

Received December 23, 2019, accepted February 6, 2020, date of publication February 12, 2020, date of current version February 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973433

\mathcal{T} -Center: A Novel Feature Extraction Approach Towards Large-Scale Iris Recognition

YIFENG CHEN¹, CHENG WU¹, (Member, IEEE), AND YIMING WANG¹, (Member, IEEE)

School of Rail Transportation, Soochow University, Suzhou 215137, China

Corresponding author: Cheng Wu (cwu@suda.edu.cn)

This work was supported by the scientific research project funding of Suzhou Rail Transit Group Company, Ltd., under Grand SZZG06YJ6000017.

ABSTRACT For large-scale iris recognition tasks, the determination of classification thresholds remains a challenging task, especially in practical applications where sample space is growing rapidly. Due to the complexity of iris samples, the classification threshold is difficult to determine with the increase of samples. The key issue to solving such threshold determination problems is to obtain iris feature vectors with more obvious discrimination. Therefore, we train deep convolutional neural networks based on a large number of iris samples to extract iris features. More importantly, an optimized center loss function referred to Tight Center (\mathcal{T} -Center) Loss is used to solve the problem of insufficient discrimination caused by the traditional Softmax loss function. In order to evaluate the effectiveness of our proposed method, cosine similarity is used to estimate the similarity between the features on the published iris recognition datasets ND-IRIS-0405, CASIA-Thousand and IITD. Our experiment results prove that the \mathcal{T} -Center loss can minimize intra-class variance and maximize inter-class variance, which achieve significant performance on the benchmark experiments.

INDEX TERMS Biometric, iris recognition, large-scale dataset, softmax loss, \mathcal{T} -center loss.

I. INTRODUCTION

Iris recognition is one of the most promising fields in biometrics. The first complete and automated iris recognition system was presented by Daugman [1]. Over the past few years, conventional iris recognition under homogeneous and controlled conditions has been extensively studied. The general procedures for iris recognition include four parts: iris location, iris segmentation, feature encoding and feature matching. Recently, more attention has been paid on the realization of large-scale iris recognition tasks with massive sample space. For massive iris samples, the ratio of non-ideal captured images and the probability of coming from different acquisition devices have increased significantly. Unfortunately, the non-ideal captured images increase the difficulty of iris segmentation and extraction and cross-device picture sources lead to adaptive trapping of parameters. In addition, the large-scale image dataset itself also brings about the problem of classification threshold determination, which has inevitably become a difficulty in

pattern classification. Therefore, it is still a challenging task to design robust feature extraction methods to cope with the complex intra-class changes of iris images in non-ideal uncontrollable acquisition environments and cross-system acquisitions.

As we know, the construction of features vectors used to code the iris pattern definitely influence the complexity of the learning methods and their performance. Unfortunately, iris patterns require relatively complex feature vectors, even if their size can be optimized [2]. More feature extraction functions have been proposed to solve the problem of iris coding [3], [4] since Daugman proposed 2D-Gabor filters for iris feature coding. Most of the early works are based on hand-crafted features. Some other researchers propose the corresponding optimization algorithms such as Particle Swarm Optimization [5] and Ant Colony Optimization [6] in order to estimate adaptive filter parameters, which are expected to achieve great performance on the public dataset. However, it is still hard to characterize complex iris texture features in practical applications due to the limitation on feature representation using shallow architectures. Moreover, it is more difficult to reproduce techniques and experiments

The associate editor coordinating the review of this manuscript and approving it for publication was Kim-Kwang Raymond Choo¹.

due to the lack of either sufficient implementation details or reliable shared codes.

Recent progress in deep learning, in particular, deep Convolutional Neural Networks (CNNs) have significantly improved the state-of-the-art performance for a wide variety of computer vision tasks, which makes deep CNNs a dominant machine learning approach for computer vision [7], [8]. Deep learning based approaches in iris biometrics, have been explored in recent works. Liu proposed a deep framework for iris verification, which learns relational features to measure the similarity between pairs of iris images [9]. Gangwar [10] proposed a deep learning based method for iris learning and various optimal tricks were used to avoid overfitting. Raja proposed multi-patch deep features using deep sparse filters to obtain robust features [11]. Zhao *et al.* [12] proposed a deep learning method based on the capsule network architecture in iris recognition. These typical researches have an interesting common point. That is, all deep models are trained under the supervision of the Softmax loss. The advantage of Softmax loss is to make the learned features highly separable, but it does not guarantee the features' high discrimination. Unfortunately, higher discrimination is the key to robustly identify new unseen classes without label prediction. Further, some more recent studies found that traditional Softmax is insufficient to maximize the discrimination power for classification for large-scale iris samples [13], [14]. Not only that, our experimental evaluation in Section IV also proved the Equal Error Rate of deep learning model supervised by Softmax loss [9] is similar to traditional methods.

In order to address this issue, Zhao *et al.* [15] proposed a well-designed model referred as UniNet and a specially designed Extended Triplet Loss (ETL) function is used to incorporate the bit-shifting and non-iris masking. Wang and Kumar [16] proposed a residual network with dilated convolutional kernels to optimize the training process based on ETL. However, the proposed loss function is time-consuming and require carefully designed triplet mining procedure. Thus, we propose a novel \mathcal{T} -Center loss in order to reduce computing expenses and enhance the discriminative power of the deeply learned features. The original center loss function fuse the Euclidean distance between the features and features centers into loss function to maximize the inter-class variance and minimize the intra-class variance [17], which is essentially a kind of distance metric learning. However for iris recognition task, we care more about the angular metric rather than distance metric since cosine distance between two features is used to compute the similarity score. Therefore, we redefine the loss function using L2-norm. To demonstrate the applicability of the deep features based \mathcal{T} -Center loss, we present the results by plotting Cumulative Match Characteristic (CMC) and Receiver Operating Characteristic (ROC) curves. Through the set of extensive experiments on ND-IRIS-0405, CASIA-IrisV4 and IITD2.0, we assert the intuition of robust feature representation, which gets high True Accept Rate ($TAR > 97\%$) with low Equal Error Rate ($EER < 1\%$). Especially in a large-scale system, the proposed

method is more suitable for iris recognition which encourages the minimum intra-class variance and maximum inter-class variance.

Our main contributions are summarized as follows:

- We propose a novel loss function called \mathcal{T} -Center loss to enhance the discriminate ability of deep models, which shows significant improvements compared to previous work on the ND-IRIS-0405, CASIA-Iris-Thousand and IITD cross sensor datasets. Meanwhile, it can be shown on the iris feature histogram that the inter-class variance is greatly reduced and the loss function pulls the features of the same class to their centers.
- To avoid the gradient explosions and identify the appropriate hyperparameter, our approach simultaneously normalizes the feature vectors and feature center vectors, which optimize the original loss function.

In the rest of paper, Section II describes the whole iris verification framework. Section III proposes our novel method with the \mathcal{T} -Center loss function and Section IV gives the implementation details and experimental results. Finally, we concludes our work in Section V.

II. SYSTEM

The whole iris verification process is shown in Figure 1. Different from most of computer vision tasks using deep CNNs, the input samples of iris recognition undergo several image processing steps instead of using original images as input samples. It is known to all that the effect of iris segmentation has a greater impact on the accuracy of iris recognition. The original iris images contain much interference information, including eyelashes, pupils, and eyelid, which cannot clearly represent iris feature information without segmentation. In previous work, the normalization of the iris region is done using Daugman's rubber-sheet model to reduce the impact of pupil contraction. We acknowledge the effectiveness of iris normalization. In order to effectively evaluate the performance of the model, we take three different methods to perform image segmentation during preprocessing and final images are resized as 128×128 after preprocessing. More details about image processing can be found in Section III.

In order to decrease the computational complexity and training cost, we do not use the very deep CNN models such as InceptionV4 and Resnet. Motivated by VGG [20], we propose a tiny deep model referred as TinyVGG. We randomly adjust the image contrast, brightness, and increase distortion for data augmentation instead of using dropout due to using a tiny model. This step aims to simulate real recognition environments and avoid overfitting. Meanwhile, we have noticed that some tiny deep models maintain high performance metrics while reducing computing expenses in recent work, such as MobileNet [18] and ShuffleNet [19]. Thus, we also use the above three models to evaluate the performance of proposed algorithm. The structure of models can be found in Figure 2.

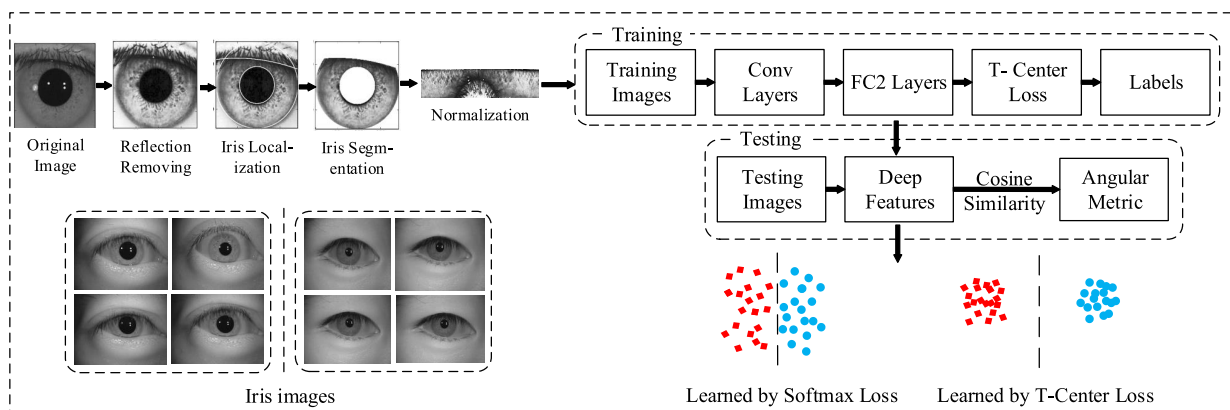


FIGURE 1. An overview of the proposed iris verification framework. At training phase, the discriminative T -Center-based iris features are learned with large margin between different classes. At testing phase, the testing data feed into framework to extract deep features which are later used to compute cosine similarity to perform iris verification.

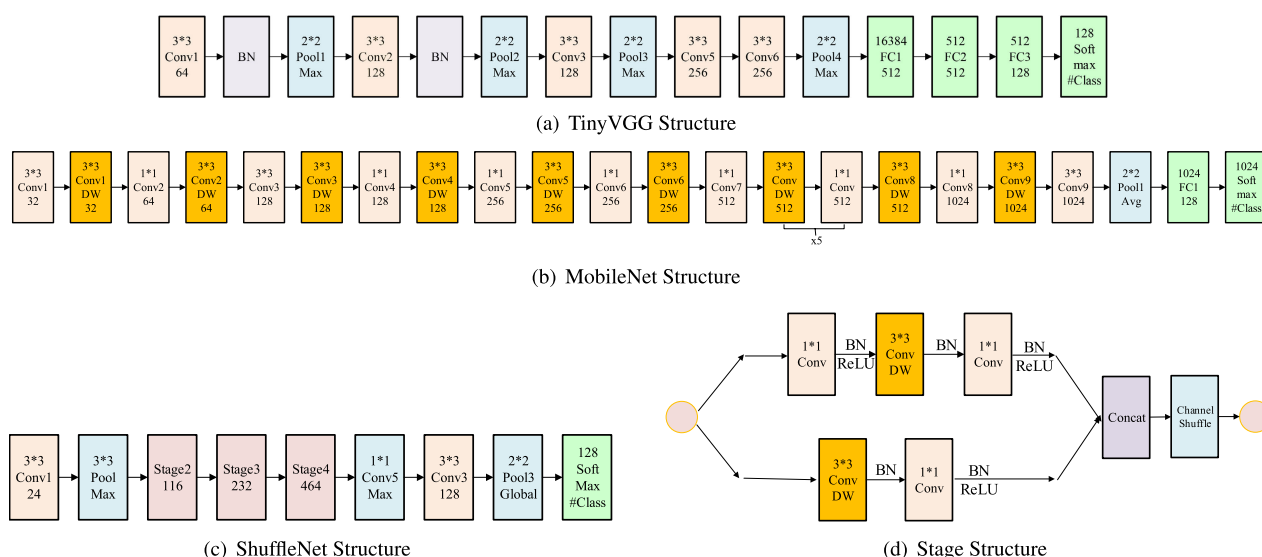


FIGURE 2. The CNN architecture used for iris recognition experiments. Each layer gives the filter size and the number of output channels. Note that Conv represents the convolution layer, Pool represents the pooling layer, FC represents the fully connected layer and DW represents the depthwise convolution layer respectively. The number of output channels in softmax layer corresponds to the number of categories. Specially, we select the last layer of FC in order to calculate the T -Center loss, which corresponds to the FC3 in TinyVGG, FC1 in MobileNet [18] and Global Pooling in ShuffleNet [19] respectively.

III. METHOD

A. PREPROCESSING

During image preprocessing, the segmentation is done followed by three different methods, including *Hough*, *Viterbi* and *Viterbi + Norm* algorithm as the figure shown in Figure 3. In Figure 3(a), we firstly remove the reflection points in iris image by using Fast Matching Algorithm [21]. The segmentation is done in 2 steps: (i) a rough localization of iris contours is performed by using Circular Hough Transform, (ii) these two circles are then used to detect pupils and eyelid following the method proposed by He *et al.* [22] in order to refine iris contours; In Figure 3(b), we closely follow the preprocessing method in OSIRISv4.0 [23]. The contours of the iris under this method correspond to an optimal path retrieved by the Viterbi algorithm for joining in an optimal

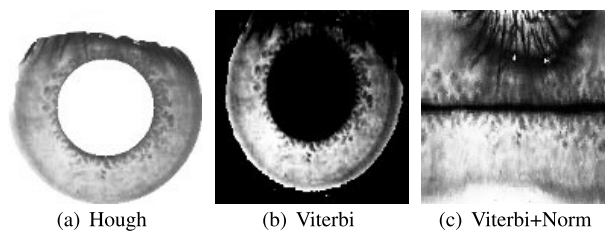


FIGURE 3. The final iris samples after preprocessing.

way so that segmentation part has been greatly improved; Notice that Daugman’s rubber sheet model can reduce the impact of pupil contraction, so that we perform normalization on upper and lower halves respectively. The result is shown in Figure 3(c).

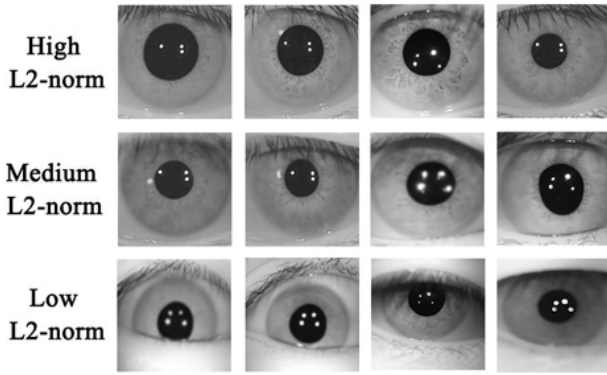


FIGURE 4. Sample images from CASIA dataset with high, medium and low L2-norm.

B. METHOD MOTIVATION

Deep learning uses the loss function to measure the degree of optimization. It is clear that designing an appropriate loss function can enhance the ability of discrimination. The most commonly used loss function is Softmax loss. Assuming that the input learned feature vectors x_i and its label y_i , the original Softmax loss function is as Equation (1):

$$L_s = - \sum_{i=1}^m \log p_i = - \sum_{i=1}^m \log \frac{e^{f_{y_i}}}{\sum_{j=1}^n e^{f_j}} \quad (1)$$

where p_i denotes the posterior probability of x_i being correctly classified, m is the size of a mini-batch, n the number of classes and f_j the j -th element of the class score vector f . f_j is usually denoted as activation of a fully-connected layer with weight vector W_j and bias b_j . We fix the bias $b_j = 0$ for simplicity and as a result f_j is given by:

$$f_j = W_j^T x = \|W_j\| \|x\| \cos \theta_j \quad (2)$$

where θ_j is the angle between W_j and x . This formula suggests that both norm and angle of vectors contribute to the posterior probability. At test time, feature descriptors x_i and x_j are extracted for the pair of test face images i, j respectively using the trained deep model, and normalized to unit length. Then, a similarity score is computed on the feature vectors which provides how close the features lie in the embedded space. If the similarity score is greater than a set threshold, the iris pairs referred as positive pairs are decided to be of the same person and the same eye. Usually, the similarity score is computed by using cosine similarity, as given by Equation (3):

$$\cos \langle x_i, x_j \rangle = \frac{x_i^T \cdot x_j}{\|x_i\| \|x_j\|} \quad (3)$$

There are two major issues with this pipeline. On one hand, the training and testing steps for iris verification task are decoupled. Training with Softmax loss doesn't necessarily ensure the positive pairs to be closer and the negative pairs to be far separated in the embedded angular space. We revisit the

softmax loss by looking into its decision criteria. In binary-case, the posterior probabilities obtained by softmax loss are:

$$p_1 = \frac{\|W_1\| \|x\| \cos \theta_1}{\|W_1\| \|x\| \cos \theta_1 + \|W_2\| \|x\| \cos \theta_2}$$

$$p_2 = \frac{\|W_2\| \|x\| \cos \theta_2}{\|W_1\| \|x\| \cos \theta_1 + \|W_2\| \|x\| \cos \theta_2} \quad (4)$$

The predicted label will be assigned to class 1 if $p_1 > p_2$ and class 2 else if $p_1 < p_2$. Note that p_1 and p_2 share the same x , the decision boundary is defined by:

$$\|W_1\| \cos \theta_1 = \|W_2\| \cos \theta_2 \quad (5)$$

Thus, its boundary depends on both magnitudes of weight vectors and cosine of angles, which results in an overlapping decision area in the embedded cosine space. As noted in our article, in the testing stage it is a common strategy to only consider cosine similarity between testing feature vectors of irises. Consequently, the trained classify testing samples. To encourage better discriminating performance, many research studies have been carried out. For example:

- **N-Softmax** [24] normalizes both weight W_1 and W_2 so that they have constant magnitude-one, which results in a decision boundary by:

$$\cos \theta_1 = \cos \theta_2 \quad (6)$$

However, it is not quite robust to noise because there is no decision margin: any small perturbation around the decision boundary can change the decision.

- **A-Softmax** [13] improves the softmax loss by performing an extra margin, so that its decision boundary is given by:

$$C_1 : \cos m\theta_1 \geq \cos \theta_2$$

$$C_2 : \cos \theta_1 \leq \cos m\theta_2 \quad (7)$$

Thus, for C_2 is requires $\theta_1 \leq \theta_2/m$, and similarly for C_2 . The comparison of decision margin under different loss functions are illustrated in Figure 5.

On the other hand, the Softmax classifier is weak in modeling difficult or extreme samples. In a typical training batch with data quality imbalance, the Softmax loss gets minimized by increasing the L2-norm of the features for easy samples, and ignoring the hard samples. In order to solve the problem, recent approach such as Center loss $L_c = \frac{1}{2} \sum_{i=1}^m \|x_i - c_{y_i}\|^2$, where c_{y_i} denotes the y_i th class center of deep features, tries to constrain the hard samples, which is one of the metric learning. This enforces to pull the feature descriptors to their feature center. However in the pre-experiment, we found that the Center loss is not as effective as expected, and somehow even not outperform than Softmax. Thus, we infer that it may respond to the quality of the iris samples. We found some low-quality images in the iris dataset as shown in Figure 4, which are blurred and occluded. These low-quality images appear even more frequently in the actual environment. We get the feature descriptors under the CNNs supervised by Center loss and then evaluate the distribution of

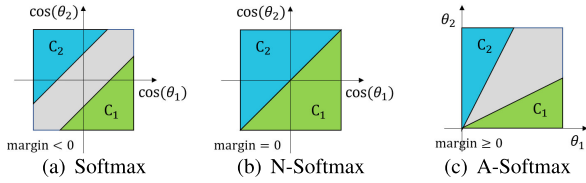


FIGURE 5. The comparison of decision margins for different loss functions the binary-classes scenarios. Gray areas are decision margins.

TABLE 1. Performance on CASIA under Center loss.

Datasets	Accuracy(%)	Angular Fisher Score
Low L2-Norm	87.39	0.3871
Medium L2-Norm	93.76	0.2613
High L2-Norm	96.03	0.2118
Total	95.21	0.2392

deep features. We found an interesting point: The low-quality iris samples usually have smaller L2-norm values. It can be explained that we regard the CNNs as multi-filters. If the edge of low-quality image is fuzzy, the value of corresponding pixel gradient will be small which leads to the small value of feature descriptors obtained by multi-filters.

To further validate this phenomenon, we perform a simple experiment on CASIA dataset which we divide the testing samples into three different sets based on the L2-norm value of feature descriptors. Samples with $L2\text{-norm} \leq 10$ are assigned to set 1. Samples with $L2\text{-norm} > 10$ but ≤ 20 are assigned to set 2, while others with $L2\text{-norm} > 20$ are assigned to set 3. Then we calculate the accuracy of different sets to evaluate the binary classification performance based on cosine similarity. Meanwhile, we use an angular Fisher score for evaluating the feature discriminativeness in angular margin feature learning. The angular Fisher score (AFS) is defined by:

$$AFS = \frac{S_w}{S_b} \tag{8}$$

where the inter-class scatter value is defined as $S_w = \sum_i \sum_{x_j \in X_i} (1 - \cos \langle x_j, m_i \rangle)$ and the intra-class scatter value is defined as $S_b = \sum_i n_i (1 - \cos \langle m_i, m \rangle)$. X_i is the i -th class samples, m_i is the mean vector of features from class i , m is the mean vector of the whole dataset, and n_i is the sample number of class i . What's more, the lower the AFS values is, the more discriminative the feature are.

As it shown in Table 1, the difference in performance between each set is quite significant, which confirms that the L2-norm of the feature descriptor is informative of its quality.

C. \mathcal{T} -CENTER LOSS

To solve these issues mentioned in Section III-B, we proposed a novel loss function referred as \mathcal{T} -Center loss. We enforce the L2-norm of the features for each iris samples. Specially, we add an L2-constraint to the features and feature centers in order to fix them on the unit embedded sphere. Thus, we have

the Equation (9) as:

$$\begin{aligned}
 L_{tc} &= \frac{1}{2} \sum_{i=1}^m \|x_i^* - c_{y_i}^*\|^2 \\
 x_i^* &= \frac{x_i}{\|x_i\|} \\
 c_{y_i}^* &= \frac{c_{y_i}}{\|c_{y_i}\|}
 \end{aligned} \tag{9}$$

where $c_{y_i}^*$ and x_i^* denote the feature center and feature vector after L2-norm respectively. For each feature x_i we can get the value of \mathcal{T} -Center loss L_i , which is given by $L_i = \frac{1}{2} \|x_i^* - c_{y_i}^*\|^2$. At training phase, we backpropagte the gradients according to the computation of L_i as the Equation (10) given below.

$$\begin{aligned}
 \frac{\partial L_i}{\partial x_i} &= \frac{\partial L_{tc}}{\partial x_i^*} \cdot \frac{\partial x_i^*}{\partial x_i} \\
 &= (x_i - c_{y_i}) \cdot \frac{\partial x_i^*}{\partial x_i} \\
 &= (x_i - c_{y_i}) \cdot \frac{\|x\| - \frac{\partial \|x_i\|}{\partial x_i}}{\|x_i\|^2} \\
 &= (x_i - c_{y_i}) \cdot \frac{\|x\|^2 - x_i^2}{\|x_i\|^3}
 \end{aligned} \tag{10}$$

Instead of updating the feature centers by back propagation algorithm, the centers are computed by averaging the features of the corresponding classes in each iteration. The update equation of c_{y_i} is computed using Equation (11):

$$\begin{aligned}
 \Delta c_j^* &= \frac{\sum_{i=1}^m \delta(y_i = j) \cdot (c_j^* - x_i^*)}{1 + \sum_{i=1}^m \delta(y_i = j)} \\
 c_j^{*t+1} &= c_j^{*t} - \alpha \cdot \Delta c_j^*
 \end{aligned} \tag{11}$$

where t is the number of iteration. The feature centers update only if y_i and j share the same class, where $\delta(\text{condition}) = 1$. A scalar α is used to control the learning rate of the centers to avoid large perturbations caused by image noise. Note that these learned features and centers would degrade to zeros when the \mathcal{T} -Center loss is used separately for deep networks. It also would lead to an overfitting model. So it is necessary to adopt the joint supervision of Softmax loss and \mathcal{T} -Center loss to train. The discrimination cannot be achieved by using any of the loss functions alone. The final loss function formulation is given in Equation (12).

$$\begin{aligned}
 L &= L_s + \lambda L_{tc} \\
 &= - \sum_{i=1}^m \log \frac{e^{W_{y_i}^T x_i^* + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i^* + b_j}} + \frac{\lambda}{2} \sum_{i=1}^m \|x_i^* - c_{y_i}^*\|^2
 \end{aligned} \tag{12}$$

where λ is a scalar used for balancing the significance of two loss functions.

D. SUBSET EXAMPLE

We study the effect of \mathcal{T} -Center loss function on the subset of dataset CASIA-IrisV4 using the DCNN mentioned in Section II where the last fully connection layer output is

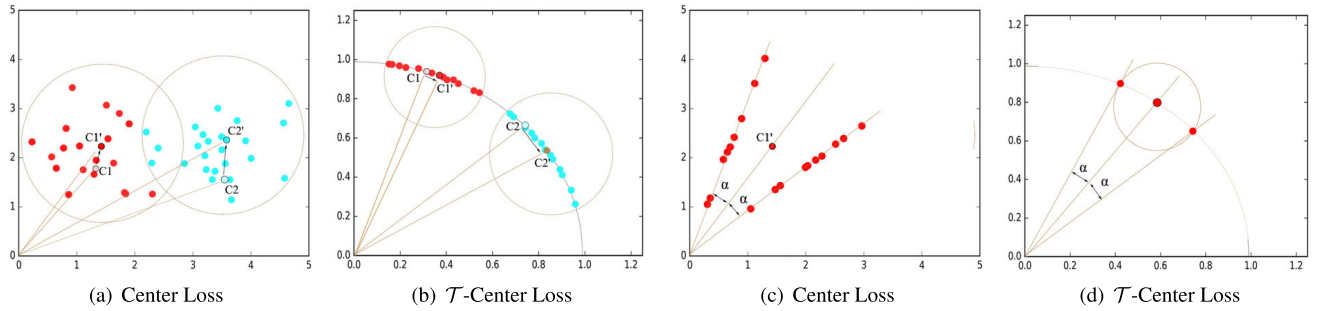


FIGURE 6. Geometry Interpretation of Euclidean margin loss(e.g. center loss and \mathcal{T} -center loss.) Note that C and C' represent the category feature centers of the last and present iteration respectively.

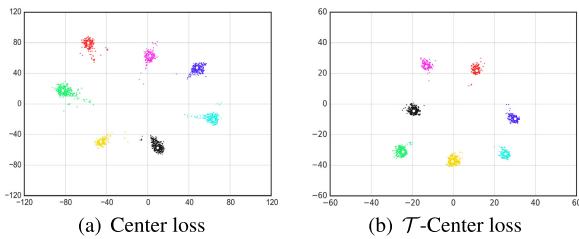


FIGURE 7. Toy example about 2-dimensional iris features under different loss functions.

restricted to 2-dimension for easy visualization. Due to the limitation of dataset, we could not get enough original testing samples so that we perform data augmentation. We train an end-to-end network and the feature results are shown in Figure 7. Each point shown in the figure represents 2-D features of the iris sample and we compute the iris center of each class. Specially, the iris features shown in figure were obtained before performing normalization.

We find two clear differences between the features learned using the two different loss functions discussed above. First, the intra-class variance is large when using the original Center loss, which can be estimated by the average width of the cluster for each class. On the other hand, the features obtained with \mathcal{T} -Center loss have lower intra-class variability, represented by narrower class radius. Second, the magnitudes of the features are much higher with the Center loss (ranging up to 100), since larger feature norms result in a higher probability for a correctly classified class. In contrast, the feature norm has minimal effect on the \mathcal{T} -Center loss since every feature is normalized to the unit circle before computing the loss. Hence, the network focuses on bringing the features from the same class closer to their corresponding centers and separating the features from different classes in the embedding space. Table 2 lists the accuracy obtained with the \mathcal{T} -Center loss. By comparing Table 1 with Table 2, \mathcal{T} -Center loss achieves the better performance reducing the classification error by more than 40%. Note that the accuracy is lower compared to the following experimental results since we are using only 2-dimensional features for classification.

E. NORMALIZATION ON FEATURES

In the proposed \mathcal{T} -Center loss, our approach simultaneously normalizes both the feature vectors and feature center vectors.

TABLE 2. Performance on CASIA under \mathcal{T} -Center loss.

Datasets	Accuracy(%)	Angular Fisher Score
Low L2-Norm	91.39	0.3071
Medium L2-Norm	95.86	0.2001
High L2-Norm	98.25	0.1618
Total	97.08	0.1840

The necessity of feature normalization can be explained in two ways. From the perspective of optimization, the original Center loss actually forces the hard sample features to be pulled to the corresponding feature center, which leads to updating the feature centers with large margin through the training phase. This approach makes the feature centers more inclined to extreme samples. As shown in the Figure 6(a) above in the 2-dimensional feature space, the Center loss essentially corresponds to a larger distance margin under the same range of angles. In this case, the loss value is not convergent for a long time. Although we can use a more conservative hyperparameters λ to balance the effects of Center loss, this approach substantially reduces the effect of center loss, and the learned feature centers could not represent the true centers. For extreme samples(with low L2-norm), the performance of Center loss is poor. In order to solve the issues, we have a L2-norm constraint on the feature descriptors and their feature centers. Minimizing the \mathcal{T} -Center loss is equivalent to maximizing the cosine similarity for the positive pairs and minimizing it for the negative pairs, which strengthens the verification signal of the features. Moreover, the \mathcal{T} -Center loss is able to model the extreme and difficult irises better, since all the iris features have same L2-norm. Besides, as for the distribution of features, we project its features onto a two-dimensional space in order to simplify the analysis in Figure 6(c)&6(d). Specially we fix the center of feature descriptors and the corresponding angle α . For \mathcal{T} -Center, the deep features are located at the intersection of the circle (center C_i and radius $2 \sin \frac{1}{2} \alpha$) and the unit circle. As for Center loss, the features are on the line with an angle of α from the center C_i . In other words, any feature that located at the line satisfies the specified angle requirement, so that there is no effect on improving the distribution of the inter-class samples. Finally, the experimental results in Table 2 prove our theoretical analysis.

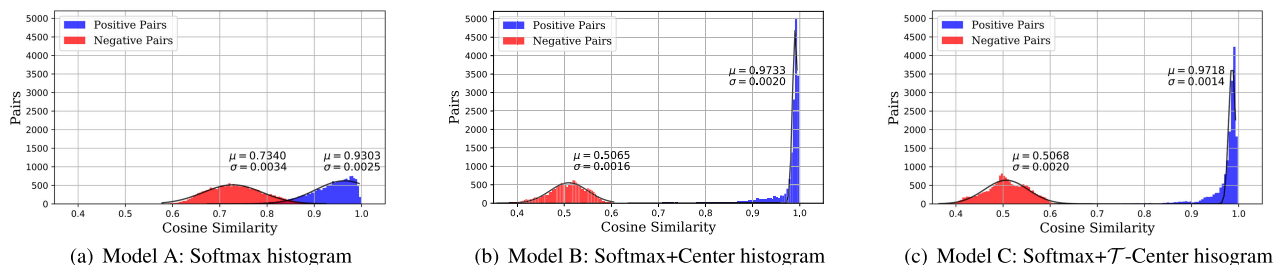


FIGURE 8. Toy example about features extraction. The figure shows the cosine similarity distributions of both positive pairs and negative pairs. Blue area indicates positive pairs while red indicates negative pairs.

IV. EXPERIMENTS

In order to evaluate the effectiveness of the proposed method in this section, three public datasets (ND-Iris, CAISA and IITD) are selected for cross-database verification. The deep learning method and the traditional feature extraction operator are selected as the experimental comparisons. We plot the CMC and ROC curves for the methods used, calculate the EER rates and analyze the model performance under large sample sizes.

A. DATASETS AND PROTOCOLS

- ND-IRIS-0405 Iris Image Dataset** The ND 2004-2005 iris image dataset [25] contains 64980 images corresponding to 356 subjects and 712 unique irises, which is the most popular iris datasets in literature. After removing some incorrectly segmented samples, we ended up with 63290 image samples. Then, we divided samples into a training set, a validation set, and a test set in a ratio of 8:1:1 by using stratified sampling (*i.e.* For each class, 80% of samples are used for training, 10% for validating and 10% for testing). In testing phase, we randomly generated positive pairs (same class) and negative pairs (different classes) for verification, which totally contained 28234 positive pairs and 89279 negative pairs.
- CASIA Iris Image Dataset V4-Thousand** CASIA-Iris-Thousand contains 20000 iris images from 1000 subjects. It is a challenging task that the main source of intra-class variations in dataset are eyeglasses and specular reflection. The testing set generates 38753 positive pairs and 107589 negative pairs after removing incorrectly segmented samples.
- IITD Iris Image Dataset** IITD [26] contains 2240 images samples from 224 subjects. Similar to the previous work, we got 9234 positive pairs and 23902 negative pairs. Specially, we trained the deep model under the ND-IRIS dataset and then directly applied on CASIA and IITD without any further tuning during testing.

Note that the deep CNN model was developed based on the Tensorflow. We use Leaky Rectified Linear Units (LReLU) activation function in all hidden layers [27] to maintain the variance of input data and output data consistent. Each class of sample center is initialized to zero. The deep models were

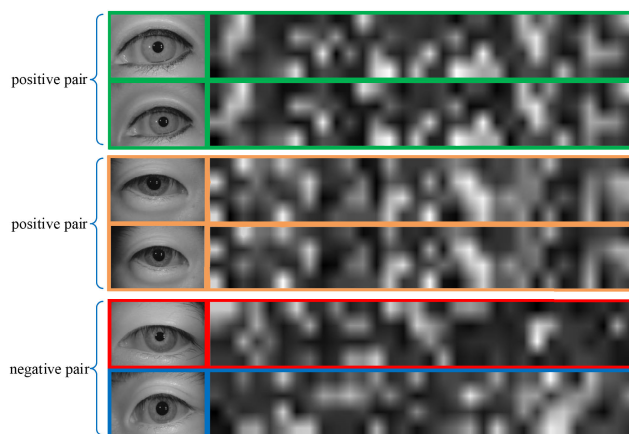


FIGURE 9. Examples of the learned 128-dimensional DeepID. The left column shows test pairs in CASIA. The first two pairs are of the same identity, the third one is of different identities. The corresponding features extracted from each patch are shown in the right. The features are in 128 dimensions. We rearrange them as 4×32 for convenience of illustration. The brighter areas indicate higher values.

trained on GTX1080 with SGD algorithm, with the batch size of 64 (*i.e.* $m = 64$). We fix $\alpha = 0.5$, $\lambda = 0.1$ and weight decay is set to 0.0005. Each model is trained with 60 epochs. For the case of training on the small dataset, the learning rate is started from 0.1 and divided by 10 at every 6K iterations. The training process stops at 20K iterations. After feature extraction using DCNN model, we got the final DeepID with 128-dimensions as it shown in Figure 9.

B. EFFECT OF LOSS FUNCTIONS

In order to visually demonstrate the ability of the proposed method to distinguish iris feature vectors in Figure 9, we compute the cosine similarity distributions of both positive pairs and negative pairs and the histogram results are shown in Figure 8 & Table 3. The results show that compared with Model A (supervised by Softmax only) and Model B (supervised jointly by Softmax and center loss), Model C (supervised jointly by Softmax and \mathcal{T} -Center loss) has 43.8% and 30.2% decrease on the intra-class variance respectively. It indicates that using the \mathcal{T} -Center loss results in a more discriminative distribution in the feature space and a larger angular margin which has lower AFS. Therefore, it is proven that the proposed method can obtain the iris feature vectors

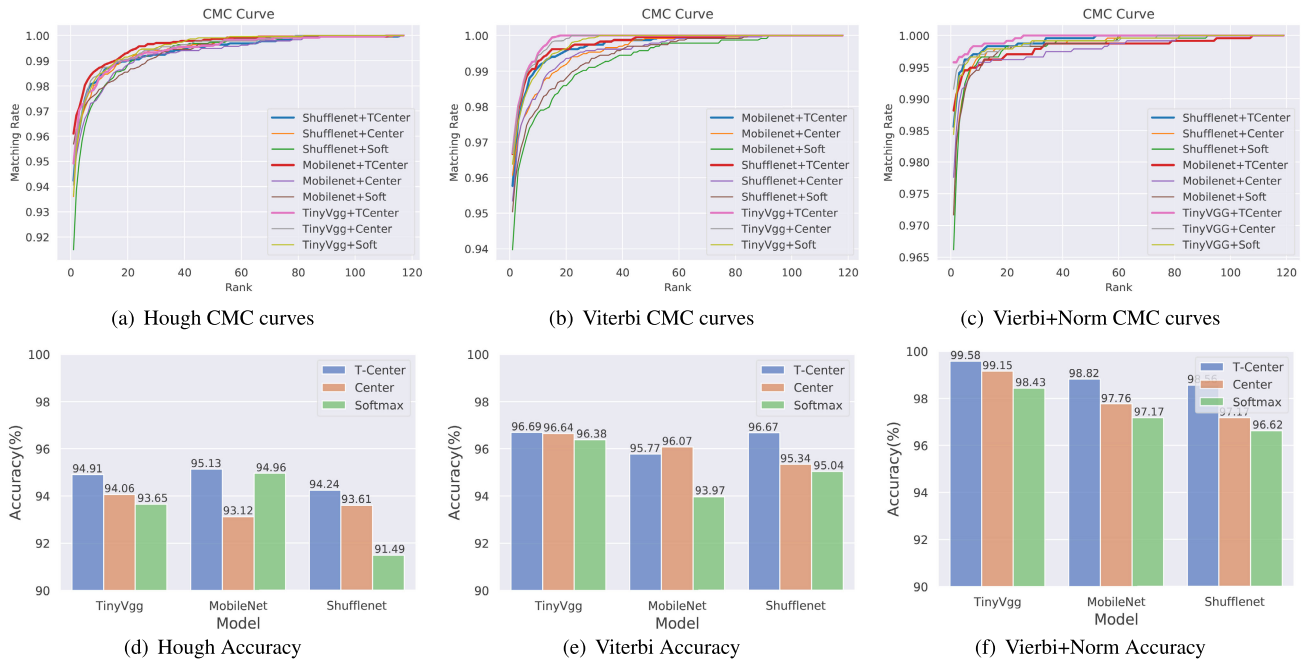


FIGURE 10. Visualization of performance of deep models under different structures and loss functions. The first column shows the CMC curves under different preprocessing methods. Specially, we bolded the T-Center CMC curves in order to facilitate visualization. The second column shows the Rank-1 accuracy, which blue, green and orange represent supervised by T-Center, Center and Softmax respectively.

TABLE 3. Variance of positive and negative pairs under different models.

Method	Positive variance($\times 10^{-3}$)	Negative variance($\times 10^{-3}$)	AFS
Model A	2.51	3.42	0.3942
Model B	2.02	1.61	0.2119
Model C	1.41	1.98	0.1638

with more distinct discrimination, which is very suitable for large-scale iris recognition tasks.

C. OVERALL BENCHMARK COMPARISON

1) EXPERIMENTS ON IRIS IDENTIFICATION

Iris identification aims to match a given probe image to the ones with the same person in gallery. For iris identification experiments, we present the results by CMC curves. It reveals the probability that a correct gallery image is ranked on top-K. The experiment is performed on CASIA dataset, including 1924 images from 1000 subjects. The results are shown in Table 4 & Figure 10.

We have the following conclusions from these results. Firstly, in contrast to other deep models, employing TinyVGG architectures achieve the better performance due to their sophisticated network design. The parameters of three models are 78M, 16M, 4.7M respectively. However, using ShuffleNet takes the least computational cost. Secondly, for better evaluation, we trained CNN architectures under different preprocessing method and loss functions. T-Center also shows significant and consistent improvements, which has 24.7% and 32.9% decrease on the ERR in average compared with Center and Softmax respectively. These results

TABLE 4. Comparison of our proposed method with others on CASIA-IrisV4 under different preprocessing and models. "Rank 1" refers to rank-1 iris identification error rate and "Rank 5" refers to rank-5 error rate.

Preprocessing	Model	Loss	Rank1 ERR.%	Rank5 ERR.%
Hough	TinyVGG	Softmax	6.35	2.61
		Softmax+Center	5.94	2.52
		Softmax+T-Center	5.09	2.66
	MobileNet	Softmax	5.04	3.12
		Softmax+Center	6.88	3.68
		Softmax+T-Center	4.87	2.44
	ShuffleNet	Softmax	8.51	3.60
		Softmax+Center	6.39	2.88
		Softmax+T-Center	5.76	2.52
Viterbi	TinyVGG	Softmax	3.62	1.79
		Softmax+Center	3.36	1.35
		Softmax+T-Center	3.31	1.29
	MobileNet	Softmax	6.03	3.16
		Softmax+Center	3.93	2.31
		Softmax+T-Center	4.23	1.75
	ShuffleNet	Softmax	4.96	2.91
		Softmax+Center	4.66	2.31
		Softmax+T-Center	3.33	1.41
Viterbi+Norm	TinyVGG	Softmax	1.57	0.47
		Softmax+Center	0.85	0.42
		Softmax+T-Center	0.42	0.30
	MobileNet	Softmax	2.83	0.89
		Softmax+Center	2.24	0.55
		Softmax+T-Center	1.18	0.55
	ShuffleNet	Softmax	3.38	0.77
		Softmax+Center	2.83	0.85
		Softmax+T-Center	1.44	0.38

convincingly demonstrate that the proposed method is well designed for open-set iris identification.

2) EXPERIMENTS ON IRIS VERIFICATION

As for iris verification, it is a two-choice decision task (i.e. same versus different). The comparison experiments are conducted on three public datasets (i.e. ND-IRIS-0405, CASIAv4-Thousand and IITDv2) and the division of testing

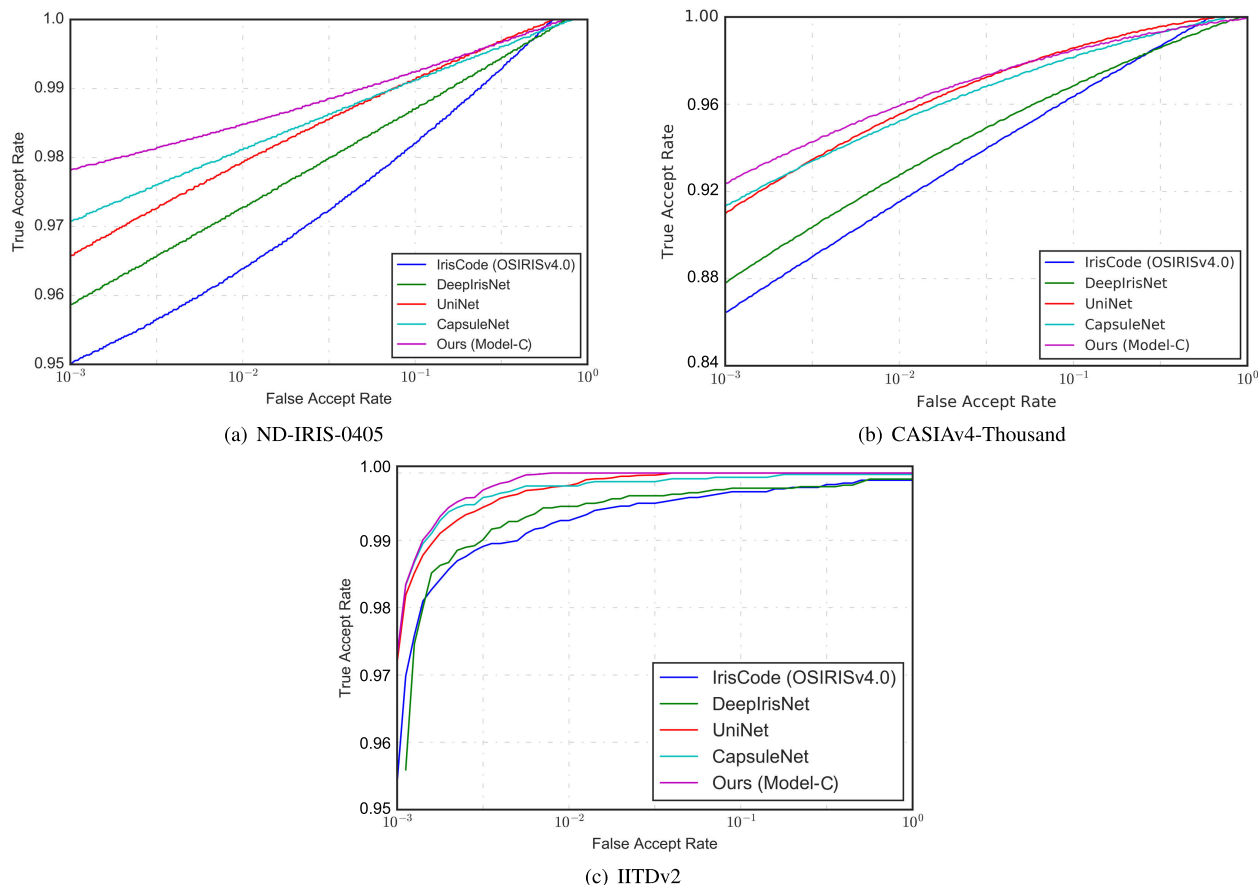


FIGURE 11. ROCs for comparison with other state-of-the-art methods on employed databases.

TABLE 5. Summary of performance for the comparison on public datasets.

	ND-IRIS-0405			CASIAv4-Thousand			IITDv2		
	TAR(%)	EER(%)	AUC(%)	TAR(%)	EER(%)	AUC(%)	TAR(%)	EER(%)	AUC(%)
IrisCode(OSIRISv4.0)	95.02	2.34	98.89	85.51	4.69	97.53	95.67	1.39	98.94
DeepIrisNet	95.84	1.93	99.01	86.86	4.01	97.93	95.03	1.17	99.02
UniNet	96.61	1.79	99.13	90.61	2.79	98.67	97.35	0.89	99.20
CapsuleNet	97.12	1.03	99.17	91.04	3.05	98.60	97.39	0.76	99.27
Ours(Moedel-C)	97.87	0.91	99.29	92.54	2.36	98.75	97.43	0.74	99.30

samples have been given in pervious Section III. In the comparison methods, we first compared with the mainstream classic iris algorithm IrisCode based [28] on 2D-Gabor filter. As we know, the majority of recent works on iris recognition focus on improving segmentation or normalization models, applying multi-score fusion or feature bits selection. In other words, in the context of iris feature representations, IrisCode is definitely a fair benchmark for the performance evaluation and we select OSIRISv4.0 [23] system, which is an open source tool for iris recognition. What’s more, we also compare with the latest SOTA algorithm based on deep learning, including DeepIrisNet [10], UniNet [15] and CapsuleNet [16]. It should be noted that we use the ND-IRIS-0405 as the training set and the trained model is directly applied on CASIAv4-Thousands and IITDv2.

On one hand, we are more concerned about the generalization capability of the proposed framework under challenging practical application to predict unseen labels. On the other hand, considering that the sample quantity of ND-Iris-0405 dataset is quite suitable as the training set, the other datasets may not achieve similar results. Finally, hyperparameters of the training processes for above architectures have been carefully investigated to achieve best performance in validation sets.

Note that all benchmark experiments are based on the same preprocessing method as shown in Section III-A. We choose TAR (at 0.1% FAR), EER and AUC as the performance evaluation indicators and the specific comparison results are shown in Figure 11 & Table 5. Consistent improvements from our method over others can be observed on all of three databases. For ND-IRIS, we trained deep model based on training set

and evaluated performance on testing set. As the results shown in Table 5, we achieve 0.75% and 2.85% increase on TAR compared with CapsuleNet and IrisCode.

For CASIA and IITD, we directly evaluate the performance on testing sets without any further tuning which is kind of cross database testing. This evaluation aims to validate generalization capability of the framework when there are limited or no training samples accessible from the target iris database. Specifically, for the IITD database, the image samples are of high quality, with clear edges and few eyelashes so that all methods have achieved good performance on the testing dataset. For the CASIA dataset, it is a challenging task that the main source of intra-class variations in dataset are eyeglasses and specular reflection, which is similar to the application in the real environment. Thus the TAR rate is lower than the ND-IRIS and IITD. In order to solve the issue, we analyzed the iris images which failed from our approach. We found that these failed cases can be largely attributed to degradation in the iris image quality (*i.e.* fuzzy or light reflection), which was a large noise interference for the fine-grained feature extraction of iris recognition. However, the proposed method still shows the significant and encouraging improvements with 1.50% TAR increase on CASIA. These results convincingly demonstrate that the proposed loss function is well designed for iris recognition and has greatly improved the robust performance of iris recognition across the datasets and over various devices.

V. CONCLUSION

Large-scale iris recognition tasks remains a challenging task in practical applications due to the difficulty of classification threshold determination. The paper proposes a novel feature extraction approach for iris recognition, which can obtain iris feature vectors with more obvious discrimination. We refer the novel loss function to \mathcal{T} -Center loss. Jointly supervised by the linear combination of \mathcal{T} -Center loss and softmax, the discriminative power of the deep features based on CNNs can be highly enhanced. Extensive experiments on cross-database and large-scale iris samples show that the performance improvement for iris verification task, which proves the effectiveness of the \mathcal{T} -Center loss function. We wish that our substantial explorations on learning discriminative features via \mathcal{T} -Center loss will benefit the iris recognition community.

REFERENCES

- [1] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1148–1161, Nov. 1993.
- [2] M. De Marsico, A. Petrosino, and S. Ricciardi, "Iris recognition through machine learning techniques: A survey," *Pattern Recognit. Lett.*, vol. 82, pp. 106–115, Oct. 2016.
- [3] W. W. Boles and B. Boashash, "A human identification technique using images of the iris and wavelet transform," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1185–1188, Apr. 1998.
- [4] R. P. Wildes, "Iris recognition: An emerging biometric technology," *Proc. IEEE*, vol. 85, no. 9, pp. 1348–1363, Sep. 1997.
- [5] C.-H. Chen and C.-T. Chu, "High performance iris recognition based on 1-D circular feature extraction and PSO-PNN classifier," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10351–10356, Sep. 2009.
- [6] A. Kumar and A. Kumar, "Adaptive management of multimodal biometrics fusion using ant colony optimization," *Inf. Fusion*, vol. 32, pp. 49–63, Nov. 2016.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
- [8] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.
- [9] N. Liu, M. Zhang, H. Li, Z. Sun, and T. Tan, "DeepIris: Learning pairwise filter bank for heterogeneous iris verification," *Pattern Recognit. Lett.*, vol. 82, pp. 154–161, Oct. 2016.
- [10] A. Gangwar and A. Joshi, "DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 2301–2305.
- [11] K. B. Raja, R. Raghavendra, S. Venkatesh, and C. Busch, "Multi-patch deep sparse histograms for iris recognition in visible spectrum using collaborative subspace for robust verification," *Pattern Recognit. Lett.*, vol. 91, pp. 27–36, May 2017.
- [12] T. Zhao, Y. Liu, G. Huo, and X. Zhu, "A deep learning iris recognition method based on capsule network architecture," *IEEE Access*, vol. 7, pp. 49691–49701, 2019.
- [13] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jul. 2017, p. 1.
- [14] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 5265–5274.
- [15] Z. Zhao and A. Kumar, "Towards more accurate iris recognition using deeply learned spatially corresponding features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 3829–3838.
- [16] K. Wang and A. Kumar, "Toward more accurate iris recognition using dilated residual features," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 12, pp. 3233–3245, Dec. 2019.
- [17] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis. Amsterdam, The Netherlands: Springer*, Oct. 2016, pp. 499–515.
- [18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: <http://arxiv.org/abs/1704.04861>
- [19] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 116–131.
- [20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*. [Online]. Available: <https://arxiv.org/abs/1409.1556>
- [21] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, Jan. 2004.
- [22] Z. He, T. Tan, Z. Sun, and X. Qiu, "Toward accurate and fast iris segmentation for iris biometrics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1670–1684, Sep. 2009.
- [23] N. Othman, B. Dorizzi, and S. Garcia-Salicetti, "OSIRIS: An open source iris recognition software," *Pattern Recognit. Lett.*, vol. 82, pp. 124–131, Oct. 2016.
- [24] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1041–1049.
- [25] K. W. Bowyer and P. J. Flynn, "The ND-IRIS-0405 iris image dataset," *arXiv:1606.04853*, 2016. [Online]. Available: <https://arxiv.org/abs/1606.04853>
- [26] A. Kumar and A. Passi, "Comparison and combination of iris matchers for reliable personal authentication," *Pattern Recognit.*, vol. 43, no. 3, pp. 1016–1026, Mar. 2010.
- [27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *J. Mach. Learn. Res.*, vol. 9, pp. 249–256, May 2010.
- [28] J. Daugman, "How iris recognition works," in *Proc. Int. Conf. Image Process.*, vol. 1, 2004, pp. 1–33–1–36.



YIFENG CHEN received the B.E. degree in electronic engineering and automation from Soochow University, in 2017. He is currently pursuing the M.Sc. degree with the Department of Intelligent Transportation and the Cognitive Computation Laboratory, Soochow University. His research interests are in biometrics, computer vision, and pattern recognition. He was involved in several projects in which these techniques are being applied in the fields of computer vision.



CHENG WU (Member, IEEE) received the M.Sc. degree in computer engineering from Concordia University, Canada, in 2005, and the Ph.D. degree from Northeastern University, USA, in 2010. He is currently an Associate Professor with the School of Urban Rail Transportation, Soochow University, China. He authored or coauthored over 20 publications in *International Journal of Intelligent and Fuzzy System* and International Conference on Autonomous Agents and Multiagent

Systems. He was also involved in several projects in which these techniques are being applied in the fields of communication, robotics, and computer vision. His research interests include neural and fuzzy systems, dynamic multiagent systems, intelligent human-computer interaction, and big data technology and applications.



YIMING WANG (Member, IEEE) received the M.Sc. degree in computer engineering from Soochow University, China, and the Ph.D. degree from the Nanjing University of Posts and Communications. She is currently a Full Professor with the School of Urban Rail Transportation, Soochow University. Her research interests include wireless communications, cognitive wireless sensor networks, and intelligent transportation technology and applications. She authored or coauthored over

60 publications. She was also involved in several projects in which these techniques are being applied in the fields of communication, robotics, and computer vision. She is also an active Reviewer for *Wireless Communications and Mobile Computing*, and *System and Signal Processing*.

...