International Journal of Computer Vision (2019) 127:668–683
https://doi.org/10.1007/s11263-018-01142-4

A Comprehensive Study on Center Loss for Deep Face Recognition

Yandong Wen[1] · Kaipeng Zhang[1] · Zhifeng Li[2] · Yu Qiao[3]

Received: 23 February 2018 / Accepted: 15 December 2018 / Published online: 17 January 2019

Abstract

Deep convolutional neural networks (CNNs) trained with the softmax loss have achieved remarkable successes in a number of close-set recognition problems, e.g. object recognition, action recognition, etc. Unlike these close-set tasks, face recognition is an open-set problem where the testing classes (persons) are usually different from those in training. This paper addresses the open-set property of face recognition by developing the *center loss*. Specifically, the center loss simultaneously learns a center for each class, and penalizes the distances between the deep features of the face images and their corresponding class centers. Training with the center loss enables CNNs to extract the deep features with two desirable properties: inter-class separability and intra-class compactness. In addition, we extend the center loss in two aspects. First, we adopt parameter sharing between the softmax loss and the center loss, to reduce the extra parameters introduced by centers. Second, we generalize the concept of center from a single point to a region in embedding space, which further allows us to account for intra-class variations. The advanced center loss significantly enhances the discriminative power of deep features. Experimental results show that our method achieves high accuracies on several important face recognition benchmarks, including Labeled Faces in the Wild, YouTube Faces, IJB-A Janus, and MegaFace Challenging 1.

Keywords Convolutional neural networks · Face recognition · Discriminative feature learning · Center loss

1 Introduction

Convolutional neural networks (CNNs) together with the softmax loss have achieved significant successes in computer vision, substantially improving the state of the art in a series of close-set recognition tasks, such as object (Krizhevsky et al. 2012; He et al. 2015a), scene (Zhou et al. 2014a, b), action (Baccouche et al. 2011; Wang et al. 2015b) and so on. In these tasks, the possible categories of the testing samples are predefined in the training set and the predicted labels determine the performance. As a result, the softmax loss is widely adopted by many CNNs (Krizhevsky et al. 2012; Simonyan and Zisserman 2014; Szegedy et al. 2015; He et al. 2015a) due to its simplicity, good performance, and probabilistic interpretation. As shown in Fig. 1a, the commonly used CNNs perform feature extraction and label prediction, mapping the input data to deep features, then to the predicted labels.

However, the conventional softmax loss may not be particularly effective for face recognition, which by nature is an open-set problem. Despite its excellent performance on close-set recognition, the softmax loss does not explicitly encourage the intra-class compactness of the features (Liu et al. 2016). The resulting features are separable but not

Communicated by Xiaoou Tang.

Yandong Wen and Kaipeng Zhang have contributed equally to this work.

In this paper, we define the softmax loss as the combination of a fully-connected layer, a softmax function, and a cross-entropy loss.

✉ Yu Qiao
yu.qiao@siat.ac.cn

Yandong Wen
yandongw@andrew.cmu.edu

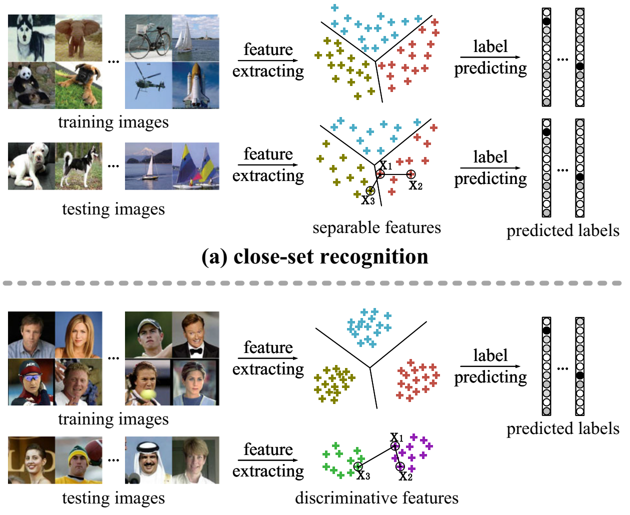
Kaipeng Zhang
kpzhang@cmlab.csie.ntu.edu.tw

Zhifeng Li
michaelzi@tencent.com

[1] Shenzhen Key Lab on Computer Vision and Pattern Recognition, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

[2] Tencent AI Lab, Shenzhen, China

[3] SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China



(a) close-set recognition

(b) open-set recognition

Fig. 1 Comparison between close set and open set recognitions. **a** The CNN frameworks for close-set problems. The decision boundaries are available in both the training and testing. **b** The CNN frameworks for open-set problems. The decision boundary is not available in the testing because of the new unseen classes

discriminative, leading to inferior performance for open-set task, like face recognition. The discriminative power here characterizes features in both the inter-class separability and the intra-class compactness.

Considering a real-world scenario, it is impractical to pre-collect all the possible testing identities in the training phase. The label prediction in CNNs is not always applicable, especially when identifying a new unseen identity in testing. Such evidence indicates that our recognition results can only be derived from the deeply learned features, e.g. the distance (or similarity) between two features. From Fig. 1a we can easily see that separable features do NOT guarantee the correctness on open-set recognition. Given x_1, x_2 belonging to class 1, and x_3 belonging to class 2, the intra-class distance $\mathrm{d}(x_1, x_2)$ can be larger than inter-class distance $\mathrm{d}(x_1, x_3)$, leading to failed recognition. On the other hand, discriminative (not just separable) features (Fig. 1b) perform much better, since they have both the intra-class compactness and inter-class separability. In particular, if *the maximal intra-class distance is smaller than the minimal inter-class distance*, we can correctly perform open-set face recognition for pairs of samples.

In practice, it is challenging to develop an appropriate loss function for discriminative feature learning in CNNs. Due to the huge scale of training set for CNNs, it is extremely time-consuming to extract the deep features of all training samples in each iteration. Traditional loss functions like linear discriminant analysis (LDA) (Mika et al. 1999) need to take the entire training set into account. As a result, they cannot be simply applied to CNNs, which usually work on samples of a mini-batch. As alternative approaches, contrastive loss (Hadsell et al. 2006; Sun et al. 2014a) and triplet loss (Schroff et al. 2015) design intuitive learning objectives based on image pairs and triplet, respectively. However, compared to the image samples, the number of the possible 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 makes training process inconvenient.

In this paper, we propose a novel loss function, called *center loss*. The center loss learns a center (a vector with the same dimension as a feature) for deep features of each class. Although the global distribution of deep features are not available, it could be approximately characterized by the learned centers, since center/mean provides the first order statistics of the distribution. In the course of training, we simultaneously update the centers 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 the center loss, with a hyper parameter to balance the two supervision signals. Intuitively, the softmax loss forces the deep features of different identities 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. The discriminative power of the deeply learned features are significantly enhanced. Our main contributions are summarized below.

- We propose a novel loss function, called *center loss* to minimize the intra-class distances of the deep features. By adding a branch of center loss in parallel with the existing softmax loss, the CNNs yield highly discriminative features for robust face recognition, as supported by our experimental results.
- We show that the center loss is trainable and can be optimized by stochastic gradient descent (SGD). More importantly, the center loss enjoys the same requirement as the softmax loss, and needs no complex samples mining in the training.
- We present extensive experiments on Labeled Faces in the Wild (LFW), YouTube Faces (YTF), IJB-A Janus, and MegaFace Challenging 1. Significant improvements demonstrate the effectiveness of the proposed center loss.

A preliminary version of this manuscript has been published in ECCV 2016 (Wen et al. 2016b). Since then, the center loss has witnessed other applications beyond face recognition, such as image retrieval (Yao et al. 2017), person re-identification (Jin et al. 2017), document semantic structure extraction (Yang et al. 2017), autoencoder (Chu and Cai 2017), and speaker recognition (Bredin 2017; Wisniewksi et al. 2017). In spite of its usefulness, the center loss still has some limitations: (1) The center loss introduces extra parameters in training CNNs, increasing the model size. (2) The center loss makes a strong assumption that the deep features of an identity should be as closed as possible to their class center, which is a single point in embedding space. The following describes how we address the limitations from the conference version.

- We adopt parameter sharing between the softmax loss and the center loss, greatly reducing the network parameters and leading to better performance. It is particularly useful for face datasets with more than thousands of identities. More importantly, the recognition performance can be further improved with this technique.
- We show that the concept of center is not limited to a single point in embedding space. Instead, it can be generalized to a region. We present an Euclidean distance based region to make the center loss more flexible and general.
- We present comprehensive ablation study for the center loss, and new experiments on large-scale benchmarks - IJB-A Janus (Klare et al. 2015). The state-of-the-art results on these challenging benchmarks show clear advantages of the proposed methods. The code and models are publicly available on https://github.com/ydwen/centerloss.

2 Related Work

Loss functions for close-set recognition Softmax loss is widely used in close-set recognition tasks, like object (Krizhevsky et al. 2012), scene (Zhou et al. 2014a), action (Wang et al. 2015b), etc. Alternatively, Vinyals et al. (2012), Nagi et al. (2012), Tang (2013) applied hinge loss in CNNs, which provided more geometric interpretation. Magnet loss (Rippel et al. 2015) is proposed to explicitly address the deep metric learning with the concept of *center*. However, it is not as scalable as center loss because it defines multiple centers for each class. To train the CNNs, the samples in minibatches have to be carefully selected and the K-means algorithm is adopted to update the centers.

Traditional face recognition Early works focused on subspace-based face recognition. Belhumeur et al. (1997), Wang and Tang (2004), Prince and Elder (2007) decomposed the face representations into several components to facilitate the subsequent recognition. Sparse representation based classification (Wright et al. 2009) and its variants (Zhang et al. 2011) encode faces over the subspace of template images. The recognition is performed based on the reconstruction error of each class. To improve the face representations, several papers investigated SIFT (Lowe 2004), HoG (Dalal and Triggs 2005), LBP (Ahonen et al. 2006) and its variants (Lu et al. 2015; Duan et al. 2017), and learned features (Cao et al. 2010). To obtain compact features, many efforts (Chen et al. 2012, 2013; Simonyan et al. 2013) focused on discriminative dimension reduction. These approaches further boost the accuracy and improve the efficiency. Our center loss can be related to these works in sense of discriminative feature learning, but with more focus on deep CNN framework.

Deep face recognition Face recognition via deep learning has witnessed a series of breakthrough in recent years (Sun et al. 2013; Taigman et al. 2014; Sun et al. 2014a; Schroff et al. 2015; Parkhi et al. 2015; Wang et al. 2018a, b). The idea of learning a neural network that maps a pair of face images to a distance starts from Chopra et al. (2005). They train siamese networks for driving the similarity metric to be small for positive pairs, and large for the negative pairs. Hu et al. (2014) learn a nonlinear transformations and yield discriminative deep metric with a margin between positive and negative face image pairs.

DeepFace (Taigman et al. 2014; Sun et al. 2014b) supervised the learning process in CNNs by challenging identification signal, which can bring richer identity-related information to deeply learned features. After that, joint identification-verification supervision signal was adopted in Sun et al. (2014a); Wen et al. (2016a), leading to more discriminative features. Sun et al. (2015) enhanced the supervision by adding a fully-connected layer and loss functions to each convolutional layer. The effectiveness of triplet loss had been demonstrated in Schroff et al. (2015); Parkhi et al. (2015); Liu et al. (2015). With the deep embedding, the distance between an anchor and a positive is minimized, while the distance between an anchor and a negative is maximized until the margin is met. Song et al. (2016), Sohn (2016), Tadmor et al. (2016) explored the sample combination in the minibatch, leading to faster convergence and better performance.

Very recently, Liu et al. (2017b), Wang et al. (2017), Ranjan et al. (2017) propose to directly optimize the cosine similarity between deep features. The weight parameters in the softmax loss are normalized, and the deep features are normalized then rescaled. This simple idea achieves better performance than the softmax loss. Liu et al. (2017a) imposes discriminative constraints on a hypersphere manifold to learn discriminative deep features with angular margin. It is worth mentioning that the aforementioned works aim to learn discriminative feature in angular perspective, while center loss pays more attention to the Euclidean space. These two branches of approaches can be complementary with each other.

3 The Proposed Approach

We elaborate our approach in this section. A toy example is presented to intuitively visualize and analyze the distribu-

Table 1 The detailed network configurations of LeNets and LeNets++

Layer	LeNet	LeNet++
Conv1	$(5, 20)_{/1,0}$	$(5, 32)_{/1,2} \times 2$
Pool1	$2_{/2,0}$	$2_{/2,0}$
Conv2	$(5, 50)_{/1,0}$	$(5, 64)_{/1,2} \times 2$
Pool2	$2_{/2,0}$	$2_{/2,0}$
Conv3	–	$(5, 128)_{/1,2} \times 2$
Pool3	–	$2_{/2,0}$
FC1	500	2
Loss	10-way softmax loss	10-way softmax loss

$(5, 32)_{/1,2} \times 2$ denotes 2 cascaded convolutional layers with 32 filters of size 5×5, where the stride and padding are 1 and 2, respectively. $2_{/2,0}$ denotes the max-pooling layers with the grid of 2×2, where the stride and padding are 2 and 0 respectively. We use the Parametric Rectified Linear Unit (PReLU) (He et al. 2015b) as nonlinear unit

tions of the CNN features learned by softmax loss. Inspired by the distribution, we develop the center loss to improve the discriminative power of the deeply learned features. Additionally, we propose two improvements for the center loss, followed by discussions.

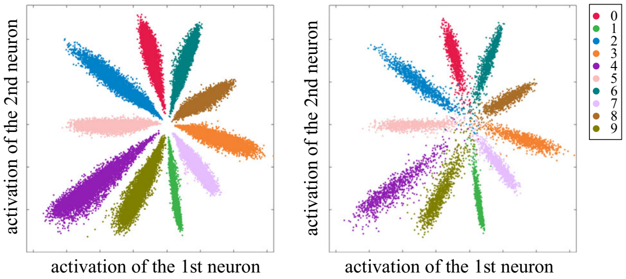
3.1 A Toy Example

A toy example on MNIST (LeCun et al. 1998b) dataset is presented. Note that the output number of the last hidden layer is reduced to 2 (means that the deep features are 2-dimentional vectors), which enables us to directly plot the features on 2-dimensional plane for visualization. To prevent the underfitting caused by low-dimensional features, we propose a deeper and wider variant of LeNets (LeCun et al. 1998a), called LeNets++ for this experiment. LeNets++ are essentially constructed by stacking convolutional layers, optionally followed by max-pooling layers. The details of the network architecture are given in Table 1. We train the LeNets++ on MNIST with the conventional softmax loss, as written in:

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

Here $\boldsymbol{x}_i \in \mathbb{R}^d$ is the deep feature of ith sample, belonging to the y_ith class. $\boldsymbol{w}_j \in \mathbb{R}^d$ is the jth column of the weights parameters $W = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n] \in \mathbb{R}^{d \times n}$ in the softmax loss and $\boldsymbol{b} \in \mathbb{R}^n$ is the bias term. m, n, and d are the number of samples in mini-batch, total number of classes, and feature dimension, respectively.

The resulting 2-D deep features are plotted in Fig. 2. This illustration provides several insights: (1) The softmax loss acts like a linear classifier for deep features, deriving linear decision boundaries between the features of different classes; (2) Under the supervision of softmax loss, the deeply learned features are prone to be separable; and (3) The deep features are not discriminative enough, since they still show significant intra-class variations. As a result, it is not suitable to directly use these features for face recognition, where the testing classes are different from those in training set.



(a) training examples **(b)** testing examples

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

3.2 Center Loss

The above analysis motivates us 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. The above toy example shows that the softmax loss contributes to enlarge the inter-class distance, exposing the considerable intra-class variation as a bottleneck. Inspired by the feature distribution in Fig. 2, we propose the *center loss*, as formulated in:

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

where $\boldsymbol{c}_{y_i} \in \mathbb{R}^d$ is the center for deep features $\boldsymbol{x}_i$ of y_ith class. This simple formulation effectively characterizes the intra-class variations by penalizing the distances between the deep features and their centers. However, optimizing Eq. 3.2 is non-trivial because both $\boldsymbol{x}_i$ and $\boldsymbol{c}_{y_i}$ are unknown. Specifically, $\boldsymbol{x}_i$ is determined by the learnable parameters in CNNs and $\boldsymbol{c}_{y_i}$ is given by the average of deep features belonging the y_ith class. Ideally, $\boldsymbol{c}_{y_i}$ should be updated as the deep features $\boldsymbol{x}_i$ changed in the training process. In other words, we need to take the entire training set into account and average the deep features of every class in each iteration, which is inefficient even impractical. Therefore, the center loss cannot be used directly.

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 based on the sam-

Algorithm 1 CNNs training algorithm with joint supervision

```
Input: Training data $\{x_i\}$. Initialized parameters $\theta_C$ in convolutional layers. Parameters $\{w_j | j = 1, 2, \ldots, n\}$ and $\{c_j | j = 1, 2, \ldots, n\}$ in loss layers, respectively. Hyper-parameter $\lambda$ 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 $\mathcal{L}^t = \mathcal{L}^t_S + \lambda \mathcal{L}^t_C$.
    Compute the backpropagation gradients $\frac{\partial \mathcal{L}^t}{\partial x^t_i}$ for each $i$ by $\frac{\partial \mathcal{L}^t}{\partial x^t_i} = \frac{\partial \mathcal{L}^t_S}{\partial x^t_i} + \lambda \cdot \frac{\partial \mathcal{L}^t_C}{\partial x^t_i}$.
    Update $w_j$ for each $j$ by $w^{t+1}_j = w^t_j - \mu^t \cdot \frac{\partial \mathcal{L}^t_S}{\partial w^t_j}$.
    Update $c_j$ for each $j$ by $c^{t+1}_j = c^t_j - \mu^t \cdot \Delta c^t_j$.
    Update the CNNs parameters $\theta_C$ by $\theta^{t+1}_C = \theta^t_C - \mu^t \sum^m_{i=1} \frac{\partial \mathcal{L}^t}{\partial x^t_i} \cdot \frac{\partial x^t_i}{\partial \theta^t_C}$.
end while
```

ples in mini-batch. In each iteration, the centers are updated by averaging the features of the same class in the mini-batch. This strategy works well in CNNs with SGD training, where only deep features of one mini-batch are exposed. Note that some of the centers may not be updated if the mini-batch does not include the features of corresponding classes. Second, to avoid large perturbations on centers caused by few mislabeled samples, we use moving average strategy to control the learning of the centers. The gradients of $\mathcal{L}_C$ with respect to $\boldsymbol{x}_i$ and $\boldsymbol{c}_{y_i}$ are given by:

$$\frac{\partial \mathcal{L}_C}{\partial \boldsymbol{x}_i} = \frac{1}{m}(\boldsymbol{x}_i - \boldsymbol{c}_{y_i}), \tag{3}$$

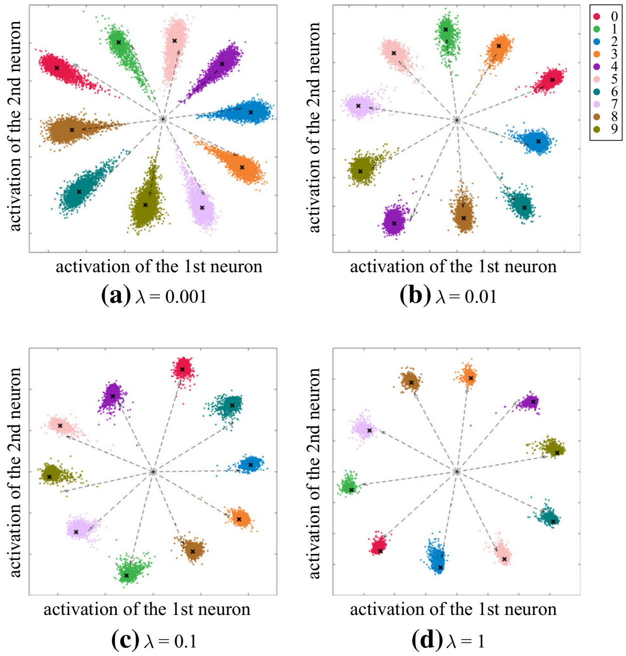
$$\Delta \boldsymbol{c}_j = \frac{\sum^m_{i=1} \delta(y_i = j) \cdot (\boldsymbol{c}_j - \boldsymbol{x}_i)}{\epsilon + \sum^m_{i=1} \delta(y_i = j)}, \tag{4}$$

where $\delta(condition) = 1$ if the *condition* is satisfied, and $\delta(condition) = 0$ if not. ϵ is a small positive number to avoid zero denominator, e.g. $\epsilon = 1e^{-5}$.

Note that the center loss cannot be used independently, otherwise the deeply learned features and centers will degraded to zeros (at this point, the center loss is very small). On the other hand, if we only use the softmax loss as supervision signal, the resulting deeply learned features contain large intra-class variations. Simply using either of them could not achieve discriminative feature learning. Hence it is necessary to combine them to jointly supervise the CNNs, as confirmed by our experiments. The formulation is given in:

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$
$$= -\frac{1}{m} \sum^m_{i=1} \log \frac{e^{\boldsymbol{w}^T_{y_i} \boldsymbol{x}_i + b_{y_i}}}{\sum^n_{j=1} e^{\boldsymbol{w}^T_j \boldsymbol{x}_i + b_j}} + \frac{\lambda}{2m} \sum^m_{i=1} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|^2_2, \tag{5}$$

where a scalar λ is used for balancing the two loss functions. The conventional softmax loss can be considered as a special case when λ is set to 0. Clearly, the CNNs under the joint supervision are trainable and can be optimized by standard SGD. We summarize the learning details in Algorithm 1.



(a) λ = 0.001 **(b)** λ = 0.01

(c) λ = 0.1 **(d)** λ = 1

Fig. 3 The distributions of deeply learned features under the joint supervision of softmax loss and center loss. λ is the loss weight for center loss. The points with different colors denote features from different classes. Different λs lead to different deep feature distributions. The **×** Marks are the learned centers, and the vectors with dotted line are the learned weight parameters in the softmax loss. Best viewed in color (Color figure online)

The initialization of centers has two options: (A) Each $\boldsymbol{c}_j$ is initialized by the average of the y_ith class deep features $\{\boldsymbol{x}_i\}$, which are extracted by randomly initialized CNNs; (B) Each $\boldsymbol{c}_j$ is randomly initialized by common initializers like *Gaussian* or*Xavier* (Glorot and Bengio 2010). We empirically compare them and could not see any difference. So we adopt the option B for center initialization in all the experiments since it is computationally efficient.

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 Parameter Sharing

We note that the two loss functions in joint supervision share lots of commons in learning deep features. Briefly, the softmax loss maximizes $\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i$ while the center loss minimizes $\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$. In the course of learning, both $\boldsymbol{w}_{y_i}$ and $\boldsymbol{c}_{y_i}$ gradually close to $\boldsymbol{x}_i$ in terms of angular and Euclidean distance, respectively. *It implies that* $\boldsymbol{w}_{y_i}$ *and* $\boldsymbol{c}_{y_i}$ *may have similar directions in well-trained CNNs*, which can be empirically verified in Fig. 3. Similar idea is also reported and analyzed by Wang et al. (2017); Liu et al. (2017b). Such evidences indicate that we may have over parameterized the CNNs in the joint supervision framework.

To reduce the redundancy, we adopt the parameter sharing between the softmax loss and the center loss. The center $\boldsymbol{c}_i$ is hence reparametrized as $\gamma_i \boldsymbol{w}_i$, as given by Eq. 6:

$$\mathcal{L}_{C_\cap} = \frac{1}{2m} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \gamma_{y_i} \boldsymbol{w}_{y_i}\|_2^2. \tag{6}$$

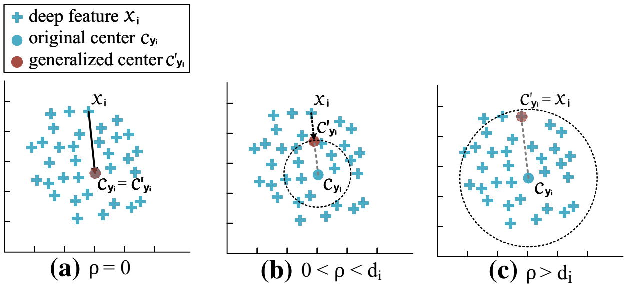
γ_i is a scaling parameter, controlling the center magnitude. $\boldsymbol{w}_i$ is reused to specify the center direction. With this strategy, the number of the learnable parameters in center loss is now reduced from nd $(\{\boldsymbol{c}_j\}_{j=1}^n)$ to n $(\{\gamma_j\}_{j=1}^n)$. Based on Eq. 6, we can derive the updating rule for γ_j.

$$\Delta\gamma_j = \frac{\sum_{i=1}^m \delta(y_i = j) \cdot (\gamma_j - \boldsymbol{x}_i^T \boldsymbol{w}_{y_i} / \boldsymbol{w}_{y_i}^T \boldsymbol{w}_{y_i})}{\epsilon + \sum_{i=1}^m \delta(y_i = j)}. \tag{7}$$

Moreover, we can impose constraint on $\{\gamma_j\}_{j=1}^n$ to make them equivalent to each other. It further reduces the number of learnable parameters from n $(\{\gamma_j\}_{j=1}^n)$ to 1 (γ), which is negligible. The formulation and its updating rule are given by:

$$\begin{aligned} \mathcal{L}_{C_\cap^+} &= \frac{1}{2m} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \gamma \boldsymbol{w}_{y_i}\|_2^2, \\ \Delta\gamma &= \gamma - \frac{\sum_{i=1}^m \boldsymbol{x}_i^T \boldsymbol{w}_{y_i}}{\epsilon + \sum_{i=1}^m \boldsymbol{w}_{y_i}^T \boldsymbol{w}_{y_i}}. \end{aligned} \tag{8}$$

For convenience, we term the vanilla center loss (Eq. 2) as CL. The advanced CLs ($\mathcal{L}_{C_\cap}$ from Eq. 6 and $\mathcal{L}_{C_\cap^+}$ from Eq. 8) with parameter sharing are called ACL and ACL-γ, respectively.



(a) $\rho = 0$ **(b)** $0 < \rho < d_i$ **(c)** $\rho > d_i$

Fig. 4 The illustrations for the generalized center

3.4 Generalized Center

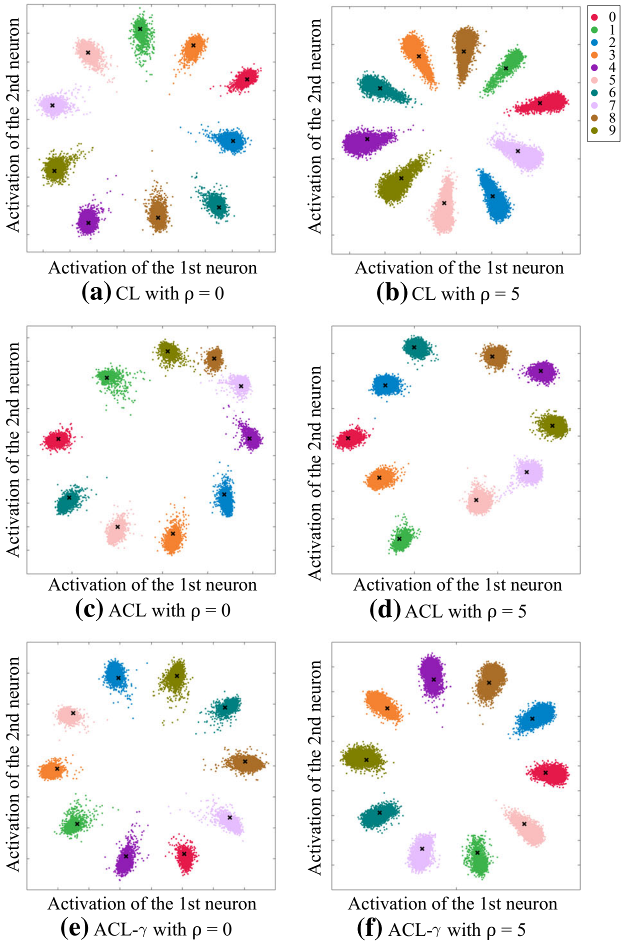
The center loss makes a strong assumption that the deep features from the same class should be as closed as possible to their center, which is a single point in embedding space. It overestimates the learning ability of CNNs, and ignores the fact that face images from the same identity may exhibit inherent large variation due to pose, age, illumination, occlusion and other factors. Consequently, we can slightly relax the learning objective of center loss by allowing reasonable intra-class variations.

In embodying this philosophy, we generalize the center from a single point to a region, as illustrated in Fig. 4. This allows the deep features to preserve necessary intra-class variations. The generalized center can be seen as a circle in 2-dimensional case, or a hypersphere in high-dimensional case. ρ is introduced as a hyper-parameter to specify the radius of the regions. The distance between feature $\boldsymbol{x}_i$ and its center region is defined as the distance between $\boldsymbol{x}_i$ and a dynamic center $\boldsymbol{c}'_{y_i}$, where the $\boldsymbol{c}'_{y_i}$ has the minimal distance to $\boldsymbol{x}_i$ within the region, as shown in Fig. 4. The formulation is given by:

$$\boldsymbol{c}'_{y_i} = \boldsymbol{c}_{y_i} + \kappa_i (\boldsymbol{x}_i - \boldsymbol{c}_{y_i}). \tag{9}$$

The κ_i is given by $\min(\frac{\rho}{d_i}, 1)$, and d_i is the distance between $\boldsymbol{x}_i$ and $\boldsymbol{c}_{y_i}$, given by $\sqrt{(\boldsymbol{x}_i - \boldsymbol{c}_{y_i})^T (\boldsymbol{x}_i - \boldsymbol{c}_{y_i})}$. κ is proportional to ρ. For the advanced center loss ACL or ACL-γ, $\boldsymbol{c}_{y_i}$ could also be replaced $\gamma_{y_i} \boldsymbol{w}_{y_i}$ or $\gamma \boldsymbol{w}_{y_i}$, respectively. With different ρs, generalized centers have different properties.

- When $\rho = 0$, the generalized center $\boldsymbol{c}'_{y_i}$ degrades to vanilla center $\boldsymbol{c}_{y_i}$. Center in Eqs. 2 and 6 be considered as a special case of the generalized center.
- When $0 < \rho < d_i$, the generalized center $\boldsymbol{c}'_{y_i}$ is given by $\boldsymbol{c}_{y_i} - \rho \cdot \frac{\boldsymbol{x}_i - \boldsymbol{c}_{y_i}}{\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2}$. Intuitively, $\boldsymbol{c}'_{y_i}$ locates at the edge of the center region (a hypersphere centered at $\boldsymbol{c}_{y_i}$ with radius ρ), and has the minimal distance to $\boldsymbol{x}_i$.



(a) CL with ρ = 0 **(b)** CL with ρ = 5

(c) ACL with ρ = 0 **(d)** ACL with ρ = 5

(e) ACL-γ with ρ = 0 **(f)** ACL-γ with ρ = 5

Fig. 5 The distributions of deeply learned features of CL, ACL, and ACL-γ with different radius ρ. Different ρ lead to different deep feature distributions. Best viewed in color (Color figure online)

- When $d_i \leq \rho$, the generalized center $\boldsymbol{c}'_{y_i}$ locates on the point $\boldsymbol{x}_i$. For this case, we do not penalize the distance between $\boldsymbol{x}_i$ and $\boldsymbol{c}_{y_i}$.

We perform experiments of the generalized center with different ρs on MNIST, shown in Fig. 5. It is known that MNIST is a less challenging dataset, where only a small portion of the data are considered as hard samples. It may not be easy for us to clearly observe how the hard samples distribute in Fig. 5. To make it clearer and more intuitive, we present the distributions of the intra-class distances of two models ($\rho = 0$ and $\rho = 5$) in Fig. 6. The *long tail* on the right shows the hard samples with large intra-class distances. Compared with the vanilla center loss (red), generalized center (blue) greatly depresses the *tail* part, indicating that hard samples are better handled. Also, the mean of the intra-class distances increases with larger ρ. It indicates that more intra-class variations are preserved.

Note that we are making an isotropic Gaussian assumption here for the learned features, i.e. learning generalized centers as the means and identity matrix as the covariance. Mahalanobis distance could be an alternative in order to capture the correlations of different dimensions in the learned features. In practice, however, we may not have sufficient data to estimate an accurate covariance matrix for each class. So the assumption we made achieves a good trade-off. It not only simplifies the optimization, but also reduces the risk of overfitting.

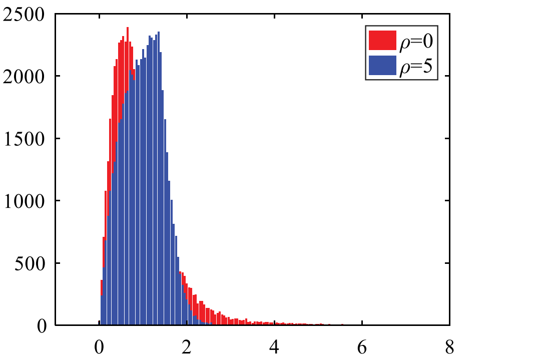


Fig. 6 The distributions of intra-class distances with different ρs on MNIST dataset

An alternative interpretation for generalized center By substituting Eqs. 9 into 5, ρ can be extracted outside the summation, as shown in the following:

$$\begin{aligned}\mathcal{L}_C &= \frac{1}{2m}\sum_{i=1}^{m}\|\boldsymbol{x}_i - \boldsymbol{c}'_{y_i}\|_2^2\\ &= \frac{1}{2m}\sum_{i=1}^{m}\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i} + \kappa(\boldsymbol{x}_i - \boldsymbol{c}_{y_i})\|_2^2\\ &= \frac{(1-\kappa)^2}{2m}\sum_{i=1}^{m}\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2\\ &= \frac{\left(1-\min\left(\frac{\rho}{d_i},1\right)\right)^2}{2m}\sum_{i=1}^{m}\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2\end{aligned} \tag{10}$$

Since κ depends on the deep feature $\boldsymbol{x}_i$ and its center $\boldsymbol{c}_{y_i}$, the term of $\frac{(1-\min(\frac{\rho}{d_i},1))^2}{2}$ can be interpreted as an adaptive multiplier depending on the position of x_i, rather than a fixed one. By introducing κ, we are able to adaptively relax the objective of center loss.

3.5 Discussions

- *Compared to contrastive loss and triplet loss* Contrastive loss (Sun et al. 2014a; Wen et al. 2016a) and triplet loss (Schroff et al. 2015) are proposed to enhance the discriminative power of the deeply learned face features. However, they both suffer from dramatic data expansion when constituting effective sample pairs or sample

Table 2 Comparisons between center loss and coco loss

	W/o center loss term	W/ a center loss term
w/o L2 normalization on $\boldsymbol{x}$ and $\boldsymbol{w}$	Softmax loss $-\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{\boldsymbol{w}_{y_i}^T\boldsymbol{x}_i+b_{y_i}}}{\sum_{j=1}^{n}e^{\boldsymbol{w}_j^T\boldsymbol{x}_i+b_j}}$	Softmax loss + advanced center loss $-\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{\boldsymbol{w}_{y_i}^T\boldsymbol{x}_i+b_{y_i}}}{\sum_{j=1}^{n}e^{\boldsymbol{w}_j^T\boldsymbol{x}_i+b_j}}+\frac{\lambda}{2m}\lVert\boldsymbol{x}_i-\gamma_{y_i}\boldsymbol{w}_{y_i}\rVert_2^2$
w/ L2 normalization on $\boldsymbol{x}$ and $\boldsymbol{w}$, where $\tilde{\boldsymbol{w}}=\frac{\boldsymbol{w}}{\lVert\boldsymbol{w}\rVert_2},\tilde{\boldsymbol{x}}=\frac{\boldsymbol{x}}{\lVert\boldsymbol{x}\rVert_2}$	coco loss $-\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{\tilde{\boldsymbol{w}}_{y_i}^T\tilde{\boldsymbol{x}}_i+b_{y_i}}}{\sum_{j=1}^{n}e^{\tilde{\boldsymbol{w}}_j^T\tilde{\boldsymbol{x}}_i+b_j}}$	coco loss + advanced center loss $-\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{\tilde{\boldsymbol{w}}_{y_i}^T\tilde{\boldsymbol{x}}_i+b_{y_i}}}{\sum_{j=1}^{n}e^{\tilde{\boldsymbol{w}}_j^T\tilde{\boldsymbol{x}}_i+b_j}}+\frac{\lambda}{2m}\lVert\tilde{\boldsymbol{x}}_i-\gamma_{y_i}\tilde{\boldsymbol{w}}_{y_i}\rVert_2^2$

triplets from the training set. On the contrary, the center loss enjoys the same requirement as the softmax loss and does not need complex recombination of the training samples. Consequently, the supervised learning of our CNNs is more efficient and easier to implement. Moreover, our loss function targets more directly on the learning objective of the intra-class compactness, which is beneficial to the discriminative feature learning.

- *Compared to the softmax loss family* Recently, (Wang et al. 2017; Ranjan et al. 2017; Liu et al. 2017a) reformulate the softmax loss as metric learning approaches for face recognition. In order to learn discriminative features on hypersphere, they explicitly model and optimize the angular distances between the samples. They achieve impressive performance on public benchmarks with simple modifications. Different from the aforementioned works, we take an alternative view by introducing an Euclidean distance based loss for deep metric learning. These two branches of approaches are complementary to each other and can be used at the same time.
- *Compared to coco loss* The formulations of coco loss (Liu et al. 2017b) and our approach are compared in Table 2. It can be observed that our approach is not euivalent to coco loss even if weight $\boldsymbol{w}_{y_i}$ and feature $\boldsymbol{x}_i$ are normalized with L2 length. In fact, they focus on different perspectives in learning discriminative features. Specifically, our approach explicitly formulates and minimizes the intra-class distance as an additional term, while coco loss integrates their merits into softmax loss itself. As an additional term, center loss is the key for our approach to yield discriminative features.

4 Experiments

Extensive experiments are conducted on several public domain face datasets to verify the effectiveness of the proposed approach. Section 4.1 describes the necessary experimental details. Section 4.2 provides a comprehensive ablation study for center loss. Section 4.3 investigates a number of prevalent loss functions for face recognition. Finally, we compare our face recognition approach to the state of the arts.

Table 3 The numbers of overlap identities in training and testing datasets we used

	LFW	YTF	IJB-A	MegaFace
CASIA	17	7	22	42
VGGFace2	594	245	31	18

4.1 Experimental Details

Training data VGGFace2 (Cao et al. 2017) is a web-collected face images dataset with manually identity labelling. It consists of two splits: `vggface2_train` and `vggface2_test`. For `vggface2_train`, we remove 271,554 images (of 637 identities) with failed face detections or labeled identities appearing in the testing datasets. Finally, 2,870,336 images from 7994 identities are remained as our training data. The face images are horizontally flipped for data augmentation.

We also use `CASIA-WebFace` (Yi et al. 2014) as another training set, in order to explore the performance of the proposed approach with small-scale training set. `CASIA-WebFace` includes 494,414 images from 10,752 identities, where 59 identities appearing in testing set are manually removed. The testing set includes LFW, YTF, IJB-A, and MegaFace Challenge 1. The overlapping identities between training and testing datasets are given in Table 3.

Preprocessing All the faces in images and their landmarks are detected by MTCNN (Zhang et al. 2016). We use 5 landmarks (two eyes, nose and two 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 (Sun et al. 2014b), each pixel (in $[0, 255]$) in RGB images is normalized by subtracting 127.5 then dividing by 128.

CNN configurations We implement the CNNs using the Caffe (Jia et al. 2014) library with our modifications. The details of CNN configurations are given in Table 4. The 20-layer net is the backbone network. We also explore architectures with different number of layers, such as 4-layer net, 10-layer net, 36-layer net, or 64-layer which are constructed by adding (or removing) a few residual blocks to (or from) the

Table 4 Our CNN architectures with different convolutional layers

Layer	4-layer	10-layer	20-layer	36-layer	64-layer
Conv1.x	$(3, 64)_{/2,1} \times 1,$	$(3, 64)_{/2,1} \times 1,$	$(3, 64)_{/2,1} \times 1, \begin{bmatrix} (3, 64)_{/1,1} \\ (3, 64)_{/1,1} \end{bmatrix} \times 1,$	$(3, 64)_{/2,1} \times 1, \begin{bmatrix} (3, 64)_{/1,1} \\ (3, 64)_{/1,1} \end{bmatrix} \times 2,$	$(3, 64)_{/2,1} \times 1, \begin{bmatrix} (3, 64)_{/1,1} \\ (3, 64)_{/1,1} \end{bmatrix} \times 3,$
Conv2.x	$(3, 128)_{/2,1} \times 1,$	$(3, 128)_{/2,1} \times 1, \begin{bmatrix} (3, 128)_{/1,1} \\ (3, 128)_{/1,1} \end{bmatrix} \times 1,$	$(3, 128)_{/2,1} \times 1, \begin{bmatrix} (3, 128)_{/1,1} \\ (3, 128)_{/1,1} \end{bmatrix} \times 2,$	$(3, 128)_{/2,1} \times 1, \begin{bmatrix} (3, 128)_{/1,1} \\ (3, 128)_{/1,1} \end{bmatrix} \times 4,$	$(3, 128)_{/2,1} \times 1, \begin{bmatrix} (3, 128)_{/1,1} \\ (3, 128)_{/1,1} \end{bmatrix} \times 8,$
Conv3.x	$(3, 256)_{/2,1} \times 1,$	$(3, 256)_{/2,1} \times 1, \begin{bmatrix} (3, 256)_{/1,1} \\ (3, 256)_{/1,1} \end{bmatrix} \times 2,$	$(3, 256)_{/2,1} \times 1, \begin{bmatrix} (3, 256)_{/1,1} \\ (3, 256)_{/1,1} \end{bmatrix} \times 4,$	$(3, 256)_{/2,1} \times 1, \begin{bmatrix} (3, 256)_{/1,1} \\ (3, 256)_{/1,1} \end{bmatrix} \times 8,$	$(3, 256)_{/2,1} \times 1, \begin{bmatrix} (3, 256)_{/1,1} \\ (3, 256)_{/1,1} \end{bmatrix} \times 16,$
Conv4.x	$(3, 512)_{/2,1} \times 1,$	$(3, 512)_{/2,1} \times 1, \begin{bmatrix} (3, 512)_{/1,1} \\ (3, 512)_{/1,1} \end{bmatrix} \times 1,$	$(3, 512)_{/2,1} \times 1, \begin{bmatrix} (3, 512)_{/1,1} \\ (3, 512)_{/1,1} \end{bmatrix} \times 1,$	$(3, 512)_{/2,1} \times 1, \begin{bmatrix} (3, 512)_{/1,1} \\ (3, 512)_{/1,1} \end{bmatrix} \times 2,$	$(3, 512)_{/2,1} \times 1, \begin{bmatrix} (3, 512)_{/1,1} \\ (3, 512)_{/1,1} \end{bmatrix} \times 3,$
FC1	512	512	512	512	512

Conv1.x, Conv2.x, Conv3.x, and Conv4.x denote convolution units that may contain multiple convolutional layers, and residual units are shown in double-row brackets. E.g., $(3, 64)_{/2,1} \times 1$ denotes convolutional layer with 64 filters of size 3 × 3, where the stride and padding are 1 and 2 respectively. Each convolutional layer is followed by a PReLU (He et al. 2015b) nonlinear unit. FC1 is the first fully-connected layer

backbone network. These models are trained with batch size of 128 on four GPUs. The learning rate is started from 0.1, and divided by 10 at the 25 K, 38 K iterations. A complete training is finished at 45 K iterations.

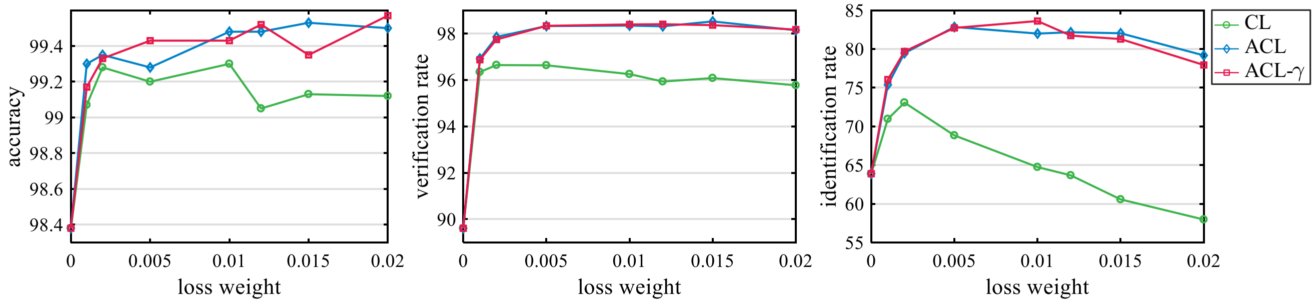
Testing details The deep features are extracted from the output of the FC1 layer. We extract the features for each image and its horizontally flipped one, and concatenate them as the final identity representation. The score is computed by the cosine similarity of two identity representations. We follow the protocols of the evaluation benchmarks and report the performance accordingly.

4.2 Ablation Studies

Evaluation dataset and protocols In this section, we use LFW (Huang et al. 2007; Huang and Learned-Miller 2014) as the evaluation dataset. It contains 13,233 web-collected face images from 5749 identities with large variations in pose, expression and illuminations. We evaluate our approaches based on two protocols. The first one is the standard protocol of *unrestricted with labeled outside data* (Huang and Learned-Miller 2014), where we evaluate 6000 pairs of face images (3000 positive pairs and 3000 negative pairs). 6000 pairs are divided into 10 splits. We compute the accuracies on them and report the average. The second one is the protocol of *benchmark of large-scale unconstrained face recognition* (BLUFR), where we evaluate 47,117,778 pairs of face images (156,915 positive pairs and 46,960,863 negative pairs). The verification rates (VR) at false accept rate (FAR) of 0.1% and the open-set detection and identification rates (DIR) at rank-1 and FAR = 1% are reported. More details are described in Liao et al. (2014).

We first present a number of ablation studies to analyze the behavior of CL, ACL and ACL-γ with varying loss weights λ. Here we fix the training set and CNN architecture to `vggface2_train` and 20-layer net, respectively. Second, we conduct exploratory experiments for radius ρ with fixed loss weights. In the following, we fix the hyperparameters λ and ρ, and perform experiments with various network architectures (see Table 4) and training sets (`vggface2_train` and `CASIA-WebFace`). From the experimental results, we have the following observations.

Parameter sharing The results in Fig. 7 show that the parameter sharing strategy not only reduces the model size, but also significantly improves the performance. Specifically, the best accuracies on 6000 pairs are 99.30%, 99.53%, and 99.57% for CL, ACL, and ACL-γ, respectively. Similarly, the best VRs and DIRs on BLUFR are (96.64% vs. 98.52% vs. 98.4%) and (73.07% vs. 82.84% vs. 83.61%). The performance improvements support our hypothesis that the parameters in softmax loss and center loss are redundant and reusable. Moreover, the advanced center loss is less sen-



(a) LFW accuracy on 6000 pairs

(b) LFW verification rate under BLUFR protocol

(c) LFW identification rate under BLUFR protocol

Fig. 7 Experimental results of varying loss weight λ

Table 5 Experimental results on LFW with varying radius ρ

Radius ρ	6000 pairs Acc. (%)	BLUFR		Average dist. ($\bar{d}$)	Normalized range
		VR (%)	DIR (%)		
0	99.30	96.25	64.74	12.05	[0.35, 3.31]
5	99.13	97.03	71.83	13.82	[0.36, 3.07]
10	99.33	97.33	73.42	16.18	[0.39, 2.95]
15	99.35	97.49	75.83	18.84	[0.44, 2.68]
20	99.35	**97.79**	77.15	22.09	[0.45, 2.48]
25	**99.35**	97.75	**79.32**	25.39	[0.46, 2.32]

Bold values indicate the best results

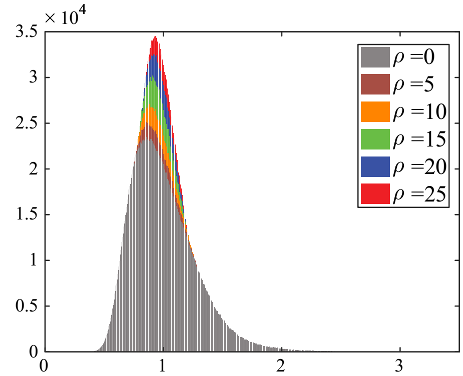


Fig. 8 The distributions of the intra-class distances with different ρs on LFW dataset

sitive to various loss weights, since its performance remains stable across a wide range of λ.

Loss weight λ A proper loss weight is important for balancing two loss functions in joint supervision. As can be seen in Fig. 7, simply using either the softmax loss or center loss is not a good choice. Comparing the models of $\lambda = 0$ and $\lambda = 0.01$, the accuracy, VR, and DIR are improved from (98.38%, 89.61%, and 63.88%) to (99.3%, 96.25%, and 64.74%). This is because deeply learned features remain considerable intra-class variations when only softmax loss is adopted. As λ increases, we are able to achieve an appropriate trade-off on intra-class and inter-class distances. Based on the experimental results, we fix the loss weight to 0.01 unless otherwise stated.

Radius ρ We present the experimental results of CL with different ρ in Table 5. Comparing the models of $\rho = 0$ and $\rho = 20$, the performances are boosted from (99.3%, 96.25%, and 64.74%) to (99.35%, 97.75%, and 79.32%), with (0.05%, 1.54%, and 12.43%) improvements. The protocol of 6000 pairs is considered to be saturated owing to its small size. On the more challenging protocol of BLUFR, the improvements are more significant. This experiment verifies that the region-based center could be a kind of effective generalized center, which is beneficial to the performance.

To better understand the generalized center, the average distances $\bar{d}$s between features and their centers with different ρs are reported in Table 5. Also, we present the normalized ranges of intra-class distance d_i, which are given by $[\frac{\min(d_i)}{\bar{d}}, \frac{\max(d_i)}{\bar{d}}]$. From Table 5 we observe that $\bar{d}$ increases with larger ρ. It is consistent with the learning objective of generalized center, where more intra-class variations are preserved. On the other hand, the normalized ranges of intra-class distances are smaller with larger ρ. It indicates that the hard samples (with large intra-class distances) are effectively optimized, leading to better performance. The distributions of intra-class distances shown in Fig. 8 provide intuitive illustrations and further verify our observations.

Training set As observed from Table 6, all the results using larger-scale training set (`vggface2_train`, 3 M) are better than their counterparts using small-scale dataset (`CASIA-WebFace`, 0.5 M). This indicates that our approac-

Table 6 LFW Results obtained by different training sets

Method	Training set (M)	6000 pairs	BLUFR	
		Acc. (%)	VR (%)	DIR (%)
CL	0.5	98.58	93.24	61.71
CL	3	99.30	96.25	64.74
ACL	0.5	98.93	94.69	72.33
ACL	3	**99.48**	98.34	81.99
ACL-γ	0.5	98.90	94.86	72.32
ACL-γ	3	99.43	**98.39**	**83.61**

Bold values indicate the best results

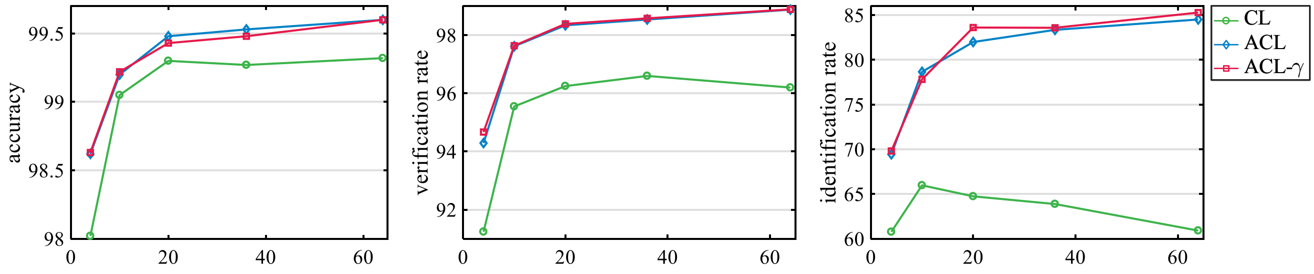
hes are scalable and far from saturated on the scale of current training set. We believe our performance can be further improved by more and more training data.

Depth of CNNs The joint supervision of softmax loss and center loss defines a challenging but useful learning objective for face recognition. It makes full use of the learning ability of deep CNNs to boost the performance. Comparing the results of 4-layer and 64-layer nets in Fig. 9, the best performance of ACL-γ are improved from (98.63%, 94.67%, and 69.81%) to (99.6%, 98.89%, and 85.28%). However, the results of CL are not consistently improving. We argue that this is because of overfitting, caused by the increasing convolutional layers. With the parameter sharing strategy, our ACL and ACL-γ models can easily enjoy performance gains from increased depth, avoiding overfitting to a great extent.

4.3 Experiments on Different Loss Functions

We compare our approaches with other state-of-the-art loss functions, including softmax loss, contrastive loss (Sun et al. 2014a), NormFace (Wang et al. 2017), coco loss (Liu et al. 2017b), and SphereFace (Liu et al. 2017a). We implement these methods using the open-source codes they released. Moreover, we also design the experiments on the combinations of state-of-the-art loss functions and center loss to see whether they are complementary. For fair comparison, all the experimental settings in training and testing are the same except the loss functions. The training set and network architecture are fixed to `vggface2_train` and 20-layer net.

We have two major observations from the results in Table 7. First, CL, ACL and ACL-γ substantially improve the



(a) LFW accuracy on 6000 pairs **(b)** LFW verification rate under BLUFR protocol **(c)** LFW identification rate under BLUFR protocol

Fig. 9 Experimental results of varying the number of layers in CNNs

Table 7 Performance of different loss functions on LFW and IJB-A

Loss function	LFW			IJB-A		
	6000 pairs	BLUFR		TAR (%) at FAR's of		
	Acc. (%)	VR (%)	DIR (%)	0.001	0.01	0.1
Softmax loss	98.38	89.61	63.88	78.01	88.79	95.40
Softmax loss + contrastive loss (Sun et al. 2014a)	99.08	95.93	68.45	78.95	89.81	96.22
NormFace (Wang et al. 2017)	97.02	83.73	48.82	68.24	83.82	93.62
coco loss (Liu et al. 2017b)	99.10	95.70	72.58	84.77	92.43	97.01
SphereFace(Liu et al. 2017a)	99.42	99.18	91.34	91.78	95.87	**98.22**
Softmax loss + CL	99.35	97.75	79.32	79.02	91.41	97.68
Softmax loss + ACL	99.48	98.34	81.99	87.12	94.07	98.12
Softmax loss + ACL-γ	99.43	98.39	83.61	87.61	94.33	98.14
coco loss + ACL-γ	**99.48**	97.96	76.22	84.24	93.45	97.55
SphereFace + ACL-γ	99.45	**99.27**	**91.45**	**91.87**	**95.93**	98.09

Bold values indicate the best results

Table 8 Comparisons with the state-of-the-art methods on LFW and YTF

Method	Model	Input size	Training set	LFW (%)	YTF (%)
DeepFace (Taigman et al. 2014)	3	152 × 152	4M*	97.35	91.4
DeepID2+ (Sun et al. 2015)	25	47 × 55	300K*	99.47	93.2
FaceNet (Schroff et al. 2015)	1	224 × 224	200M*	99.65	95.1
Deep Embedding (Liu et al. 2015)	1	–	1.3M*	99.13	–
coco loss (Liu et al. 2017b)	1	235 × 235	3M*	99.86	–
ReST (Wu et al. 2017)	1	112 × 112	0.5M	99.03	95.4
NormFace (Wang et al. 2017)	1	112 × 96	0.5M	99.19	94.7
SphereFace (Liu et al. 2017a)	1	112 × 96	0.5M	99.42	95.0
VDNet (Sohn et al. 2017)	1	100 × 100	1M	–	91.4
DAN (Rao et al. 2017)	1	112 × 96	1M	–	94.3
NAN (Yang et al. 2016)	1	224 × 224	3M	–	95.7
Softmax loss	1	112 × 96	3M	98.55	94.7
Softmax loss + CL	1	112 × 96	3M	99.30	95.2
Softmax loss + ACL-γ	1	112 × 96	3M	99.57	96.1
Softmax loss + CL ($\rho = 5$)	1	112 × 96	3M	99.33	95.4
Softmax loss + ACL-γ ($\rho = 5$)	1	112 × 96	3M	99.60	96.2
SphereFace + ACL-γ ($\rho = 5$)	1	112 × 96	3M	**99.60**	**96.6**

Bold values indicate the best results
*Indicates the private training dataset

softmax loss, with more than 15%, 18%, 19% and accuracy gains on DIR, respectively. They also clearly outperform the performance of softmax loss + contrastive loss (Sun et al. 2014a), NormFace (Wang et al. 2017), and coco loss (Liu et al. 2017b). However, they are marginally inferior than SphereFace. Second, we conduct experiments to combine ACL-γ with coco loss and SphereFace. The resulting models perform better than the single coco loss or the single SphereFace. Specifically, SphereFace + ACL-γ achieves the best result on BLUFR with 99.27% and 91.45%. Additional experiments on IJB-A are also presented in Table 7. Similar observations are shown in the results. Besides, we notice that ACL-γ can be added to many existing loss functions to improve their performance, especially on challenging protocols, like BLUFR on LFW and lower FAR on IJB-A. These facts verify that our approaches are useful, and complementary with the prevalent loss functions.

4.4 Experiments on LFW and YTF

Evaluation dataset and protocols In this section, we compare the performance of our methods to those of the state of the arts on LFW and YTF. YTF dataset (Wolf et al. 2011) consists of 3425 videos of 1595 different identities, with an average of 2.15 videos per person. The clip durations vary from 48 frames to 6070 frames, with an average length of 181.3 frames. For both LFW and YTF benchmarks, we follow the *unrestricted with labeled outside data* protocol (Huang and Learned-Miller 2014). The verification rates on 6000 image pairs in LFW and 5000 video pairs are reported.

Following the best practices we investigated, we train several 64-layer nets on `vggface2_train` using different combinations of our two contributions, and compare them with the state-of-the-art approaches. The results are summarized in Table 8. we make several observations based on the results of our models. First, parameter sharing (softmax loss + ACL-γ, 99.57% on LFW and 96.1% on YTF) and generalized center (softmax loss + CL($\rho = 5$), 99.33% on LFW and 95.4% on YTF) individually improve the performance (softmax loss + CL, 99.30% on LFW and 95.2% on YTF). Second, combining parameter sharing and generalized center (softmax loss + ACL($\rho = 5$)) can further achieve better results. The improvements are more significant on challenging dataset, as validated on IJB-A in Sect. 4.5.

Comparing to the state of the arts, both softmax loss + ACL-γ ($\rho = 5$) and SphereFace + ACL-γ ($\rho = 5$) achieve the accuracy of 99.6% on LFW, outperforming the other methods. Noted that the comparisons may not be direct since different methods use different network architectures and training datasets. Moreover, it is worth noting that our good results are achieved by only using the relative small image size (112 × 96, compared to 224 × 224 for coco loss), which greatly reduces the computation and memory cost. Moreover, our methods achieve 96.6% accuracy on YTF. It is higher than the performance of the video-based recognition approaches (95.7% for NAN, 94.3% for DAN, and 91.4% for VDNet) in Table 8. These results show that only using still images in the training, our models can yield highly discriminative face representations that is easily generalized to video data.

Table 9 State-of-the-art results on IJB-A datasets

Method	Input size	1:1 Verification (%)			1:N Identification (%)				
		TAR @ FAR			TPIR @ FPIR		TPIR @ Rank		
		0.001	0.01	0.1	0.01	0.1	1	5	10
LSFS (Wang et al. 2015a)	100 × 100	51.4	73.3	89.5	38.3	61.3	82.0	92.9	–
DCNN-fusion (Chen et al. 2016)	100 × 100	–	83.8	96.7	–	–	90.3	96.5	97.7
VDNet (Sohn et al. 2017)	100 × 100	64.9	86.4	97.0	–	–	89.5	95.7	96.8
DR-GAN (Tran et al. 2017)	100 × 100	53.9	77.4	–	–	–	85.5	94.7	–
PAMs (Masi et al. 2016)	224 × 224	65.2	82.6	–	–	–	84.0	92.5	94.6
NAN (Yang et al. 2016)	224 × 224	88.1	94.1	97.8	81.7	91.7	95.8	98.0	98.6
TPE (Sankaranarayanan et al. 2016)	224 × 224	81.3	90.0	96.4	75.3	86.3	93.2	–	97.7
Template Adaptation (Crosswhite et al. 2017)	224 × 224	83.6	93.9	97.9	77.4	88.2	92.8	97.7	98.6
VGGFace2 (Cao et al. 2017)	224 × 224	90.4	95.8	**98.5**	84.7	93.0	**98.1**	**99.4**	**99.6**
Softmax loss	112 × 96	81.43	90.62	96.41	76.62	89.72	95.92	98.18	98.69
Softmax loss + CL	112 × 96	78.51	92.31	97.92	71.15	92.10	96.71	98.74	99.07
Softmax loss + ACL-γ	112 × 96	89.07	95.59	98.46	84.63	95.16	97.42	98.81	99.09
Softmax loss + CL ($\rho = 5$)	112 × 96	84.09	94.15	98.30	81.94	94.48	97.01	98.72	99.11
Softmax loss + ACL-γ ($\rho = 5$)	112 × 96	89.19	95.22	98.46	86.71	95.29	97.30	98.80	99.17
SphereFace + ACL-γ ($\rho = 5$)	112 × 96	**93.67**	**96.90**	98.43	**93.57**	**97.75**	97.66	98.63	98.90

Bold values indicate the best results

4.5 Experiments on IJB-A Janus

Evaluation dataset and protocols IJB-A dataset (Klare et al. 2015) consists of 5712 face images and 2085 videos from 500 identities. They are collected from unconstrained environment and show large variations on poses. We evaluate the 1:1 face verification and report the verification rates at FAR of 0.001, 0.01, and 0.1. Again, we follow the best practices that we already obtained to train the models.

From the results in Table 9, we have several observations. First, the improvements obtained by parameter sharing and generalized center are consistent with those on LFW and YTF. In particular, combing these two (softmax loss + ACL-γ ($\rho = 5$)) achieves 2%–3% improvements on the most challenging protocol, i.e. TPIR at FPIR of 0.01. Second, our SphereFce + ACL-γ model ($\rho = 5$) achieves the best verification rates of 93.67% at 0.001 FAR and 96.90% at 0.01 FAR. Again, it indicates that our models work well in the challenging scenarios. For FAR of 0.1, our performance (98.43%) is slightly inferior than what VGGFace2 achieves (98.5%). Besides, for identification evaluation, we also observe that our SphereFce + ACL-γ model ($\rho = 5$) significantly improves the state-of-the-art results by 8.87% and 4.75% on TPIR at FPIR of 0.01 and 0.1. Moreover, we obtain competitive results using the relative small input size (112 × 96), while other state-of-the-art methods generally adopt larger input size (224 × 224). It demonstrates the superiority of our approaches, and indicates that there is substantial room for improvement.

Table 10 Comparison of performance on MegaFace Challenge 1

Method	Identification Rate (%)	Verification Rate (%)
Softmax loss	52.01	60.41
Softmax loss + CL	60.45	76.98
Softmax loss + ACL-γ	64.79	79.81
Softmax loss + CL ($\rho = 5$)	64.03	78.45
Softmax loss + ACL-γ ($\rho = 5$)	65.27	80.20

4.6 Experiments on MegaFace Challenging 1

Evaluation dataset and protocols MegaFace Challenge 1 (Miller et al. 2015) is released as a testing benchmark. It 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 690 K different individuals. The probe set for general face recognition is Facescrub (Ng and Winkler 2014). Facescrub dataset is a publicly available dataset, containing 100 K photos of 530 unique individuals (55,742 images of males and 52,076 images of females). Here we evaluate our model under the protocol of large-scale training set and report two results: (1) rank-1 identification rate with 1 M distractors, (2) verification rate at FAR $= 10^{-6}$ with 1 M distractors.

The results are summarized in Table 10 which shows unsurprising consistency with those on LFW, YTF, and IJB-

A datasets. Compared to the original center loss, parameter sharing and generalized center individually achieves 3%–4% and 2%–3% improvements on identification and verification, respectively. Combing these two proposed strategies can improve the performance further. These results demonstrate that the proposed approaches are useful and complementary with each other.

5 Conclusions

In this paper, we have proposed a new loss function, referred to as center loss for training deep face recognition networks. By combining center loss with traditional losses like softmax loss, the discriminative power of the deeply learned features can be highly enhanced for robust face recognition. In addition, we introduce parameter sharing strategy and generalized center to extend and strengthen center loss. We conduct extensive experiments to examine the performance of our loss on several large-scale face benchmarks. Empirical results convincingly demonstrated the effectiveness of the proposed approach.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (U1613211, 61633021) and Shenzhen Research Program (JCYJ20170818164704758, JCYJ20150925163005055, ZDSYS201605101739178).

References

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 2037–2041.

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). Sequential deep learning for human action recognition. In A. A. Salah & B. Lepri (Eds.), *Human behavior understanding* (pp. 29–39). New York: Springer.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, *19*(7), 711–720.

Bredin, H. (2017). Tristounet: triplet loss for speaker turn embedding. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5430–5434). IEEE.

Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2017). Vggface2: A dataset for recognising faces across pose and age. arXiv:1710.08092.

Cao, Z., Yin, Q., Tang, X., & Sun, J. (2010). Face recognition with learning-based descriptor. In *2010 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 2707–2714). IEEE.

Chen, D., Cao, X., Wang, L., Wen, F., & Sun, J. (2012). Bayesian face revisited: A joint formulation. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, & C. Schmid (Eds.), *Computer vision-ECCV 2012* (pp. 566–579). New York: Springer.

Chen, D., Cao, X., Wen, F., & Sun, J. (2013). Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *2013 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3025–3032). IEEE.

Chen, J. C., Patel, V. M., & Chellappa, R. (2016). Unconstrained face verification using deep CNN features. In *2016 IEEE winter conference on applications of computer vision (WACV)* (pp. 1–9). IEEE.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005* (Vol. 1, pp. 539–546). IEEE.

Chu, W., & Cai, D. (2017). Stacked similarity-aware autoencoders. In *Proceedings of the 26th international joint conference on artificial intelligence* (pp. 1561–1567). New Orleans: AAAI Press.

Crosswhite, N., Byrne, J., Stauffer, C., Parkhi, O., Cao, Q., & Zisserman, A. (2017). Template adaptation for face verification and identification. In *2017 12th IEEE international conference on automatic face and gesture recognition (FG 2017)* (pp. 1–8). IEEE.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005* (Vol. 1, pp. 886–893). IEEE.

Duan, Y., Lu, J., Feng, J., & Zhou, J. (2017). Learning rotation-invariant local binary descriptor. *IEEE Transactions on Image Processing*, *26*(8), 3636–3651.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249–256).

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition* (Vol. 2, pp. 1735–1742). IEEE.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv:1512.03385.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034)

Hu, J., Lu, J., & Tan, Y. P. (2014). Discriminative deep metric learning for face verification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1875–1882).

Huang, G. B., & Learned-Miller, E. (2014). Labeled faces in the wild: Updates and new reporting procedures. In *Technical Report* (pp 14–003). Amherst, MA, USA: Department of Computer Sciences, University of Massachusetts Amherst.

Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Technical report* Amherst: University of Massachusetts.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM international conference on multimedia* (pp. 675–678). ACM.

Jin, H., Wang, X., Liao, S., & Li, S. Z. (2017). Deep person re-identification with improved embedding. arXiv:1705.03332.

Klare, B. F., Klein, B., Taborsky, E., Blanton, A., Cheney, J., Allen, K., Grother, P., Mah, A., & Jain, A. K. (2015). Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1931–1939).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

LeCun, Y., Cortes, C., & Burges, C. J. (1998). The MNIST database of handwritten digits.

Liao, S., Lei, Z., Yi, D., Li, S. Z. (2014). A benchmark study of large-scale unconstrained face recognition. In *2014 IEEE international joint conference on biometrics (IJCB)* (pp. 1–8). IEEE.

Liu, J., Deng, Y., & Huang, C. (2015). Targeting ultimate accuracy: Face recognition via deep embedding. arXiv:1506.07310.

Liu, W., Wen, Y., Yu, Z., & Yang, M. (2016). Large-margin softmax loss for convolutional neural networks. In *ICML* (pp. 507–516).

Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE conference on computer vision and pattern recognition (CVPR)* (Vol. 1).

Liu, Y., Li, H., & Wang, X. (2017). Rethinking feature discrimination and polymerization for large-scale recognition. arXiv:1710.00870.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lu, J., Liong, V. E., Zhou, X., & Zhou, J. (2015). Learning compact binary face descriptor for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(10), 2041–2056.

Masi, I., Rawls, S., Medioni, G., & Natarajan, P. (2016). Pose-aware face recognition in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4838–4846).

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., & Mullers, K. R. (1999). Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop* (pp. 41–48). IEEE.

Miller, D., Kemelmacher-Shlizerman, I., & Seitz, S. M. (2015). Megaface: A million faces for recognition at scale. arXiv:1505.02108.

Nagi, J., Di Caro, G. A., Giusti, A., Nagi, F., & Gambardella, L. M. (2012). Convolutional neural support vector machines: Hybrid visual pattern classifiers for multi-robot systems. In *2012 11th international conference on machine learning and applications (ICMLA)* (Vol. 1, pp. 27–32). IEEE.

Ng, H. W., & Winkler, S. (2014). A data-driven approach to cleaning large face datasets. In *2014 IEEE international conference on image processing (ICIP)* (pp. 343–347). IEEE.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. *Proceedings of the British Machine Vision*, *1*(3), 6.

Prince, S. J., & Elder, J. H. (2007). Probabilistic linear discriminant analysis for inferences about identity. In *IEEE 11th international conference on computer vision, 2007. ICCV 2007* (pp. 1–8). IEEE.

Ranjan, R., Castillo, C. D., & Chellappa, R. (2017). L2-constrained softmax loss for discriminative face verification. arXiv:1703.09507.

Rao, Y., Lin, J., Lu, J., & Zhou, J. (2017). Learning discriminative aggregation network for video-based face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3781–3790).

Rippel, O., Paluri, M., Dollar, P., & Bourdev, L. (2015). Metric learning with adaptive density discrimination. arXiv:1511.05939.

Sankaranarayanan, S., Alavi, A., Castillo, C. D., & Chellappa, R. (2016). Triplet probabilistic embedding for face verification and clustering. In *2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS)* (pp. 1–8). IEEE.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815–823)

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Simonyan, K., Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2013). Fisher vector faces in the wild. In *BMVC* (vol. 2, p. 4).

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems* (pp. 1857–1865).

Sohn, K., Liu, S., Zhong, G., Yu, X., Yang, M. H., Chandraker, M. (2017). Unsupervised domain adaptation for face recognition in unlabeled videos. arXiv:1708.02191.

Song, H. O., Xiang, Y., Jegelka, S., & Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4004–4012). IEEE.

Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems* (pp. 1988–1996).

Sun, Y., Wang, X., & Tang, X. (2013). Hybrid deep learning for face verification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1489–1496).

Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1891–1898).

Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2892–2900).

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

Tadmor, O., Rosenwein, T., Shalev-Shwartz, S., Wexler, Y., & Shashua, A. (2016). Learning a metric embedding for face recognition using the multibatch method. In *Advances in neural information processing systems* (pp. 1388–1389).

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

Tang, Y. (2013). Deep learning using linear support vector machines. arXiv:1306.0239.

Tran, L., Yin, X., & Liu, X. (2017). Disentangled representation learning gan for pose-invariant face recognition. In *CVPR* (Vol 3, p. 7).

Vinyals, O., Jia, Y., Deng, L., & Darrell, T. (2012). Learning with recursive perceptual representations. In *Advances in neural information processing systems* (pp. 2825–2833).

Wang, D., Otto, C., & Jain, A. K. (2015a). Face search at scale: 80 million gallery. arXiv:1507.07242.

Wang, F., Xiang, X., Cheng, J., & Yuille, A. L. (2017). Normface: *l*_2 hypersphere embedding for face verification. arXiv:1704.06369.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Li, Z., Gong, D., Zhou, J., & Liu, W. (2018a). Cosface: Large margin cosine loss for deep face recognition. arXiv:1801.09414.

Wang, L., Qiao, Y., & Tang, X. (2015b). Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4305–4314).

Wang, X., & Tang, X. (2004). A unified framework for subspace face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(9), 1222–1228.

Wang, Y., Gong, D., Zhou, Z., Ji, X., Wang, H., Li, Z., Liu, W., & Zhang, T. (2018b). Orthogonal deep features decomposition for age-invariant face recognition. arXiv:1810.07599.

Wen, Y., Li, Z., & Qiao, Y. (2016). Latent factor guided convolutional neural networks for age-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4893–4901).

Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *European conference on computer vision* (pp. 499–515). New York: Springer.

Wisniewksi, G., Bredin, H., Gelly, G., & Barras, C. (2017). Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization. *Proceedings of Interspeech*, *2017*, 3582–3586.

Wolf, L., Hassner, T., & Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *2011 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 529–534). IEEE.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 210–227.

Wu, W., Kan, M., Liu, X., Yang, Y., Shan, S., & Chen, X. (2017). Recursive spatial transformer (rest) for alignment-free face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3772–3780).

Yang, J., Ren, P., Chen, D., Wen, F., Li, H., & Hua, G. (2016). Neural aggregation network for video face recognition. arXiv:1603.05474.

Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., & Lee Giles, C. (2017). Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Yao, J., Yu, Y., Deng, Y., & Sun, C. (2017). A feature learning approach for image retrieval. In *International conference on neural information processing* (pp. 405–412). New York: Springer.

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

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multi-task cascaded convolutional networks. arXiv:1604.02878.

Zhang, L., Yang, M., & Feng, X. (2011). Sparse representation or collaborative representation: Which helps face recognition? In *2011 IEEE international conference on computer vision (ICCV)* (pp. 471–478). IEEE.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2014). Object detectors emerge in deep scene cnns. arXiv:1412.6856.

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems* (pp. 487–495).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.