LOW-SHOT PALMPRINT RECOGNITION BASED ON META-SIAMESE NETWORK

Xuefeng Du[1], Dexing Zhong[1,2], Pengna Li[1]

[1]School of Electronic and Information Engineering, Xi'an Jiaotong University Xi'an, Shaanxi 710049, P.R. China
[2]Research Institute of Xi'an Jiaotong University, Zhejiang, Hangzhou, 311215, P.R. China
bell@xjtu.edu.cn

ABSTRACT

Palmprint is one of the discriminant biometrical features of humans. Recognizing palmprints in complex environments is a significant multimedia task, which is highly suitable for applications in information security and forensics. Recently, deep learning-based recognition methods have improved the accuracy and robustness of recognition results to a new level. However, obtaining the required large amount of training data and labels is impracticable in practical scenarios. Therefore, in this paper, we exploit few-shot learning for palmprint recognition. We propose Meta-Siamese network based on Siamese network. Specifically, we train this network episodically with a more flexible framework to learn both the feature embedding and the deep similarity metric function. Moreover, we extend our model to zero-shot recognition tasks based on deep hashing network. Experiment result shows competitive improvements compared to baseline methods in eight different datasets.

Index Terms— Palmprint recognition, low-shot learning, biometrics, information security.

1. INTRODUCTION

Palmprint recognition, as one of the emerging technologies of biometrics, has received more research attention recently [1, 2, 22]. Generally, researchers applied signal processing methods to analyze the pattern of palmprints for personal authentication, which is suitable for safety and forensics appliations. So far, principal lines [3] and local texture [4] have been exploited for feature representation. They are time-invariant with large inter-class variance and low intra-class variance. Therefore, promising recognition results have been achieved with high universality, stability and uniqueness compared to other biometrical features, such as iris and face.

Typical procedure of palmprint recognition consists of images acquisition, preprocessing, feature extraction and matching, where feature representation is the most important step. After obtaining the Region of Interest (ROI), several categories of extraction methods have been proposed to separate different identities, namely encoding-based methods, structure-based methods, statistics methods, subspace methods and deep learning methods. So far, deep learning techniques have emerged as effective tools for automatic visual understanding. Several deep learning models for palmprint recognition are proposed [5, 6] which outperform traditional methods by a large margin.

One of the major problems when applying these models is large amounts of training data and labels. On the one hand, labelling is a heavy work and sometimes the identities are missing in the acquisition stage. On the other hand, training on large amounts of data is computationally expensive with poor generalization ability. Therefore, constructing an accurate recognition system where less training data are required is a significant task. The general definition of such a small-sample-size (SSS) problem, which is also called low-shot recognition, is formulated as follows: given a training set S consisting of palmprint images from N categories, $S = \{x_0^0, ..., x_{k-1}^0, ..., x_0^{N-1}, ..., x_{k-1}^{N-1}\}$. For each category, we have k samples (k is a small number). The task is how can we recognize the rest images in each category using the trained neural network without overfitting. One effective method is meta-learning, which aims to train the neural networks for generalizing on different tasks, such as classification and regression [7]. These algorithms intend to learn a good initialization for each task such that when facing new data, the neural network can update to an optimal state through a few gradient descent steps.

In this paper, we propose a low-shot palmprint recognition method called Meta-Siamese network based on [5], which uses Siamese Network for comparing two images. In order to help neural network generalize to new data, we follow the original rationale of Siamese network which first extracts features from image pairs using weight-shared Convolutional Neural Network (CNN). Then we input the concatenated feature vector to decision network (fully connected layers, denoted as FC layers) to obtain their similarity. The losses are then backpropagated so that given any palmprint image pairs, we can verify whether they are from the same identity.

Different from the original training method, we randomly sample N categories during training, and then select k samples from each of these categories for comparison with the remaining images (denoted as N-way, k-shot episode training [8]). In the testing stage, we use unseen categories to test the effectiveness of our method. We also randomly select N categ-

978-1-5386-1737-3/18/$31.00 ©2019 IEEE

ories and k samples from each. Given only these $N*k$ images with known identities, we test if the neural network can distinguish other images either from the same categories or from a different category. Specially, to increase the flexibility, we incorporate the convolutional blocks in the decision network instead of pure stacked FC layers such that the entire neural network can adapt quickly. Besides, we also introduce zero-shot palmprint recognition where no image samples are seen by the neural network. Experiments on eight different datasets reveals the outperforming recognition accuracy and generalization ability of our model.

The rest of the paper is organized as follows: Section 2 reviews the related works. Section 3 discusses our proposed low-shot learning method. Section 4 describes comparative experiments and results. Section 5 concludes the paper.

2. LITERATURE REVIEW

Palmprint recognition has been researched over many years. Traditional computer vision techniques have been researched thoroughly. PCA [9], LBP [10] and Competitive Code [11] are popular feature representation methods which focuses on the local or global structure of the palmprint. Sometimes, hand-engineered features cause the information loss, so feature fusion has been introduced, which combines different recognition object [10], matching scores [12], *etc*.

For better image understanding, CNN has been proposed and applied in many vision tasks, such as classification, object detection, segmentation, *etc*. It uses spatial filters to compute the weighted sum in image patches. Then the FC layers are employed for dimension transformation and regression, which causes huge computation complexity. As an application of meta-learning, few-shot learning is introduced to solve the problem of overfitting when deep neural networks are faced with sparse data, which samples auxiliary tasks for helping neural networks generalize. There exist three types of methods, namely Recurrent Neural Network (RNN)-based methods, initialization-based and metric-based models. RNN-based methods take the advantage of memory. It trains RNN models to remember the seen tasks. When facing tasks with a different distribution, it compares with its memory [13]. The second model aims to let neural network learn to update its parameters within a few gradient steps, therefore the sampled tasks in training resemble the task setting in testing [8]. The metric-based models learn the image embedding functions such that after embedding, the subspace features are easier to classify.

The improvement of our work compared to [14] and [5] is that we add meta episode training to improve the generalization ability. Besides, we add CNN blocks in the decision network which enables more flexible training. Compared to [13], our model doesn't require designing complex memory encoding and decoding structure. Besides, we extend our model to zero-shot palmprint recognition.by modifying the embedding structure similar to [15].

3. METHOD

3.1. Task description

The low-shot palmprint recognition can be formulated as training a classifier to recognize the remaining images given a few training images in each category. Suppose we have M image samples in total. We denote the small support set as $(x_i, y_i), i = 1,...,N*k$ for training, where we have N classes and k samples in each class. The testing set with the same label space is denoted as $(x_i, y_i), i = N*k+1,...,M$. Therefore, if directly trained, the model will suffer from overfitting. Hence, we introduce the meta-train set which is separated to sample set and query set to simulate the low-shot recognition setting in the meta-test set. Note that the label space of the sample/query set is different from that of support/testing set. We randomly sample N-way, k-shot tasks into the sample set and test on the query set. The losses we obtain on the sample/query set are backpropagated to make the system easier to adapt to new tasks. Therefore, after the episode-based training in the meta-train set, we apply our model to the meta-test set for low-shot palmprint recognition.

3.2. One-shot palmprint recognition

For one-shot palmprint recognition, we firstly begin by describing our model structure. The Meta-Siamese network is shown in Fig.1.

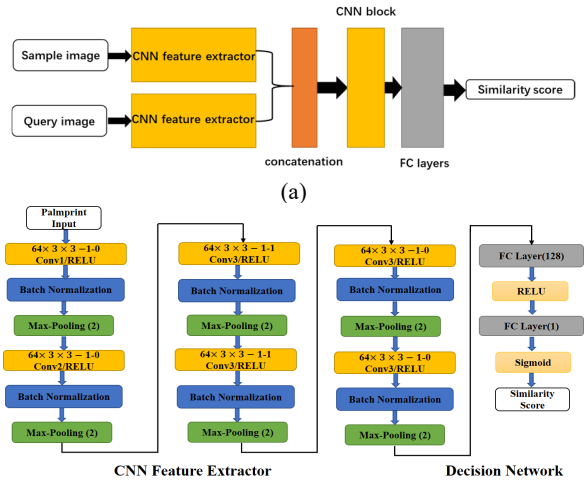


Fig.1. Framework of our Meta-Siamese network
(a)Structure of our Meta-Siamese network. (b) Structure of two sub-modules, namely CNN feature extractor and the decision network. The configuration of each layer is shown in the box. For instance, the "64×3×3-1-0" means we have 64 convolutional filters with 3×3 kernel size, 1 stride and 0 padding. "FC layer (128)" means we have 128 hidden units in the FC layer. Activation functions and batch normalization layer are also demonstrated in each box.

We first sample one image from N classes in meta-train set to form the sample set $\{(x_0^0, y_0^0), (x_0^1, y_0^1), ..., (x_0^{N-1}, y_0^{N-1})\}$, the remaining images of N-1 classes are denoted as the query set $\{(x_i^j, y_i^j)\}, j = 0,...,N-1, i = 1,...,K-1$. Here we suppose each

class consists of K images. Every time, we input one image in the sample set and one image in the query set to the CNN feature extractor module $f(\cdot)$ with the same weights. Then, we obtain two feature vectors, $f(x_0^j), f(x_i^s)$. We concatenate them together in the depth dimension and obtain concatenated feature map $C(f(x_0^j), f(x_i^s)), i \neq 0$. By down-sampling and dimension transformation in the decision network $D(\cdot)$, we obtain a similarity score with respect to the current image pair. When the score is higher, it means the two images are more likely to belong to the same class. Therefore, during one episode training, we compare the similarity between different image pairs for $N \times N \times (K\text{-}1)$ times.

For parameter optimization, we construct a one-hot vector denoting the ground-truth value in every training episode, which is a $N \times N \times (K\text{-}1)$ dimensional vector Y. The t-th value of Y is formulated as follows:

$$Y_t = \begin{cases} 1 & \text{if } t\text{ th image pair is from the same class} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We let the predicted similarity to be as follows:

$$\hat{Y}_t = D(C(f(x_0^j), f(x_i^s))) . \quad (2)$$

Therefore, the overall loss in one episode is as follows:

$$Loss = \sum_{t=1}^{N \times N \times (K-1)} (\hat{Y}_t - Y_t)^2 . \quad (3)$$

Where we simply calculate the Mean Squared Error (MSE) for gradient descent.

At the meta-test stage, we construct the support set and testing set which is similar to the sample set and query set. For each of the $N \times (K\text{-}1)$ testing images, we compare it to every image in the support set and select the image in the support set which generates the highest similarity score. We then compare the label of the support set image with that of the testing image to decide if our model generates the right recognition result.

3.3. *k*-shot palmprint recognition

For k-shot palmprint recognition, similar to one-shot recognition, we sample k instead of one image from each class and regard the remaining images as the query images. The difference is that we input k sample images along with query images to the Meta-Siamese network at each episode training. In order to let neural network see more information of a certain class, we sum over all k feature maps of the sample images and then obtain the averaged feature map. Afterwards, we concatenate the averaged feature vector with feature from the query images. After that, the following matching and comparison procedure are the same as the one-shot palmprint recognition.

3.4. Zero-shot palmprint recognition

To further extend the performance of our model, we modify it to deal with zero-shot palmprint recognition tasks. Instead of letting Meta-Siamese network see one or k sample images, we follow the common procedure in zero-shot recognition [16]. Typical zero-shot tasks usually take advantage of labelled training data and the knowledge that how some of the unseen classes are semantically related to the class being learned. It aims to distill the information from the semantic attributes and project the semantic vectors to the feature map space such that classification can be performed using unsupervised methods.

In our scenario, we first obtain the semantic embeddings for every class in the dataset based on deep hashing network [17]. We denoted the obtained attributes as $\{A_0, ..., A_{N-1}\}$, which is a 128-dimensional vector that has large inter-class variance and small intra-class variance. In the training stage, we use CNN module $f_1(\cdot)$ to obtain the feature map for query set and we introduce a warping module $f_2(\cdot)$ which are two FC layers to project the attributes of sample set to feature map space. Then we apply feature concatenation and use the following decision network for generating the similarity score, which is formulated in Equation (4).

$$Similarity\ \ Score = D(C(f_1(x_i^s), f_2(A^j))) . \quad (4)$$

Where similarity score means the similarity between the j-th class and the i-th image in the s-th category.

The entire structure is demonstrated in Fig. 2.

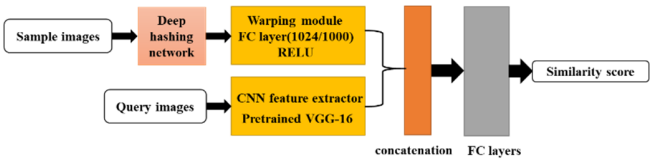


Fig.2. Structure of zero-shot palmprint recognition. Unspecified module has the same configuration as the few-shot recognition. Note that the warping module consists of two fully connected layers with the hidden units to be 1024 and 1000, respectively.

In the testing stage, we test two versions of zero-shot recognition as described in [15], which is pure zero-shot recognition and generalized zero-shot recognition. The testing and support set of the former one consists of only unseen classes while the latter one adds the attributes of seen classes into the support set.

To illustrate the generation of the semantic embeddings, we apply deep hashing network (DHN) [18] whose input is a palmprint image with the output to be a binary vector consisting of 1 and -1. The general architecture of DHN is a stack of convolutional layers plus fully connected layers. Then a sign function is added to force the output values to be either -1 or 1. Specially, the two loss functions of DHN are hash loss and quantization loss where the first one forces the neural net to encode the palmprint image discriminatively and the second one forces every bit of the output vector equals -1 or 1, which are shown as follows:

Hash loss.

$$J(U_i, U_j, T_{ij}) = \min[\frac{1}{2} T_{ij} D_h(U_i, U_j) + \frac{1}{2}(1 - T_{ij}) \max(M - D_h(U_i, U_j), 0)] \quad (5)$$

=

Where U_i denotes the binary outputs of *i*-th image. $D_h(U_i,U_j)$ represents the Euclidean distance between two hash codes. T_{ij} is 1 if *i*-th and *j*-th image belongs to the same category. The left term in Equation (5) attempts to minimize the distance between samples within the same class while the right term tries to maximize the distance between samples in different classes. M is the threshold parameter for invalidating the hash loss if the distance is larger than M . Here we set the value of M to be 180. The final hash loss is formulated as follows:

$$J_H = \sum_{i=1}^{N}\sum_{j=1}^{N} J(U_i,U_j,T_{ij}) \ . \tag{6}$$

Quantized loss.

When transforming the probability vector into the discrete binary hash codes, we use the function (7) which generates error between the image features and the final codes. To force the output to be hash code, the quantization loss is formulated as follows:

$$J_Q = \sum_{i=1}^{N} \frac{1}{2}(\| 1-|U_i| \|_2) \ . \tag{7}$$

Where $|U_i|$ denotes the element-wise absolute value of DHN output U_i and $\|\cdot\|_2$ stands for L2-norm of vector. By minimizing the following quantization loss J_Q, the output is closer to 1 or -1.

By combing the above two losses, the final loss of DHN is formulated as Equation (8). Note that α is a hyperparameter weighting the two losses.

$$\min J = \alpha J_H + J_Q \ . \tag{8}$$

4. EXPERIMENTS AND RESULTS

4.1. Datasets

4.1.1. Multispectral palmprint dataset

In this paper, we use PolyU multispectral palmprint dataset which contains four spectral bands, *i.e.* blue, red, green and NIR. Under each spectral band, 6000 grayscale images from 500 different palms of 250 people are collected [19], including 195 males and 55 females. For each palm, 12 images are collected in two sessions, among which 6 of them is collected in the first session and the remaining are collected in the second session. The palmprint images in this dataset is oriented and cropped to form the ROI (Region of Interest) whose size is 128×128 pixels.

4.1.2. Palmprint dataset in the open environment

To simulate the image acquisition environment in the unguided scenario (with no fixed pegs), we set up a palmprint dataset where the users can freely move their hands upon acquisition. The PA image dataset is built in the lab environment with proper illumination and low noise level while the PB is constructed outdoors so the quality of the images is lower. The images in dataset PC and PD is collected by mobile phones with and without flash light in the RGB form.

In PA, we use CMOS camera to collect 57 people's grayscale images of both left and right palms. For each palm, we collect 10 images. In PB, we collect 1000 RGB images from 100 different people. The ROI extraction method is implemented using the same method in [5]. In PC, we collected 1960 images from 196 different people with flash light. In PD, we collected 1970 images from 197 different people. The exemplary images and their ROIs are demonstrated in Fig.3.

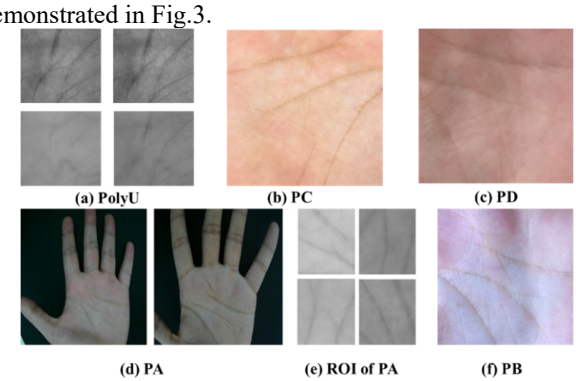


Fig.3. Different datasets used in our experiments

4.2. Few-shot recognition

First, we randomly split the dataset into the meta-train set and meta-test set with the ratio of 4:1. In the training stage, we test our model using 8 different datasets on different task setting, which are 5-way/1-shot, 5-way/3-shot, 5-way/5-shot, 15-way/1-shot, 15-way/3-shot and 15-way/5-shot recognition tasks. For episode-based training, we first train Meta-Siamese network by sampling tasks in sample/query set. Every 5000 episodes, we start testing on the support/testing set for 600 episodes and calculate the mean recognition accuracy with the confidence level to be 0.95.

For 5-way/1-shot recognition scenario, the size of sample set is 5 while the size of the query set is 55 for PolyU dataset and 45 for our datasets because we have 12 images in a class in PolyU dataset and 10 images in each class for others. The recognition results in presented in Table.1.

Table 1. Few-shot recognition results

Dataset	5-way			15-way		
	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
PolyU-B	97.2%	99.2%	99.4%	96.8%	97.9%	97.3%
PolyU-R	96.9%	98.9%	99.3%	96.1%	96.2%	95.6%
PolyU-G	95.4%	97.3%	98.7%	94.7%	95.8%	95.9%
PolyU-N	94.3%	95.6%	99.7%	94.1%	93.9%	95.3%
PA	93.5%	94.6%	95.3%	91.6%	95.4%	94.8%
PB	87.8%	88.8%	93.4%	84.2%	86.9%	92.7%
PC	97.3%	96.3%	98.8%	95.6%	96.6%	98.3%
PD	94.2%	95.8%	96.4%	91.5%	91.9%	94.7%

From the given table, our recognition accuracy is close to the state-of-the-art's [5] with only a few seen samples in training. When the number of seen images during training increases, the testing accuracy increases as well because more

knowledge of a certain class is obtained by Meta-Siamese network. In the PolyU palmprint dataset, the testing accuracy is higher than that of real-world dataset. The reason is that the images in the former database is collected in a controlled environment where the illumination and noise level is almost fixed between classes. While in our dataset, the images either from the same class or not vary a lot in illumination, noise level, viewpoint, *etc.*, which causes a potential loss in recognition accuracy. Besides, the model behaves less effectively in 15-way few-shot recognition because the knowledge acquired by sampling tasks when training may not be specific enough for handling classification problem among larger amounts of classes.

For comparison, we present the results of some baseline methods in the 5-way/1-shot recognition, namely Siamese network (S-Net) [5], Model-Agnostic Meta-Learning (Maml) [7], Prototypical Nets [20] (P-Net), DHN [17] and Matching Net (M-Net) [21]. Their corresponding recognition results are shown in Table.2.

Table 2. Comparative results of few-shot recognition

Dataset/ Shot	S-Net	Maml	P-Net	**Ours**	M-Net	DHN
	1	1	1	1	1	1
PolyU-B	66.7%	94.1%	94.3%	**97.2%**	83.6%	54.3%
PolyU-R	64.6%	93.8%	92.8%	**96.9%**	85.9%	62.4%
PolyU-G	69.5%	93.5%	91.7%	**95.4%**	88.0%	66.5%
PolyU-N	70.1%	96.7%	90.5%	**94.3%**	87.7%	64.1%
PA	71.5%	94.5%	88.5%	**93.5%**	91.2%	62.9%
PB	50.0%	84.5%	86.0%	**87.8%**	89.3%	43.8%
PC	61.2%	94.4%	93.7%	**97.3%**	88.5%	59.9%
PD	58.9%	98.9%	90.3%	**94.2%**	90.2%	47.3%

From the table, our model achieves a competitive result compared to popular low-shot recognition methods, namely M-Net, Maml and P-Net. Compared to models without special design for low-shot recognition, our model performs much better in all eight datasets. Note that all of modules are implemented using similar hyperparameters with a slight difference in each model to reach the best performance respectively.

4.3. Zero-shot recognition

We apply three different application scenarios in zero-shot recognition as in [8, 16]. The first is the support/testing set contains different identities from sample/query set. The second one is that the support set contains semantic attributes of all classes and the testing set only contains the feature maps from unseen classes. The third is that the testing set contains all the seen classes while the support set consists of semantic attributes of all classes. We denoted all three setting as pure zero-shot, generalized zero-shot with unseen classes and generalized zero-shot with seen classes.

We first obtain the feature maps of testing set using the pretrained VGG-16. Then, the semantic attributes of all classes are generated by pretrained DHN. Every image is encoded as a 128-bit hash code which contains invariant semantic information. After every 2000 episodes, we start testing on the support/testing set for 1000 episodes and calculate the recognition accuracy with the confidence level to be 0.95. The recognition performance is presented in Table.3.

Table 3. Zero-shot recognition results

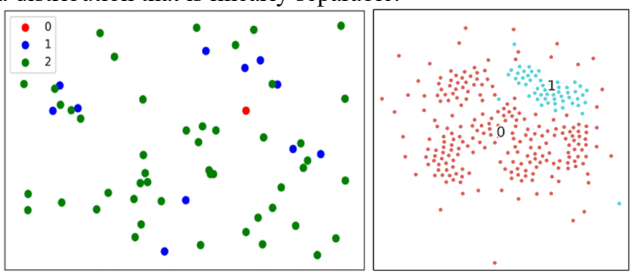
Dataset	Pure zero-shot	Generalized zero-shot	
		Unseen	Seen
PolyU-B	65.9%	45.9%	90.4%
PolyU-R	71.3%	54.6%	89.4%
PolyU-G	69.7%	33.7%	92.1%
PolyU-N	71.4%	49.5%	88.1%
PA	35.1%	24.9%	79.3%
PB	36.8%	21.1%	58.9%
PC	42.9%	41.5%	79.4%
PD	48.2%	39.9%	81.6%

From the given table, zero-shot recognition performance drops sharply compared to few-shot recognition. Under the worst experiment setting, the recognition performance reaches only 35.1% in pure zero-shot setting and 11.1% in unseen generalized zero-shot recognition. However, the Meta-Siamese network works much better when the samples are seen in the training stage because it has learned how to project the semantic attributes to feature map space and how to compare two feature maps with large domain shift. Similar to few-shot palmprint recognition, the experiment accuracy in PolyU dataset is higher than that of our self-built dataset, which validates our previous analysis.

4.4 Visualization

Palmprint recognition in the practical environment is a highly complex and non-linear problem. We illustrate this problem by plotting one palmprint image and other matched and un-matched images in 5-way/1-shot recognition using t-SNE visualization tool in Fig.4(a). It can be seen that the images from the same category don't cluster together in the latent two-dimensional space, which makes the metric learning and image separation process quite challenging.

In order to show the distinguishability of our model, we visualize the outputs of the same image pairs from the first FC layer, which is demonstrated in Fig.4. (b). It can be seen that after the flexible feature extraction and similarity computation, the outputs of unmatched images and matched images follow a distribution that is linearly separable.



(a) (b)

Fig.4. Visualization results. (a) Matched and unmatched results in the image level. Where '0' and '1' belong to the same category. '2' denotes samples from a different class. (b) Matched and unmatched feature representations in the first FC layer. Where '0' denotes the output of unmatched images and '1' denotes the matched outputs.

5. CONCLUSIONS

In this paper, we propose a low-shot model for palmprint recognition using a few labelled images. We build on the classical Siamese network and introduce meta episode training for better generalization performance. We also modify the initial model to compare two palmprint images flexibly by introducing CNN blocks in the similarity metric learning stage. Furthermore, we extend our model to zero-shot recognition by obtaining the semantic hash codes using Deep Hash Net. Therefore, our model is very suitable for hand-based practical personal authentication when the size of the database is small or when only a small part of the palmprints are labelled in the acquisition stage. Future works will surround on zero-shot palmprint recognition by using more meaningful semantic embeddings to obtain a better recognition performance.

7. REFERENCE

[1] A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognition,* vol. 42, no. 7, pp. 1408-1418, 2009.

[2] D. Zhong, X. Du, and K. Zhong, "Decade progress of palmprint recognition: A brief survey," Neurocomputing, vol. 328, pp. 16-28, 2019.

[3] N. B. Mahfoudh, Y. B. Jemaa, and F. Bouchhima, "A Robust Palmprint Recognition System Based on Both Principal Lines and Gabor Wavelets," International Journal of Image, Graphics and Signal Processing, vol. 5, no. 7, 2013.

[4] Y. Wang and S. Guang-Hua, "Palmprint recognition using Palm-line direction field texture feature," in 2012 International Conference on Machine Learning and Cybernetics, 2012, vol. 3, pp. 1130-1134.

[5] D. Zhong, Y. Yang, and X. Du, "Palmprint Recognition Using Siamese Network," in 13th Chinese Conference on Biometric Recognition, CCBR, 2018, pp. 48-55.

[6] D. Zhong, M. Li, H. Shao, and S. Liu, "Palmprint And Dorsal hand Vein Dualmodal Biometrics," in IEEE International Conference on Multimedia and Expo (ICME), 2018, pp. 1-8.

[7] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," presented at International Conference on Machine Learning (ICML), 2017. pp. 1126-1135

[8] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to Compare: Relation Network for Few-Shot Learning,", presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. pp. 1199-1208.

[9] X. Pan and Q. Q. Ruan, "Palmprint recognition using Gabor feature-based (2D) 2 PCA," Neurocomputing, vol. 71, no. 13, pp. 3032-3036, 2008.

[10] Z. Guo, L. Zhang, D. Zhang, and X. Mou, "Hierarchical multiscale LBP for face and palmprint recognition," in IEEE International Conference on Image Processing, 2010, pp. 4521-4524.

[11] F. Yue, W. Zuo, D. Zhang, and K. Wang, "Orientation selection using modified FCM for competitive code-based palmprint recognition," Pattern Recognition, vol. 42, no. 11, pp. 2841-2849, 2009.

[12] G. S. Badrinath, "Score level fusion of voting strategy of geometric hashing and SURF for an efficient palmprint-based identification," Journal of Real-Time Image Processing, vol. 8, no. 3, pp. 265-284, 2013.

[13] T. Munkhdalai and H. Yu, "Meta Networks," presented at International Conference on Machine Learning, 2017. pp. 2554-2563.

[14] G. Koch, Richard Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition,", ICML Deep Learning Workshop, vol. 2, 2015.

[15] L. Zhang, T. Xiang, and S. Gong, "Learning a Deep Embedding Model for Zero-Shot Learning," presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. pp. 3010-3019

[16] E. Kodirov, T. Xiang, and S. Gong, "Semantic Autoencoder for Zero-Shot Learning," presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. pp. 4447-4456.

[17] D. Zhong, H. Shao, and Y. Liu, "Hand Dorsal Vein Recognition Based on Deep Hash Network," PRCV 2018, 2018, pp. 26-37.

[18] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep Learning to Hash by Continuation," presented at IEEE International Conference on Computer Vision, 2017. pp. 5609-5618.

[19] D. Zhang, Z. Guo, G. Lu, L. Zhang, and W. Zuo, "An Online System of Multispectral Palmprint Verification," IEEE Transactions on Instrumentation & Measurement, vol. 59, no. 2, pp. 480-490, 2010.

[20] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical Networks for Few-shot Learning," presented at Conference and Workshop on Neural Information Processing Systems, 2017. pp. 4080-4090

[21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," presented at Conference and Workshop on Neural Information Processing Systems, 2016. pp. 3630-3638.

[22] D. Zhong and J. Zhu, "Centralized Large Margin Cosine Loss for Open-set Deep Palmprint Recognition," IEEE Transactions on Circuits and Systems for Video Technology, 2019, DOI: 10.1109/TCSVT.2019.2904283.

=