A Discriminative Deep Feature Learning Approach for Face Recognition

Anonymous ECCV submission

Paper ID 1458

Abstract. Convolutional neural networks (CNNs) have been widely used in computer vision community, significantly improving the state-of-the-art. In most of the available CNNs, the softmax loss function is used as the supervision signal to train the deep model. In order to enhance the discriminative power of the deeply learned features, this paper proposes a new supervision signal, called center loss, for face recognition task. Specifically, the center loss simultaneously learns a center for deep features of each class and penalizes the distances between the deep features and their corresponding class centers. More importantly, we prove that the proposed center loss function is trainable and easy to optimize in the CNNs. With the joint supervision of softmax loss and center loss, we can train a robust CNNs to obtain the deep features with the two key learning objectives, inter-class dispension and intra-class compactness as much as possible, which are very essential to face recognition. It is encouraging to see that our CNNs (with such joint supervision) achieve the state-of-the-art accuracy on several important face recognition benchmarks, Labeled Faces in the Wild (LFW), YouTube Faces (YTF), and MegaFace Challenge. Especially, our new approach achieves the best results on the largest public domain face benchmark (The MegaFace Benchmark: 1 Million Faces for Recognition at Scale [1]) under the protocol of small training set (contains under 500000 images and under 20000 persons), significantly improving the previous results and setting new state-of-the-art for both face recognition and face verification tasks.

Keywords: Convolutional Neural Networks, face recognition, discriminative feature learning, center loss

1 Introduction

Convolutional neural networks (CNNs) have achieved great success on vision community, significantly improving the state of the art in classification problems, such as object [2–6], scene [7, 8], action [9–11] and so on. It mainly benefits from the large scale training data [12, 13] and the end-to-end learning framework. The most commonly used CNNs perform feature learning and label prediction, mapping the input data to deep features (the output of the last hidden layer), then to the predicted labels, as shown in Figure 1.

In generic object, scene or action recognition, the classes of the possible testing samples are within the training set, which is also referred to close-set

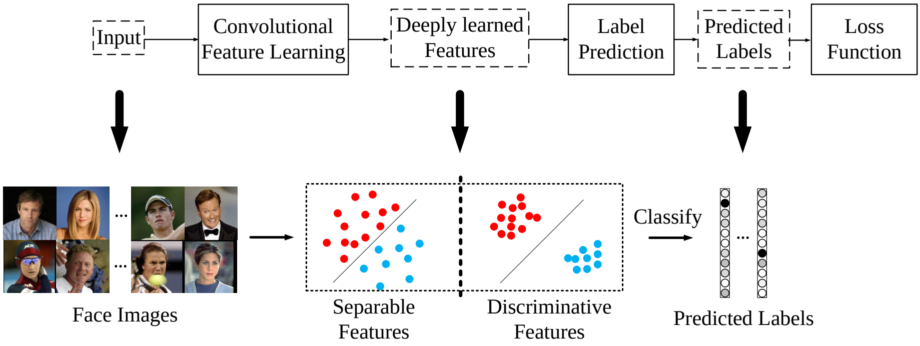


Fig. 1. The typical framework of convolutional neural networks. **Best viewed in color.**

identification. Therefore, the predicted labels dominate the performance and softmax loss is able to directly address the classification problems. In this way, the label prediction (the last fully connected layer) acts like a linear classifier and the deeply learned features are prone to be separable.

Face recognition can be categorized as face identification and face verification [14, 15]. Face identification is to match a probe face image to one of the persons in gallery, while face verification aims to decide whether a given pair of face images is from the same person or not. For face recognition task, the deeply learned features need to be not only separable but also discriminative. Since it is impractical to pre-collect all the possible testing identities for training, the label prediction in CNNs is not always applicable. The deeply learned features are required to be discriminative and generalized enough for identifying new unseen classes without label prediction. Discriminative power characterizes features in both the compact intra-class variations and separable inter-class differences, as shown in Figure 1. Discriminative features can be well-classified by nearest neighbor (NN) [16] or k-nearest neighbor (k-NN) [17] algorithms, which do not necessarily depend on the label prediction. However, the softmax loss only encourage the separability of features. Therefore, the resulting features are not sufficiently effective for face recognition.

Constructing highly efficient loss function for discriminative feature learning in CNNs is non-trivial. Because the stochastic gradient descent (SGD) [18] optimizes the CNNs based on mini-batch, which can not reflect the global distribution of deep features very well. Due to the huge scale of training set, it is impractical to input all the training samples in every iteration. As alternative approaches, contrastive loss [19, 20] and triplet loss [21] respectively construct loss functions for image pairs and triplet. However, compared to the image samples, the number of training pairs or triplets dramatically grows. It inevitably results in slow convergence and instability. By carefully selecting the image pairs or triplets, the problem may be partially alleviated. But it significantly increases the computational complexity and the training procedure becomes inconvenient.

In this paper, we propose a new loss function, namely center loss, to efficiently enhance the discriminative power of the deeply learned features in neural networks. Specifically, we learn a center (a vector with the same dimension as a feature) for deep features of each class. In the course of training, we simultaneously update the center and minimize the distances between the deep features and their corresponding class centers. The CNNs are trained under the joint supervision of the softmax loss and center loss, with a hyper parameter to balance the two supervision signals. Intuitively, the softmax loss forces the deep features of different classes staying apart. The center loss efficiently pulls the deep features of the same class to their centers. With the joint supervision, not only the inter-class features differences are enlarged, but also the intra-class features variations are reduced. Hence the discriminative power of the deeply learned features can be highly enhanced. Our main contributions are summarized as follows.

- We propose a new loss function (called center loss) to minimize the intra-class distances of the deep features. To be best of our knowledge, this is the first attempt to use such a loss function to help supervise the learning of CNNs. With the joint supervision of the center loss and the softmax loss, the highly discriminative features can be obtained for robust face recognition, as supported by our experimental results.
- We show that the proposed loss function is very easy to implement in the CNNs. Our CNN models are trainable and can be directly optimized by the standard SGD.
- We present extensive experiments on the datasets of MegaFace Challenge [1] (the largest public domain face database with 1 million faces for recognition) and set new state-of-the-art under the evaluation protocol of small training set. We also verify the excellent performance of our new approach on Labeled Faces in the Wild (LFW) [22] and YouTube Faces (YTF) datasets [23].

2 Related work

Face recognition via deep learning has achieved a series of breakthrough in these years. The idea of mapping a pair of face images to a distance starts from [24]. They train siamese networks for driving the similarity metric to be small for positive pairs, and large for the negative pairs. [25] proposes a hybrid network structure for face verification, where the CNNs are used to extract hierarchical features and the Restricted Boltzmann Machine (RBM)[26] model is using for feature fusion. The hybrid networks can be jointly optimized for better performance. Hu *et al.* [27] learn a nonlinear transformations and yield discriminative deep metric with a margin between positive and negative face image pairs. There approaches are required image pairs as input.

Very recently, [28] and [29] supervise the learning process in CNNs by identification signal (softmax loss function). Since CNNs form identity-related features through challenging identification task. After that, joint identification-verification supervision signal is adopted in [20], leading to more discriminative features. [30] enhances the supervision by adding a fully connected layer and loss functions to each convolutional layer. The effectiveness of triplet loss has been demonstrated in [21], [31] and [32]. With the deep embedding, the distance between an anchor and a positive are minimized, while the distance between an anchor and a negative are maximized until the margin is met. They achieve state-of-the-art performance in LFW and YTF datasets.

3 Our Approach

In this Section, we elaborate our approach. We first use a toy example to intuitively show the distributions of the deeply learned features. Inspired by the distribution, we propose the center loss to improve the discriminative power of the deeply learned features, followed by some discussions.

Table 1. The CNNs architecture we use in toy example, called LeNets++. Some of the convolution layers are followed by max pooling. $(5,32)_{/1,2} \times 2$ denotes 2 cascaded convolution layers with 32 filters of size 5×5, where the stride and padding are 1 and 2 respectively. In LeNets++, we use the Parametric Rectified Linear Unit (PReLU) [5] as the nonlinear unit.

	stage 1		stage 2		stage 3		stage 4
Layer	conv	pool	conv	pool	conv	pool	FC
LeNets	$(5,20)_{/1,0}$	$2_{/2,0}$	$(5,50)_{/1,0}$	$2_{/2,0}$			500
LeNets++	$(5,32)_{/1,2} \times 2$	$2_{/2,0}$	$(5,64)_{/1,2} \times 2$	$2_{/2,0}$	$(5,128)_{/1,2} \times 2$	$2_{/2,0}$	**2**

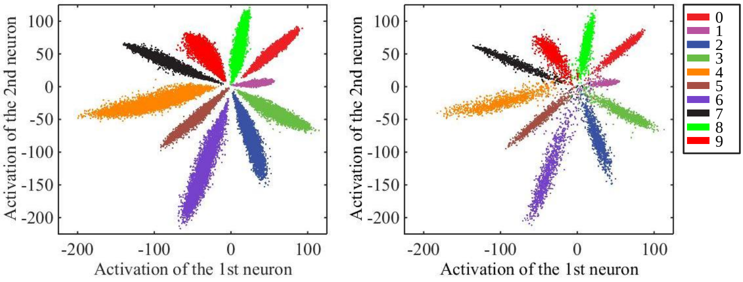
3.1 A toy example

In this section, a toy example on MNIST [33] dataset is presented. We modify the LeNets [18] to a deeper and wider network, but reduce the output number of the last hidden layer to 2 (It means that the dimension of the deep features is 2). More details of the network architecture are given in Table 1. The softmax loss function is presented as follows.

$$\mathcal{L}_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}} \tag{1}$$

In Equation 1, the $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes the ith deep feature, belonging to the $\boldsymbol{y}_i$th class. d is the feature dimension. $W_j \in \mathbb{R}^d$ denotes the jth column of the weights $W \in \mathbb{R}^{d \times n}$ in the last fully connected layer and $\boldsymbol{b} \in \mathbb{R}^n$ is the bias term. The size of mini-batch and the number of class is m and n, respectively.

The resulting 2-D deep features are plotted in Figure 2 to illustrate the distribution. Since the last fully connected layer acts like a linear classifier, the deep features of different classes are distinguished by decision boundaries. From Figure 2 we can observe that: i) under the supervision of softmax loss, the deeply learned features are separable, and ii) the deep features are not discriminative enough, since they still show significant intra-class variations. Consequently, it is not suitable to directly use these features for recognition.



(a) (b)

Fig. 2. The distribution of deeply learned features in (a) training set (b) testing set, both under the supervision of softmax loss, where we use 50K/10K train/test splits. The points with different colors denote features from different classes. **Best viewed in color.**

3.2 The center loss

So, how to develop an effective loss function to improve the discriminative power of the deeply learned features? Intuitively, minimizing the intra-class variations while keeping the features of different classes separable is the key. To this end, we propose the center loss function, as formulated in Equation 2.

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2 \tag{2}$$

The $\boldsymbol{c}_{y_i} \in \mathbb{R}^d$ denotes the y_ith class center of deep features. The formulation effectively characterizes the intra-class variations. Ideally, the $\boldsymbol{c}_{y_i}$ should be updated as the deep features changed. In other words, we need to take the entire training set into account and average the features of every class in each iteration, which is inefficient even impractical. Therefore, the center loss can not be used directly. This is possibly the reason that such a center loss has never been used in CNNs until now.

To address this problem, we make two necessary modifications. First, instead of updating the centers with respect to the entire training set, we perform the update base on mini-batch. In each iteration, the centers are computed by averaging the features of the corresponding classes (In this case, some of the centers may not update). Second, to avoid large perturbations caused by few mislabelled samples, we use a scalar α to control the learning rate of the centers. The gradients of $\mathcal{L}_C$ with respect to $\boldsymbol{x}_i$ and update equation of $\boldsymbol{c}_{y_i}$ are computed as:

$$\frac{\partial \mathcal{L}_C}{\partial \boldsymbol{x}_i} = \boldsymbol{x}_i - \boldsymbol{c}_{y_i} \tag{3}$$

$$\varDelta \boldsymbol{c}_j = \frac{\sum_{i=1}^{m} \delta(y_i = j) \cdot (\boldsymbol{c}_j - \boldsymbol{x}_i)}{\sum_{i=1}^{m} \delta(y_i = j) + 1} \tag{4}$$

where $\delta(condition) = 1$ if the *condition* is satisfied, and $\delta(condition) = 0$ if not. α is restricted in $[0, 1]$. We adopt the joint supervision of softmax loss and center loss to train the CNNs for discriminative feature learning. The formulation is given in Equation 5.

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_S + \lambda\mathcal{L}_C \\ &= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2\end{aligned} \tag{5}$$

Clearly, the CNNs supervised by center loss are trainable and can be optimized by standard SGD. A scalar λ is used for balancing the two loss functions. The conventional softmax loss can be considered as a special case of this joint supervision, if λ is set to 0. In Algorithm 1, we summarize the learning details in the CNNs with joint supervision.

Algorithm 1 The discriminative feature learning algorithm

```
Input: Training data $\{\boldsymbol{x}_i\}$. Initialized parameters $\theta_C$ in convolution layers. Parameters $W$ and $\{\boldsymbol{c}_j | j = 1, 2, ..., n\}$ in loss layers, respectively. Hyperparameter $\lambda$, $\alpha$ and learning rate $\mu^t$. The number of iteration $t \leftarrow 0$.
Output: The parameters $\theta_C$.
while not converge do
    $t \leftarrow t + 1$.
    Compute the joint loss by $\mathcal{L}^t = \mathcal{L}_S^t + \mathcal{L}_C^t$.
    Compute the backpropagation error $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t}$ for each $i$ by $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} = \frac{\partial \mathcal{L}_S^t}{\partial \boldsymbol{x}_i^t} + \lambda \cdot \frac{\partial \mathcal{L}_C^t}{\partial \boldsymbol{x}_i^t}$.
    Update the parameters $W$ by $W^{t+1} = W^t - \mu^t \cdot \frac{\partial \mathcal{L}^t}{\partial W^t} = W^t - \mu^t \cdot \frac{\partial \mathcal{L}_S^t}{\partial W^t}$.
    Update the parameters $\boldsymbol{c}_j$ for each $j$ by $\boldsymbol{c}_j^{t+1} = \boldsymbol{c}_j^t - \alpha \cdot \Delta\boldsymbol{c}_j^t$.
    Update the parameters $\theta_C$ by $\theta_C^{t+1} = \theta_C^t - \mu^t \sum_i^m \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} \cdot \frac{\partial \boldsymbol{x}_i^t}{\partial \theta_C^t}$.
end while
```

We also conduct experiments to illustrate how the λ influences the distribution. Figure 3 shows that different λ lead to different deep feature distributions. With proper λ, the discriminative power of deep features can be significantly enhanced. Moreover, features are discriminative within a wide range of λ. Therefore, the joint supervision benefits the discriminative power of deeply learned features, which is crucial for face recognition.

3.3 Discussion

- **The necessity of joint supervision.** If we only use the softmax loss as supervision signal, the resulting deeply learned features would contain large intra-class variations. On the other hand, if we only supervise CNNs by the center loss, the deeply learned features and centers will degraded to zeros (At this point, the center loss is very small). Simply using either of them could not achieve discriminative feature learning. So it is necessary to combine them to jointly supervise the CNNs, as confirmed by our experiments.

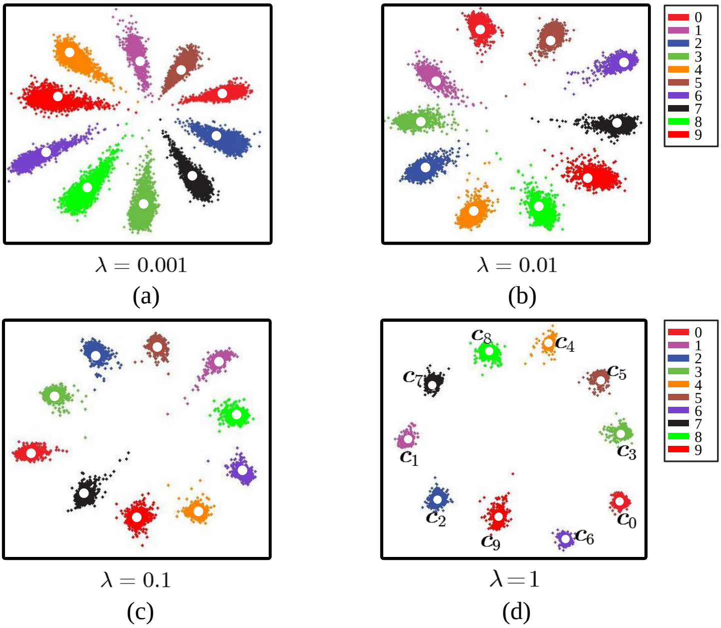


Fig. 3. The distribution of deeply learned features under the joint supervision of softmax loss and center loss. The points with different colors denote features from different classes. Different λ lead to different deep feature distributions ($\alpha = 0.5$). The white dots ($\boldsymbol{c}_0$, $\boldsymbol{c}_1$,...,$\boldsymbol{c}_9$) denote 10 class centers of deep features. **Best viewed in color.**

- **Compared to contrastive loss and triplet loss.** Recently, contrastive loss [19, 20] and triplet loss [21] are also proposed to enhance the discriminative power of the deeply learned face features. However, both contrastive loss and triplet loss suffer from dramatic data expansion when constituting the sample pairs or sample triplets from the training set. Our center loss enjoys the same requirement as the softmax loss and needs no complex recombination of the training samples. Consequently, the supervised learning of our CNNs is more efficient and easy-to-implement. Moreover, our loss function targets more directly on the learning objective of the intra-class compactness, which is very beneficial to the discriminative feature learning.

4 Experiments

The necessary implementation details are given in Section 4.1. Then we investigate the sensitiveness of the parameter λ and α in Section 4.2. In Section 4.3 and 4.4, extensive experiments are conducted on several public domain face datasets (LFW [22], YTF [23] and MegaFace Challenge [1]) to verify the effectiveness of the proposed approach.

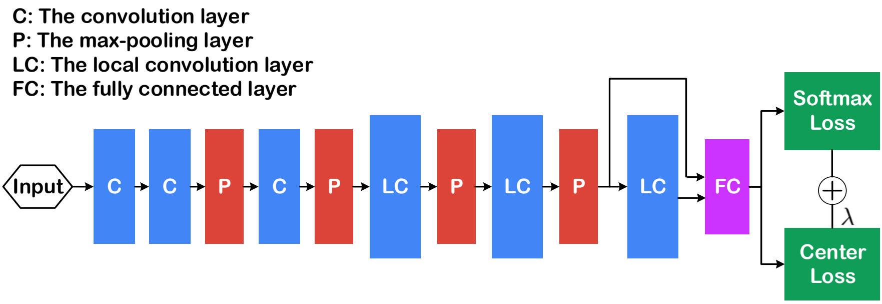


Fig. 4. The CNN architecture using for face recognition experiments. Joint supervision is adopted. The kernels in both convolution and local convolution layers are 3×3 with stride 1, followed by PReLU [5] nonlinear units. Weights in three local convolution layers are locally shared in the regions of 4×4, 2×2 and 1×1 respectively. The number of the feature maps are 128 for the convolution layers and 256 for the local convolution layers. The max-pooling grid is 2×2 and the stride is 2. The output of the 4th pooling layer and the 3th local convolution layer are concatenated as the input of the 1st fully connected layer. The output dimension of the fully connected layer is 512. **Best viewed in color.**

4.1 Implementation Details

Preprocessing. All the faces in images and their landmarks are detected by the recently proposed algorithms [34–36]. We use 5 landmarks (two eyes, nose and mouth corners) for similarity transformation. When the detection fails, we simply discard the image if it is in training set, but use the provided landmarks if it is a testing image. The faces are cropped to 112×96 RGB images. Following a previous convention, each pixel (in $[0, 255]$) in RGB images is normalized by subtracting 127.5 then dividing by 128.

Training data. We use the web-collected training data, including CASIA-WebFace [37], CACD2000 [38], Celebrity+ [39]. After removing the images with identities appearing in testing datasets, it roughly goes to 0.7M images of 17,189 unique persons. In Section 4.4, we only use 0.49M training data, following the protocol of small training set. The images are horizontally flipped for data augmentation. Compared to [21] (200M), [28] (4M) and [31] (2M), it is a small scale training set.

Detailed settings in CNNs. We implement the CNN model using the Caffe [40] library with our modifications. All the CNN models in this Section are the same architecture and the details are given in Figure 4. For fair comparison, we respectively train three kind of models under the supervision of softmax loss (**model A**), softmax loss and contrastive loss (**model B**), softmax loss and center loss (**model C**). These models are trained with batch size of 256 on two GPUs (TitanX). For model A and model C, the learning rate is started from 0.1, and divided by 10 at the 16K, 24K iterations. A complete training is finished at 28K iterations and roughly costs 14 hours. For model B, we find that it converges slower. As a result, we initialize the learning rate to 0.1 and switch it at the 24K, 36K iterations. Total iteration is 42K and costs 22 hours.

Detailed settings in testing. The deep features are taken from the output of the FC layer. We extract the features for each image and its horizontally flipped one, and concatenate them as the representation. The score is computed by the Euclidean Distance of two features after PCA. Nearest neighbor [16] and threshold comparison are respectively used for identification and verification tasks. Note that, we only use single model for all the testing.

4.2 Experiments on the parameter λ and α

The hyper parameter λ dominates the intra-class variations and α controls the learning rate of center c in model C. Both of them are essential to our model. So we conduct two experiments to investigate the sensitiveness of the two parameters.

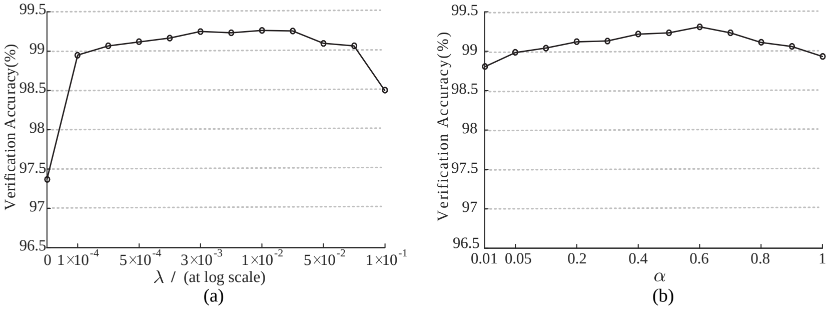
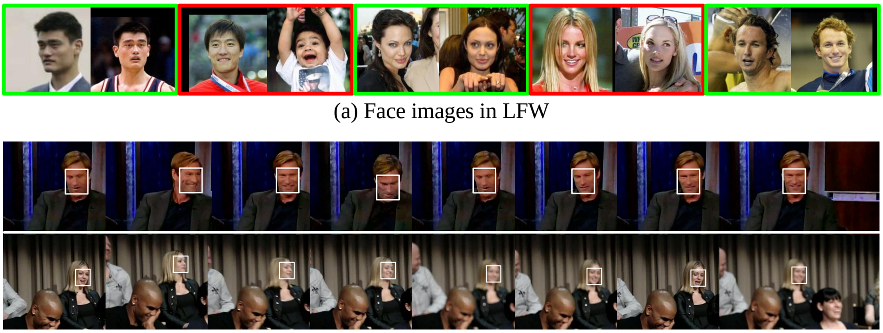


Fig. 5. Face verification accuracies on LFW dataset, respectively achieve by (a) models with different λ and fixed $\alpha = 0.5$. (b) models with different α and fixed $\lambda = 0.003$.

In the first experiment, we fix α to 0.5 and vary λ from 0 to 0.1 to learn different models. The verification accuracies of these models on LFW dataset are shown in Figure 5. It is very clear that simply using the softmax loss (in this case λ is 0) is not a good choice, leading to poor verification performance. Properly choosing the value of λ can improve the verification accuracy of the deeply learned features. We also observe that the verification performance of our model remains largely stable across a wide range of λ. In the second experiment, we fix $\lambda = 0.003$ and vary α from 0.01 to 1 to learn different models. The verification accuracies of these models on LFW are illustrated in Figure 5. Likewise, the verification performance of our model remains largely stable across a wide range of α.

4.3 Experiments on the LFW and YTF datasets

In this part, we evaluate our single model on two famous face recognition benchmarks in unconstrained environments, LFW and YTF datasets. They are ex-



(a) Face images in LFW

(b) Face videos in YTF

Fig. 6. Some face images and videos in LFW and YTF datasets. The face image pairs in green frames are the positive pairs (the same person), while the ones in red frames are negative pairs. The white bounding box in each image indicates the face for testing.

cellent benchmarks for face recognition in image and video. Some examples of them are illustrated in Figure 6. Our model is trained on the 0.7M outside data, with no people overlapping with LFW and YTF. In this section, we fix the λ to 0.003 and the α is 0.5 for model C.

LFW dataset contains 13,233 web-collected images from 5749 different identities, with large variations in pose, expression and illuminations. Following the standard protocol of *unrestricted with labeled outside data* [14]. We test on 6,000 face pairs and report the experiment results in Table 2.

YTF dataset consists of 3,425 videos of 1,595 different people, with an average of 2.15 videos per person. The clip durations vary from 48 frames to 6,070 frames, with an average length of 181.3 frames. Again, we follow the *unrestricted with labeled outside data* protocol and report the results on 5,000 video pairs in Table 2.

Table 2. Verification performance of different methods on LFW and YTF datasets

Method	Images	Networks	Acc. on LFW	Acc. on YTF
DeepFace [28]	4M	3	97.35%	91.4%
DeepID-2+ [30]	-	1	98.70%	-
DeepID-2+ [30]	-	25	99.47%	93.2%
FaceNet [21]	200M	1	99.63%	95.1%
Deep FR [31]	2.6M	1	98.95%	97.3%
Baidu [32]	1.3M	1	99.13%	-
model A	0.7M	**1**	97.37%	91.1%
model B	0.7M	**1**	99.10%	93.8%
model C (Proposed)	**0.7M**	**1**	**99.28%**	**94.9%**

From the results in Table 2, we have the following observations. First, model C (jointly supervised by the softmax loss and the center loss) beats the baseline one (model A, supervised by the softmax loss only) by a significant margin, improving the performance from (97.37% on LFW and 91.1% on YTF) to (99.28% on LFW and 94.9% on YTF). This shows that the joint supervision can notably enhance the discriminative power of deeply learned features, demonstrating the effectiveness of the center loss. Second, compared to model B (supervised by the combination of the softmax loss and the contrastive loss), model C achieves better performance (99.10% *v.s.* 99.28% and 93.8% *v.s.* 94.9%). This shows the advantage of the center loss over the contrastive loss in the designed CNNs. Last, compared to the state-of-the-art results on the two databases, the results of the proposed model C (much less training data and simpler network architecture) are consistently among the top-ranked sets of approaches based on the two databases, outperforming most of the existing results in Table 2. This shows the advantage of the proposed CNNs.

4.4 Experiments on the dataset of MegaFace Challenge

MegaFace datasets are recently released as a testing benchmark. It is a very challenging dataset and aims to evaluate the performance of face recognition algorithms at the **million scale of *distractors*** (people who are not in the testing set). MegaFace datasets include gallery set and probe set. The gallery set consists of more than 1 million images from 690K different individuals, as a subset of Flickr photos [41] from Yahoo. The probe set using in this challenge are two existing databases: Facescrub [42] and FGNet [43]. Facescrub dataset is publicly available dataset, containing 100K photos of 530 unique individuals (55,742 images of males and 52,076 images of females). The possible bias can be reduced by sufficient samples in each identity. FGNet dataset is a face aging dataset, with 1002 images from 82 identities. Each identity has multiple face images at different ages (ranging from 0 to 69).

There are several testing scenarios (identification, verification and pose invariance) under two protocols (large or small training set). The training set is defined as *small* if it contains less than 0.5M images and 20K subjects. Following the protocol of small training set, we reduce the size of training images to 0.49M but maintaining the number of identities unchanged (i.e. 17,189 subjects). The images overlapping with Facescrub dataset are discarded. For fair comparison, we also train three kinds of CNN models on small training set under different supervision signals. The resulting models are called model A-, model B- and model C-, respectively. Following the same settings in Section 4.3, the λ is 0.003 and the α is 0.5 in model C-. We conduct the experiments with the provided code [1], which only tests our algorithm on one of the three gallery (Set 1).

Face Identification. Face identification aims to match a given probe image to the ones with the same person in gallery. In this task, we need to compute the similarity between each given probe face image and the gallery, which includes at least one image with the same identity as the probe one. Besides, the gallery contains different scale of *distractors*, from 10 to 1 million, leading to increasing challenge in testing. More details can be found in [1]. In face identification experiments, we present the results by Cumulative Match Characteristics (CMC)

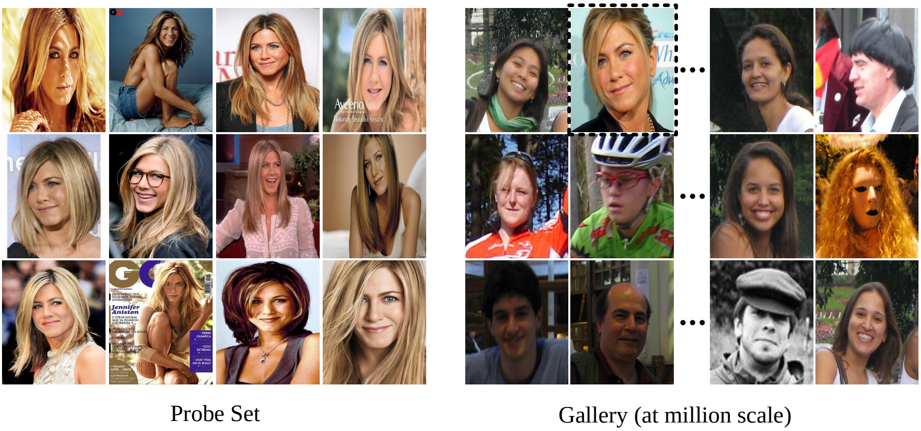


Fig. 7. Some example face images in MegaFace dataset, including probe set and gallery. The gallery consists of at least one correct image and millions of distractors. Because of the great intra-variations in each subject and varieties of distractors, the identification and verification task become very challenging.

curves. It reveals the probability that a correct gallery image is ranked on top-K. The results are shown in Figure 8.

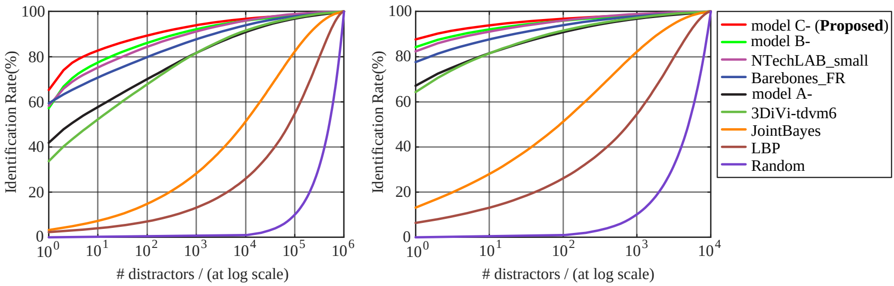


Fig. 8. CMC curves of different methods (under the protocol of small training set) with (a) 1M and (b) 10K distractors on Set 1. The results of other methods are provided by MegaFace team.

Face Verification. For face verification, the algorithm should decide a given pair of images is the same person or not. 4 billion negative pairs between the probe and gallery datasets are produced. We compute the True Accept Rate (TAR) and False Accept Rate (FAR) and plot the Receiver Operating Characteristic (ROC) curves of different methods in Figure 9.

We compare our method against many existing ones, including i) LBP [44] and JointBayes [45], ii) our baseline deep models (model A- and model B-), and iii) deep models submitted by other groups. As can be seen from Figure 8

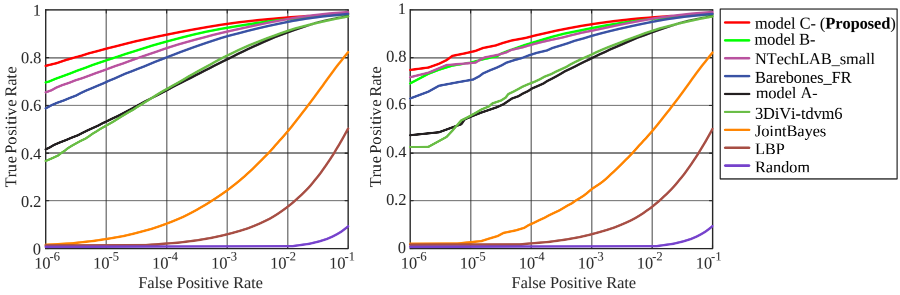


Fig. 9. ROC curves of different methods (under the protocol of small training set) with (a) 1M and (b) 10K distractors on Set 1. The results of other methods are provided by MegaFace team.

and Figure 9, the hand-craft features and shallow model perform poorly. Their accuracies drop sharply with the increasing number of distractors. In addition, the methods based on deep learning perform better than the traditional ones. However, there is still much room for performance improvement. Finally, with the joint supervision of softmax loss and center loss, model C- achieves the best results, not only surpassing the model A- and model B- by a clear margin but also significantly outperforming the other published methods.

To meet the practical demand, face recognition models should achieve high performance against millions of distractors. In this case, only Rank-1 identification rate with at least 1M distractors and verification rate at low false accept rate (e.g., 10^{-6}) are very meaningful [1]. We report the experimental results of different methods in Table 3 and 4.

Table 3. Identification rates of different methods on MegaFace dataset with 1M distractors.

Method	protocol	Identification Acc. (Set 1)
NTechLAB - facenx_large	large	73.300%
Google - FaceNet v8	large	70.496%
Beijing Faceall Co. - FaceAll_Norm_1600	large	64.803%
Beijing Faceall Co. - FaceAll_1600	large	63.977%
Barebones_FR - cnn	small	59.363%
NTechLAB - facenx_small	small	58.218%
3DiVi Company - tdvm6	small	33.705%
model A-	small	41.863%
model B-	small	57.175%
model C- (Proposed)	**small**	**65.234%**

From these results we have the following observations. First, not surprisingly, model C- consistently outperforms model A- and model B- by a significant mar-

Table 4. Verification TAR of different methods at 10^{-6} FAR on MegaFace dataset with 1M distractors.

Method	protocol	Verification Acc. (Set 1)
Google - FaceNet v8	large	86.473%
NTechLAB - facenx_large	large	85.081%
Beijing Faceall Co. - FaceAll_Norm_1600	large	67.118%
Beijing Faceall Co. - FaceAll_1600	large	63.960%
Barebones_FR - cnn	small	59.036%
NTechLAB - facenx_small	small	66.366%
3DiVi Company - tdvm6	small	36.927%
model A-	small	41.297%
model B-	small	69.987%
model C- (Proposed)	**small**	**76.516%**

gin in both face identification and verification tasks, confirming the advantage of the designed loss function. Second, under the evaluation protocol of small training set, the proposed model C- achieves the best results in both face identification and verification tasks, outperforming the $2nd$ place by **5.97%** on face identification and **10.15%** on face verification, respectively. Moreover, it is worth to note that model C- even surpasses some models trained with large training set (e.g., Beijing Facecall Co.). Last, the models from Google and NTechLAB achieve the best performance under the protocol of large training set. Note that, their private training set (500M for Google and 18M for NTechLAB) are much larger than ours (0.49M).

5 Conclusions

In this paper, we have proposed a new loss function, referred to as center loss. By combining the center loss with the softmax loss to jointly supervise the learning of CNNs, the discriminative power of the deeply learned features can be highly enhanced for robust face recognition. Extensive experiments on several large-scale face benchmarks have convincingly demonstrated the effectiveness of the proposed approach.

References

1. Miller, D., Kemelmacher-Shlizerman, I., Seitz, S.M.: Megaface: A million faces for recognition at scale. arXiv preprint arXiv:1505.02108 (2015)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

4. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9
5. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1026–1034
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
7. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in neural information processing systems. (2014) 487–495
8. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. arXiv preprint arXiv:1412.6856 (2014)
9. Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Sequential deep learning for human action recognition. In: Human Behavior Understanding. Springer (2011) 29–39
10. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4305–4314
11. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **35**(1) (2013) 221–231
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 248–255
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3) (2015) 211–252
14. Huang, G.B., Learned-Miller, E.: Labeled faces in the wild: Updates and new reporting procedures. Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep (2014) 14–003
15. Learned-Miller, E., Huang, G., RoyChowdhury, A., Li, H., Hua, G., Huang, G.B.: Labeled faces in the wild: A survey
16. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. Information Theory, IEEE Transactions on **13**(1) (1967) 21–27
17. Fukunaga, K., Narendra, P.M.: A branch and bound algorithm for computing k-nearest neighbors. Computers, IEEE Transactions on **100**(7) (1975) 750–753
18. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
19. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Computer vision and pattern recognition, 2006 IEEE computer society conference on. Volume 2., IEEE (2006) 1735–1742
20. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Advances in Neural Information Processing Systems. (2014) 1988–1996
21. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 815–823
22. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst (2007)

23. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matched background similarity. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE (2011) 529–534
24. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Volume 1., IEEE (2005) 539–546
25. Sun, Y., Wang, X., Tang, X.: Hybrid deep learning for face verification. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1489–1496
26. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted boltzmann machines for collaborative filtering. In: Proceedings of the 24th international conference on Machine learning, ACM (2007) 791–798
27. Hu, J., Lu, J., Tan, Y.P.: Discriminative deep metric learning for face verification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1875–1882
28. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1701–1708
29. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1891–1898
30. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2892–2900
31. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. Proceedings of the British Machine Vision **1**(3) (2015) 6
32. Liu, J., Deng, Y., Huang, C.: Targeting ultimate accuracy: Face recognition via deep embedding. arXiv preprint arXiv:1506.07310 (2015)
33. LeCun, Y., Cortes, C., Burges, C.J.: The mnist database of handwritten digits (1998)
34. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Robust discriminative response map fitting with constrained local models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3444–3451
35. Asthana, A., Zafeiriou, S., Cheng, S., Pantic, M.: Incremental face alignment in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1859–1866
36. Zhang, Z., Luo, P., Loy, C.C., Tang, X.: Learning deep representation for face alignment with auxiliary attributes. (2015)
37. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Learning face representation from scratch. arXiv preprint arXiv:1411.7923 (2014)
38. Chen, B.C., Chen, C.S., Hsu, W.H.: Face recognition and retrieval using cross-age reference coding with cross-age celebrity dataset. Multimedia, IEEE Transactions on **17**(6) (2015) 804–815
39. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 3730–3738
40. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, ACM (2014) 675–678

41. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: The new data and new challenges in multimedia research. arXiv preprint arXiv:1503.01817 (2015)
42. Ng, H.W., Winkler, S.: A data-driven approach to cleaning large face datasets. In: Image Processing (ICIP), 2014 IEEE International Conference on, IEEE (2014) 343–347
43. : Fg-net aging database. In: http://www.fgnet.rsunit.com/. (2010)
44. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on **28**(12) (2006) 2037–2041
45. Chen, D., Cao, X., Wang, L., Wen, F., Sun, J.: Bayesian face revisited: A joint formulation. In: Computer Vision–ECCV 2012. Springer (2012) 566–579