it is impractical. So we decide to employ the mini-batch based SGD algorithm to update the parameters. The denominator in log part is computed by summing all the inter-class distances between a sample and other class centers appear in the mini-batch. This approach seems to be a natural choice with the probability interpretation, the same to softmax loss.

Some existing similar DML methods express that a sample quite far away from the corresponding class center should vanish from its term in our objective, approximating the denominator of Eq. (5) with a small number of nearest classes. Variance standardization also renders the objective invariant to the characteristic length scale of the problem. Whereas, all these benefits are based on a superb neighborhood sampling strategy for each class to keep the local distribution. Different from the strategy exploited in [35] which sampling the nearest K clusters in each class, we decide to use the Q nearest class centers to obtain the objective. The improved objective loss function is formulated as follows:

$$\mathcal{L}(\Theta) = \frac{1}{M}\sum_{n=1}^{M}\left\{-log\frac{exp(-\frac{1}{2\sigma^2}\|f_n - \mu_{C(f_n)}\|_2^2 - \alpha)}{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n - \mu_c\|_2^2)}\right\}_+, \tag{6}$$

where Q is an effectively selected number of different inter-class distances between a sample and other class centers in a mini-batch, and these distances are sorted in ascending order. We can choose a proper Q according to different training datasets to acquire the best performance. One can notice that the sophisticated off-line nearest clusters sampling strategy is avoided, and the mini-batch based SGD works well for our training. Besides, the too large inter-class distances are removed to accelerate the convergence, which is especially valid for the datasets with many classes. Subsequent results will show that the proposed method can greatly improve the training efficiency without sacrificing speed, since these auxiliary loss layers are removed in the classification step.

When we set $Q = 1$ and $\sigma^2 = 0.5$, Eq. (6) immediately reduces to Eq. (7).

$$\mathcal{L}(\Theta) = \frac{1}{M}\sum_{n=1}^{M}\left\{\|f_n - \mu_{C(f_n)}\|_2^2 + \alpha - \min_{c\neq C(f_n)}\|f_n - \mu_c\|_2^2\right\}_+. \tag{7}$$

It is clear that this formula is a variant of the efficient center loss and triplet loss. This loss function seems more appropriate to reflect the characteristics of our proposed method. It apparently forces the minimum inter-class distance larger than the intra-class distance with a margin α.

The effectiveness of our method is shown in Fig. 2. The visualization of 2-D features on training and testing sets sufficiently reflects the relative intra-class compactness and inter-class separability of IE loss, compared to softmax loss. One can also find that L-Softmax loss obviously amplifies the angle of features between different classes, and center loss seriously shrinks the intra-class distances such that the deeply learned features are discriminative in a small subspace.

Considering the classical back-propagation algorithm, the entire parameter updating process of IE loss is summarized in Algorithm 1. Softmax loss is incorporated to accelerate the con-

Algorithm 1 The parameter updating algorithm of IE loss.

```
Input: training set $\mathcal{D} = \{x_n, y_n\}_{n=1}^{M}$, initialized parameters $\theta_c$ in
    convolutional layers, $W$, $\sigma^2$ and $\mu_q (q = 0, 1, \ldots, Q)$ in loss
    layer where $q = 0$ corresponds to the case of $\mu_{C(f_n)}$, hyperparameters
    $\alpha$ and $\lambda$, learning rate $\eta^t$ and total iterative steps $T$.
Output: model parameters $\theta_c$.
for $t = 1, 2, \ldots, T$ do
    compute the loss function
    $\mathcal{L}^t = \mathcal{L}^t_{softmax} + \lambda\mathcal{L}^t_{IE}$
    compute the gradients
    $\frac{\partial\mathcal{L}^t}{\partial f^t_n} = \frac{\partial\mathcal{L}^t_{softmax}}{\partial f^t_n} + \lambda\frac{\partial\mathcal{L}^t_{IE}}{\partial f^t_n}$
    $\frac{\partial\mathcal{L}^t}{\partial W^t} = \frac{\partial\mathcal{L}^t_{softmax}}{\partial W^t} + \lambda\frac{\partial\mathcal{L}^t_{IE}}{\partial W^t} = \lambda\frac{\partial\mathcal{L}^t_{IE}}{\partial W^t}$
    $\frac{\partial\mathcal{L}^t}{\partial \mu^t_q} = \frac{\partial\mathcal{L}^t_{softmax}}{\partial \mu^t_q} + \lambda\frac{\partial\mathcal{L}^t_{IE}}{\partial \mu^t_q} = \lambda\frac{\partial\mathcal{L}^t_{IE}}{\partial \mu^t_q}$
    $\frac{\partial\mathcal{L}^t}{\partial \sigma^2_t} = \frac{\partial\mathcal{L}^t_{softmax}}{\partial \sigma^2_t} + \lambda\frac{\partial\mathcal{L}^t_{IE}}{\partial \sigma^2_t} = \lambda\frac{\partial\mathcal{L}^t_{IE}}{\partial \sigma^2_t}$
    update parameters
    $W^{t+1} = W^t - \eta^t \cdot \frac{\partial\mathcal{L}^t}{\partial W^t} = W^t - \eta^t \cdot \lambda \cdot \frac{\partial\mathcal{L}^t_{IE}}{\partial W^t}$
    $\mu^{t+1}_q = \mu^t_q - \eta^t \cdot \frac{\partial\mathcal{L}^t}{\partial \mu^t_q} = \mu^t_q - \eta^t \cdot \lambda \cdot \frac{\partial\mathcal{L}^t_{IE}}{\partial \mu^t_q}$
    $\sigma^2_{t+1} = \sigma^2_t - \eta^t \cdot \frac{\partial\mathcal{L}^t}{\partial \sigma^2_t} = \sigma^2_t - \eta^t \cdot \lambda \cdot \frac{\partial\mathcal{L}^t_{IE}}{\partial \sigma^2_t}$
    $\theta^{t+1}_c = \theta^t_c - \eta^t \sum_{n=1}^{M} \frac{\partial\mathcal{L}^t}{\partial f^t_n} \cdot \frac{\partial f^t_n}{\partial \theta^t_c}$
end for
```

Table 1
Some normal CNN architectures for different benchmark datasets. Conv1.x, Conv2.x and Conv3.x denote structures that may contain multiple successive convolutional layers. Batch normalization is used in these networks.

MNIST (for Fig.2)	Conv0.x	Conv1.x	Pool1	Conv2.x	Pool2	Conv3.x	Pool3	Fully connected
Num Layer	–	2	1	2	1	2	1	1
Filt Dim	–	5	2	5	2	5	2	1
Num Filt	–	32	–	64	–	128	–	2
Stride	–	1	2	1	2	1	2	1
Pad	–	2	–	2	–	2	–	–
MNIST	Conv0.x	Conv1.x	Pool1	Conv2.x	Pool2	Conv3.x	Pool3	Fully Connected
Num Layer	1	3	1	3	1	3	1	1
Filt Dim	3	3	2	3	2	3	2	1
Num Filt	64	64	–	64	–	64	–	256
Stride	1	1	2	1	2	1	2	1
Pad	1	1	–	1	–	1	–	–
CIFAR10	Conv0.x	Conv1.x	Pool1	Conv2.x	Pool2	Conv3.x	Pool3	Fully Connected
Num Layer	1	4	1	4	1	4	1	1
Filt Dim	3	3	2	3	2	3	2	1
Num Filt	64	64	–	96	—	128	–	256
Stride	1	1	2	1	2	1	2	1
Pad	1	1	–	1	–	1	—	–
CIFAR100	Conv0.x	Conv1.x	Pool1	Conv2.x	Pool2	Conv3.x	Pool3	Fully Connected
Num Layer	1	4	1	4	1	4	1	1
Filt Dim	3	3	2	3	2	3	2	1
Num Filt	96	96	–	192	–	384	–	512
Stride	1	1	2	1	2	1	2	1
Pad	1	1	–	1	–	1	–	–

verge of our training process. λ is the weighting parameter between softmax loss and IE loss in our final objective, to keep the balance between these two supervision symbols.

To alleviate the computational complexity of real gradients, we assume f_n, μ_c, σ^2 are three independent variables. One can refer to Appendix A for the complete derivation process. The gradients of $\mathcal{L}_{IE}(\Theta)$ with respect to f_n, μ_c, σ^2 are estimated as follows:

$$\frac{\partial \mathcal{L}_{IE}(\Theta)}{\partial f_n} = \frac{1}{M}\sum_{n=1}^{M}\left(\frac{f_n - \mu_{C(f_n)}}{\sigma^2} - \frac{f_n}{\sigma^2 Q} + \frac{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n - \mu_c\|_2^2)\cdot \mu_c}{\sigma^2 Q \sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n - \mu_c\|_2^2)}\right), \tag{8}$$

$$\frac{\partial \mathcal{L}_{IE}(\Theta)}{\partial \mu_q} = \begin{cases} \frac{1}{M}\sum_{n=1}^{M}\left(\frac{exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_q\|_2^2)\cdot \frac{f_n-\mu_q}{\sigma^2 Q}}{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)}\right), & q \neq C(f_n) \\ -\frac{1}{M}\sum_{n=1}^{M}\frac{f_n-\mu_q}{\sigma^2}, & q = C(f_n) \end{cases}, \tag{9}$$

$$\frac{\partial \mathcal{L}_{IE}(\Theta)}{\partial \sigma^2} = \frac{1}{M}\sum_{n=1}^{M}\left(\frac{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n - \mu_c\|_2^2)\cdot \frac{\|f_n-\mu_c\|_2^2}{2\sigma^4 Q}}{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n - \mu_c\|_2^2)} - \frac{\|f_n - \mu_{C(f_n)}\|_2^2}{2\sigma^4}\right). \tag{10}$$

4. Experiments

The concrete implementation details are given in Section 4.1. In Section 4.2, three kinds of CNNs with different capacity are given to validate the effectiveness of our algorithm on object recognition databases (MNIST [41], CIFAR10 [42] and CIFAR100 [42]). Some experiments on face recognition databases (LFW [43] and YTF [44]) are also performed in Section 4.3.

4.1. Implementation details

We use the Caffe library [45] to implement our experiments, and a speed-up parallel computing technique by two Tesla K80 GPUs is exploited. All the networks in this part are based on some existing CNNs. We partition them into three classes: the lighter, the normal and the powerful. We will refer to [L], [N] and [P] as their respective notations in the following experiments. The normal networks are shown in Tables 1 and 5 which are inspired by [12,40]. Also, the powerful ones are similar to [4,46]. We adopt ReLU [1] as the default activation function except in Table 1 where the PReLU [16] is used. The weight decay and momentum is set to 0.0005 and 0.9. Note that the mean subtraction image preprocessing is performed if not mentioned. The normally used SGD works well for the training. The lighter networks are some known structures built in Caffe library, and we comply with their original settings. In all these cases, we set α as 0.1 and Q as the entire interclass distances in the mini-batch, if not specified. The joint supervision of softmax loss and IE loss is necessary to accelerate the convergence of training process. When testing, the softmax classifier is used for object recognition, and cosine similarity metric is computed to obtain the face verification accuracies. For a fair comparison, we train four kinds of models in each experiment, namely under the supervision of softmax loss, softmax loss and L-Softmax loss, softmax loss and center loss, softmax loss and IE loss. For simplicity, we refer to the four original loss names as their corresponding methods. The details of every experiment about the training setups will be presented in their respective subsections subsequently. In all the experiments, only a single model is used to achieve the final performance.

4.2. Object recognition

MNIST This handwritten dataset has 60,000 training images and 10,000 testing images. In this section, we use two CNNs to validate the generalization of our algorithm. One is the lighter LeNet included in Caffe library. We train it according to the default updating strategy of learning rate and parameter initialization, eventually terminate it at 12k. The normal one is depicted in Table 1. This model is trained with the batch size of 256, and the learning rate is started from 0.01, divided by 10 at 12 k and 15 k iterations,

Table 2
Recognition error rate (%) on MNIST dataset.

Method	Error rate (%)
DropConnect [20]	0.57
CNN [47]	0.53
Maxout [15]	0.45
DSN [48]	0.39
R-CNN [49]	**0.31**
GenPool [50]	**0.31**
Softmax [L]	0.83
L-Softmax [L]	0.74
Center [L]	0.76
IE [L]	**0.49**
Softmax [N]	0.61
L-Softmax [N]	0.47
Center [N]	0.58
IE [N]	**0.31**

Table 3
Recognition error rate (%) on CIFAR10 dataset.

Method	Error rate (%)
Maxout [15]	11.68
DSN [48]	9.69
DropConnect [20]	9.41
All-CNN [51]	9.08
R-CNN [49]	8.69
GenPool [50]	**7.62**
Softmax [L]	21.88
L-Softmax [L]	-
Center [L]	19.40
IE [L]	**18.98**
Softmax [N]	11.56
L-Softmax [N]	9.59
Center [N]	10.25
IE [N]	**8.77**
Softmax [P]	6.59
L-Softmax [P]	6.46
Center [P]	6.17
IE [P]	**5.97**

Table 4
Recognition error rate (%) on CIFAR100 dataset.

Method	Error rate (%)
Maxout [15]	38.57
DSN [48]	34.57
All-CNN [51]	33.71
R-CNN [49]	**31.75**
Softmax [N]	33.31
L-Softmax [N]	30.79
Center [N]	29.39
IE [N]	**28.42**
Softmax [P]	27.06
L-Softmax [P]	26.21
Center [P]	26.15
IE [P]	**25.32**

eventually terminated at 20k iterations. In all these experiments, we only preprocess the images by dividing by 256 to provide them in range [0,1] as inputs. Some existing best results and the compared methods are shown in Table 2. It is obvious that IE loss not only outperforms other DML methods under the same setings, but also among the top performance compared to other state-of-the-art methods.

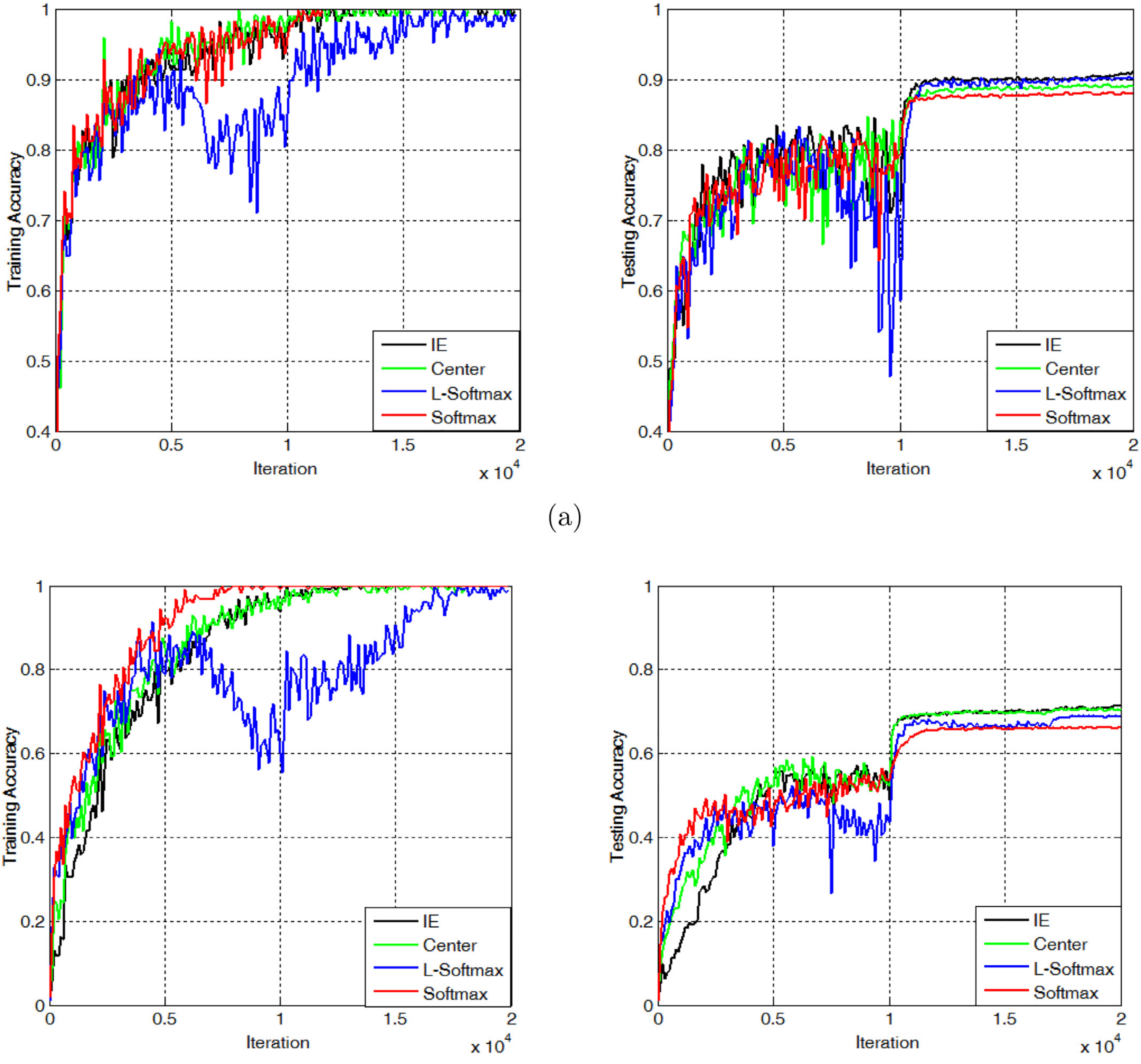
CIFAR10 This dataset has 10 classes of objects with 50k for training and 10k for testing. The experiments on three CNNs are carried out here. The lighter one is the Cifar10 network built in Caffe library. The updating strategy and initialization of parameters follow the original settings. The normal one is depicted in Table 1. We start with a learning rate of 0.01, divide it by 10 at 10k and 17 k iterations, and eventually terminate it at 22 k iterations. Simple mean/std normalization and horizontal flips are used to preprocess the dataset. The powerful one is WRN-28-10 as illustrated in [46], with some differences. The WRN-28-10 network is said to achieve a comparable accuracy with more than 1000 layers raw ResNet [4] on CIFAR10. To speed up the training process, we fine-tune the other three compared DML methods from the softmax baseline model. In this experiment, the dataset is preprocessed by global contrast normalization and mean/std normalization. We follow the standard data augmentation [40] for training, and the batch size is 128. The results are listed in Table 3. We can observe that our method always achieves the best performance among the four compared DML methods regardless of the size of CNNs.

CIFAR100 The final part of this section, we will verify the effectiveness of IE loss on CIFAR100 dataset. This dataset is just like the CIFAR10, except it has 100 classes containing 600 images per class, where 500 for training and 100 for testing. The 100 classes in CIFAR100 are grouped into 20 superclasses. Each image comes with a “fine” label (the class to which it belongs) and a “coarse” label (the superclass to which it belongs). We use the former protocol here. By convention, the normal network is shown in Table 1, and the powerful one is WRN-28-10. Also, the training strategy is the same as which described in CIFAR10. For the powerful WRN-28-10, we fine-tune the other three compared DML methods from the softmax baseline model. Differently, to better inspect the effectiveness of the compared methods with the capacity of networks growing, we preprocess the dataset in the same way on the normal and powerful networks, only by simple mean/std normalization and horizontal flips to augment data. In Table 4, we can clearly find that our method consistently performs better than other compared approaches.

From the results presented above, one can find that our IE loss always achieves the best results among the four compared DML methods on three object recognition datasets. Specifically, the performance of center loss and L-Softmax loss fluctuates significantly with different network structures. In Fig. 3, the training and testing process on CIFAR10 and CIFAR100 with the normal CNNs are displayed. It can be seen that the convergence rate of our IE loss is comparable with other compared loss functions, avoiding the notoriously slow convergence of triplet loss. Considering the performance gap between training and testing, one can observe that IE loss can mitigate the serious overfitting of softmax loss and the difficult convergence of L-Softmax loss. The testing accuracies of our method about different λ and α, and the best settings of them on the normal networks are shown in Appendix B.

4.3. Face verification

Different from object recognition, face verification is to compute the feature similarity of two images, and threshold comparison is exploited to decide whether the same person or not. Specifically, we use softmax classifier and metric loss functions to jointly supervise the training process, and the cosine similarity of two features is used to obtain the testing accuracy (Fig. 4). In this section, we evaluate our approach for face verification on LFW and YTF datasets. These two face datasets are the recognized benchmarks for face image and video, respectively. We use the publicly available CASIA-WebFace [52] as the training set, which originally has 494,414 labeled face images from 10,575 individuals. After removing the images failing to detect and mislabeled, the resulting dataset for our training is just over 430 K images. The cropped faces of all images are detected by [53], and 5 facial landmarks are labeled to globally align the face images by similarity transformation [54]. The normal network is depicted in Table 5, which is



(a)

(b)

Fig. 3. Accuracy vs. iteration curves using the normal networks on (a) CIFAR10 dataset and (b) CIFAR100 dataset.

a reduced version of ResNet [4] with 27 convolutional layers. The input faces are cropped to 112 × 96 RGB images, and the batch size is 256. Besides, the images are normalized by subtracting the mean image and dividing by 128. We start the training with a learning rate of 0.1, and divide it by 10 at 16 K, 24 K iterations, then terminate it at 28 K iterations. For face images, we find that using wider ResNet with fewer layers like WRN-28-10 does not bring so many benefits, and accompanied by rapidly growing memory space. So we decide to widen the network listed in Table 5 to obtain the powerful one. Specifically, we widen all the convolutional layers between Conv1 and Conv4 with a widening factor 2. When testing, we extract the features from both the frontal face and its mirror image, and merge the two features by element-wise summation. All the evaluations are based on the similarity scores of image pairs, which are computed by the cosine similarity of two representations after PCA.

Considering the difference from previous experiments, we select Q as the first 20% inter-class distances in every mini-batch to calculate the objective here. The reason is that some datasets like CASIA-WebFace have too many subjects, most of the inter-class distances tend to be very large in our method, thus leading to the difficult convergence of training process. Fig. 5a shows the verification accuracies on LFW with Q ranging from 0 to 100% of the number of inter-class distances. The importance of choosing a proper Q is displayed clearly. Here, we regard the case when $Q = 0$ as the original softmax method.

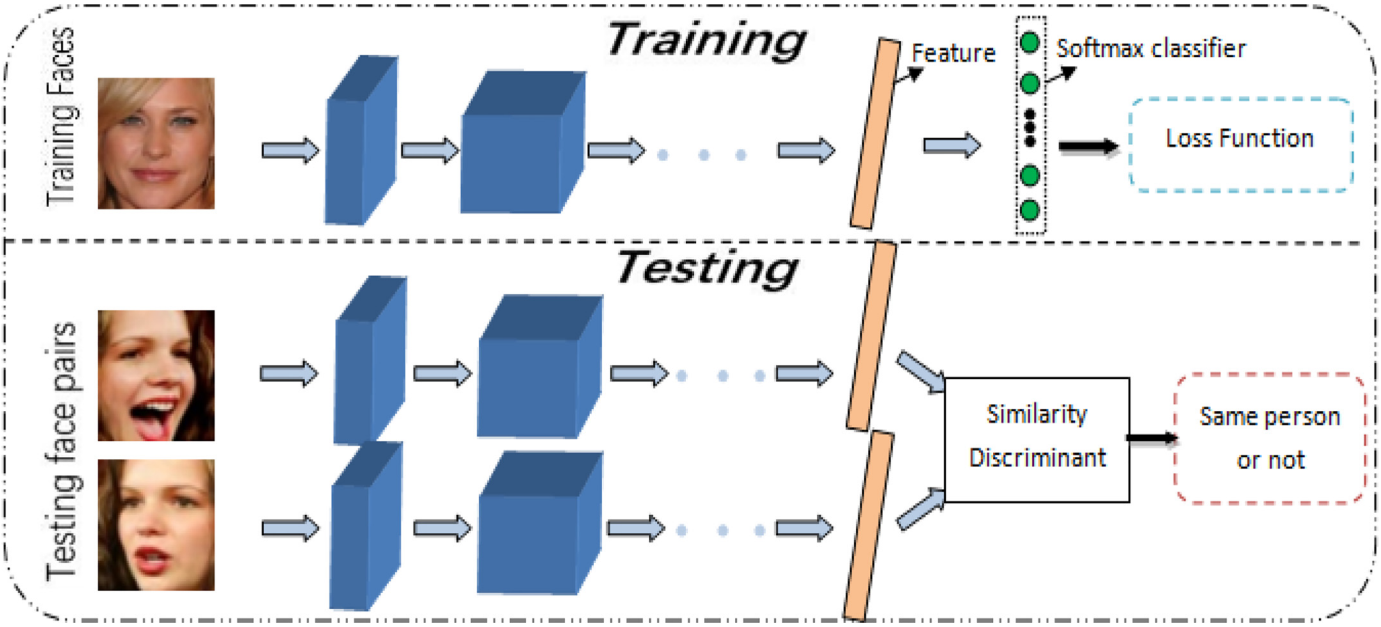


Fig. 4. The general pipeline for face verification in this paper, where classifier loss function is used to train and similarity discriminant is used to obtain the final verification accuracy.

Table 5
The normal ResNet architecture used for face verification. Resblock is the classical Residual unit which consists of two consecutive convolutional layers and a unit mapping.

Layer	Type	Filter Size/Stride	Output Size	Depth	Params
Conv0	Convolution	3 × 3/1	110 × 94 × 32	1	0.86K
Conv1	Convolution	3 × 3/1	108 × 92 × 64	1	18K
Pool1	Max pooling	2 × 2/2	54 × 46 × 64	0	–
Resblock1	Convolution	3 × 3/1	54 × 46 × 64	2	73K
Conv2	Convolution	3 × 3/1	52 × 44 × 128	1	73K
Pool2	Max pooling	2 × 2/2	26 × 22 × 128	0	–
Resblock2	Convolution	3 × 3/1	26 × 22 × 128	2	294K
Resblock3	Convolution	3 × 3/1	26 × 22 × 128	2	294K
Conv3	Convolution	3 × 3/1	24 × 20 × 256	1	294K
Pool3	Max pooling	2 × 2/2	12 × 10 × 256	0	–
Resblock4	Convolution	3 × 3/1	12 × 10 × 256	2	1179K
Resblock5	Convolution	3 × 3/1	12 × 10 × 256	2	1179K
Resblock6	Convolution	3 × 3/1	12 × 10 × 256	2	1179K
Resblock7	Convolution	3 × 3/1	12 × 10 × 256	2	1179K
Resblock8	Convolution	3 × 3/1	12 × 10 × 256	2	1179K
Conv4	Convolution	3 × 3/1	10 × 8 × 512	1	1179K
Pool4	Max pooling	2 × 2/2	5 × 4 × 512	0	–
Resblock9	Convolution	3 × 3/1	5 × 4 × 512	2	4718K
Resblock10	Convolution	3 × 3/1	5 × 4 × 512	2	4718K
Resblock11	Convolution	3 × 3/1	5 × 4 × 512	2	4718K
Fc5	Fully connection	–	1 × 1 × 512	1	5242K

Table 6
Face verification performance (%) on LFW and YTF datasets.

Method	Points for Alig.	Outside data	Networks	Acc. on LFW (%)	Acc. on YTF (%)
High-dim LBP [55]	27	100 K	–	95.17	–
DeepFace[7]	73	4 M	3	97.35	91.40
Gaussian Face [8]	–	20 K	1	98.52	–
DeepID [9]	5	200 K	1	97.45	–
DeepID-2+ [10]	18	300 K	25	99.47	93.20
FaceNet [11]	–	200 M	1	**99.63**	**95.10**
DCNN [56]	7	490 K	1	97.45	–
CASIA-WebFace [52]	2	490 K	1	97.73	90.60
Softmax [N]	5	430 K	1	97.42	91.52
Triplet Loss [N]	5	430 K	1	98.20	92.16
L-Softmax [N]	5	430 K	1	98.86	**94.14**
Center [N]	5	430 K	1	98.91	93.80
IE [N]	5	430 K	1	**99.10**	94.12
Softmax [P]	5	430 K	1	97.73	92.42
Triplet Loss [P]	5	430 K	1	98.23	91.98
L-Softmax [P]	5	430 K	1	98.67	92.66
Center [P]	5	430 K	1	99.01	**94.12**
IE [P]	5	430 K	1	**99.15**	**94.12**

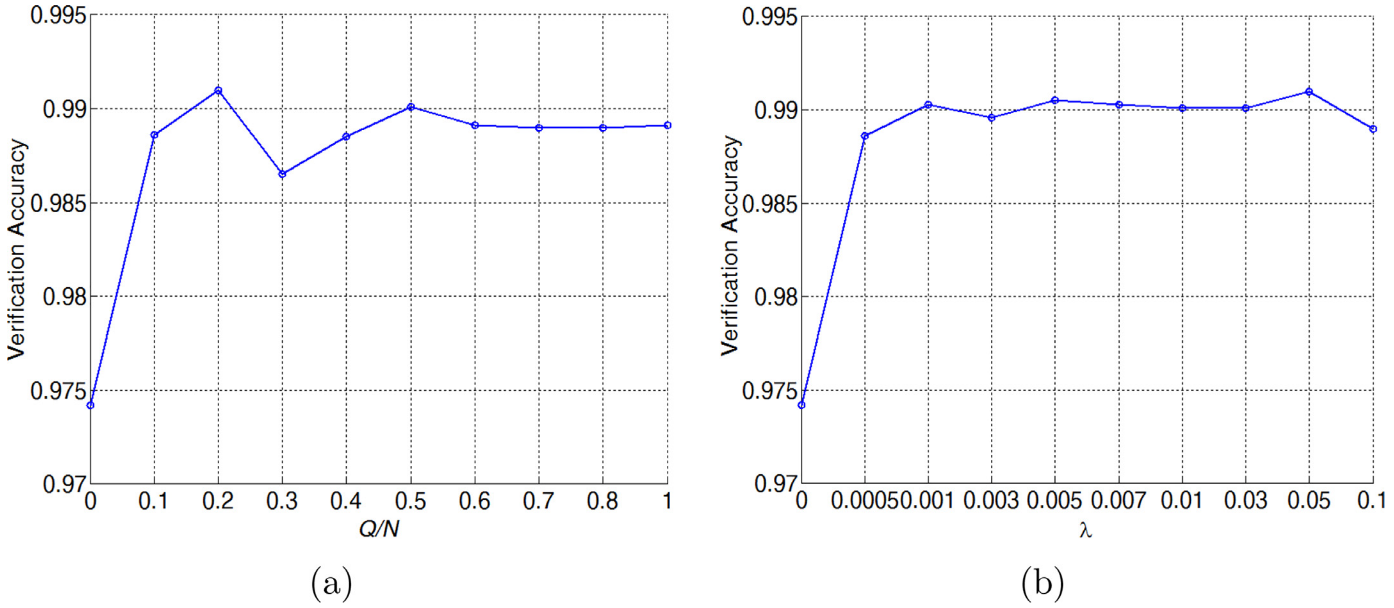


Fig. 5. (a) Verification accuracies of IE loss with different Q/N on LFW using the normal network, where N is the number of inter-class distances regarding a sample in a mini-batch. (b) Face verification accuracies of IE Loss on LFW with different λ using the normal network.

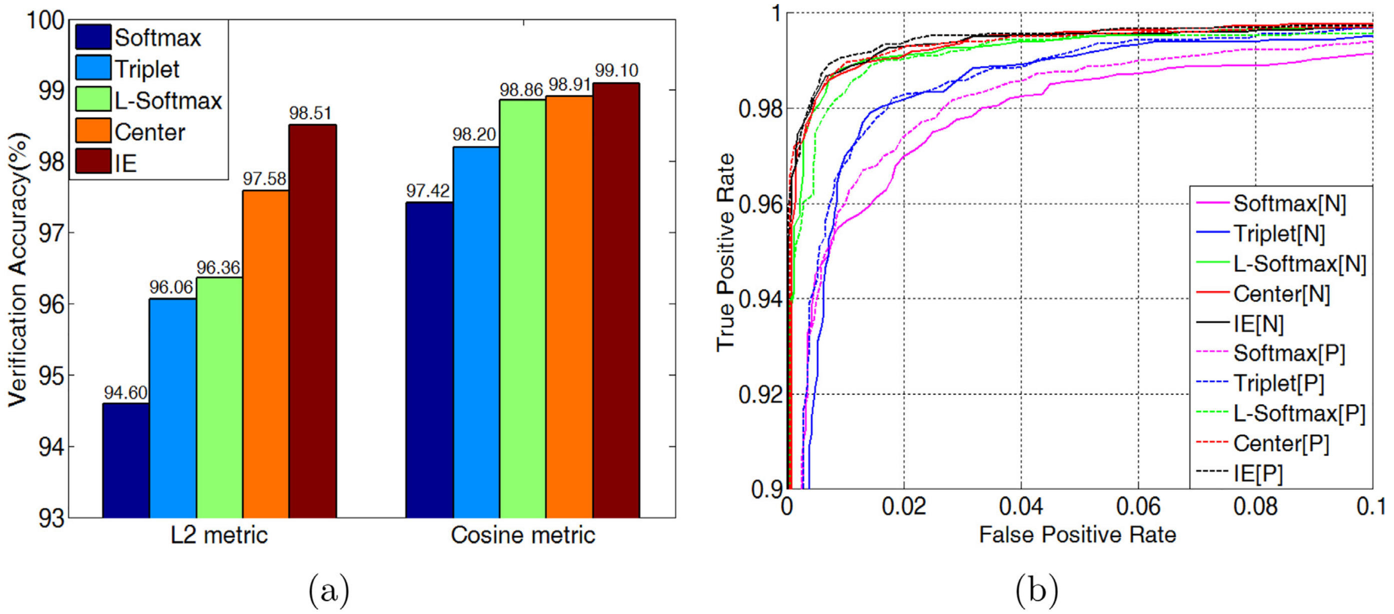


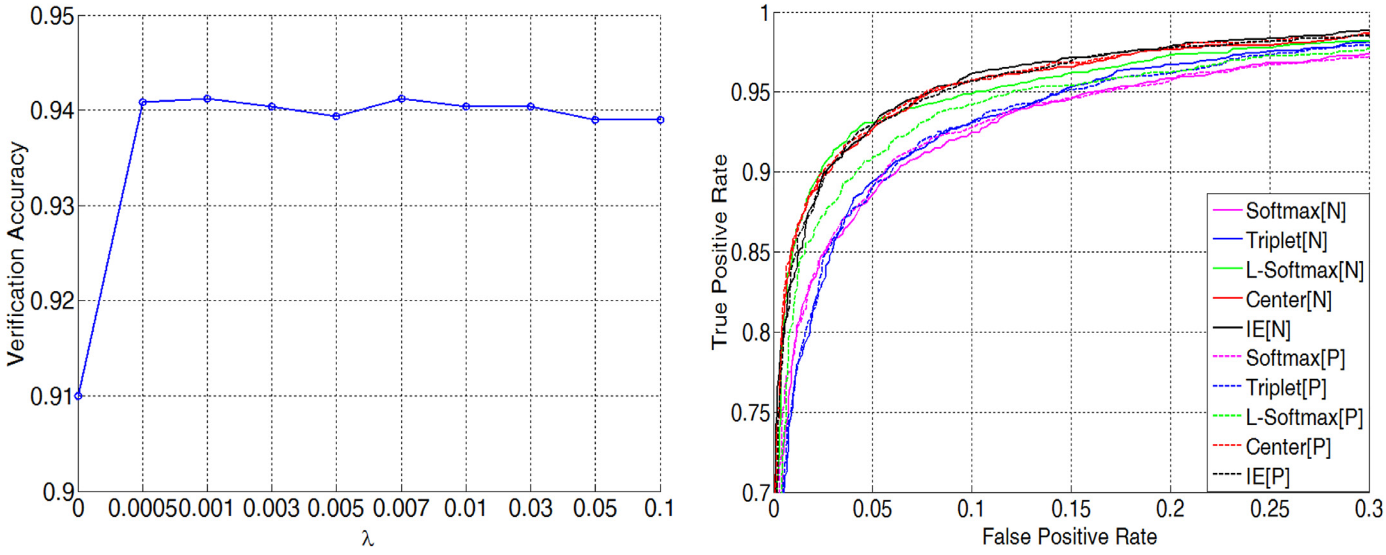
Fig. 6. (a)Verification accuracies of compared loss functions with two different similarity metrics on LFW using the normal network. (b) ROC curves of five compared loss functions on LFW.

LFW This dataset contains 13,233 face images of 5749 different identities from the Internet, with large variations in pose, expression and illumination. For comparison purpose, algorithms typically report the mean face verification accuracies and the ROC curves on 6000 given face pairs, following the standard protocol of unrestricted with labeled outside data [43]. According to previous experience, we find that a properly chosen λ which balances the weight between softmax loss and IE loss can improve the performance. So we experiment our method across a wide range of λ from 0 to 0.1 to select the best setting. The results on LFW are shown in Fig. 5b. It can be seen that IE loss is stable with different λ, and the best setting is 0.05.

Fig. 6a illustrates the verification accuracies of five loss functions with two different similarity metrics for testing. The results show that cosine similarity is more suitable than L2 similarity for our feature representations. Obviously, our method is robust to both cases, and always achieves the best performance.

YTF This dataset consists of 3,425 videos from 1,595 different people, with an average of 2.15 videos for everyone. Besides, the average length of a video clip is 181.3 frames, with each clip duration varying from 48 frames to 6070 frames. Just as the experiments on LFW, we report the results on 5000 video pairs in Table 6, according to the unrestricted protocol with labeled outside data in [44]. Also, Fig. 7 shows the accuracy of IE loss in regard to different λ ranging from 0 to 0.1 and the ROC curves of five compared loss functions.

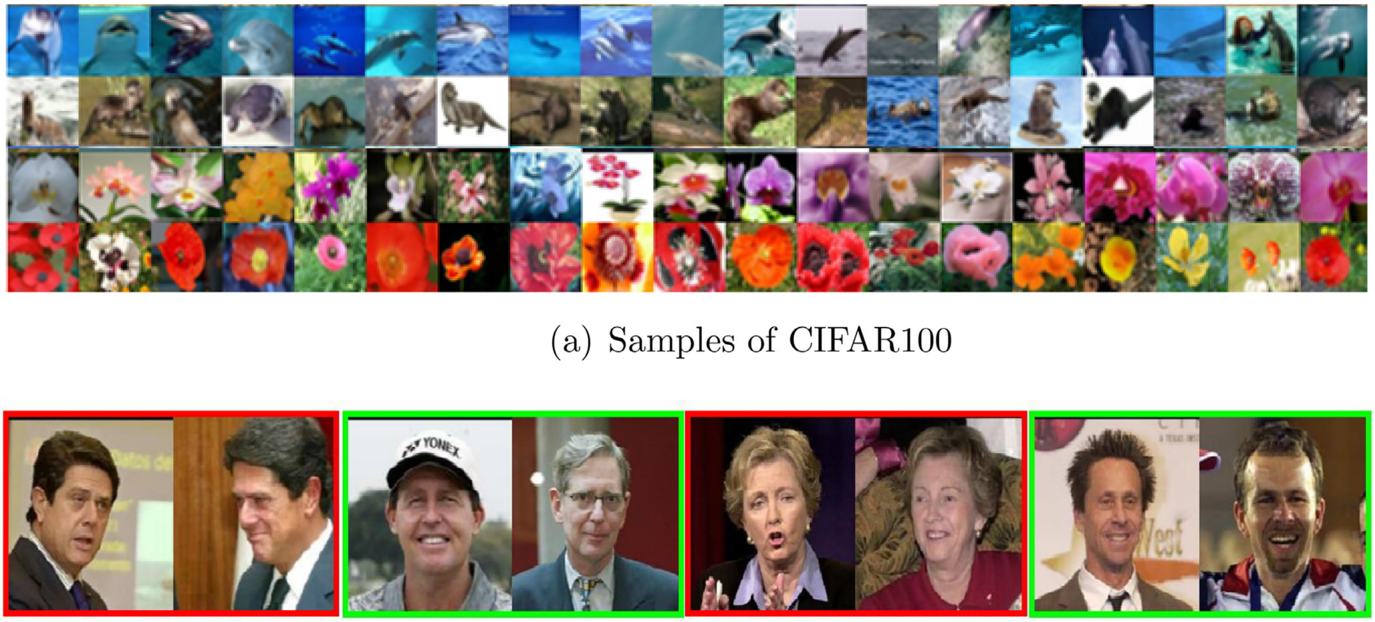
From the verification results in Table 6 and ROC curves on these two datasets, we can find that the performance on the



(a)

(b)

Fig. 7. (a) Face verification accuracies of IE Loss on YTF with different λ using the normal ResNet. (b) ROC curves of five compared loss functions on YTF.



(a) Samples of CIFAR100

(b) Face images in LFW

Fig. 8. Some examples of the datasets in our experiments. The image pairs in red are those positive pairs that our method succeeds to recognize, while the softmax method fails. Likewise, the green ones are some negative pairs. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

powerful network is consistently superior to which on the normal one except the L-Softmax loss. IE loss is always outstanding in the five loss functions under a small training dataset of CASIA-WebFace, and competitive with the state-of-the-art methods using larger training datasets or model ensemble. Noticeably, the results of triplet loss and L-Softmax loss are not satisfactory, and there exhibits a large margin of triplet loss compared to the results in [11]. This convincingly demonstrates the difficult convergence and big data dependence of triplet loss. We conjecture that maybe the performance of our method can be improved considerably if a larger training set or a more powerful network is used. Anyway, the excellent performance undoubtedly verify the great generalization of IE loss. The visualization of some datasets is shown in Fig. 8.

5. Conclusion and future work

In this paper, we propose a powerful and replicable DML method, which enforces the mean inter-class distance larger than the intra-class distance with a margin, to enhance the discriminability of the deeply learned features in object recognition and face verification. Extensive experiments on several public datasets have convincingly demonstrated the effectiveness of our method. The results also exhibit the excellent generalization of IE loss in various size of CNNs. Instead of requiring a superior neighborhood sampling strategy, our approach only uses mini-batch based SGD to conduct the experiments, avoiding the exponentially increased computational complexity of image pairs or triplets. Maybe a better hard sample mining strategy could improve the performance further. Inspired by the outstanding performance of IE loss in object recognition and face recognition, we will explore its extension in the case where the swarm intelligent methods are exploited to optimize the clustering algorithm [57,58] in the following work. In the future, we will delve into DML to explore its extensive applications to other tasks.

Acknowledgments

The authors would like to thank Kun Shang, Mengya Zhang, Ruipeng Shen and Wenjuan Li for their helpful advices. This research was supported by the National Science Foundation of China.

Appendix A

In this section, we concretely describe the deduction of gradient formulas (9)∼(11) listed in Section 3.2. First, we rewrite Eq. (6) as follows:

$$\mathcal{L} = \frac{1}{M}\sum_{n=1}^{M}\left\{-log\frac{exp(-\frac{1}{2\sigma^2}\|f_n-\mu_{C(f_n)}\|_2^2-\alpha)}{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)}\right\}_+ . \tag{A.1}$$

We need to compute the gradient formulas of $\mathcal{L}$ with respect to f_n, μ_c and σ^2. Note that directly computing the real gradients of them leads to costly computational complexity in training. So we will consider f_n, μ_c and σ^2 as three independent variables. If the value in $\{\cdot\}$ is positive, then

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial f_n} &= -\frac{1}{M}\cdot\frac{\partial}{\partial f_n}\left(\sum_{n=1}^{M} log\frac{exp(-\frac{1}{2\sigma^2}\|f_n-\mu_{C(f_n)}\|_2^2-\alpha)}{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)}\right)\\ &= \frac{1}{M}\cdot\frac{\partial}{\partial f_n}\left(\frac{\|f_n-\mu_{C(f_n)}\|_2^2}{2\sigma^2}\right.\\ &\quad\left.+\alpha+\log\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)\right)\\ &= \frac{1}{M}\sum_{n=1}^{M}\left(\frac{f_n-\mu_{C(f_n)}}{\sigma^2}-\frac{f_n}{\sigma^2 Q}\right.\\ &\quad\left.+\frac{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)\cdot\mu_c}{\sigma^2 Q\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)}\right).\end{aligned} \tag{A.2}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mu_q} &= \frac{1}{M}\cdot\frac{\partial}{\partial\mu_q}\left(\frac{\|f_n-\mu_{C(f_n)}\|_2^2}{2\sigma^2}\right.\\ &\quad\left.+\alpha+\log\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)\right).\end{aligned} \tag{A.3}$$

When $q\neq C(f_n)$, we have

$$\frac{\partial\mathcal{L}}{\partial\mu_q} = \frac{1}{M}\sum_{n=1}^{M}\left(\frac{exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_q\|_2^2)\cdot\frac{f_n-\mu_q}{\sigma^2 Q}}{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)}\right). \tag{A.4}$$

When $q=C(f_n)$, we have

$$\frac{\partial\mathcal{L}}{\partial\mu_q} = -\frac{1}{M}\sum_{n=1}^{M}\frac{f_n-\mu_q}{\sigma^2}. \tag{A.5}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\sigma^2} &= \frac{1}{M}\cdot\frac{\partial}{\partial\sigma^2}\left(\frac{\|f_n-\mu_{C(f_n)}\|_2^2}{2\sigma^2}\right.\\ &\quad\left.+\alpha+\log\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)\right)\\ &= \frac{1}{M}\sum_{n=1}^{M}\left(\frac{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)\cdot\frac{\|f_n-\mu_c\|_2^2}{2\sigma^4 Q}}{\sum_{c=1,c\neq C(f_n)}^{Q} exp(-\frac{1}{2\sigma^2 Q}\|f_n-\mu_c\|_2^2)}\right.\\ &\quad\left.-\frac{\|f_n-\mu_{C(f_n)}\|_2^2}{2\sigma^4}\right).\end{aligned} \tag{A.6}$$

Appendix B

Here we describe the accuracy results about different hyperparameters and the optimal settings on object recognition using the little and normal networks in details. All the experiments in this part obey the following steps. First, we fix α to 0.1 and vary λ according to its corresponding range in different databases. Then, we fix λ to the best setting from the previous results and vary α to find the final optimal setting. Both the optimal values of λ and α are displayed in bold.

Table B.1
The recognition accuracy of IE loss on MNIST regarding different value of λ and α, respectively with (a) LeNet built in Caffe library and (b) MNIST network depicted in Tab.1.

(a)				(b)			
λ	Accuracy	α	Accuracy	λ	Accuracy	α	Accuracy
0.110	0.9939	0.01	0.9945	0.001	0.9964	0.01	0.9961
0.115	0.9936	0.03	0.9939	0.004	0.9958	0.03	0.9965
0.120	0.9940	0.05	0.9938	0.007	0.9952	0.05	0.9962
0.125	0.9949	0.07	0.9943	0.010	0.9963	0.07	0.9967
0.130	0.9944	**0.10**	**0.9951**	0.030	0.9961	0.09	0.9962
0.135	0.9940	0.15	0.9950	0.050	0.9962	**0.10**	**0.9969**
0.140	0.9938	0.20	0.9936	0.070	0.9958	0.13	0.9956
0.150	0.9937	0.25	0.9945	0.090	0.9961	0.15	0.9959
0.170	0.9935	0.30	0.9943	0.110	0.9961	0.18	0.9958
0.190	0.9938	0.35	0.9945	0.130	0.9961	0.21	0.9966
0.210	0.9943	**0.40**	**0.9951**	**0.150**	**0.9969**	0.24	0.9967
0.230	0.9945	0.45	0.9938	0.170	0.9963	0.27	0.9963
0.250	0.9945	0.50	0.9942	0.190	0.9955	0.30	0.9958
0.270	0.9945	0.55	0.9947	0.210	0.9963		
0.290	0.9946	0.60	0.9941	0.230	0.9960		
0.310	0.9944	0.65	0.9937	0.250	0.9957		
0.330	0.9945	0.70	0.9948	0.270	0.9959		
0.350	0.9943	0.75	0.9946				
0.370	0.9947	0.80	0.9942				
0.390	0.9938	0.85	0.9945				
0.410	0.9943	0.90	0.9940				
0.430	**0.9951**	0.95	0.9940				
0.450	0.9945	1.00	0.9938				
0.470	0.9945						
0.500	0.9948						
0.550	0.9946						
0.600	0.9943						
0.650	0.9942						
0.700	0.9937						
0.750	0.9945						
0.800	0.9942						
0.850	0.9944						
0.900	0.9941						
0.950	0.9949						
1.000	0.9938						

Table B.2
The recognition accuracy of IE loss on CIFAR10 regarding different value of λ and α, respectively with (a) CIFAR10 built in Caffe library and (b) CIFAR10 network depicted in Table 1.

(a)				(b)			
λ	Accuracy	α	Accuracy	λ	Accuracy	α	Accuracy
0.001	0.8028	0.001	0.8057	0.001	0.9086	0.001	0.9093
0.004	0.8054	0.005	0.8018	0.005	0.9102	0.005	0.9087
0.008	0.8064	0.010	0.8068	0.008	0.9109	0.010	0.9075
0.010	0.8063	0.050	0.8029	0.011	0.9108	0.050	0.9066
0.040	0.8011	0.100	0.8093	0.015	0.9088	**0.100**	**0.9123**
0.080	0.7950	0.150	0.8032	0.030	0.9111	0.200	0.9100
0.100	0.8033	0.200	0.7972	0.050	0.9078	0.250	0.9088
0.130	0.8012	0.250	0.7989	**0.070**	**0.9123**	0.300	0.9073
0.160	0.8064	0.300	0.7996	0.100	0.9099		
0.190	0.7959	0.350	0.8059	0.150	0.9111		
0.210	0.7998	**0.40**	**0.8102**	0.200	0.9057		
0.240	0.8002	0.450	0.8064	0.250	0.9043		
0.270	**0.8093**	0.500	0.8031	0.300	0.9035		
0.300	0.8073	0.550	0.8075	0.350	0.9078		
0.330	0.8055	0.600	0.7954	0.400	0.9061		
0.370	0.8049	0.650	0.8045	0.450	0.9091		
0.400	0.8078	0.700	0.8015	0.500	0.9095		
0.430	0.8042	0.750	0.8051	0.550	0.9084		
0.470	0.8019	0.800	0.8027	0.600	0.9070		
0.500	0.8066	0.850	0.8072				
0.530	0.8044	0.900	0.8058				
0.570	0.8028						
0.600	0.8005						
0.650	0.7911						
0.700	0.8074						
0.750	0.8018						
0.800	0.8082						
0.850	0.8022						
0.900	0.8006						
0.950	0.8076						
1.000	0.8094						

Table B.3
The recognition accuracy of IE loss on CIFAR100 with the CIFAR100 network depicted in Tab.1, in regard to different value of λ and α, respectively.

λ	0.001	0.003	0.005	0.007	0.010	0.030	0.050	0.070	0.100	0.200
Accuracy	70.41	71.41	71.14	**71.58**	70.72	70.72	70.85	71.11	70.80	70.62
α	0.007	0.005	0.001	0.010	**0.100**	0.200				
Accuracy	71.06	70.97	71.16	70.82	**71.58**	71.18				

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of International Conference on Learning Representations, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Object detectors emerge in deep scene CNNS, in: Proceedings of International Conference on Learning Representations, 2014.

[6] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, in: Proceedings of Advances in Neural Information Processing systems, 2014, pp. 487–495.

[7] Y. Taigman, M. Yang, M. Ranzato, 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] C. Lu, X. Tang, Surpassing human-level face verification performance on LFW with Gaussianface, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2014.

[9] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1891–1898.

[10] Y. Sun, X. Wang, 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.

[11] F. Schroff, D. Kalenichenko, 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.

[12] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, in: Proceedings of European Conference on Computer Vision, Springer, 2016, pp. 499–515.

[13] G. Levi, T. Hassner, Age and gender classification using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 34–42.

[14] H. Liu, J. Lu, J. Feng, J. Zhou, Group-aware deep feature learning for facial age estimation, Pattern Recogn. 66 (2016) 82–94.

[15] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A.C. Courville, Y. Bengio, Maxout networks., in: Proceedings of the 30th International Conference on Machine Learning, 28, 2013, pp. 1319–1327.

[16] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[17] W. Shang, K. Sohn, D. Almeida, H. Lee, Understanding and improving convolutional neural networks via concatenated rectified linear units, in: Proceedings of International Conference on Machine Learning, 2016.

[18] D.C. Cireşan, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, High-performance neural networks for visual object classification, arXiv preprint arXiv:1102.0183 (2011).

[19] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580 (2012).

[20] L. Wan, M. Zeiler, S. Zhang, Y.L. Cun, R. Fergus, Regularization of neural networks using dropconnect, in: Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 1058–1066.

[21] S. Han, H. Mao, W.J. Dally, Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding, in: Proceedings of International Conference on Learning Representations, 2015.

[22] Y. Sun, X. Wang, 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.

[23] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Proceedings of Advances in Neural Information Processing Systems, 2014, pp. 1988–1996.

[24] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.

[25] B. Moghaddam, T. Jebara, A. Pentland, Bayesian face recognition, Pattern Recogn. 33 (11) (2000) 1771–1782.

[26] X. Wang, X. Tang, A unified framework for subspace face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1222–1228.

[27] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: a joint formulation, Proceedings of European Conference on Computer Vision, Springer, 2012, pp. 566–579.

[28] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, in: Proceedings of Advances in Neural Information Processing Systems, 2005, pp. 1473–1480.

[29] M. Kan, S. Shan, Y. Su, D. Xu, X. Chen, Adaptive discriminant learning for face recognition, Pattern Recogn. 46 (9) (2013) 2497–2509.

[30] J. Song, L. Gao, Y. Yan, D. Zhang, N. Sebe, Supervised hashing with pseudo labels for scalable multimedia retrieval, Proceedings of the 23rd ACM International Conference on Multimedia, ACM, 2015, pp. 827–830.

[31] L. Gao, J. Song, F. Zou, D. Zhang, J. Shao, Scalable multimedia retrieval by deep learning hashing with relative similarity learning, Proceedings of the 23rd ACM International Conference on Multimedia, ACM, 2015, pp. 903–906.

[32] L. Gao, J. Song, X. Liu, J. Shao, J. Liu, J. Shao, Learning in high-dimensional multimedia data: the state of the art, Multimed. Syst. 23 (3) (2017) 303–313.

[33] J. Wang, T. Zhang, N. Sebe, H.T. Shen, et al., A survey on learning to hash, IEEE Trans. Pattern Anal. Mach. Intell. PP (2017) 1-1.

[34] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[35] O. Rippel, M. Paluri, P. Dollar, L. Bourdev, Metric learning with adaptive density discrimination, in: Proceedings of International Conference on Learning Representations, 2016.

[36] J. Song, Binary generative adversarial networks for image retrieval, in: Proceedings of the 32th AAAI Conference on Artificial Intelligence, 2018.

[37] J. Song, L. Gao, L. Li, X. Zhu, N. Sebe, Quantization based hashing: a general framework for scalable image and video retrieval, Pattern Recogn. 75 (2018) 178–187.

[38] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2013, pp. 6645–6649.

[39] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1631–1642.

[40] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, in: Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 507–516.

[41] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[42] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report, Citeseer, 2009.

[43] G.B. Huang, E. Learned-Miller, Labeled Faces in the Wild:Updates and New Reporting Procedures, Department of Computer Science, University of Massachusetts Amherst, Amherst, MA, USA, Technical Report (2014) 14–23.

[44] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 529–534.

[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675–678.

[46] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proceedings of British Machine Vision Conference, 2016.

[47] K. Jarrett, K. Kavukcuoglu, Y. Lecun, et al., What is the best multi-stage architecture for object recognition, Proceedings of 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 2146–2153.

[48] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, Z. Tu, Deeply-supervised nets, in: Proceedings of AISTATS, volume 2, 2015, p. 5.

[49] M. Liang, X. Hu, Recurrent convolutional neural network for object recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3367–3375.

[50] C.-Y. Lee, P.W. Gallagher, Z. Tu, Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree, in: Proceedings of International Conference on Artificial Intelligence and Statistics, 2016.

[51] J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: the all convolutional net, in: Proceedings of International Conference on Learning Representations, 2015.

[52] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, arXiv preprint arXiv:1411.7923 (2014).

[53] S. Wu, M. Kan, Z. He, S. Shan, X. Chen, Funnel-structured cascade for multi--view face detection with alignment-awareness, Neurocomputing 221 (2017) 138–145.

[54] J. Zhang, S. Shan, M. Kan, X. Chen, Coarse-to-fine auto-encoder networks (CFAN) for real-time face alignment, Proceedings of European Conference on Computer Vision, Springer, 2014.

[55] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3025–3032.

[56] J.-C. Chen, V.M. Patel, R. Chellappa, Unconstrained face verification using deep CNN features, in: Proceedings of Winter Conference on Applications of Computer Vision, 2016, pp. 1–9.

[57] N. Zeng, Z. Wang, H. Zhang, W. Liu, F.E. Alsaadi, Deep belief networks for quantitative analysis of a gold immunochromatographic strip, Cogn. Comput. 8 (4) (2016) 684–692.

[58] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, A.M. Dobaie, Facial expression recognition via learning deep sparse autoencoders, Neurocomputing 273 (2018) 643–649.

Bowen Wu is currently a Ph.D. candidate who majors in applied mathematics in Nankai University. His research interests include combinatorics, graph theory, machine learning, face recognition, object recognition and deep learning.

Zhangling Chen is currently a Ph.D. candidate who majors in applied mathematics in Tianjin University. Her research interests include face recognition and deep learning.

Jun Wang received the M.S. degree in fundamental mathematics from Beijing Institute of Technology, Beijing, China, in 2007. From 2007 to 2016, he acted as the experiment technology division in Nankai University. Currently, he is engaged in technical R&D in Tianjin University. His research interests include pattern recognition, face recognition, machine learning and parallel programming.

Huaming Wu received the B.E. and M.S. degrees from Harbin Institute of Technology, China in 2009 and 2011, respectively, both in electrical engineering. He received the Ph.D. degree with the highest honor in computer science at Free University of Berlin, Germany in 2015. He is currently an assistant professor in the Center for Applied Mathematics, Tianjin University. His research interests include mobile cloud computing, face recognition and deep learning. In these areas, he has published over 10 papers in leading international journals or conference proceedings. He has served as a student forum chair and program committee member in many international conferences/workshops.