Image and Vision Computing 79 (2018) 123–132

Contents lists available at ScienceDirect

Image and Vision Computing

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

Improving face representation learning with center invariant loss ☆

Yue Wu[a,*], Hongfu Liu[b], Jun Li[a], Yun Fu[a,c]

[a] Department of Electrical & Computer Engineering, Northeastern University, MA, USA
[b] Michtom School of Computer Science, Brandeis University, MA, USA
[c] College of Computer & Information Science, Northeastern University, MA, USA

ARTICLE INFO

Article history:
Received 29 September 2017
Received in revised form 18 May 2018
Accepted 12 September 2018
Available online 22 September 2018

Keywords:
Face recognition
Convolutional Neural Network
Center invariant loss

ABSTRACT

In this paper, we address on the deep face representation learning with imbalanced data. With a large number of available face images of different people for training, Convolutional Neural Networks could learn deep face representation through classifying these people. However, uniformed distributed data for all people are hard to get. Some people come with more images but some come with less. In learning the deep face representation, the imbalanced images between people introduce the bias towards these people that have more images. Existing methods focus on the intra-class and inter-class variations but not well address the imbalanced data problem. To generate a robust and discriminative face representation for all people, we propose a center invariant loss which adds penalty to the differences between each center of classes. The center invariant loss could align the center of each person to the mean of all centers, which could force the deeply learned face features to have a good representation for all people with better generalization ability. Extensive experiments well demonstrate the effectiveness of the proposed approach. Many existing methods in learning deep face representation are further improved after adding the proposed center invariant loss.

1. Introduction

Convolutional Neural Networks (CNNs) lead to many great achievements on a series of vision tasks such as object recognition [1-7], object segmentation [8], image retrieval [9,10], video analysis [11,12] and many other applications [13-17]. Face Recognition [18-20] has also witnessed great success with the development of CNNs. DeepFace [21], DeepID series [22-24], FaceNet [25], and many other approaches have been proven to be effective.

As mentioned by [1], abundant training data and well-designed training strategies are necessary to train a robust deep model and get a discriminative face representation. Many approaches collected and labeled more images to better train CNNs. Facebook [21] and Google [25] used 4 million and 200 million images to train the network, respectively. But these data are private. For public available data, CelebFaces [22] was collected with 10K people and 200K images. In [26], CASIA-WebFace was released as the largest publicly available face dataset with 0.5 million images of 10,575 identities at that time, and gained 97.73% accuracy on Labeled Faces in the Wild (LFW) benchmark [27]. Recently, MS-Celeb-1M [28] was published with 8.9 million faces of 100 thousand identities. However, in these datasets, long-tail distribution of data [29,30] are observed. In Fig. 1(a), the per-subject image number of CASIA-WebFace is shown. We can see that only limited number of classes appear frequently. With imbalanced data for training, the learned classifier in Fig. 1 (c) has a bias on these classes with more samples compared with the classifier learned with balanced data in Fig. 1 (b). It is hard to learn features that are not biased for all classes with the imbalanced distributed data.

To better learn a face representation, Masi et al. [29] utilized domain specific methods for face image synthesis to generate more images. They inflated the size of CASIA-WebFace dataset to several times its size to avoid the long-tail effect. However, the face synthesizing is highly relied on the accurate facial landmark detection, which may introduce poor synthesizing data. And large amount synthesizing samples slow the training procedure. In [30], Zhang et al. proposed range loss that utilized the harmonic range in one class to measure the intra-class variations and add a penalty term to reduce these variations. Models trained on a balanced dataset achieved state-of-the-art performance. However, to ensure that there are enough samples for one class to calculate the range loss, the mini-batch needs to be carefully constructed, which introduce more labors. Hariharan et al. [31] introduced the concept of low-shot visual learning problem and a Squared Gradient Magnitude loss to penalize the difference between classifiers learned on large and small classes. They

☆ This paper has been recommended for acceptance by Vitomir Štruc.
* Corresponding author.
E-mail address: yuewu@ece.neu.edu (Y. Wu).

https://doi.org/10.1016/j.imavis.2018.09.010
0262-8856/

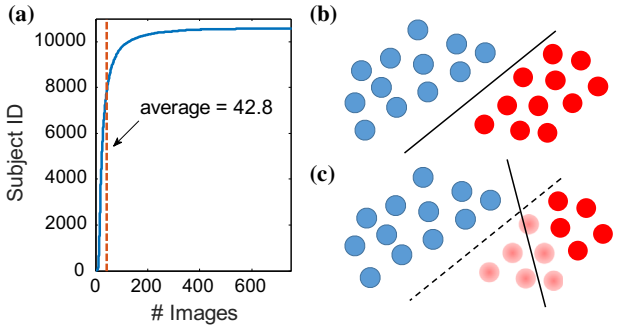


Fig. 1. (a) Long-tail distribution of per-subject image numbers in the CASIA-WebFace dataset [26], (b) the classifier learned with balanced data, (c) the classifier learned with imbalanced data.

also hallucinated additional samples by transferring modes of variation from base classes to improve the low-shot learning performance. However, in the problem of feature learning given long-tail distribution data, both small classes and large classes are used to learn the features instead of only using the large classes in the low-shot visual learning problem. Wen et al. [32] proposed to use center loss to minimize the intra-class variations to learn a discriminative deep face representation. Liu et al. [33] proposed to explicitly encourage a large margin between classification hyperplanes of different classes, which can achieve large inter-class separability in the deep face representation. Further, Liu et al. [34] extended the large margin method [33] to the deep hypersphere embedding. Similarly, the deep hypersphere embedding tackles the inter-class separability in a hypersphere manifold. However, these methods did not consider the dispersion between different classes due to imbalanced training data. Guo et al. [35] proposed an underrepresented-classes promotion (UP) loss to promote these classes that have few samples. Nevertheless, these methods target heavily unbalanced classification problem and did not take the feature representation learning into consideration.

To solve these problem, inspired by the idea of penalizing the difference between large and small classes in [31,35], we propose a center invariant (CIA) loss which aligns the center of each class to the mean of all centers, which could force the deeply learned face features to have a good representation for all people with better generalization ability. Center invariant loss penalizes the difference between the *L2* norm of each center of classes and the mean *L2* norm of all centers and explicitly encourages an unbiased deep face representation for all classes, despite different training samples. The extensive experiments well demonstrate the effectiveness of the proposed center invariant loss in learning deep face representation.

The contributions of this paper are mainly summarized as:

- We propose a new loss function, namely center invariant loss to enhance the generalization ability of deeply learned features by penalizing the difference between large classes and small classes.
- We verify that the center invariant loss could help deeply learned features to separate the feature space equally for all classes given imbalanced training data and improve the classification performance. With joint supervision of multiple state-of-the-art supervision signals that aim at inter-class and intra-class variations, the center invariant loss can further improve the generalization ability of deeply learned face representation.
- Our results show that one single model with 22-layer ResNet model trained with CASIA-WebFace achieves state-of-the-art performance, 99.28% on LFW [27] and 94.92% on YTF [37] benchmarks.

This paper is the journal extension from our recent conference paper [36]. The conference version mainly focused on the illustration of the motivation and showed primary results with joint supervision of center loss [32]. Compared to the conference version, in the journal extension, we address on multiple state-of-the-art supervision signals (i.e., large-margin softmax [33] and angular softmax [34]). After adding the proposed center invariant loss, face representation learned by these methods could be further improved.

The rest of this paper is organized as follows. In Section 2, we review the related works of face recognition and multiple supervision signals in learning deep face representation. In Section 3, we describe the proposed center invariant loss and joint training with other state-of-the-art supervision signals. Experimental results and discussions are shown in Section 4. Finally, we draw the conclusion in Section 5.

2. Related work

In this section, we first introduce the related works in deep face recognition. Second, we review the works of multiple supervision signals in learning discriminative face representation. We also highlight the differences between our work and the related work.

Face recognition is one of the central problems in pattern recognition. Performance was greatly improved with the recent development of deep learning techniques and CNNs. CNNs have been used for face recognition as far back as [38]. State-of-the-art performances are achieved recently due to the available massive amounts of data and the application of GPU. Facebook DeepFace [21] utilized over 4 million images used for training and achieved state-of-the-art at that time. Google FaceNet [25] got better results with training on 200 million faces later. Other CNNs based recognition systems, e.g. the Deep-ID series systems [22-24], achieved similar performance with far less data, but more elaborate network architectures. More data and novel network designs can both lead to better results.

Instead of collecting more data, several approaches utilized on the different supervision signals to improve the representation ability of deeply learned features. In [23], Sun et al. took both face identification and verification signals as supervision to increase inter-personal variations and to reduce the intra-personal variations at the same time. In [32], a center loss was proposed to penalize the distances between deep features and their corresponding class centers. Similarly, Zhang et al. [30] introduced a range loss to identify the maximum Euclidean distance between all sample pairs. Although these approaches try to minimize intra-class variations using various methods, increasing the inter-class variations is designated to the Softmax loss. Softmax loss is a widely used loss function for multi-class classification due to its simplicity and excellent performance. In [39], to solve the expensive computation cost for large scale inference, Michalis utilized the form of a rigorous lower bound on the exact probability to efficiently approximate the original Softmax probabilities. In [40], a fast locality-sensitive hashing technique is employed to approximate the dot product in Softmax function. This enabled scaling up the training and inference to millions of output classes using Softmax function. To explicitly encourage discriminative learning of features, Liu et al. [33,34] proposed generalized large-margin Softmax loss in terms of angular margin constraint.

The main difference between our center invariant loss and existing works is that we explicitly encourage an unbiased deep face representation for all classes despite different training samples. This compensates the terms both in improving inter-class separability and reducing intra-class variations. By penalizing the differences between centers of all classes, the deeply learned face representation can represent faces better regardless of the number of available training samples.

3. Deep representation learning

In this section, we first investigate the distributions of deeply learned features with softmax loss on a toy dataset with imbalanced data. The formulation of the proposed center invariant loss is presented later, followed by the optimization using stochastic gradients descent.

3.1. Imbalanced data phenomenon

In this subsection, distributions of deeply learned features of a Convolutional Neural Network (CNN) on MNIST [41] are investigated. The CNN model we used is LeNets++ [32] that has 6 convolutional layers and one fully-connected layer. The output number of the fully-connected layer is 2, which means that the deeply learned features have 2 dimensions. Thus, the features can be directly plotted on 2-D surface for visualization. The softmax loss function is formalized as:

$$L_s = -\log\left(\frac{e^{\mathbf{w}_y^T\mathbf{x}_i+b_y}}{\sum_{j=1}^{m} e^{\mathbf{w}_j^T\mathbf{x}_i+b_j}}\right), \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the deeply learned feature of the i-th sample with dimension d. This sample belongs to the y-th class. $\mathbf{w}_j \in \mathbb{R}^d$ is the j-th weights for the j-th class in the output fully connected layer. $b_j \in \mathbb{R}$ is the corresponding bias term. m is the number of classes.

The MNIST training set has 10 classes with total 60,000 samples. There are about 6000 samples for each class. The training set is nearly balanced. After training LeNets++ with original MNIST training set, the resulting 2D deep features are plotted in Fig. 2d and h. Fig. 2d shows the distributions of the training set and Fig. 2h shows the distributions on MNIST test set with 10,000 samples. From the two figures, we can observe that the deeply learned features for each class are nearly equitably, unbiased distributed in the feature space. Centers of each class almost form a circle. The trend on both the training set and test set holds. Then we reduce the training samples in five of ten classes. 200, 500 and 1000 samples are randomly selected from all training samples in the five classes. The rest classes use all training samples. The same CNN model and training procedure are utilized. The only difference between these CNN models is the ***number*** of training samples of the five classes. Distributions of these deeply learned features in the training set are shown in Fig. 2a, b and c for 200, 500 and 1000 samples, respectively. From these figures, we can observe the classes with more samples dominate the feat space. For example, in Fig. 2a, although these features seem separable, the feature space is actually not well occupied by all classes. The classes with small number of samples take a small area compared with the other classes. Moreover, we can observe that the centers of five classes with small samples are closer to the origin of the feature space than the other five classes. Bad splitting of the feature space leads to a poor representation of the deeply learned features, shown in Fig. 2e, f and g for the test set. Given the balanced data, the deeply learned features for every class are nearly equally split the feature space, e.g. Fig. 2h. However, with the imbalanced data, the deeply learned features are biased towards these classes with more samples, e.g. Fig. 2e, f and g, which has worse generalization ability compared with the unbiased features. The classification based on unbiased features (Fig. 2h) has 98.76% accuracy. However, the accuracy with biased features (Fig. 2e) only has 93.95%, which is much worse compared with the learned classifier with all data. To improve the generalization ability of the deeply learned features with imbalanced training data, we proposed center invariant loss of which the motivation is to align class centers for a better separation of the feature space.

3.2. Center invariant loss

We introduce the center invariant loss in this subsection. In m classes, the k-th class ($1 \le k \le m$) has n_k samples. The deeply learned

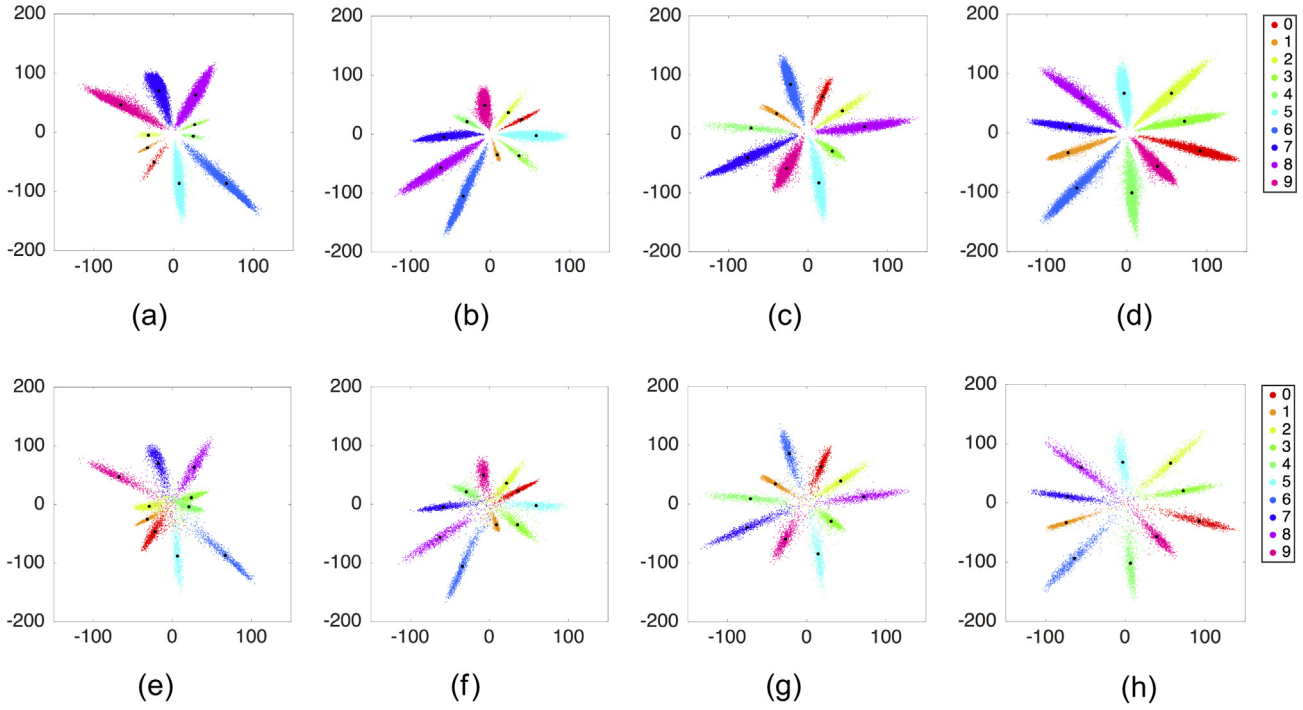


Fig. 2. Distributions of deeply learned features in different number of training samples and training/test sets. Five of ten classes have (a) (e) 200 samples, (b) (f) 500 samples, (c) (g) 1000 samples and (d) (h) full samples. (a) (b) (c) (d) are for training sets and (e) (f) (g) (h) are for test sets. Centers of each class are denoted as black dots.

feature of the i-th sample is $\mathbf{x}_i$. The k-th class center of the deeply learned features is $\mathbf{c}_k$, which can be written as:

$$\mathbf{c}_k = \frac{1}{n_k}\sum_{i=1}^{n_k}\mathbf{x}_i. \tag{2}$$

Center invariant loss of the i-th sample with label y can be formalized as:

$$L_I = \frac{1}{4}\left(\| \mathbf{c}_y \|_2^2 - \frac{1}{m}\sum_{k=1}^{m}\| \mathbf{c}_k \|_2^2\right)^2. \tag{3}$$

$$= \frac{1}{4}\left(\left\|\frac{1}{n_y}\sum_{i=1}^{n_y}\mathbf{x}_i\right\|_2^2 - \frac{1}{m}\sum_{k=1}^{m}\left\|\frac{1}{n_k}\sum_{i=1}^{n_k}\mathbf{x}_i\right\|_2^2\right)^2 \tag{4}$$

This formulation penalizes differences between centers of each class. Centers distribute at different positions in the feature space. The center invariant loss aims to make these centers to have the same Euclidean norm. A mean value of all centers Euclidean norm, written as $\frac{1}{m}\sum_{k=1}^{m}\| \mathbf{c}_k \|_2^2$, is calculated first. Given the i-th sample, the loss generates the cost from the norm of its class center to that mean value. In Fig. 3, we give an illustration of the center invariant loss in 2-D surface. We analyze the case with 200 training samples. The radius of the circle is the mean value of all centers Euclidean norm. We can see that the center invariant loss pull out the centers inside the circle and push in the centers outside the circle. Keeping the centers of each classes to equal distances from origin is a special case of our loss. However, instead of manually assigning a scalar to the radius of the circle, we set the radius to the mean of all class centers. It is worth of note that a trivial solution exists if we assign the same feature value to all the samples. The loss is still zero. However, this only happens if we only use the center invariant loss without the combination with other losses. In practice, since center invariant loss is always combined with other loss function that promotes separability between classes, the trivial solution actually never happened. The 1/4 multiplier is for the gradient simplicity in Eq. (20).

3.3. Combination with multiple supervision signals

Center invariant loss is a term to regularize the deeply learned features and can be combined with multiple training losses that target reducing intra-class variations or enhancing inter-class separability. In this subsections, we show the different combinations with multiple state-of-the-art supervision signals. The proposed loss aims to penalize the differences of the deeply learned features between classes, which is related to class labels. Thus the center invariant loss is different from a regularizer, which usually adds penalty in the model complexity, which is not related to data, e.g. l_2-norm on model weights.

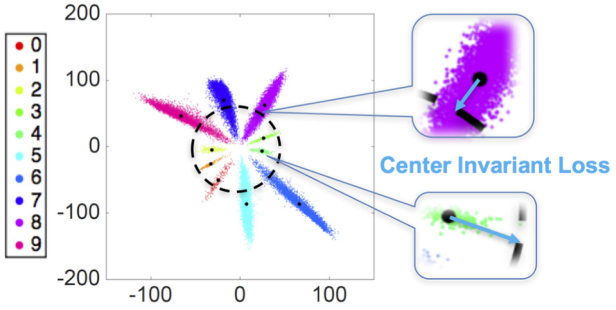


Fig. 3. Illustration of center invariant loss. The circle is the mean center Euclidean norm. The center invariant loss pull out the centers inside the circle and push in the centers outside the circle.

3.3.1. Softmax loss

The center invariant loss is added with softmax loss to jointly supervise the training of the CNN model. Softmax loss makes the features discriminative and center invariant loss forces a unified separation of feature space for a good generalization ability. The joint supervision of softmax loss and center invariant loss can be written as:

$$\begin{aligned} L &= L_s + \gamma L_I \\ &= -\log\left(\frac{e^{\mathbf{w}_y^T\mathbf{x}_i + b_y}}{\sum_{j=1}^{m} e^{\mathbf{w}_j^T\mathbf{x}_i + b_j}}\right) + \frac{\gamma}{4}\left(\| \mathbf{c}_y \|_2^2 - \frac{1}{m}\sum_{k=1}^{m}\| \mathbf{c}_k \|_2^2\right)^2, \end{aligned} \tag{5}$$

where γ is a scalar to balance the center invariant loss.

3.3.2. Large margin softmax loss

Large margin softmax loss proposed by Liu et al. [33] generalizes the softmax loss to a more general large margin softmax, which can be formulated as the following:

$$L_{ls} = -\log\left(\frac{e^{\|\mathbf{w}_y\|\|\mathbf{x}_i\|\phi(\theta_{y,i})}}{e^{\|\mathbf{w}_y\|\|\mathbf{x}_i\|\phi(\theta_{y,i})} + \sum_{j=1, j\neq y}^{m} e^{\|\mathbf{w}_j\|\|\mathbf{x}_i\|\cos(\theta_{j,i})}}\right), \tag{6}$$

in which $\phi(\theta_{j,i})$ is defined as $\phi(\theta_{j,i}) = (-1)^k\cos(m\theta_{j,i}) - 2k, \theta_{j,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right]$ and $k \in [0, m-1]$. $\theta_{j,i}$ is the angle between the vector $\mathbf{w}_j$ and $\mathbf{x}_i$. m is an integer that controls the classification margin. Softmax loss is a special case ($m = 1$) of large margin softmax.

3.3.3. Angular softmax loss

Compared with large margin softmax loss, angular softmax loss [34] normalizes the weights ($\|\mathbf{w}_j\|$) and zeros the bias ($b_j = 0$). Thus, the angular softmax loss is formulated as:

$$L_{as} = -\log\left(\frac{e^{\|\mathbf{x}_i\|\phi(\theta_{y,i})}}{e^{\|\mathbf{x}_i\|\phi(\theta_{y,i})} + \sum_{j=1, j\neq y}^{m} e^{\|\mathbf{x}_i\|\cos(\theta_{j,i})}}\right), \tag{7}$$

in which $\phi(\theta_{j,i})$ is defined the same as large margin softmax loss.

3.3.4. Center loss

Center loss [32] focuses to minimize the intra-class variations. The loss function can be expressed as:

$$L_c = \frac{1}{2}\| \mathbf{x}_i - \mathbf{c}_y \|^2. \tag{8}$$

This loss function penalizes the distances between each sample and its class center.

3.3.5. Combinations with center invariant loss

We study different combinations of center invariant loss with different other terms. First, compared with CenterFace [32] that have

softmax loss and center loss. We add center invariant loss into the objective function, the overall loss becomes:

$$L = L_s + \gamma L_I + \lambda L_c \tag{9}$$

$$= -\log\left(\frac{e^{\mathbf{w}_y^T\mathbf{x}_i+b_y}}{\sum_{j=1}^{m} e^{\mathbf{w}_j^T\mathbf{x}_i+b_j}}\right) + \frac{\gamma}{4}\left(\| \mathbf{c}_y \|_2^2 - \frac{1}{m}\sum_{k=1}^{m} \| \mathbf{c}_k \|_2^2\right)^2 + \frac{\lambda}{2} \| \mathbf{x}_i - \mathbf{c}_y\|^2, \tag{10}$$

where λ is a scalar to balance the center loss and γ is a scalar to balance the center invariant loss. When λ and $\gamma = 0$, this joint loss becomes identical to the original softmax loss. The objective function (Eq. (9)) consists of three parts: softmax loss, center invariant loss and center loss. Softmax loss (L_s) aims to maximize the inter-class variations. Center invariant loss regularizes each class to be treated equally given imbalance data. These two losses both handle inter-class dispersion. Meanwhile, center loss (L_c) tries to minimize the intra-class variations, which is also essential to learn discriminative features.

Further, we replace the softmax loss with large margin softmax loss, Eq. (9) becomes:

$$L = L_{ls} + \gamma L_I + \lambda L_c \tag{11}$$

$$= -\log\left(\frac{e^{\|\mathbf{w}_y\|\|\mathbf{x}_i\|\phi(\theta_{y,i})}}{e^{\|\mathbf{w}_y\|\|\mathbf{x}_i\|\phi(\theta_{y,i})} + \sum_{j=1,j\neq y}^{m} e^{\|\mathbf{w}_j\|\|\mathbf{x}_i\|\cos(\theta_{j,i})}}\right) + \frac{\gamma}{4}\left(\| \mathbf{c}_y \|_2^2 - \frac{1}{m}\sum_{k=1}^{m} \| \mathbf{c}_k \|_2^2\right)^2 + \frac{\lambda}{2} \| \mathbf{x}_i - \mathbf{c}_y\|^2 \tag{12}$$

As demonstrated in [34], angular loss is better than softmax loss + center loss, thus the softmax and center loss in Eq. (9) is then replaced by the angular softmax loss. The objective function can be written as:

$$L = L_{as} + \gamma L_I \tag{13}$$

$$= -\log\left(\frac{e^{\|\mathbf{x}_i\|\phi(\theta_{y,i})}}{e^{\|\mathbf{x}_i\|\phi(\theta_{y,i})} + \sum_{j=1,j\neq y}^{m} e^{\|\mathbf{x}_i\|\cos(\theta_{j,i})}}\right) + \frac{\gamma}{4}\left(\| \mathbf{c}_y \|_2^2 - \frac{1}{m}\sum_{k=1}^{m} \| \mathbf{c}_k \|_2^2\right)^2. \tag{14}$$

3.4. Optimization

In this subsection, we calculate the gradients of loss function for backward operation using stochastic gradient descent (SGD). The key terms in objective function is the loss respective to $\mathbf{x}_i$ in L_I. We first make a notation:

$$\tau = \frac{1}{m}\sum_{k=1}^{m} \| \mathbf{c}_k \|_2^2 . \tag{15}$$

Taking Eq. (15) into Eq. (3), we get:

$$L_I = \frac{1}{4}\left(\| \mathbf{c}_y \|_2^2 - \tau\right)^2. \tag{16}$$

According to the chain rule, gradients of L_I with respect to $\mathbf{x}_i$ $\frac{\partial L_I}{\partial \mathbf{x}_i}$ can be written as:

$$\gamma\frac{\partial L_I}{\partial \mathbf{x}_i} = \gamma\frac{\partial L_I}{\partial \mathbf{c}_y}\frac{\partial \mathbf{c}_y}{\partial \mathbf{x}_i}. \tag{17}$$

$\frac{\partial L_I}{\partial \mathbf{c}_y}$ can be calculated as:

$$\frac{\partial L_I}{\partial \mathbf{c}_y} = \frac{1}{2}\left(\| \mathbf{c}_y \|_2^2 - \tau\right)\left(2\mathbf{c}_y - \frac{2}{m}\mathbf{c}_y\right) = \left(\| \mathbf{c}_y \|_2^2 - \tau\right)\left(1 - \frac{1}{m}\right)\mathbf{c}_y. \tag{18}$$

Due to Eq. (2), we have:

$$\frac{\partial \mathbf{c}_y}{\partial \mathbf{x}_i} = \frac{1}{n_y}. \tag{19}$$

Taking Eqs. (18) and (19) into Eq. (17), we get:

$$\gamma\frac{\partial L_I}{\partial \mathbf{x}_i} = \gamma\left(\| \mathbf{c}_y \|_2^2 - \tau\right)\left(1 - \frac{1}{m}\right)\mathbf{c}_y\frac{1}{n_y} = \gamma\frac{1}{n_y}\left(1 - \frac{1}{m}\right)\left(\| \mathbf{c}_y \|_2^2 - \tau\right)\mathbf{c}_y. \tag{20}$$

From Eq. (20), we can see that the gradients of center invariant loss with respect to $\mathbf{x}_i$ follow either the direction or the opposite direction of the center $\mathbf{c}_y$. When the Euclidean norm of center is smaller than the mean center norm $\left(\| \mathbf{c}_y \|_2^2 - \tau \leq 0\right)$, $\frac{\partial L_I}{\partial \mathbf{x}_i}$ has the opposite direction of $\mathbf{c}_y$. It means the center $\mathbf{c}_y$ will be pulled out with $\mathbf{x}_i$ since the loss function is minimized. Similarly, the center $\mathbf{c}_y$ will be pushed in if Euclidean norm of center is greater than the mean center norm $\left(\| \mathbf{c}_y \|_2^2 - \tau \geq 0\right)$. Given these gradients, we can optimize the CNN model with stochastic gradient descent (SGD).

4. Experiments

In this section, we firstly describe implementation details about data preprocessing, training and testing. Then we show how the center invariant loss influences the distribution of deeply learned features. And we investigate the sensitiveness of the parameters on the face recognition problem. We also compare our model with other state-of-the-art method and illustrate the effects in feature learning both visually and quantitatively.

4.1. Implementation details

In this subsection, we describe data preprocessing, training/test data split and detailed settings in training CNNs.

4.1.1. Data preprocessing

All faces and landmarks are detected by Mutli-task Cascaded Convolutional Networks (MTCNN) [42]. We use five landmarks (two eyes, nose and mouth corners) for similarity transformation. These faces are cropped to 112×96 RGB images. Each pixel is normalized by subtracting 127.5 then dividing by 128.

4.1.2. Training test data

We use the CASIA-WebFace [26] dataset for training CNN models with 0.5 million faces. After removing the faces which fail to be detected by MTCNN, the CASIA-WebFace data set has 0.45 million faces with 10,575 identities.

The CNN models are evaluated on two famous face benchmarks:

Label Face in the Wild (**LFW**) [27][1] contains 13,233 web images from 5749 different identities, with large variations in pose, expression and illuminations. We follow the standard protocol of unrestricted with labeled outside data and test on 6000 face pairs and report the performance. Some faces are shown in Fig. 4. Positive

[1] http://vis-www.cs.umass.edu/lfw/.

Fig. 4. Face images in LFW and YTF datasets.

samples are at the first row and negative samples are shown in the second row.

Youtube Face Database (**YTF**) [37][2] consists of 3425 videos of 1595 different identities, with an average of 2.15 videos per person. Clip durations vary from 48 frames to 6070 frames. An average length is 181.3 frames. Again, we follow the unrestricted with labeled outside data protocol and report results on 5000 video pairs. Some face images of two different people are shown in the last two rows in Fig. 4.

4.1.3. Detailed settings in CNNs

The CNN model is a 22-layer residual network[3] implemented using Caffe [43] library. For a fair comparison, we train two kinds of models under supervision of softmax loss and softmax + center loss. The two models are trained with batch size of 256 on four PASCAL Titan X GPUs. The batches are selected randomly from all training images. 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. The training takes roughly 3 h. For the model with new loss function (Eq. (9)), it is fine-tuned from the softmax + center loss model with the same data. The learning rate is fixed to 0.001 and the fine-tuning takes 20K iterations with roughly 2 h.

The classification layer cannot be used directly since the classification layer is used to recognize the 10,575 people but is not used for face verification that tells if two images come from the same person. After the CNN models finish training, the deep features are taken from the last hidden layer. We extract features for each face and its horizontally flipped image. Then the features are concatenated as the representation, which follows [32]. The similarity score between one pair of images is computed by the Cosine Distance after PCA. The performance without PCA is slightly worse than that using PCA. Here is the recognition pipeline. We first learn deep face representations by building a classifier to recognize 10,575 people. Then the classifier is removed and we take the output of the layer before the last classification layer as the deep features. When recognizing two faces, we first extract features from two faces separately and apply PCA to the features. The cosine distance is then used as the similarity measurement to decide if the two faces are from the same person.

4.2. Experiments on MNIST

In this subsection, we conduct experiments to illustrate how the γ influences the distribution of the deeply learned features on the MNIST dataset. The loss that is used in this subsection is Eq. (5). We take the 200 samples setting in Section 3.1. In Fig. 5, we can observe that different γ leads to different deep feature distributions. With proper γ, the feature space is almost equally split by all classes, even if there is a huge difference in the number of training samples (e.g. 200 vs 6000). With the increase of γ, the penalty of differences among class center Euclidean norms gets larger and larger. As shown in Fig. 5a, b, c and d, all centers are aligned to have more and more similar magnitudes and better separation of the feature space with the increase of γ. From Fig. 5e, f, g and h for the deeply learned features on the test set, we can also see the same phenomenon. We further compare the classification accuracy between the softmax loss and softmax loss + cia loss, shown in Table 1. The result of softmax loss + cia loss boosts the performance for an average of 1.1% when varying the γ from 5e-7 to 1e-5, which is robust and stable. Thus, center invariant loss could help deeply learned features to separate the feature space equally for all classes given the imbalanced training data and improve the classification performance.

We further conduct experiments with various losses illustrated in Section 3.3 on two different settings. The first is with 1000 unbalanced samples and the other is with full training samples. The learned deep features are shown in Figs. 6 and 7. Different loss functions have their corresponding effect in learning deep representations. Center loss reduces the intra-class variations. Large margin softmax and angular loss promote the separability between classes. Adding cia loss helps alleviate the unbalanced data problem and even can help with balanced data. The combination of softmax + cia loss has been already studied and is not shown in the two figures.

[2] https://www.cs.tau.ac.il/~wolf/ytfaces/.
[3] https://github.com/ydwen/caffe-face.

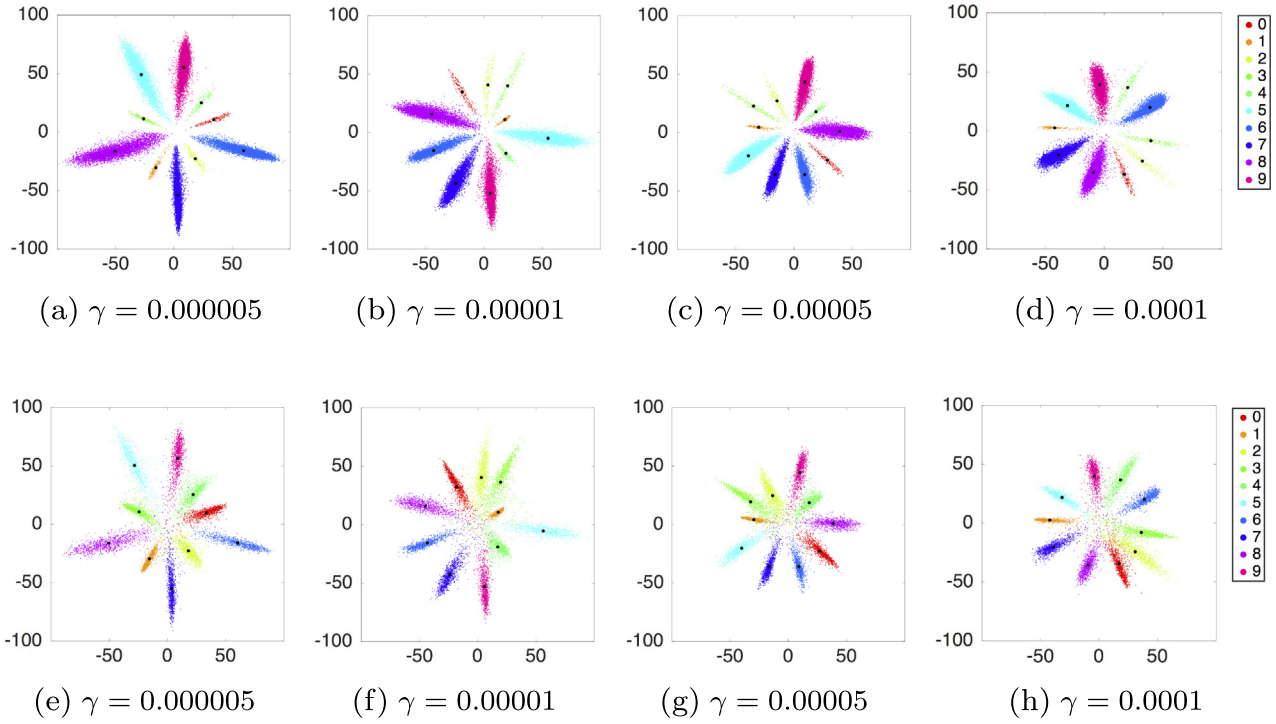


Fig. 5. Distributions of deeply learned features with different γ on train/test sets. Centers of each class are denoted as black dots. Will increasing of the γ, centers of each class get aligned gradually.

4.3. Experiments on parameters

In this subsection, we investigate the two hyperparameters λ and γ in Eq. (9) on LFW datasets. λ is for the intra-class variations in Eq. (8) and γ is for the variations between centers in Eq. (3).

We first investigate the sensitiveness of γ given fixed $\lambda = 0.008$. γ is varied from 0 to 0.02. The verification accuracies of models with different γ on LFW dataset are shown in Fig. 8a. We can observe that accuracy difference goes up from 0 to 0.001, which demonstrates that our method could boost performance. And we can see that our model is stable across a wide range of γ. When γ goes to 0.02, accuracies drop since models overfit the data.

The sensitivity of λ was initially investigated in [32]. However, since we introduce a new term called the center invariant loss (Eq. (3)), we re-investigate the sensitiveness of λ given fixed $\gamma = 0.001$. Results on LFW are shown in Fig. 8b. We can see that a proper λ is important for a good performance. Smaller λ is not a good choice since it cannot reduce large intra-class variations. A wide range of λ can also make the model stable. Large λ, e.g. 0.05, leads to overfitting and decreases the accuracy.

4.4. Results on LFW and YTF

In this subsection, we evaluate our model on LFW and YTF face recognition benchmarks. γ is fixed to 0.001 and λ is set to 0.008 in all objective functions, e.g. Eqs. (9), (11) and (13).

Table 1
Accuracy (%) of classification with softmax loss + cia loss on the MNIST test set with 200 imbalance training data in terms of different γ. When $\gamma = 0$, softmax loss + cia loss is identical to the softmax loss.

γ	0 (softmax loss)	5e-7	1e-6	5e-5	1e-5
Accuracy (%)	93.95	95.29	95.03	94.88	95.03

In Table 2, we compared our method against many state-of-the-art methods and baseline models. From the results, we can see that our model beats the centerface [32], large margin softmax [33] and sphereface [34] under the same conditions (same training data and the network structure). The performance is improved from 99.03% to 99.12% on LFW and from 93.82% to 93.88% on YTF compared with centerface. Compared with large margin softmax [33], adding center loss (Eq. (8)) could improve the results from 97.56% to 99.13% on LFW and from 93.78% to 93.82 on YTF. With center invariant loss, the results could be further improved to 99.26% on LFW and 93.82% on YTF. Compared with sphereface [34] model that is the best performance among centerface and larger margin softmax, our center invariant loss could still improve the results from 99.18% to 99.28% on LFW and from 94.48% to 94.92% on YTF. This shows that the proposed center invariant loss can improve the generalization ability of deeply learned features. Compared with other models [26,29,33,47,48] trained on CASIA-WebFace, our model achieves better result on LFW. This demonstrates the advantage of our model.

Compared with sphereface [34], our result is worse on LFW because the network structure we used is a 22-layer ResNet while 64-layer ResNet in [34]. But we achieve comparable results on YTF, with our results 94.92% compared with 95.0% in [34]. For the SphereFace Retrain model with the 22 layers, we get 99.18% accuracy. The most similar model in [34] is a 20-layer model, which has 99.26% accuracy. Both models have similar number of layers and are built based on the residual block. However, we use the plain residual network, which uses the pooling layer to reduce the size of feature maps. While the model SphereFace utilizes a different network structure, it has been proven to have better performance compared with plan residual network. For a fair comparison in terms of loss functions, we use the same network structure that is the plain residual network. For another re-trained model LMS [33], we use the same network structure but change the loss term. And though we tried our best to tune the parameters in training LMS, we cannot achieve the best results in LMS [33]. This is also might be due

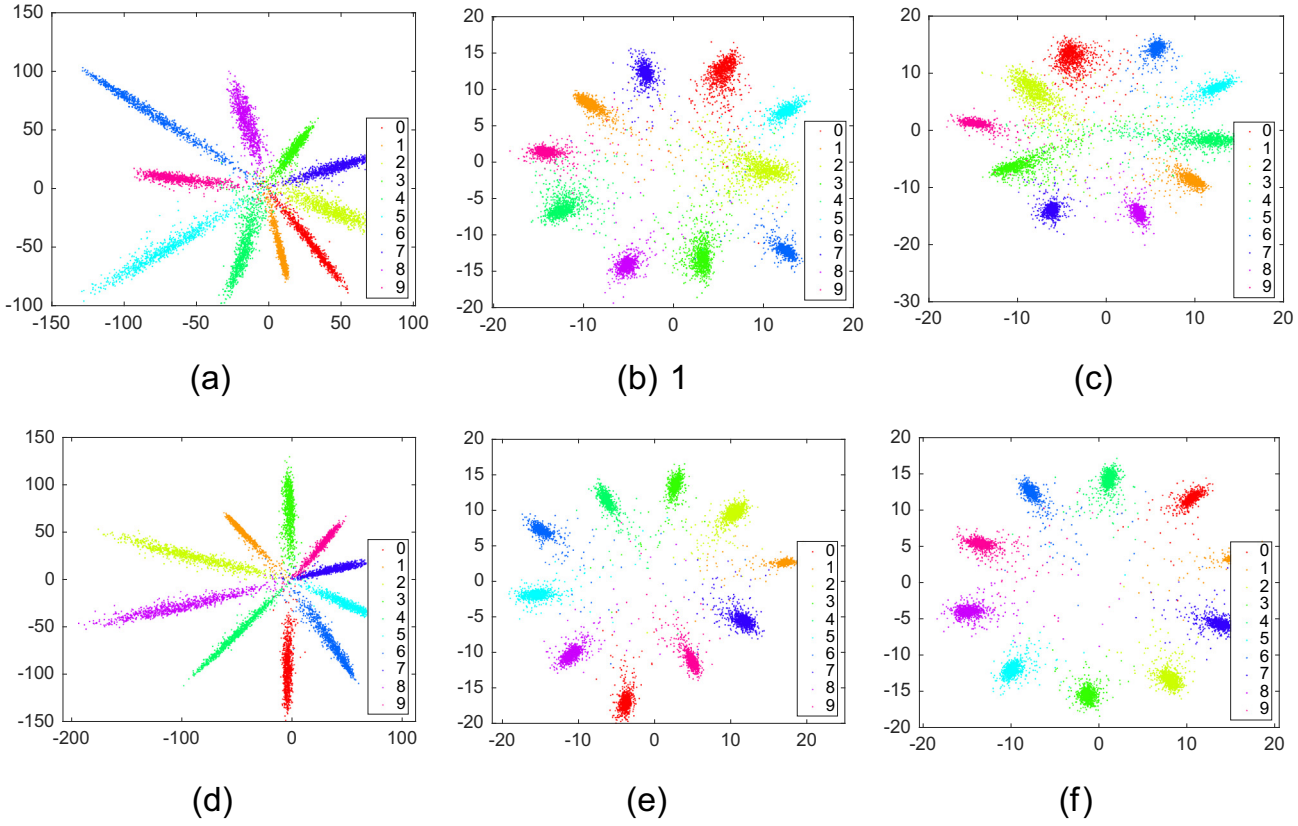


Fig. 6. Distributions of deeply learned features with different training loss functions on test sets: (a)–(c) results of softmax losss, softmax loss + center loss and softmax loss + center loss + cia loss on the 1000 training setting, (d)–(f) results of softmax losss, softmax loss + center loss and softmax loss + center loss + cia loss on the full training setting.

to the different network structure. Another reason might be that we use smaller learning rate in training LMS since we found that their loss function is easy to diverge with large learning rate. However, the combination of LMS, center loss and cia loss with the accuracy 99.26% beats LMS with the accuracy 98.06% even if the retrained LMS has 97.56% accuracy.

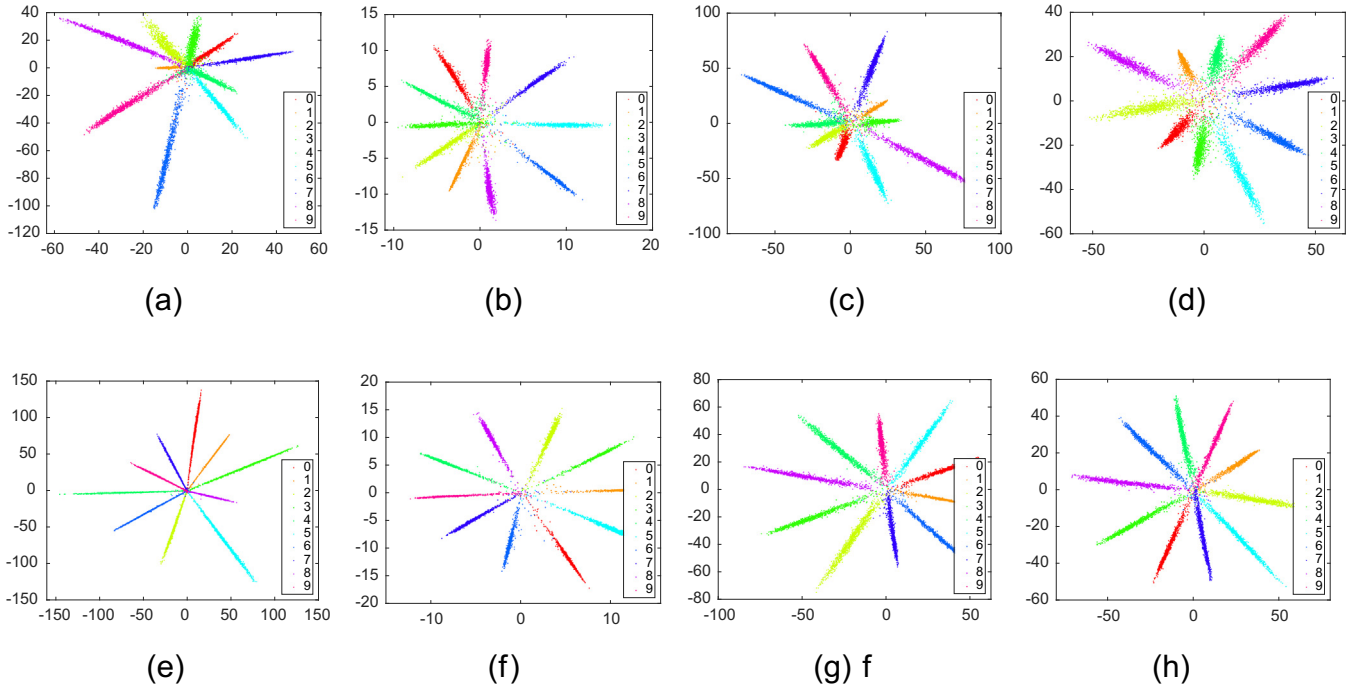
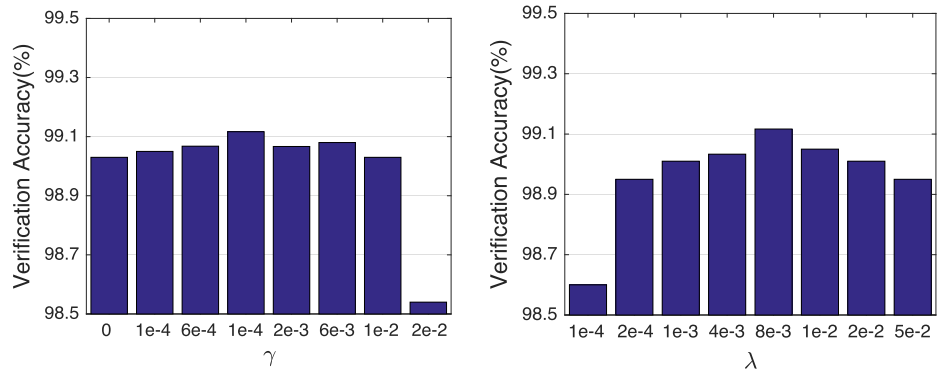


Fig. 7. Distributions of deeply learned features with different training loss functions on test sets: (a)–(d) Results of largemargin loss, largemargin + cia loss, angular loss and angular loss + cia loss on the 1000 training setting, (e)–(h) results of largemargin loss, largemargin + cia loss, angular loss and angular loss + cia loss on the full training setting.



(a) Different γ was tested given fixed $\lambda = 0.08$. (b) Different λ was tested given fixed $\gamma = 0.001$.

Fig. 8. Face verification accuracies on LFW dataset with parameters a) γ and b) λ.

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

Method	#Images/dataset	#Net	LFW	YTF
DeepFace [21]	4 M	3	97.35%	91.40%
FaceNet [25]	200 M	1	99.63%	95.10%
DeepID-2+ [24]	-	1	98.70%	-
DeepID-2+ [24]	-	25	99.47%	93.20%
VGG Face [44]	2.6 M	1	98.95%	92.80%
MFM [45]	8.9 M	1	98.80%	93.40%
Web-Scale [46]	500 M	1	98.00%	-
Web-Scale [46]	500 M	4	98.37%	-
CenterFace [32]	0.7 M	1	99.28%	94.90%
Rangeloss [30]	1.5 M	1	99.52%	93.70%
LFS [26]	CASIA-WebFace	1	97.73%	92.24%
FSS [47]	CASIA-WebFace	1	97.45%	-
MMD [48]	CASIA-WebFace	1	98.43%	-
PSEA [29]	CASIA-WebFace	1	98.06%	-
LMS [33]	CASIA-WebFace	1	98.71%	-
Sphereface [34]	CASIA-WebFace	1	99.42%	95.0%
Softmax	CASIA-WebFace	1	97.33%	90.88%
+ CIA Loss	CASIA-WebFace	1	97.85%	91.32%
+ CenterFace [32] Retrain	CASIA-WebFace	1	99.03%	93.82%
+ CenterFace + CIA Loss	CASIA-WebFace	1	99.12%	93.88%
LMS [33] Retrain	CASIA-WebFace	1	97.56%	91.12%
+ CenterFace	CASIA-WebFace	1	99.13%	93.78%
+ CenterFace + CIA Loss	CASIA-WebFace	1	99.26%	93.82%
SphereFace [34] Retrain	CASIA-WebFace	1	99.18%	94.48%
+ CIA Loss	CASIA-WebFace	1	99.28%	94.92%

Moreover, our model is trained with 0.45 million faces, which are much less than many other state-of-the-art models [21,25,44,46,30]. But we achieve comparable results, which show the advantages of the proposed model.

5. Conclusion

In this paper, we proposed a center invariant loss which aligns the center of each person to enforce the learned feature to have a good representation for all people with better generalization ability. The center invariant loss penalized the difference between each center of classes. With jointly training the center invariant loss, the generalization ability of deeply learned face representation under multiple state-of-the-art supervision signals could be further improved. Extensive experiments demonstrated the effectiveness of the proposed approach. The center invariant loss aims to regulate the distribution of the clusters belonging to different classes uniformly around a center, which alleviates the training difficulty caused by the data imbalance. The proposed method would benefit the problem that the clusters are assumed to be uniformly distributed. For some problems that the data distribution comes with different priors, it might has some different observations. We would leave this part to the future work. In the future, we would try to apply our model to other applications, such as, object detection, object recognition, scene classification, one-shot learning and low-shot learning.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 2012. pp. 1097–1105.
[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, Proceedings of the IEEE Conference on CVPR, 2016.
[3] J. Yu, Y. Jiang, Z. Wang, Z. Cao, T. Huang, UnitBox: an advanced object detection network, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 516–520.
[4] R.K. Sarvadevabhatla, S. Surya, S.S. Kruthiventi, et al. SwiDeN: convolutional neural networks for depiction invariant object recognition, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 187–191.

[5] W. Luo, J. Li, J. Yang, W. Xu, J. Zhang, Convolutional sparse autoencoders for image classification, IEEE Trans. Neural Netw. Learn. Syst. 29 (7) (2018) 3289–3294.
[6] J. Li, T. Zhang, W. Luo, J. Yang, X. Yuan, J. Zhang, Sparseness analysis in the pretraining of deep neural networks, IEEE Trans. Neural Netw. Learn. Syst. 28 (6) (2017) 1425–1438.
[7] J. Li, H. Chang, J. Yang, Sparse deep stacking network for image classification, Proc. of the AAAI Conf. on Artif. Intell., 2015. pp. 3804–3810.
[8] Y. Wang, J. Liu, Y. Li, J. Yan, H. Lu, Objectness-aware semantic segmentation, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 307–311.
[9] S. Jiang, Y. Wu, Y. Fu, Deep bi-directional cross-triplet embedding for cross–domain clothing retrieval, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 52–56.
[10] Y. Li, X. Kong, L. Zheng, Q. Tian, Exploiting hierarchical activations of neural network for image retrieval, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 132–136.
[11] L. Cabrera-Quiros, H. Hung, Who is where?: matching people in video to wearable acceleration during crowded mingling events, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 267–271.
[12] S. Yamamoto, T. Harada, Video generation using 3D Convolutional Neural Network, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 576–580.
[13] V. Vonikakis, R. Subramanian, S. Winkler, Shaping datasets: optimal data selection for specific target distributions across dimensions, Image Processing (ICIP), 2016 IEEE International Conference on, IEEE. 2016, pp. 3753–3757.
[14] Y. Dong, Y. Wu, Adaptive cascade deep convolutional neural networks for face alignment, Comput. Stand. Interfaces 42 (2015) 105–112.
[15] H. Mao, Y. Wu, J. Li, Y. Fu, Super resolution of the partial pixelated images with deep Convolutional Neural Network, Proceedings of the 2016 ACM on Multimedia Conference, ACM. 2016, pp. 322–326.
[16] S. Jiang, M. Shao, C. Jia, Y. Fu, Consensus style centralizing auto-encoder for weak style classification, AAAI, 2016. pp. 1223–1229.
[17] Z. Ding, M. Shao, Y. Fu, Deep robust encoder through locality preserving low-rank dictionary, European Conference on Computer Vision, Springer. 2016, pp. 567–582.
[18] P.J. Phillips, H. Wechsler, J. Huang, P.J. Rauss, The FERET database and evaluation procedure for face-recognition algorithms, Image Vis. Comput. 16 (5) (1998) 295–306.
[19] J. Hu, Discriminative transfer learning with sparsity regularization for single-sample face recognition, Image Vis. Comput. 60 (2017) 48–57.
[20] W. Huang, H. Yin, On nonlinear dimensionality reduction for face recognition, Image Vis. Comput. 30 (4) (2012) 355–366.
[21] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 1701–1708.
[22] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014. pp. 1891–1898.
[23] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, Advances in Neural Information Processing Systems, 2014. pp. 1988–1996.
[24] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 2892–2900.
[25] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: a unified embedding for face recognition and clustering, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 815–823.
[26] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, arXiv preprint. arXiv:1411.7923.
[27] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Tech. rep., Technical Report 07-49, Technical Report 07-49, University of Massachusetts, Amherst, 2007.
[28] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, Ms-celeb-1m: a dataset and benchmark for large-scale face recognition, European Conference on Computer Vision, Springer. 2016, pp. 87–102.
[29] I. Masi, A.T. Trân, T. Hassner, J.T. Leksut, G. Medioni, Do we really need to collect millions of faces for effective face recognition? European Conference on Computer Vision, Springer. 2016, pp. 579–596.
[30] X. Zhang, Z. Fang, Y. Wen, Z. Li, Y. Qiao, RangeLoss for Deep Face Recognition with Long-tail, arXiv preprint. arXiv:1611.08976.
[31] B. Hariharan, R. Girshick, Low-shotvisual object recognition, arXiv preprint. arXiv:1606.02819.
[32] Y. Wen, K. Zhang, Z. Li, Y. Qiao, A discriminative feature learning approach for deep face recognition, European Conference on Computer Vision, Springer. 2016, pp. 499–515.
[33] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for Convolutional Neural Networks., ICML, 2016. pp. 507–516.
[34] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, SphereFace: Deep Hypersphere Embedding for Face Recognition, arXiv preprint. arXiv:1704.08063.
[35] L. Guo, L. Zhang, One-shot Face Recognition by Promoting Underrepresented Classes, arXiv preprint. arXiv:1707.05574.
[36] Y. Wu, H. Liu, J. Li, Y. Fu, Deep face recognition with center invariant loss, ACM MM Thematic Workshop, 2017.
[37] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. 2011, pp. 529–534.
[38] S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back, Face recognition: a convolutional neural-network approach, IEEE Trans. Neural Netw. 8 (1) (1997) 98–113.
[39] M.K. Titsias, One-vs-each approximation to softmax for scalable estimation of probabilities, Advances In Neural Information Processing Systems, 2016. pp. 4161–4169.
[40] S. Vijayanarasimhan, J. Shlens, R. Monga, J. Yagnik, Deep networks with large output spaces, International Conference on Learning Representations, 2015. http://arxiv.org/abs/1412.7479.
[41] Y. LeCun, C. Cortes, C.J. Burges, The MNIST Database of Handwritten Digits, 1998.
[42] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process Lett. 23 (10) (2016) 1499–1503.
[43] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, Proceedings of the 22nd ACM International Conference on Multimedia, ACM. 2014, pp. 675–678.
[44] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition., BMVC, vol. 1, 2015. pp. 6.
[45] X. Wu, R. He, Z. Sun, T. Tan, A Light CNN for Deep Face Representation with Noisy Labels, arXiv preprint. arXiv:1511.02683.
[46] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Web-scale training for face identification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. pp. 2746–2754.
[47] D. Wang, C. Otto, A.K. Jain, A Light CNN for Deep Face Representation with Noisy Labels, arXiv preprint. arXiv:1507.07242.
[48] C. Ding, D. Tao, Robust face recognition via multimodal deep face representation, IEEE Trans. Multimedia 17 (11) (2015) 2049–2058.