Journal of
Electronic Imaging

JElectronicImaging.org

Effective deep ensemble hashing for open-set palmprint recognition

Huikai Shao
Dexing Zhong
Xuefeng Du

SPIE. IS&T imaging.org

Huikai Shao, Dexing Zhong, Xuefeng Du, "Effective deep ensemble hashing for open-set palmprint recognition," *J. Electron. Imaging* **29**(1), 013018 (2020), doi: 10.1117/1.JEI.29.1.013018

Effective deep ensemble hashing for open-set palmprint recognition

Huikai Shao,[a] **Dexing Zhong,**[a,b,*] **and Xuefeng Du**[a]
[a]Xi'an Jiaotong University, School of Electronic and Information Engineering, Xi'an, China
[b]Nanjing University, State Key Laboratory for Novel Software Technology, Nanjing, China

Abstract. Recently, palmprint recognition has made huge progress and attracted the attention of more and more researchers. However, current research rarely involves open-set palmprint recognition. We proposed deep ensemble hashing (DEH) for open-set palmprint recognition. Based on the online gradient boosting model, we trained multiple learners in DEH, which focus on identifying different samples. In order to increase the diversity between learners, activation loss and adversarial loss were introduced. Through minimizing activation loss, the neurons of different learners restrained each other, and through adversarial loss, the optimal distance between the features extracted by different learners was obtained. Palmprint identification and verification experiments were performed on PolyU multispectral database and our self-built databases. The results show the effectiveness of DEH in deal with open-set palmprint recognition. Compared to baseline models, DEH increased the recognition accuracy by up to 6.67% and reduced the equal error rate by up to 3.48%. *© 2020 SPIE and IS&T* [DOI: 10.1117/1.JEI.29.1.013018]

Keywords: palmprint recognition; deep ensemble learning; deep hashing network; activation loss; adversarial loss.

Paper 190615 received Jul. 1, 2019; accepted for publication Jan. 22, 2020; published online Feb. 11, 2020.

1 Introduction

With the development of information technology, the security of identity information has become more and more important. Compared with traditional authentication methods, such as keys and passwords, biometrics has many unique advantages as an emerging technology.[1] Palmprint as a unique biometric modality has received wide attention from researchers.[2–4] A number of databases have emerged as benchmarks for academic research, such as PolyU[5] and IITD[6] palmprint databases. And many palmprint recognition algorithms have been proposed, such as competitive code[7] and double-orientation code (DOC).[8] With the development of deep learning-based methods, many researchers have introduced convolutional neural networks (CNN) into the field of palmprint recognition, which achieved promising performance.[9,10]

However, such palmprint recognition algorithms need to align the training set with the categories in the test set. The performance tends to saturate or decline due to over-fitting when untrained classes appear in the test set, especially for deep learning-based methods.[11] To solve this problem, in this paper, we propose deep ensemble hashing (DEH) algorithm, combining the high efficiency of deep hashing network (DHN) with the high accuracy of ensemble learning algorithm. Ensemble learning has shown great capability of extracting high-dimensional features to conduct accurate and efficient classification tasks.[12] In ensemble methods, a set of models (base classifiers) is combined to obtain a more accurate and reliable model in comparison with what a single predictor can achieve.[13] In this paper, we introduce the ensemble learning into DHN. The structure of DEH is shown in Fig. 1, CNN based on VGG-16[14] is used to extract palmprint convolution features, which are inputted to several different hashing learners for encoding. In this paper, the first few layers of VGG-16 (yellow, batch 1 to batch 4 in Fig. 1) are pretrained by ImageNet and the remaining layers are fine-tuned by palmprint

*Address all correspondence to Dexing Zhong, E-mail: bell@xjtu.edu.cn

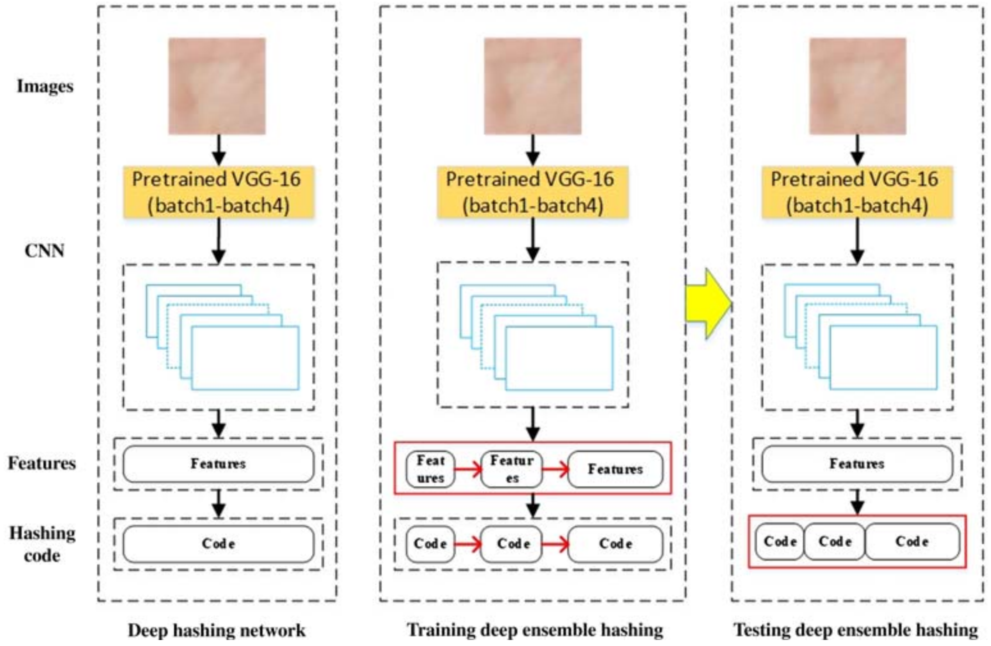

Fig. 1 Schematic diagram of DEH. In DHN (on the left), there is only a learner, and there are several learners in DEH (on the right). Several learners are trained by online gradient boosting model. Activation loss and adversarial loss are introduced to increase the diversity of learners.

images. The features of the fully connected layer are inputted to the hashing learner to obtain binary codes. Every hashing learner in DEH can be used as a standard DHN. Finally, all the hashing codes learned by different learners are connected in series to determine the final palmprint features.

The accuracy of ensemble learning depends on the accuracy of individual learners as well as the correlations between them.[15] Intuitively, training multiple learners and then tying the features in series can obtain higher accuracy than an individual feature learner with the same dimensions. But these learners treat different samples in the same way and extract highly correlated feature representations. Due to the high correlation between the learners, the learner with a shorter code length is rendered useless, which will not gain performance improvements over the traditional DHN algorithm. To overcome this weakness, the activation loss and adversarial loss were adopted.

First, different hashing learners were related to each other through boosting algorithm. Each learner updates the weights based on a reweighted training batch according to the performance of previous learner. Difficult samples will be given greater weights. Therefore, successive learners focus on the samples that the preceding learners cannot distinguish, which can improve the overall recognition performance. Second, activation loss and adversarial loss were introduced to increase the diversity of learners and reduce their relevance. Activation loss acts on the neuron level, where different learners restrain each other. In this way, the learner will learn different feature representations separately. Adversarial loss is used to constrain the features learned by each learner. On the one hand, the features extracted by different learners need to be as different as possible to maintain diversity, and on the other hand, these features are as similar as possible to maintain recognizability. DEH was adopted to perform experiments of open-set palmprint recognition on several databases and our self-built databases. The experiment results show the effectiveness of DEH.

The contributions can be briefly summarized as follows.

(1) We proposed an efficient DEH algorithm to improve the accuracy of open-set palmprint recognition. DEH maintains the efficiency of DHN and the high accuracy of ensemble learning. Based on online gradient boosting learning algorithm, DEH reweighted different samples, thus the learners can extract unique features.

(2) Activation loss and adversarial loss were introduced into DEH. These constraints increase the diversity between learners and decrease their correlation, so that each learner can learn unique discriminative features and improve the recognition accuracy.

(3) Based on DEH, open-set palmprint identification and verification experiments were implemented on several public databases and our self-built databases. The experiment results show that DEH can greatly improve the recognition accuracy by up to 6.67% and reduce the equal error rate (EER) by up to 3.48%.

This paper consists of six sections. In Sec. 2, we review the related works. Section 3 describes our proposed DEH. Section 4 presents experiments and results. Section 5 analyzes the results. Section 6 concludes the paper. Section 7 provides future works.

2 Related Work

In general, palmprint recognition mainly includes image acquisition, preprocessing, feature extraction, and feature matching.[16] In image acquisition, the commonly used devices contain complementary metal oxide semiconductor (CMOS) or charge coupled device cameras, additional lights, and handles for fixing the palm position. In currently published databases, the testers needed to place their palms in a closed environment, and additional illuminations were applied, which limits their practical application. But in this paper, we used a mobile phone to collect unconstrained databases in an open environment. The preprocessing is mainly to eliminate noise interference and extract regions of interests (ROIs), which is based primarily on valley points.[17–19]

Palmprint feature extraction and matching is based primarily on the texture and directional features.[20] Kong and Zhang[7] proposed CompCode to extract orientation information from palm lines using competitive coding scheme and angular matching. Then motivated from the success of CompCode, Jia et al.[21] proposed robust line orientation code to extract principal orientation code of a palmprint and obtained higher recognition rate and faster processing speed. Guo et al.[22] presented binary orientation co-occurrence vector (BOCV) to represent multiple orientations for a local region. Zhang et al.[23] proposed extended-BOCV by incorporating fragile bits information in the code maps generated from different palms of the same person. Dai et al.[24] quantitatively studied statistics of ridges to design a segment-based matching and fusion palmprint recognition algorithm. Fei et al.[8] proposed a novel DOC to extract the orientation feature of palmprint and adopted nonlinear angular matching score to evaluate the similarity. Zhang et al.[5] found that the performance of palmprint recognition could be improved by filtering clear images to lower sharpness. Using more accurate dominant orientation representations, Xu et al.[25] proposed a discriminative and robust competitive code-based method for palmprint authentication. Benjoudi et al.[26] proposed a simple but effective method, patch binarized statistical image features descriptor, for a meaningful local palmprint representation. With the development of deep learning, CNN-based algorithms greatly improve the accuracy of palmprint recognition. Cheng et al.[27] proposed deep convolutional features-based supervised hashing using CNN-F and principal component analysis. Zhong et al.[9] used Siamese Network to extract features and compared their similarity using by neural networks. Izadpanahkakhk et al.[28] proposed a deep mobile palmprint verification framework to solve the problems of deep learning-based methods, such as high computation cost and lack of data. Bensid et al.[29] proposed discrete cosine transform network for palmprint identification in unimodal and multimodal modes. Zhong et al.[30] encoded palmprint images as binary codes by DHN and obtained the state-of-the-art performance. In recently, researchers have tried to perform cross-domain palmprint recognition and achieved some promising results.[31,32] However, the previous algorithms need to align the training set with the categories in the test set. In this paper, we focus on the open-set palmprint recognition.

Another direction of related work is DHN, which has also widely been studied for image retrieval. Lai et al.[33] proposed deep supervised hashing and mapped the images into binary codes via carefully designed deep neural networks. Based on DHN, Zhong et al.[34] proposed an end-to-end method to further improve the performance of dorsal hand vein recognition. Inspired by alternating direction method of multipliers, Zhang et al.[35] trained very deep neural networks for supervised learning of hashing codes. Yang et al.[36] proposed supervised semantics-preserving deep hashing that constructed binary hashing codes from labeled data for large-scale image search. Recently, DHN is also used for domain adaptation. Venkateswara et al.[37] used labeled source data and unlabeled target data to learn informative hashing codes and accurately classify unseen target data.

About ensemble learning, there are mainly three famous ensemble models: bagging,[38] random forest,[15] and boosting.[39] Motivated by the success of deep learning, deep ensemble methods are broadly used in computer vision competitions.[12] Using a multilayered long short-term memory, Sutskever et al.[40] proposed a general end-to-end approach to sequence learning. Zheng et al.[12] proposed an end-to-end learning framework, deep ensemble machine, for video classification. Inspired by metalearning, Xuan et al.[41] defined a family of embedding functions as an ensemble. Opitz et al.[42] created an ensemble of embedding for image retrieval. The last embedding layer of deep network was divided into an embedding ensemble, and a boosting model was used to formulate the task of training this ensemble. Inspired by them,[42] we proposed DEH to implement open-set palmprint recognition with unseen classes. Combining ensemble learning with DHN, boosting model was adopted to learn several hashing learners of different lengths. Finally, the different codes are concatenated into one code for palmprint recognition.

3 Our Method

As shown in Fig. 1, DEH sets a group of hashing learners in the coding layer. On the one hand, each learner can obtain a distinguished hashing code. On the other hand, the correlation between each learner is minimal, so more discriminative features can be learned. Thus when the total hashing code length is the same, i.e., the dimensions of feature space are the same, DEH can achieve better performance. This section will present the details of our proposed DEH.

3.1 *Deep Hashing Network*

DHN combines deep learning and hash coding, which extracts features using CNN and converts images into binary codes through encoding.[43] DEH relies on every effective DHN learner. When individual DHN learner obtains better performance, the ensemble DHN will also obtain better results. Like metric learning, DHN also makes the positive codes in the feature space as close to each other as possible and the negative codes far away from each other. This can be mainly achieved by hashing loss, which is based on binomial deviance loss in this paper.[44]

3.1.1 *Hashing loss*

Suppose there are two palmprint images x_i and x_j, and their convolutional features extracted by learner u are $u(x_i)$ and $u(x_j)$, which are converted to binary codes by sign function. A matrix S_{ij} is defined to represent the relationship between x_i and x_j. When they are positive samples, $S_{ij} = 1$, otherwise $S_{ij} = 0$. Further, the distance between $u(x_i)$ and $u(x_j)$ is defined as $D[u(x_i), u(x_j)]$, as shown in Eq. (1). $D[u(x_i), u(x_j)]$ is similar to the cosine similarity score, and its value belongs to $[-1, +1]$

$$D[u(x_i), u(x_j)] = \frac{u(x_i)^{\mathrm{T}} \cdot u(x_j)}{\|u(x_i)\| \cdot \|u(x_j)\|}, \tag{1}$$

where $\| \cdot \|$ is the L_2-norm of vector. Then the hashing loss between the two images is

$$\begin{aligned} L\{D[u(x_i), u(x_j)]\} = {} & S_{ij} \log(1 + e^{-\beta_1\{D[u(x_i),u(x_j)]-\beta_2\}}) \\ & + (1 - S_{ij}) \log(1 + e^{\beta_1\{D[u(x_i),u(x_j)]-\beta_2\}C}), \end{aligned} \tag{2}$$

where C is used to balance the weight between positive and negative samples, set to 25.[45] β_1 and β_2 are scaling and translation parameters and are set to 2 and 0.5.[42,46]

3.1.2 *Quantization loss*

At the same time, in order to reduce the error caused by the direct conversion of the feature vector into binary code, quantization loss is introduced. Suppose the code of x_i is $b_i \in \{-1, +1\}^L$, L is the length of code. Quantization loss is

$$L_Q(x_i) = \frac{1}{2}\|b(x_i) - u(x_i)\| = \frac{1}{2}\||u(x_i)| - 1\|, \tag{3}$$

where $|\cdot|$ is the operation of obtaining element-wise absolute value and $|\cdot|$ is also the L_2-norm of vector.

The loss of image pair x_i and x_j consists of hashing loss and quantization loss, which are combined by a weight w:

$$L_{\mathrm{DHN}} = L\{D[u(x_i), u(x_j)]\} + w \times [L_Q(x_i) + L_Q(x_j)]. \tag{4}$$

3.2 *Ensemble DHN*

Single DHN can achieve relatively good results, but there are certain disadvantages in some aspects. Because the algorithm gives the same weights to different images, it is easy to identify simple samples, but difficult for complex samples. However, the illumination and noise between samples are often different, resulting in different recognition difficulties. Based on online gradient boosting, DEH is adopted to train multiple hashing learners with shorter coding lengths to solve the problem. During the learning process, each learner resets the weight of each training sample according to the gradient of loss. The samples correctly identified were set lower weights, and the misidentified samples were set larger weights. As the number of iterations increase, the subsequent learner pays more attention to the samples that the preceding learners cannot distinguish, thus improving the overall recognition performance.

Suppose there are M hashing learners in DEH, which are expressed as $\{u_1, u_2, u_3, \dots, u_M\}$. For each sample pair x_i and x_j, each learner will generate a distance $D[u(x_i), u(x_j)]$, which is given a unique weight. A fixed number of M weak learners combined by different weights are optimized by the online gradient boosting learning algorithm:[47]

$$L_E\{D[u(x_i), u(x_j)]\} = \sum_{m=1}^{M} \alpha_m w_{i,j}^m L\{D[u_m(x_i), u_m(x_j)]\}, \tag{5}$$

$$\alpha_m = \frac{2}{m+1} \cdot \prod_{n=m+1}^{M} \left(1 - \frac{2}{n+1}\right), \tag{6}$$

where α_m represents the weight of the m'th learner u_m, and $w_{i,j}^m$ represents the weight generated by pair x_i and x_j for u_m. $w_{i,j}^m$ is computed from the negative gradient $-L\{D[u(x_i), u(x_j)]\}'$ of the previous $m-1$ learners. In this way, the latter learners can be optimized according to the performances of previous learners, that is

$$w_{i,j}^{m+1} = -L(D_{i,j}^m)', \tag{7}$$

$$D_{i,j}^m = \left(1 - \frac{2}{m+1}\right) D_{i,j}^{m-1} + \frac{2}{m+1} D[u_m(x_i), u_m(x_j)], \tag{8}$$

where $D_{i,j}^0 = 0$. Compared with the previous algorithm, in the online gradient boosting learning algorithm, the distance score generated by the m'th learner will be recombined with the previous learners, so that different weights are set for different samples.

3.3 *Diversity in Ensemble Hashing Learners*

The successive different learners above focus on different samples, which will improve the performance. For ensemble, another important topic is to make its constituent models diverse.[48]

In this paper, activation loss and adversarial loss were adopted to increase the diversity between learners.[42]

3.3.1 *Activation loss*

Activation losses increase the diversity through mutual inhibition between learners. For a sample, the output of only one weak learner follows the normal value distribution, and the others are close to zero. Suppose that for a sample x_i, the feature vector obtained by the feature extractor is $f(x_i): R^k -> R^h$, where k denotes the dimensionality of input sample and h denotes the dimensionality of the last hidden layer of feature extractor. The feature extractor is the first few layers of VGG-16 pretrained on ImageNet. Then the feature $f(x_i)$ is converted to a low-dimensional feature by the learner u, i.e., $u(x_i) = u[f(x_i)] = f(x_i)^{\mathrm{T}}W$, which finally generates binary code through sign function, i.e., $b_i = \mathrm{sign}[u(x_i)]$. The activation loss between any two learners u_m and u_n can be expressed as

$$L_{\mathrm{act}_{m,n}}(x_i) = \sum_{\substack{k\in G_m \\ l\in G_n}} [u_m(x_i)_k \cdot u_n(x_i)_l]^2, \tag{9}$$

$$u_m(x_i) = f(x_i)^{\mathrm{T}} W_m, \tag{10}$$

$$u_n(x_i) = f(x_i)^{\mathrm{T}} W_n, \tag{11}$$

where G_m and G_n represent the neuron sets of learners u_m and u_n, respectively, and $u_m(x_i)_k$ denotes the k'th dimension of $u_m(x_i)$. W_m denotes the submatrix of W corresponding the u_m, and W_n denotes the submatrix of W corresponding the u_n. In order to avoid getting the trivial solution $W = 0$, the regulator L_{weight} needs to be introduced:

$$L_{\mathrm{weight}} = \sum_{h=1}^{L} (w_h^{\mathrm{T}} w_h - 1)^2, \tag{12}$$

where w_h denotes the row vectors of W, and L is equal to the length of hashing code. For N samples, the total activation loss is expressed as

$$L_{\mathrm{act}} = \sum_{i=1}^{N} \sum_{\substack{m=1 \\ n=m+1}}^{M} L_{\mathrm{act}_{m,n}}(x_i) + \gamma_w L_{\mathrm{weight}}, \tag{13}$$

where γ_w is the regularization parameter.

3.3.2 *Adversarial loss*

The aforementioned activation loss reduces the correlation between learners from the neuron level. Further, constraints can be directly applied to the feature vectors $u_m(x_i)$ and $u_n(x_i)$ extracted by the learners. Specifically, for the same sample x_i, two vectors $u_m(x_i)$ and $u_n(x_i)$ need to be as different as possible, nonetheless discriminative. Therefore, on the one hand, in order to reduce the correlation between different learners, the distance between $u_m(x_i)$ and $u_n(x_i)$ should be as large as possible. On the other hand, for a same sample, the features extracted by different learners need to be as consistent as possible, so that the features can be discriminative to obtain accurate recognition results. This is similar to the optimization goal in generative adversarial nets.[49]

In the online gradient boosting learning algorithm, the learner that focus on more complex samples will generally have longer codes, which makes it impossible to directly obtain the distance between $u_m(x_i)$ and $u_n(x_i)$ extracted by learners with different lengths. A regressor t is introduced into the network, as shown in Fig. 2. Regressor t converts $u_m(x_i)$ into $t_{m,n}(u_m)$, which has the same dimension as $u_n(x_i)$, so that the distance between them can be compared. Therefore, adversarial loss can be expressed by

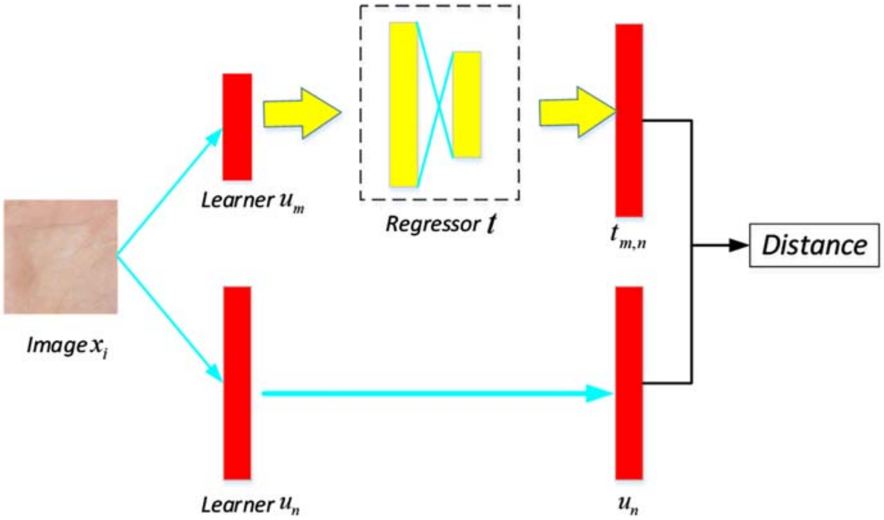

Fig. 2 An overview of adversarial loss. A palmprint image x_i is inputted into learners u_m and u_n with different dimensions. Through a regressor t, $u_m(x_i)$ was converted into $t_{m,n}(u_m)$, which is the same dimension as $u_n(x_i)$, so that the distance between them can be compared.

$$\max\nolimits_{u_m,u_n} \min\nolimits_{t_{m,n}} L_{\text{sim}}[u_m(x_i), u_n(x_i)], \tag{14}$$

where $L_{\text{sim}}[u_m(x_i), u_n(x_i)]$ represents the similarity loss between the feature vectors $u_m(x_i)$ and $u_n(x_i)$. Equation (14) shows that for $u_m(x_i)$ and $u_n(x_i)$, maximizing L_{sim} can increase the similarity between them to obtain stable semantic features; for $t_{m,n}$, minimizing L_{sim} can reduce the similarity to maintain recognizability. To achieve this, a reverse gradient layer is introduced.[42]

Suppose that the dimensions of $u_m(x_i)$ and $u_n(x_i)$ obtained are d_m and d_n, respectively. Then

$$L_{\text{sim}}[u_m(x_i), u_n(x_i)] = -\frac{1}{d_n} \sum \{u_n(x_i) \odot t_{m,n}[u_m(x_i)]\}^2, \tag{15}$$

where $\odot$ denotes the Hadamard product.

Similar to the activation loss, a regularization term L_{weight} is added:

$$L_{\text{weight}} = \sum_h (w_h^{\mathrm{T}} w_h - 1)^2 + \max(0, \hat{b}^{\mathrm{T}} \hat{b} - 1) + \sum_h (\hat{w}_h^{\mathrm{T}} \hat{w}_h - 1)^2, \tag{16}$$

where $\hat{b}$ denotes biases and $\hat{w}_h$ denotes the h'th row of weight matrix $\hat{W}$of regressor $t_{m,n}$. w_h denotes the h'th row of weight matrix W. Therefore, the adversarial loss is

$$L_{\text{adv}} = \sum_{i=1}^{N} \sum_{\substack{m=1\\ n=m+1}}^{M} L_{\text{sim}}[u_m(x_i), u_n(x_i)] + \gamma_w L_{\text{weight}}, \tag{17}$$

where γ_w is a regularization parameter.

In this paper, regressor $t_{m,n}$ consists of two fully connected layers with the leaky rectified linear unit. There are 256 neurons in the first layer, and the number of neurons in the second layer is d_n. Regressor $t_{m,n}$ updates the weights by optimizing L_{adv}.

Assuming that there are N images in training, which can be combined into $\frac{N(N-1)}{2}$ image pairs. The entire optimization goal is

$$L_{\text{DEH}} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} L_E\{D[u(x_i), u(x_j)]\} + \alpha \sum_{i=1}^{N} L_Q(x_i) + D_{\text{diversity}}, \tag{18}$$

where $D_{\text{diversity}}$ takes L_{act} and L_{adv}, respectively.

4 Experiments and Results

4.1 *Databases*

4.1.1 *PolyU multispectral palmprint database*

PolyU multispectral palmprint database was established under multispectral illuminations, i.e., blue, red, green, and near-infrared (NIR) illuminations.[50] The collection conditions are the same except for the illumination, so they can be studied as independent spectral palmprint databases, i.e., blue, red, green, and NIR databases. Each database contains 6000 palmprint images from 250 volunteers, including 195 males and 55 females. The samples were collected in two sessions, and each session collected six images. All palmprint images had been cropped to a size of 128×128 pixels, as shown in Fig. 3.

4.1.2 *XJTU_A palmprint database*

XJTU_A palmprint database was collected by us using CMOS camera. It contains 1130 images from 57 volunteers. Each individual was asked to provide 10 images of left hand and 10 images of right hand, respectively. Unlike other public databases, there are almost no restrictions on the conditions of collection. Only the indoor natural illumination was adopted without any additional extra illumination. The position of hand was not limited, and the volunteers only needed to place their palms under the CMOS camera. All samples are stored as 640×480 RGB images containing complete palms. Although the limitation of image acquisition is reduced, the background of image is still pure black, which makes the extraction of ROI simpler. First, the outline of the palm was extracted, and then the valley point P_1 between the index finger and the middle finger and the valley point P_2 between the ring finger and the little finger were obtained, respectively, as the reference points for determining ROI. Assume that the length of the line P_1P_2 was L. Then the point P, in which the distance from P_1P_2 is $\frac{L}{2}$ was set as the center of ROI. Finally, with P as the center point, a rectangular area with the side length of $\frac{3}{5}L$ was extracted as the palmprint ROI, as shown in Fig. 4.

4.1.3 *XJTU_B palmprint database*

XJTU_A palmprint database has reduced the constraints for image acquisition, but it is somewhat different from the real application scenario. In practical applications, the illumination and background vary greatly, making authentication relatively difficult. In this paper, we used the mobile phone, iPhone 6s to establish XJTU_B palmprint database in an uncontrolled way. It contains 3920 palmprint images from 98 individuals. The volunteers randomly selected the image background and the angle of hand according to their own wishes. At the same time, two kinds of illumination were adopted, the indoor natural illumination and the flash of iPhone 6s. Each volunteer was required to provide images of different palms under different illuminations with a total of 40 images. In order to increase the diversity of samples, the background and angle were changing at any time. The image was stored as the default format for iPhone 6s, an RGB

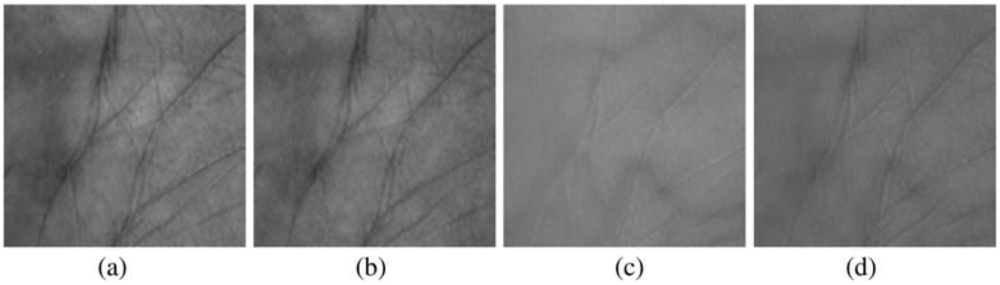

Fig. 3 Samples of multispectral palmprint database: (a) under blue, (b) under green, (c) under NIR, and (d) under red illuminations.

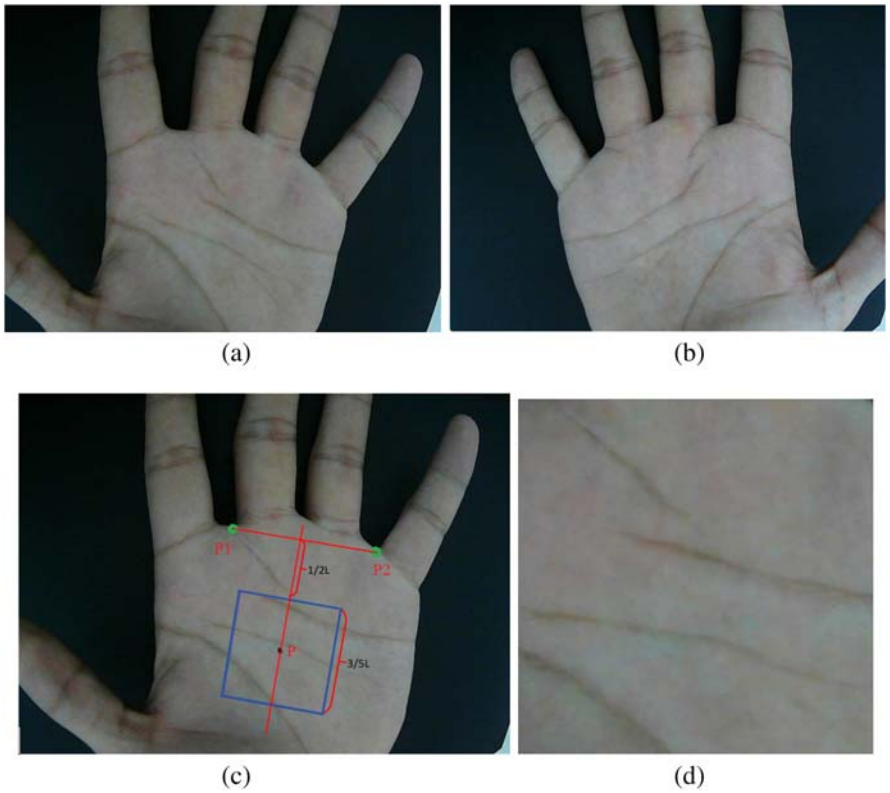

Fig. 4 Samples of XJTU_A palmprint database. (a), (b) palmprint images, (c) the setting of ROI extraction, and (d) an extracted ROI.

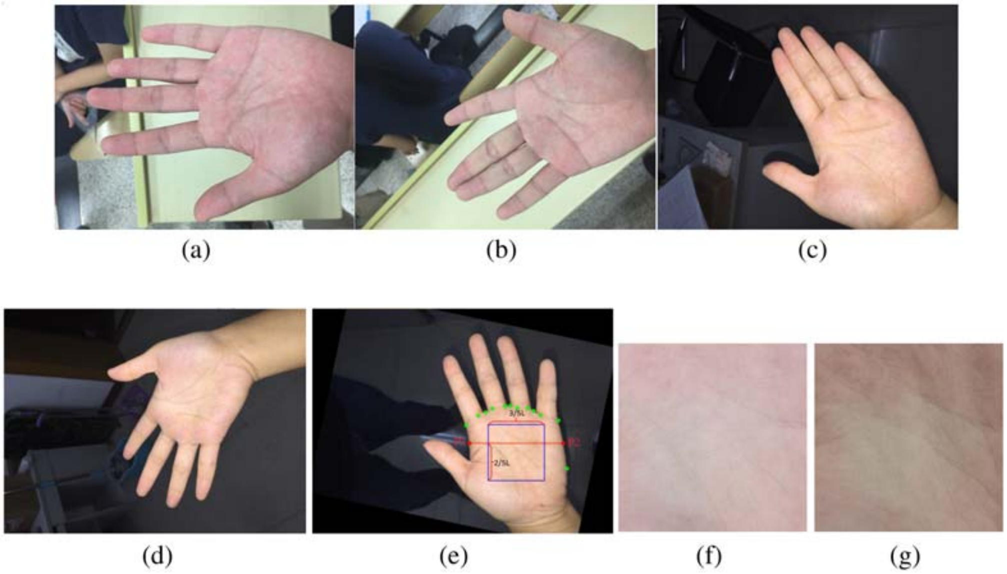

Fig. 5 Samples of XJTU_B palmprint database. (a), (b) under natural illumination (XJTU_BN); (c), (d) under flash illumination (XJTU_BF), (e) the setting of ROI extraction, and (f), (g) two ROIs extracted from the same person.

image of size 3264×2448 pixels. Some samples in different cases are shown in Fig. 5. The databases are the same except for the illuminations, so they can also be divided into two independent databases, which are defined as XJTU_BN (XJTU_B under natural illumination) and XJTU_BF (XJTU_B under flash illumination).

For samples collected in an uncontrolled environment, the boundaries between the palmprint and the background are not obvious, so it is difficult to extract ROI by obtaining the contour of palm. We extracted the ROI manually using the valley points. First, 14 points were manually labeled in each image, including the roots of fingers and the sides of palm, as shown by the green points in Fig. 5(e). Since the relative positions of fingers in the image are known to vary, resulting in unstable positions of valley points, the ROI extracted according to the above method is unstable. Therefore, we then used the relatively stable points on both sides of palm as reference points, as shown by the red points P_1 and P_2 in Fig. 5(e). Assume that the distance between P_1 and P_2 is L. Finally, P_1 and P_2 were used as reference points, and a square with length $\frac{3}{5}L$ was extracted as ROI. The specific setting is shown in Fig. 5(e).

4.2 *Experiment Principles and Results*

In this paper, we mainly focus on open-set palmprint recognition, i.e., the categories in the training and test sets are different. We set the category ratios of training set (G) and test set (P) to $1:1$ and $1:2$, respectively. During training, the training set was inputted into the DEH. After optimizing the loss, the image codes from the same class will become more and more similar, and the difference of image codes of different classes will become larger and larger. In testing, the samples in the test set were inputted into the trained DEH, and the codes obtained by different learners were connected in series as the final codes. Then the Hamming distances between every image and the remaining images in the test set were calculated. If two images with the smallest distance are from the same category, the recognition result is correct; otherwise, it is wrong. The final recognition accuracy is the ratio of the number of correctly identified images over the total number of images in the test set.

At the same time, palmprint verification was also performed. Palmprint verification means that any two images in the test set are compared by calculating the Hamming distance. When two images come from the same category, they are genuine matches, and when they come from different categories, they are imposter matches. For a specific distance threshold, the false acceptance rate (FAR) and false rejection rate (FRR) can be calculated. And when FAR is equal to FRR, EER can also be obtained. Then the detection error trade-off (DET) curve can be generated with FAR and FRR as the horizontal and vertical coordinates, respectively. In general, the smaller the FAR and FRR are, the better the performance is. In a specific algorithm, the FAR and FRR cannot be reduced at the same time. Therefore, EER is often used as a means of evaluation for verification. The smaller the EER is, the better the recognition performance is.

In order to verify the effectiveness of DEH, we conducted a series of experiments. First, we trained the 128-bit DHN as a baseline in the same way as before. Then we divide the 128-bit DHN into three learner combinations. In this paper, we set up two different combinations of learner-20-36-72 and learner-24-48-56. In addition to using the basic ensemble loss and hashing loss, L_{act} and L_{adv} were added to the optimization function, respectively. Finally, the effects of different loss functions on the recognition performance were evaluated separately. In experiments, the trade-off parameters are set to $\alpha = 1$ and $\gamma_w = 5 \times 10^{-3}$. The entire experiments were implemented under the TensorFlow framework. A quad-core Intel CPU i7-5820K and a NVIDIA GPU GTX1080 with 8G memory power were used. The learning rate was set to 0.00005, and each batch contained 64 palmprint images.

4.2.1 *Performance on controlled database*

The controlled database refers to PolyU multispectral database containing four independent subdatabases. On each database, when G:P = 1:1, the training set and the test set contain 250 classes and 3000 images, respectively. When G:P = 1:2, there are 166 classes, 1992 images in the training set, and 334 classes, 4008 images in the test set. In verification, when G:P = 1:1, the test set has $3000 \times 2999/2 = 4{,}498{,}500$ matches, in which 16,500 are genuine matches and 4,482,000 are imposter matches. The results obtained in the four independent subdatabases are shown in Tables 1–4, and the DET curves are shown in Fig. 6.

Table 1 The performances of different methods on the blue palmprint database.

		G:P = 1:1		G:P = 1:2	
Loss	Method	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
Adversarial loss	Baseline-128	98.47	4.93	97.01	4.67
	Learner-20-36-72	98.83	4.45	98.50	4.32
	Learner-24-48-56	99.03	4.30	**98.90**	4.70
Activation loss	Learner-20-36-72	98.90	**3.77**	97.41	4.46
	Learner-24-48-56	**99.30**	4.57	98.20	**4.07**

Table 2 The performances of different methods on the green palmprint database.

		G:P = 1:1		G:P = 1:2	
Loss	Method	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
Adversarial loss	Baseline-128	95.73	6.03	93.69	5.65
	Learner-20-36-72	97.53	5.47	96.23	5.91
	Learner-24-48-56	98.23	4.73	**97.43**	5.24
Activation loss	Learner-20-36-72	98.57	4.30	97.33	**4.65**
	Learner-24-48-56	**98.73**	**3.96**	97.03	4.79

Table 3 The performances of different methods on the red palmprint database.

		G:P = 1:1		G:P = 1:2	
Loss	Method	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
Adversarial loss	Baseline-128	97.63	5.50	96.36	5.47
	Learner-20-36-72	98.53	4.56	98.23	4.91
	Learner-24-48-56	99.23	4.18	**99.30**	**3.51**
Activation loss	Learner-20-36-72	98.80	3.84	98.05	4.19
	Learner-24-48-56	**99.27**	**3.69**	98.70	4.96

Table 4 The performances of different methods on the NIR palmprint database.

		G:P = 1:1		G:P = 1:2	
Loss	Method	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
Adversarial loss	Baseline-128	98.07	4.71	95.36	5.73
	Learner-20-36-72	98.87	3.78	97.75	**3.72**
	Learner-24-48-56	**99.13**	4.44	**98.43**	4.36
Activation loss	Learner-20-36-72	98.57	3.64	97.38	4.13
	Learner-24-48-56	98.87	**3.44**	98.20	4.25

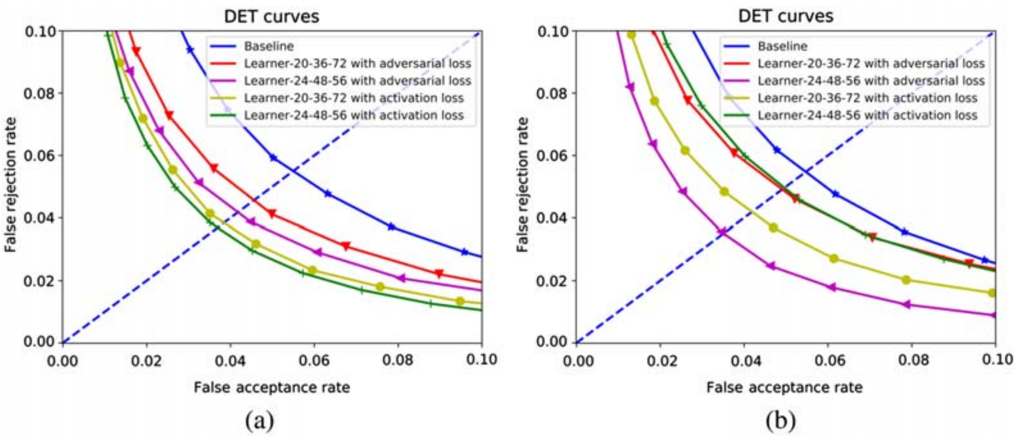

Fig. 6 The DET curves of images under red illumination in the test sets: (a) G:P = 1:1 and (b) G:P = 1:2.

4.2.2 *Performance on uncontrolled database*

In this paper, XJTU_A and XJTU_B were used as uncontrolled palmprint databases. In XJTU_A, when G:P = 1:1, there are 56 and 57 categories in the training set and test set, respectively, and the test set consists of 2565 genuine matches and 159,600 imposter matches. When G: P = 1:2, there are 37 and 76 categories in the training set and test set, respectively, and the test set consists of 3420 genuine matches and 285,000 imposter matches. The recognition results are shown in Table 5 and Figs. 7(a) and 7(b).

Table 5 The performances of different methods on the XJTU_A palmprint database.

		G:P = 1:1		G:P = 1:2	
Loss	Method	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
Adversarial loss	Baseline-128	95.79	**10.56**	85.92	14.61
	Learner-20-36-72	95.09	12.02	91.05	13.64
	Learner-24-48-56	96.14	10.88	**91.91**	13.34
Activation loss	Learner-20-36-72	**96.67**	10.77	90.79	**11.97**
	Learner-24-48-56	96.49	11.61	90.13	13.98

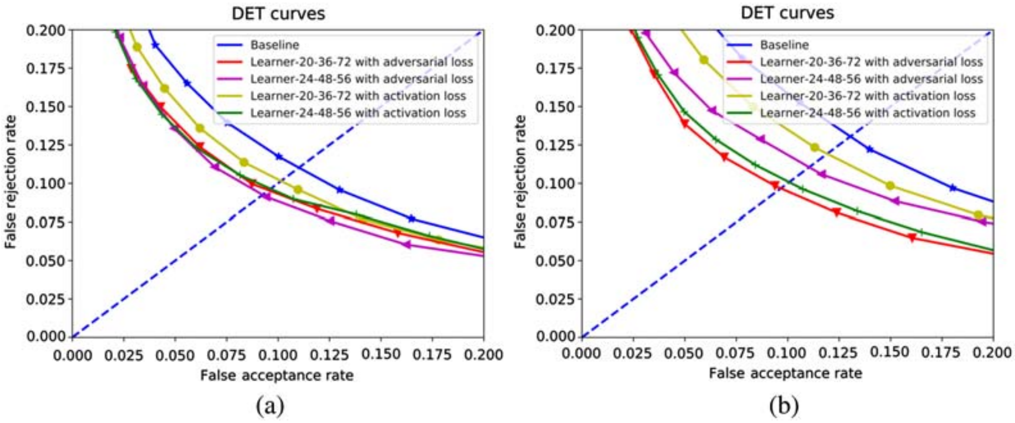

Fig. 7 The DET curves of XJTU_BN in the test sets: (a) G:P = 1:1 and (b) G:P = 1:2.

Table 6 The performances of different methods on the XJTU_BN palmprint database.

		G:P = 1:1		G:P = 1:2	
Loss	Method	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
Adversarial loss	Baseline-128	95.56	11.01	89.24	13.05
	Learner-20-36-72	**96.97**	9.55	95.45	**9.67**
	Learner-24-48-56	97.86	**9.25**	**95.91**	11.0
Activation loss	Learner-20-36-72	96.26	10.14	92.05	11.92
	Learner-24-48-56	96.06	9.63	94.85	10.07

Table 7 The performances of different methods on the XJTU_BF palmprint database.

		G:P = 1:1		G:P = 1:2	
Loss	Method	Accuracy (%)	EER (%)	Accuracy (%)	EER (%)
Adversarial loss	Baseline-128	97.24	8.93	96.34	8.00
	Learner-20-36-72	98.06	8.70	97.48	8.48
	Learner-24-48-56	**98.47**	**7.35**	97.63	6.81
Activation loss	Learner-20-36-72	97.55	8.00	**97.86**	**6.29**
	Learner-24-48-56	98.06	8.50	97.63	6.32

XJTU_B is divided into two separate subdatabases. In XJTU_BN and XJTU_BF, there are 196 categories and 1960 images, respectively. When G:P = 1:1, the training set and test set have 98 (left hand) and 98 (right hand) categories, respectively, and the test set consists of 4410 genuine matches and 475,300 imposter matches. When G:P = 1:2, there are 65 and 131 categories in the training set and test set, respectively, and the test set consists of 2925 genuine matches and 854,470 imposter matches. The recognition results are shown in Tables 6 and 7 and Figs. 7(c)–7(f).

5 Results Analysis

5.1 *Comparison Between DEH with Single DHN (Baseline)*

From the recognition results, DEH outperforms the baseline on the palmprint identification and palmprint verification for different databases. In blue, when G:P = 1:1, the recognition accuracy increased by 0.83%, and EER decreased by 1.16%; when G:P = 1:2, the recognition accuracy increased by 1.89%, and EER decreased by 0.60%. In green, when G:P = 1:1, the recognition accuracy increased by 3.00%, and EER decreased by 2.07%; when G:P = 1:2, the recognition accuracy increased by 3.74%, and EER decreased by 1.00%. In red, when G:P = 1:1, the recognition accuracy increased by 1.64%, and EER decreased by 1.81%; when G:P = 1:2, the recognition accuracy increased by 2.94%, and EER decreased by 1.96%. In NIR, when G:P = 1:1, the recognition accuracy increased by 1.06%, and EER decreased by 1.27%; when G:P = 1:2, the recognition accuracy increased by 3.07%, and EER decreased by 2.01%. In XJTU_A, when G:P = 1:1, the recognition accuracy increased by 5.88%, however, the palmprint verification performance deteriorates; when G:P = 1:2, the recognition accuracy increased by 5.99%, and EER decreased by 2.64%. In XJTU_BN, when G:P = 1:1, the recognition accuracy increased by 1.41%, and EER decreased by 1.76%; when G:P = 1:2, the recognition accuracy increased by 6.67%, and EER decreased by 3.48%. In XJTU_BF, when G:P = 1:1, the recognition accuracy increased by 1.23%, and EER decreased by 1.58%; when G:P = 1:2, the recognition accuracy increased by 1.52%, and EER decreased by 1.71%. The improvements demonstrate the great benefit by combing ensemble learning with DHN.

5.2 *Comparison Between Different Learner Groups*

It can be seen from the experiment results that in most cases, the learner group Learner-24-48-56 has a higher recognition accuracy and lower EER than the Learner-20-36-72 for different databases. This shows that the difference between learners should not be too large. A short learner has limited learning ability and plays a small role in the learner combination. However, the difference between the three learners should not be too small. This is because using online gradient boosting, DEH reweights samples based on the performance of other learners. Samples that are difficult to identify will be focused on learning and given higher weights. However, if the subsequent learner and the previous learner are set to the same, these samples still cannot be correctly identified.

5.3 *Comparison with Different Losses*

In this paper, in addition to using the basic ensemble loss and hashing loss, the effects of activation loss L_{act}, and adversarial loss L_{adv}, on the results are compared. From the experiment results, for palmprint recognition, L_{adv} is slightly better than L_{act}; whereas for palmprint verification, L_{act} is slightly better than L_{adv}. L_{act} allows the neurons of different hashing learners to restrain each other to increase the diversity of learners. L_{adv} draws on the idea of adversarial learning. Through L_{adv}, on the one hand, the distance between the features obtained by different learners is as large as possible to increase the diversity of learners. On the other hand, the distances between features become as small as possible to maintain the recognizability of the features.

5.4 *Comparisons Between Different Databases*

In experiments, there are four independent controlled databases, i.e., blue, green, red, and NIR, and three independent uncontrolled databases, i.e., XJTU_A, XJTU_BN, and XJTU_BF. From the experiment results, the performances of controlled databases are better than those of uncontrolled databases, especially for palmprint verification. The EERs in the controlled environment are around 4.00%, whereas the EERs in XJTU_A and XJTU_B are around 10.00%. There are two reasons. First, in the controlled databases, there are 6000 samples from 500 categories, far more than the uncontrolled databases. There are 98 categories of 1960 samples in the subdatabase of XJTU_B, and in XJTU_A there are only 1130 samples of 57 categories. In fact, palmprint recognition with unseen samples is more difficult. The more unseen samples in the test set, the higher the requirements for the generalization capabilities of the algorithm. It is also reflected in the experiment results of different training sets and test sets in the same database. For different training sets, in most cases, when the loss function and the learner combination are the same, the performances of G:P = 1:1 are better than that of G:P = 1:2. At G:P = 1:2, there are fewer training samples and more test samples, which increase the diversity of test samples and requires the model to have a strong generalization ability. On the other hand, uncontrolled samples are highly heterogeneous. Illuminations, angles, and noises vary widely, especially under natural illumination. As shown in Fig. 5, visually, the textures of the two ROIs collected from the same person are significantly different. The two reasons lead to the performances of uncontrolled databases that are not as good as multispectral database. However, on uncontrolled databases, the recognition accuracies are up to 98.47%, which shows the efficiency of DEH for palmprint recognition with unseen classes.

5.5 *Comparisons with Other Works*

In order to demonstrate the effectiveness of our algorithm, we conducted several experiments to compare the accuracy and speed with other methods. In benchmark dataset, blue palmprint

Table 8 The comparisons of different methods on the blue and XJTU_BF palmprint databases.

	Blue (%)		XJTU_BF (%)	
Method	G:P = 1:1	G:P = 1:2	G:P = 1:1	G:P = 1:2
Baseline-128	98.47	97.01	97.24	96.34
Alexnet	87.49	88.49	78.57	77.52
VGG-16	91.78	89.28	74.15	77.69
Adversarial loss + learner-20-36-72	98.83	98.50	98.06	97.48
Adversarial loss + learner-24-48-56	99.03	**98.90**	**98.47**	97.63
Activation loss + learner-20-36-72	98.90	97.41	97.55	**97.86**
Activation loss + learner-24-48-56	**99.30**	98.20	98.06	97.63

Table 9 The speed comparisons of different methods on the blue and XJTU_BF palmprint databases.

Method	Feature extraction (ms)	Feature matching (ms)
Baseline-128	0.01000	0.0019
Alexnet	0.00456	0.0142
VGG-16	0.01000	0.0142
Adversarial loss + learner-20-36-72	0.01002	0.0019
Adversarial loss + learner-24-48-56	0.01010	0.0020
Activation loss + learner-20-36-72	0.00997	0.0018
Activation loss + learner-24-48-56	0.01009	0.0019

database, and our self-built dataset, XJTU_BF palmprint database, we perform palmprint identification with different category ratios of training set and test set, i.e., 1:1 and 1:2. Table 8 shows the identification accuracy and Table 9 shows the speeds of different steps. For Alexnet[51] and VGG-16,[14] we adopted cross entropy loss to train the models. In the test, the outputs of last layer were used as features. From the results, our algorithms with different learners and different losses can obtain better accuracy. For the speed, our model is based on VGG-16, so the feature extraction time may be longer. However, due to the using of hashing codes and Hamming distance, feature matching will be more efficient. To sum up, DHN can obtain better performance for open-set palmprint recognition.

6 Conclusions

In this paper, for the palmprint recognition with unseen classes, we proposed an ensemble learning module, named DEH. Based on DHN, DEH adopted the online gradient boosting model to train multiple hashing learners. Each hashing learner resets the weights of samples based on the negative gradient of the loss function of previous learners. Different samples were given different weights, difficult samples were set with larger weights, and easy samples were set with small weights. In order to increase the diversity of learners and reduce their relevance, adversarial loss and activation loss were introduced to the optimization goal. In activation loss, neurons between different learners restrained each other. In adversarial loss, the distances of features extracted by different learners were constrained. On the one hand, the difference between different features should be as large as possible to increase the diversity between different weak learners. On the other hand, different features should be as similar as possible to maintain recognizability. In order to promote the practical application of palmprint recognition, we established two databases in uncontrolled environments, one of which was collected by mobile phone. Experiments were conducted on the multispectral database and our self-built databases. In experiments, sample sets with different classes were used as training sets and test sets, respectively, and different loss functions were adopted. The results show that DEH can achieve better performance than baseline, and the recognition accuracy was increased by up to 6.67%.

7 Future Work

About future work, we can further reduce the correlation between different learners and extract more discriminative features to improve the performance of DEH. On the one hand, an ambiguity penalty term may be introduced based on the deviation between the predictions of different weak learners and the ensemble prediction. In general, for a specific sample, if weak learners deviate from the ensemble prediction, the ambiguity will be high, which can be adopted to update the weights of samples for successive learners. On the other hand, weak DHN with more parameters and layers can be adopted. In this paper, DHN based on VGG-16 is used as feature extractor, which can obtain promising results. When the requirements for recognition efficiency are not high, other complex CNNs can be adopted as the main net to extract more discriminative features. In addition, our DEH has three weak learners and extracts 128-bit codes. More learners and higher dimensional codes can also be adopted as a possible way, where the diversity of learners needs to be further considered.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61105021 and 61673157) and the Natural Science Foundation of Zhejiang Province (No. LGF19F030002).

References

1. A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.* **14**(1), 4–20 (2004).
2. A. K. Jain and J. Feng, "Latent palmprint matching," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(6), 1032–1047 (2009).
3. L. Fei et al., "Low-rank representation integrated with principal line distance for contactless palmprint recognition," *Neurocomputing* **218**, 264–275 (2016).
4. L. Zhang et al., "Towards contactless palmprint recognition: a novel device, a new benchmark, and a collaborative representation based identification approach," *Pattern Recognit.* **69**, 199–212 (2017).
5. K. Zhang et al., "Improving texture analysis performance in biometrics by adjusting image sharpness," *Pattern Recognit.* **66**, 16–25 (2017).
6. A. Kumar and S. Shekhar, "Personal Identification using multibiometrics rank-level fusion," *IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev.* **41**(5), 743–752 (2011).
7. A. W. K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *17th Int. Conf. Pattern Recogn. (ICPR)*, J. Kittler, M. Petrou, and M. Nixon, Eds., pp. 520–523 (2004).
8. L. Fei et al., "Double-orientation code and nonlinear matching scheme for palmprint recognition," *Pattern Recognit.* **49**, 89–101 (2016).
9. D. Zhong, Y. Yang, and X. Du, "Palmprint recognition using siamese network," in *13th Chin. Conf. Biom. Recognit., CCBR*, pp. 48–55 (2018).
10. Z. Xie, Z. Guo, and C. Qian, "Palmprint gender classification by convolutional neural network," *IET Comput. Vision* **12**(4), 476–483 (2018).
11. H. Song et al., "Deep metric learning via lifted structured feature embedding," in *29th IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, pp. 4004–4012 (2016).

12. J. Zheng et al., "Deep ensemble machine for video classification," *IEEE Trans. Neural Networks Learn. Syst.* **30**(2), 553–565 (2019).
13. O. Araque et al., "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.* **77**, 236–246 (2017).
14. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, California, pp. 1–14 (2015).
15. L. Breiman, "Random forests," *Mach. Learn.* **45**(1), 5–32 (2001).
16. A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognit.* **42**(7), 1408–1418 (2009).
17. D. Zhang et al., "Palmprint recognition using 3-D information," *IEEE Trans. Syst. Man Cybern. Part C-Appl. Rev.* **39**(5), 505–519 (2009).
18. F. Yue et al., "Orientation selection using modified FCM for competitive code-based palmprint recognition," *Pattern Recognit.* **42**(11), 2841–2849 (2009).
19. H. Wang et al., "Person recognition by fusing palmprint and palm vein images based on 'Laplacianpalm' representation," *Pattern Recognit.* **41**(5), 1514–1527 (2008).
20. D. Zhong, X. Du, and K. Zhong, "Decade progress of palmprint recognition: a brief survey," *Neurocomputing* **328**, 16–28 (2019).
21. W. Jia, D. Huang, and D. Zhang, "Palmprint verification based on robust line orientation code," *Pattern Recognit.* **41**(5), 1504–1513 (2008).
22. Z. Guo et al., "Palmprint verification using binary orientation co-occurrence vector," *Pattern Recognit. Lett.* **30**(13), 1219–1227 (2009).
23. L. Zhang, H. Li, and J. Niu, "Fragile bits in palmprint recognition," *IEEE Signal Process. Lett.* **19**(10), 663–666 (2012).
24. J. Dai, J. Feng, and J. Zhou, "Robust and efficient ridge-based palmprint matching," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(8), 1618–1632 (2012).
25. Y. Xu et al., "Discriminative and robust competitive code for palmprint recognition," *IEEE Trans. Syst. Man Cybern. -Syst.* **48**(2), 232–241 (2018).
26. S. Benjoudi et al., "Palmprint identification performance improvement via patch-based binarized statistical image features," *J. Electron. Imaging* **28**(5), 053009 (2019).
27. J. Cheng et al., "Supervised hashing with deep convolutional features for palmprint recognition," in *12th Chin. Conf. Biom. Recogn., CCBR*. pp. 259–268 (2017).
28. M. Izadpanahkakhk et al., "Joint feature fusion and optimization via deep discriminative model for mobile palmprint verification," *J. Electron. Imaging* **28**(4), 043026 (2019).
29. K. Bensid et al., "Deep learning feature extraction for multispectral palmprint identification," *J. Electron. Imaging* **27**(3), 033018 (2018).
30. D. Zhong, H. Shao, and X. Du, "A hand-based multi-biometrics via deep hashing network and biometric graph matching," *IEEE Trans. Inf. Forensic Secur.* **14**(12), 3140–3150 (2019).
31. H. Shao, D. Zhong, and X. Du, "Cross-domain palmprint recognition based on transfer convolutional autoencoder," in *IEEE Int. Conf. Image Process.* pp. 1153–1157 (2019).
32. H. Shao, D. Zhong, and Y. Li, "PalmGAN for Cross-domain Palmprint Recognition," in *IEEE Int. Conf. Multimedia and Expo (ICME)*, pp. 1390–1395 (2019).
33. H. Lai et al., "Simultaneous feature learning and hash coding with deep neural networks," in *28th IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3270–3278 (2015).
34. D. Zhong, H. Shao, and Y. Liu, "Hand dorsal vein recognition based on deep hash network," in *First Chin. Conf. Pattern Recognit. and Comput. Vision (PRCV)*, pp. 26–37 (2018).
35. Z. Zhang, Y. Chen, and V. Saligrama, "Efficient training of very deep neural networks for supervised hashing," in *29th IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1487–1495 (2016).
36. H. Yang, K. Lin, and C. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(2), 437–451 (2018).
37. H. Venkateswara et al., "Deep hashing network for unsupervised domain adaptation," in *30th IEEE Conf. Comput. Vision and Pattern Recognit.* pp. 5385–5394 (2017).
38. L. Breiman, "Bagging predictors," *Mach. Learn.* **24**(2), 123–140 (1996).

39. Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *J. Comput. Syst. Sci.* **55**(1), 119–139 (1997).
40. I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *28th Conf. Neural Inf. Process. Syst. (NIPS)*, Z. Ghahramani et al., Eds., pp. 3104–3112 (2014).
41. H. Xuan, R. Souvenir, and R. Pless, “Deep randomized ensembles for metric learning,” in *15th Eur. Conf. Comput. Vision*, pp. 751–762 (2018).
42. M. Opitz et al., “Deep metric learning with BIER: boosting independent embeddings robustly,” *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(2), 276–290 (2020).
43. Z. Cao et al., “HashNet: deep learning to hash by continuation,” in *16th IEEE Int. Conf. Comput. Vision (ICCV)*, pp. 5609–5618 (2017).
44. C. Burges et al., “Learning to rank using gradient descent,” in *Proc. 22nd Int. Conf. Mach. Learn.*, ACM, Bonn, Germany, pp. 89–96 (2005).
45. U. Evgeniya and L. Victor, “Learning deep embeddings with histogram loss,” *Adv. Neural Inform. Process. Syst.* **29**, 4170–4178 (2016).
46. E. Ustinova and V. Lempitsky, “Learning deep embeddings with histogram loss,” in *30th Conf. Neural Inf. Process. Syst.*, Barcelona Spain, pp. 1–9 (2016).
47. A. Beygelzimer et al., “Online gradient boosting,” in *29th Annu. Conf. Neural Inf. Process. Syst.*, C. Cortes et al., Eds., pp. 1–18 (2015).
48. G. Brown et al., “Diversity creation methods: a survey and categorisation,” *Inf. Fusion* **6**(1), 5–20 (2005).
49. I. J. Goodfellow et al., “Generative adversarial nets,” in *28th Conf. Neural Inf. Process. Syst. (NIPS)*, Z. Ghahramani et al., Eds., pp. 1–9 (2014).
50. D. Zhang et al., “An online system of multispectral palmprint verification,” *IEEE Trans. Instrum. Meas.* **59**(2), 480–490 (2010).
51. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *26th Annu. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, Nevada, pp. 1106–1114 (2012).

Huikai Shao received his BSc degree from Chongqing University in 2017. He is a PhD candidate at the School of Electronic and Information Engineering of Xi’an Jiaotong University. His main research interests include biometrics and computer vision.

Dexing Zhong received his BSc and PhD degrees from Xi’an Jiaotong University, China, in 2005 and 2010, respectively. He is an associate professor at the School of Electronic and Information Engineering of Xi’an Jiaotong University. He was a visiting scholar at the University of Illinois at Urbana–Champaign, USA. His main research interests include biometrics and computer vision.

Xuefeng Du is currently pursuing his bachelor’s degree at the School of Electronic and Information Engineering of Xi’an Jiaotong University. He is participating in the Information-Technology Talent Program sponsored by the school.