Contactless Palmprint Identification using Deeply Learned Residual Features

Yang Liu and Ajay Kumar

Abstract—Contactless and online palmprint identification offers improved user convenience, hygiene, user-security and is highly desirable in a range of applications. This paper proposes an accurate and generalizable deep learning-based framework for the contactless palmprint identification. Our network is based on fully convolutional network that generates deeply learned residual features. We design a soft-shifted triplet loss function to more effectively learn discriminative palmprint features. Online palmprint identification also requires a contactless palm detector, which is adapted and trained from faster-R-CNN architecture, to detect palmprint region under varying backgrounds. Our reproducible experimental results on publicly available contactless palmprint databases suggest that the proposed framework consistently outperforms several classical and state-of-the-art palmprint recognition methods. More importantly, the model presented in this paper offers superior generalization capability, unlike other popular methods in the literature, as it does not essentially require database-specific parameter tuning, which is another key advantage over other methods in the literature.

Index Terms—Biometrics, Contactless Palmprint Matching, Contactless Palmprint Detection, Personal Identification, Deep Learning

1 Introduction

Automated personal identification using palmprint images has been widely studied and employed for a range of law-enforcement and e-security applications. However *contactless* palmprint identification is relatively new area of research and offers more attractive solution for the deployments as it can address serous concerns relating to the hygiene while offering significantly higher user convenience and security. In addition, the contactless palmprint imaging also enables deformation free acquisition of palmprint features, or the ground truth information, which can enable higher matching accuracy than those acquired using contact-based imaging.

Several challenges need to be addressed by the contactless palmprint researchers. Firstly, the contactless palmprint matching accuracy is known to significantly degrade, as compared to those from the contact-based palmprint images, as such contactless images often present significantly higher imaging variations. Therefore more advanced matching techniques need to be developed to improve the matching accuracy from the contactless palmprint images. Secondly, the automated detection of contactless palmprint images (region of interest) from the presented hands is quite challenging as the background during such imaging is expected to be dynamic or less stable. Available research on contactless palmprint images addresses such challenges by acquiring contactless palmprint images with fixed background that can enable key point detection using pixel-wise operators to segment the palmprint images. Deep learning capabilities offer enormous potential to address these two challenges and are considered in this paper.

In recent years, deep learning has emerged as the dominant approach for a range of computer vision related problems and has delivered state-of-the-art performance for the face recognition [4], [7], iris recognition [8] and image classification. However, compared to face recognition, there has so far been relatively little effort to explore deep learning for palmprint identification.

This paper proposes a new, deep learning based, contactless palmprint identification framework which not only offers accurate matching capabilities but also exhibits outstanding generalization capabilities on different public databases. With the design of effective residual feature network, our model can enlarge the receptive field [9] for matching contactless palmprint images and learn comprehensive palmprint features which generalize very well on other databases. We develop a soft-shifted triplet loss function to accommodate frequent contactless palmprint imaging variations and offer meaningful supervision for learning effective palmprint features from a limited number of training samples. We also introduce an automatic contactless palm detector intended to handle complex real-world backgrounds. Design of such detectors is critical for the success of contactless palmprint identification during deployments.

The main *contributions* from this paper can be summarized as follows: (a) We develop a new deep learning based contactless palmprint identification framework with high generalization capability for operating on different contactless palmprint databases that can represent diverse deployment scenarios. A new *Soft-Shifted Triplet Loss* (SSTL) function has been developed to successfully address the nature of contactless palmprint patterns for learning comprehensive palm features (please see more details in section 2.3). Our work therefore presents significant advances to bridge the gap between deep learning and contactless palmprint matching techniques available today; (b) Under fair comparison, our approach consistently outperforms several state-of-the-art methods on publicly available contactless palmprint databases. Even under the challenging scenario of not incorporating any parameter tuning on the target

• *Authors are with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.*
Corresponding author email: ajay.kumar@polyu.edu.hk

Manuscript received June 5, 2019; revised September 18, November 27, 2019, January 10, 2020.

dataset, our model can achieve superior or competitive performance over the state-of-art methods that have had extensive parameter tuning. This paper also demonstrates how the faster-R-CNN [5] architecture can be adapted to build an online palm detector, which can robustly detect palm images from the presented hands under complex backgrounds. Such advancements are highly desirable, with reference to the current literature, for the success of online and contactless palmprint identification applications.

1.1 Related Work

Completely automated matching for contactless palmprint images has received lot of attention and a range of palmprint matchers have been introduced in the literature. Detected or segmented palm images can be characterized by major/minor curved lines and creases that can be observed even from low resolution (~ 100 dpi) images and additional flexion ridges [1] that are observed from high resolution (~ 500 dpi, not the focus of this work like for [10]) images. Therefore a range of texture matching methods have been introduced in the literature [11], [12], [13], [14], [17], [18]. Encoding palmprint features using the dominant orientation of lines/creases in [19], [21] is one of the most effective method for matching palmprint images. More recent work in matching contactless palmprint images appear in [14] where an ordinal measurement based descriptor, i.e., difference of normal (DoN), has shown to outperform a range of methods introduced for matching contactless palmprint images using publicly available databases. This approach benefits from the contactless palm image acquisition modeling and introduces specialized masks to encode projective ordinal measurements. Therefore, this method has also been used to ascertain the effectiveness of approach developed in this paper and serves as a reasonable choice as other methods [2], [20] have not yet shown to offer superior performance than from [14] in the best of our knowledge.

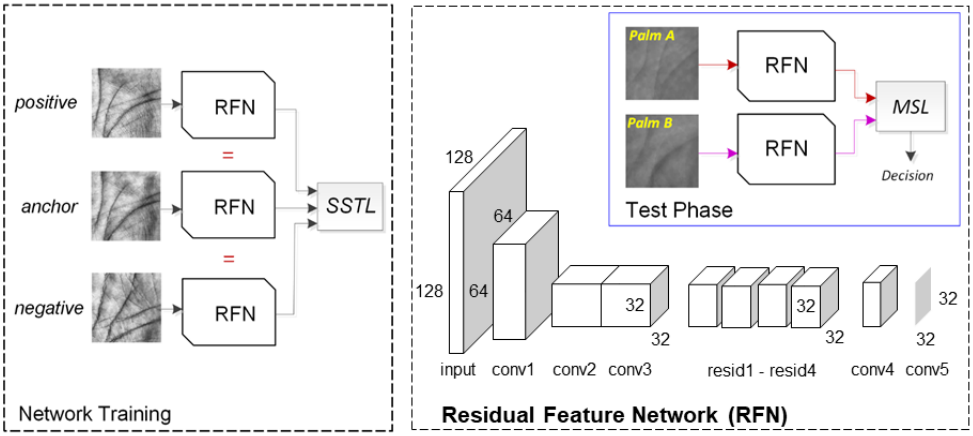
Automated detection of palm images, or the region of interest from the hands presented by users, is inherently required for the success of contactless palmprint identification systems during real deployments. Most popular methods for palmprint detection are based on the extraction of key-points representing finger joints and extract a fixed region of interest relative to the orientation and/or the distance **[?]** between the key points. This approach works very well for the contact-based imaging setups but poses a range of problems for contactless palmprint images as it is very difficult to robustly detect these key-points under background changes which are inherent during the contactless imaging even with the cooperative users attempting access. Therefore developed contactless palmprint databases [22], [24], [25] (in public domain) have been acquired using relatively fixed or stable background to primarily address the open problem of detecting palm images under user friendly contactless imaging setup. Advancements to detect contactless palmprints under real-world backgrounds is highly desirable and is also considered in our work.

1.2 Open Problems and Challenges

Despite promising performance indicated in the literature for matching palmprint images, conventional palmprint descriptors have several limitations. Summary of earlier work presented in [2] indicates that existing methods offer quite accurate performance but this performance needs to be further improved (especially on large contactless databases *e.g.* [25]) to meet expectations for a wide range of deployments. Conventional palmprint descriptors, such as CompCode [19] or DoN [14], RLOC [21] or Ordinal [17], are based on empirical models, which apply hand crafted filters for the generation of features. Therefore these models heavily rely on the parameter selection when incorporated for matching performance for other/different databases or those acquired under different imaging environments. This situation can also be observed from [14], where eight different combination of parameters on 4 different databases are employed by extensive tuning. Commonly employed techniques in the the palmprint literature [21], [26] for the automated detection of palm images, or the region of interest from the hands presented by users interested to access the system, often fails when the hand images are acquired under complex backgrounds. Such failure can be attributed to the nature of algorithms that relies on the detection of key-point using pixel-based operators that are dependent to differentiate gray-levels from skin and the background.

The deep learning-based approaches have potential to address above outlined limitations with the conventional palmprint matching methods. As compared to the empirical selection of hand-crafted filter parameters for palmprint matching, the parameters in deep neural networks can be self-learned from the data. Deeply learned architectures are known [3]-[4] to offer higher generalization capabilities for a range of computer vision problems. However, any direct application of such architectures, e.g. [2], [27], is expected to deliver limited performance or cannot match performance offered from state-of-art techniques such as those from [14]. This is due to the fact that new challenges emerge while incorporating typical deep learning architectures (e.g. CNN) for the palmprint recognition, which can primarily be attributed to the nature of palmprint patterns. Unlike the face, palmprint patterns are known to reveal little structured information or meaningful hierarchies. Palmprint texture-based methods are widely considered to be more accurate methods in the literature [2], [14], [17], [18], [19], [21] which mainly employed small sized filters or block based operators to extract palmprint features. Therefore, we can infer that the most discriminative information from palmprint patterns is extracted from the local intensity distributions in region of interest (palm) images rather than from (if any) global features. The CNNs are known to be effective in recovering features from low level to the high level, and from local to global, due to the combination of convolutional and fully connected layers [28]. However as outlined earlier, the high level and global features extracted from such networks may not be optimal for the accurate matching of palmprint patterns.

This paper attempts to develop a more accurate and robust deep learning based palmprint feature representation framework [40]. Such advancements to uncover the potential from deep learning capabilities are highly desirable to realize full potential from the palmprint biometric. Different from [4], [8], [27], a new network architecture and a customized loss function is developed to extract discriminative palmprint features. The experimental results presented in



Layer	Type	Kernel Size	Stride	#Filters
conv1	Convolution	5 × 5	2	32
conv2	Convolution	3 × 3	2	64
conv3	Convolution	3 × 3	1	128
resid1	Residual Block	3 × 3	1	128
resid2	Residual Block	3 × 3	1	128
resid3	Residual Block	3 × 3	1	128
resid4	Residual Block	3 × 3	1	128
conv4	Convolution	3 × 3	1	64
conv5	Convolution	3 × 3	1	1

Fig. 1. An overview of our Residual Feature Network (RFN) architecture for contactless palmprint matching. The RFN contains three consecutive convolutional layers followed by four classical residual blocks. The first and the second convolutional layer down-sample the input which results in the feature map that is of one quarter the size of input as illustrated at right. Instance normalization is employed in the residual block instead of batch normalization. The RFN generates a single-channel feature map for each of the input images. The RFN is trained using Soft Shifted Triplet Loss (SSTL) as detailed in Section 2.3.

section 3, on four different contactless palmprint databases, validate the effectiveness of our framework.

2 Matching Contactless Palmprint Images

2.1 Network Architecture

We develop a highly optimized deep learning architecture, referred to as residual feature network (RFN) in this paper, to accurately match real-world contactless palmprint images. Different from the residual network [4], RFN does not have fully connected layers which results in pure feature map outputs (Figure 1) that can preserve spatial-correspondences among the most discriminative palmprint features. We replace all of the batch normalization layers [29] with the instance normalization [30]. Our key motivation is to enhance the robustness of RFN in learning low/mid/high level features [31] as the contactless palmprint images present high intra-class variations not just due to deformations [32] but also due to the pose and illumination changes [25].

2.2 Network Training

The convolutional kernels of RFN were trained using a triplet network [7]. As shown in Figure 1, this triplet network consists of three identical RFNs and their weights are kept identical during the training. These RFNs are interconnected in parallel to enable the forward and backward propagation of the data and gradients for anchor, positive and negative samples respectively. The triplet loss function in such architecture is expected to help the network learn in generating the feature maps that can reduce the anchor-positive distances while increase the anchor-negative distances.Formulation of network loss that can accommodate high intra-class variations is highly desitable and can train the network in generating more robust feature maps. We therefore soften the matching loss and improve the original loss function to accommodate frequent translational changes in the contactless palmprint images from the same class/subject. This new loss function is referred to as *Soft Shifted Triplet Loss* (SSTL) and is detailed in section 2.3.

2.3 Soft-Shifted Triplet Loss Function

The triplet networks [7] have been conventionally trained using the original loss function which can be written as follows:

$$L = \sum_{i}^{N} [\|\mathcal{F}(I_i^a) - \mathcal{F}(I_i^p)\|^2 - \|\mathcal{F}(I_i^\alpha) - \mathcal{F}(I_i^n)\|^2 + m]_+ \quad (1)$$

where function $\mathcal{F}(I)$ represents the embedding of the input image I into a high dimensional feature space, N is the number of triplet samples in a mini-batch, $\mathcal{F}(I_i^a)$, $\mathcal{F}(I_i^p)$ and $\mathcal{F}(I_i^n)$ are the feature representations of anchor, positive and negative image samples in the i-th triplet respectively. The symbol $[\circ]_+$ is equivalent to $max(\circ, 0)$. m is preset parameter to control the desired distance between anchor-positive and anchor-negative. For simplicity, we denote these three feature maps from the input I_i^a, I_i^p, I_i^n as $\mathcal{F}_i^a$, $\mathcal{F}_i^p$, $\mathcal{F}_i^n$ respectively.

Contactless palmprint images from the same class generally depict high translational changes along the two axes. In order to accommodate such translations, we now introduce a new loss function SSTL and is defined as follows:

$$SSTL = \frac{1}{N} \sum_{i=1}^{N} [\mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^p) - \mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^n) + m]_+ \quad (2)$$

where $\mathcal{L}$ represents the *Minimum Shifted Loss*(MSL). Any loss function to train the network should be differentiable along the shift directions. We now systematically formulate such requirements in the following.

Let us denote the width and height of the feature map from RFN by W and H respectively. We use W_s and H_s to define extent of maximum expected spatial shifts along the horizontal and vertical directions. The *MSL* is defined to accommodate frequent translational shifts in the input or among segmented contactless palmprint images, as follows:

$$\mathcal{L}(\mathcal{F}_1, \mathcal{F}_2) = \min_{-W_s \le w \le W_s, -H_s \le h \le H_s} \{D_{w,h}(\mathcal{F}_1, \mathcal{F}_2)\} \quad (3)$$

$$D_{w,h}(\mathcal{F}_1, \mathcal{F}_2) = \frac{1}{|C_{w,h}|} \sum_{(x,y) \in C_{w,h}} (\mathcal{F}_1^{(w,h)}[x,y] - \mathcal{F}_2[x,y])^2 \quad (4)$$

$$C_{w,h} = \{(x,y)|\max(w,0) \leq x \leq \min(W+w,W), \\ \max(h,0) \leq y \leq \min(H+h,H)\} \quad (5)$$

where C represents the common region between two matched feature maps with valid (non-zero) values for each of the (w,h) combinations while x and y denotes the spatial coordinates. The *MSL* in (3) attempts to compute the minimum distance between the two feature map that can be achieved after translation by w and h pixels along the horizontal and vertical directions respectively. The superscript (w,h) in (4) denotes such translational operation on feature map $\mathcal{F}_1$ and the resulting shifted feature map has following spatial correspondence with the original one:

$$\mathcal{F}^{(w,h)}[x_w, y_h] = \begin{cases} \mathcal{F}[x,y], & (x,y) \in C_{w,h} \\ 0, & \text{otherwise} \end{cases} \\ x_w = (x - w + W) \bmod W \\ y_h = (y - h + H) \bmod H \quad (6)$$

x_w is obtained by shifting the feature values to the left (horizontal translation) in a step of w and y_h is obtained by shifting the feature values upward (vertical translation) in a step of h. As illustrated in (6), the void generated due to the translation of feature map values are automatically assigned as zeros. The training of RFN requires us to compute the gradients (or partial derivatives) of the soft shifted triplet loss, between the anchor-positive and anchor-negative feature maps. The resulting loss is back propagated iteratively during the network training. Let us firstly consider the loss between the feature map from the anchor and its respective positive feature map $\mathcal{F}_i^p$ and compute its derivative for one sample pair in the batch:

$$\frac{\partial SSTL}{\partial \mathcal{F}_i^p} = \begin{cases} 0, & \text{if } SSTL = 0 \\ \frac{\partial SSTL}{\partial \mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^p)} \frac{\partial \mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^p)}{\partial \mathcal{F}_i^p}, & \text{otherwise} \end{cases} \quad (7)$$

Since $\frac{\partial SSTL}{\partial \mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^p)} = \frac{1}{N}$, above equation can be further simplified as

$$\frac{\partial SSTL}{\partial \mathcal{F}_i^p} = \begin{cases} 0, & \text{if } SSTL = 0 \\ \frac{1}{N} \frac{\partial \mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^p)}{\partial \mathcal{F}_i^p}, & \text{otherwise} \end{cases} \quad (8)$$

Let us firstly define the shifting offsets for the anchor-positive and anchor-negative image pairs that can meet requirements for *MSL* as follows:

$$(w_{ap}, h_{ap}) = \underset{-W_s \leq w \leq W_s, -H_s \leq h \leq H_s}{\arg\min} \{D_{w,h}(\mathcal{F}_i^a, \mathcal{F}_i^p)\} \\ (w_{an}, h_{an}) = \underset{-W_s \leq w \leq W_s, -H_s \leq h \leq H_s}{\arg\min} \{D_{w,h}(\mathcal{F}_i^a, \mathcal{F}_i^n)\} \quad (9)$$

The gradient of the distance $\mathcal{L}$ in (8) can be computed from the following pixel-wise derivatives using (3) and (4):

$$\frac{\partial \mathcal{L}(\mathcal{F}_i^a, \mathcal{F}_i^p)}{\partial \mathcal{F}_i^p[x,y]} = \frac{D_{w_{ap},h_{ap}}(\mathcal{F}_i^a, \mathcal{F}_i^p)}{\partial \mathcal{F}_i^p[x,y]} \\ = \begin{cases} 0, \text{ if } (x,y) \notin C_{w_{ap},h_{ap}} \text{ or } SSTL = 0 \\ \frac{-2(\mathcal{F}_i^a[x_{w_{ap}}, y_{h_{ap}}] - \mathcal{F}_i^p[x,y])}{N|C_{w_{ap},h_{ap}}|}, \text{ otherwise} \end{cases} \quad (10)$$

The partial derivative of SSTL with respect to the positive feature map $\mathcal{F}_i^p$ can be computed as follows:

$$\frac{\partial SSTL}{\partial \mathcal{F}_i^p} = \begin{cases} 0, \text{ if } (x,y) \notin C_{w_{ap},h_{ap}} \text{ or } SSTL = 0 \\ \frac{-2(\mathcal{F}_i^a[x_{w_{ap}}, y_{h_{ap}}] - \mathcal{F}_i^p[x,y])}{N|C_{w_{ap},h_{ap}}|}, \text{ otherwise} \end{cases} \quad (11)$$

We can similarly compute the required partial derivative with respect to the negative feature map:

$$\frac{\partial SSTL}{\partial \mathcal{F}_i^n} = \begin{cases} 0, \text{ if } (x,y) \notin C_{w_{an},h_{an}} \text{ or } SSTL = 0 \\ \frac{-2(\mathcal{F}_i^a[x_{w_{an}}, y_{h_{an}}] - \mathcal{F}_i^n[x,y])}{N|C_{w_{an},h_{an}}|}, \text{ otherwise} \end{cases} \quad (12)$$

Our final requirement is to compute the partial derivatives for the feature map from anchor. It can be observed from (3)-(6) that shifting or translation of the first map towards the left by w pixels and towards the top by h pixels is equivalent to shifting the second map towards the right by w pixels and towards the bottom by h pixels. We can therefore rewrite (4) as follows:

$$D_{w,h}(\mathcal{F}_1, \mathcal{F}_2) = \frac{1}{|C_{w,h}|} \sum_{(x,y) \in C_{w,h}} (\mathcal{F}_1^{(w,h)}[x,y], \mathcal{F}_2[x,y])^2 \\ = \frac{1}{|C_{w,h}|} \sum_{(x,y) \in C_{w,h}} (\mathcal{F}_1[x,y], \mathcal{F}_2^{(-w,-h)}[x,y])^2 \quad (13)$$

It is now quite straightforward to compute the partial derivative for the anchor positive feature map using (7)-(10) and (13):

$$\frac{\partial SSTL}{\partial \mathcal{F}_i^a[x,y]} = -\frac{\partial SSTL}{\partial \mathcal{F}_i^p[x_{-w_{ap}}, y_{-h_{ap}}]} + \frac{\partial SSTL}{\partial \mathcal{F}_i^n[x_{-w_{an}}, y_{-h_{an}}]} \quad (14)$$

The rest of the back-propagation process is the same as for common end-to-end convolutional network. Above derivation shows that during the matching of feature maps, from the translated palmprint images, the gradients that only lie in the overlapped regions will be back-propagated. This enables more accurate matching of feature maps from the contactless palmprint images that are not strictly aligned. The network is trained using *SSTL* while *MSL* is used during the test or the evaluation phase.

3 Experiments and Results

We performed thorough experiments using publicly available databases to ascertain various aspects of the performance from our approach. In the following sections, we detail on the experimental protocols, along with the reproducible results [33], employed for the extensive evaluation of the model proposed in this paper.

Our experiments are firstly organized to ascertain within database performance (*WithinDB*) which uses some part of the database for the training while using some other independent part of this database for the performance evaluation. Also, the cross-database performance evaluation (*CrossDB*) is highly desirable to address limitations of currently available palmprint recognition methods in the literature. Therefore *CrossDB* performance evaluation results are also presented in this paper which uses the network that is trained on some part of publicly available database while the test performance are reported using other independent publicly available database with the respective protocols which have been used in the literature (to ensure fairness in the performance comparison). It should be noted that for both *WithinDB* and *CrossDB* configurations, training set and test set are totally separated, i.e., none of the palmprint images are overlapping between training set and the test set. Since our focus is mainly on extensive *CrossDB* performance evaluation, we incorporated the largest subjects database

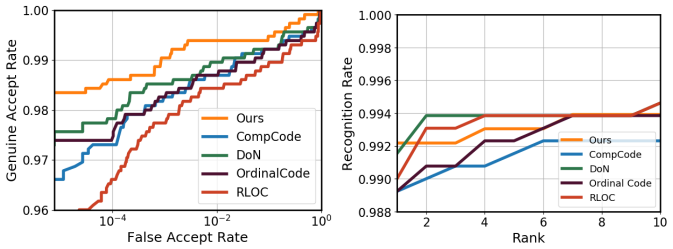


Fig. 2. The ROC curves (**Left**) and CMC curves (**Right**) for different methods on the IITD Right contactless palmprint database.

TABLE 1
Summary of accuracy (average rank-one recognition rate) and equal error rate (EER) on three different contactless palmprint databases.

	IITD		PolyU-IITD		CASIA
	Acc(%)	EER(%)	Acc(%)	EER(%)	EER(%)
DoN(TPAMI16)	99.15	0.68	98.3	0.329	0.53
RLOC	99.00	0.88	98.45	0.557	1.0
Comp Code	98.85	1.0	98.45	0.435	0.76
Ordinal Code	98.92	1.25	98.48	0.451	0.79
Ours-CrossDB	/	/	98.6	0.267	**0.51**
Ours-WithinDB	**99.20**	**0.60**	**98.7**	**0.153**	/

from 600 different subjects for this task as detailed in the next section.

During our *CrossDB* performance evaluation, all of the test configurations uses the IITD Left [22] (all left hand palmprint images in this dataset) as the training set. The trained model is used for the performance evaluation using IITD Right (all the right hand palmprint images) which indicates *WithinDB* performance. During the *WithinDB* configuration, we used the left palmprint images for the training set and the right palmprint images for test or performance evaluation as it allows us to perform fair comparison, with the respective results from more recent approach DoN in [14] which has shown outperforming results over several state of art methods.A) Other baseline methods in our experiments represent faster version of these methods instead of their original ones, e.g. *fast-CompCode*, or *fast-RLOC* as these methods have shown to offer superior performance over [19], [21] and theoretically justified in [15]. The same trained model which is trained using IITD left hand palmprint images is used for the *CrossDB* performance evaluation using the largest subjects database made available from [25] and also using the CASIA contactless palmprint image database from [24]. Thus the *CrossDB* performance evaluation can illustrate the generalization capability of the proposed model when few or the training samples from other databases are incorporated. The *WithinDB* performance evaluation using [25], in addition to results from IITD [22], is also presented for comparative performance evaluation.

3.1 Databases and Protocols

IITD Palmprint Database. The IITD touchless palmprint database [22] provides contactless palmprint images from the right and left hands of 230 subjects. There are 5 samples for each right hand or left hand. This database also provides 150×150 pixels segmented palmprint images. In our experiments, the left hand palmprint images are used to train our model detailed in section 2 and all the 1300 right hand palmprint images are used for the performance evaluation. This protocol for test performance evaluation is exactly the same as in [14] and results in 1150 genuine matches and 263,350 imposter matches. The comparative evaluation results using ROC, CMC (Figure 2) and EER (Table 1) is presented to ascertain the performance. The ROC, EER and the average rank-one recognition rate achieved from our approach indicates outperforming results.

3.2 Cross-Database Performance Evaluation

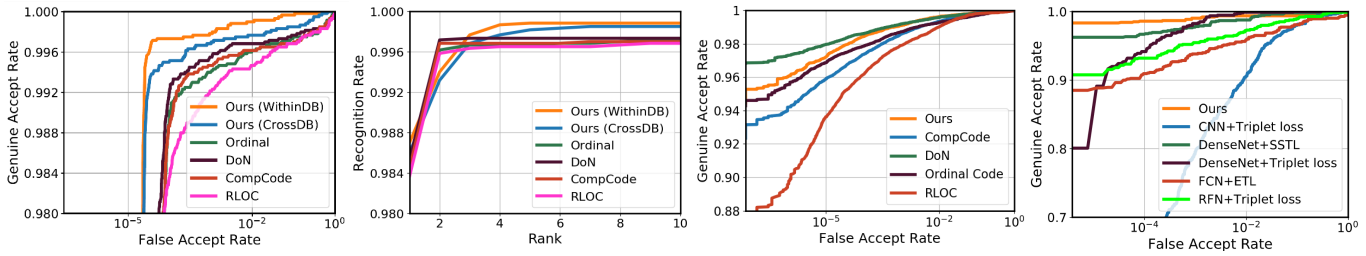
Our *CrossDB* performance evaluation is firstly focused on more recent contactless palmprint database in [25], which has been acquired from over 600 subjects, and is the largest in the best of our knowledge. In our experiments, 6,000 palmprint images from the first 600 subjects left hands were used for the test performance evaluation and the protocol is exactly the same as used for Figure 2 or the protocol used in [14]. Therefore, the test set for this *CrossDB* performance generated 6,000 genuine and 3,594,000 imposter matches. Figure 3 illustrates comparative ROC, CMC and respective EER is presented in Table 1. The results in figure 3 also illustrates *WithinDB* performance which is achieved by training our model using other or all the right hand images for the same database. The results in Figure 2 (a)-(b) indicates that our model can achieve outperforming results and the performance is further improved for the *WithinDB* case or when the model trained from the right hand images for the same database is used for the performance evaluation.

Another contactless palmprint database available in public domain is from [24]. This CASIA palmprint database contains 5239 palmprint images from 301 individuals. We also employ this database for the *CrossDB* performance evaluation and used the model trained on IITD database (same as for results in Figure 2 or *CrossDB* in Figure 3) for the performance evaluation. All experiments on this CASIA database use the same matching protocol as used in [14] to ensure fairness in the comparison. Therefore as in [14], we also generate 13,692,466 match scores, which consisted of 20,567 genuine and 13,689,899 imposter match scores. Figure 3 (c) illustrates comparative ROC performance and Table 1 provides respective EER from the *CrossDB* performance. It is worth to underline that comparison here is with the same result as in [14], which uses heavy tuning of parameters while our results are on unseen or *CrossDB* evaluation protocol. Due to small number of images (only three) per subject, *WithinDB* evaluation was not performed and is of least interest. It can be observed from these results that in terms of EER our model performs better than best of results in [14] while the performance from Figure 3 is otherwise but quite competing.

3.3 Discussion

We also perform comparative performance evaluation from our method against other popular deep learning architectures that are widely used for various recognition tasks. The details on the such configurations considered for the performance evaluation is provided in the following.

CNN+Triplet Loss. Pre-trained CNN based methods are most widely employed in the deep learning configurations for the recognition tasks [7], [34] and therefore also be interesting and worth evaluating. We use VGG-16 model,



(a) (b) (c) (d)

Fig. 3. (a) Comparative ROC and (b) corresponding CMC for *WithinDB* and *CrossDB* tests (600 subjects). (c) The ROC for *CrossDB* evaluation for CASIA palmprint database [24] and other methods using respective/best parameters. (d) The ROC curves for typical deep learning architecture in the literature using contactless palmprint database in [22].

as our test architecture, which has been widely used in many other recognition tasks. We replace the last fully connected class layer with another fully connected feature layer for matching the features. We freeze the basic feature extraction layers in VGG-16 during the training phase and fine-tune the newly added fully connected layer using the given training dataset, i.e. IITD Left palmprint images in our experiment.

Fully convolutional network+Extended triplet loss. The fully convolutional network (FCN) was originally developed for semantic segmentation [35]. Recently [8] combines FCN and extended triplet loss (ETL) to achieve the state-of-art performance for the iris recognition task. Since this work also employs bit-shifting in the original triplet loss function, it is important to comparatively ascertain the performance from our model over this method.

Residual feature network(RFN)+Triplet loss. Comparative evaluation has also been performed using the RFN used in our model and the original triplet loss function instead of the soft shifted triplet loss introduced in section 2.3. Such comparison is performed to ascertain the merit of SSTL for the problem considered in this paper.

DenseNet+Soft shifted triplet loss/triplet loss. We also compared our method against a very popular deep learning architecture, densely connected convolutional network (DenseNet) which has shown to offer significant performance improvement over the state-of-the-art on many/most recognition tasks. In our experiments on palmprint image datasets, we use a basic DenseNet-BC structure with three dense blocks on 128×128 input images and replace the last fully connected layer with one 1×1 convolutional layer to perform SSTL. The initial convolution layer uses 5×5 convolution kernels with stride of two.

The comparison with the other deep learning based methods was performed on IITD dataset, which we employed for *WithinDB* configuration with the same protocol as for the results in Figure 2 or in [14]. All above discussed models are trained on the IITD Left palmprint images and evaluated using the IITD Right palmprint images. The test set generate 1150 genuine match scores and 263,350 imposter match scores which is consistent for the comparisons. The hyper-parameters for all training processes have been carefully investigated to achieve best performance. Comparative performances using ROC are presented in Figure 3 (d) while comparative storage and matching complexity for these methods is summarized in Table 2.

It can be observed from Figure 3 (d) that our newly developed architecture together with newly developed soft shifted loss outperforms other deep learning configurations. It should be noted that the architecture introduced in this work, e.g. RFN or FCN, provide new insights on the performance and have not been investigated for the palmprint matching in the literature. The CNN based configurations suggest that directly using global and high level features extracted by CNN may not be suitable for the palmprint recognition problem. Relatively poor performance from (RFN + Triplet) illustrates that a soft matching term introduced by SSTL offers great benefit in addressing inherent variations in the feature map for the contactless palmprint identification. Comparative performance between (DenseNet + SSTL) and (DenseNet + Triplet) also supports such observation.

In all of our experiments, we train our network using Stochastic Gradient Descent (SGD) with standard backprop [36], [37] and Adam [38]. We start with a learning rate of 0.001 and the models are randomly initialized. The predefined margin m is set to 0.2. The maximum vertical shift size H_s and horizontal shifting size W_s are both fixed as 5.

TABLE 2
The Comparison of time and space complexity of different contactless palmprint matching methods (evaluated on Linux Ubuntu 14.04 x86_64 with Quadro M6000 GPU under 10K average runs. The default shift size was set as 5 for *SSTL*).

Approaches	#Parm	Feature extraction	Matching	Template size
CNN +Triplet loss	∼449M	0.00745s	0.00140s	4096-d
DenseNet +SSTL	∼3.1M	0.0235s	0.049s	32×32
DenseNet +Triplet loss	∼3.1M	0.0235s	0.00040s	32×32
FCN +ETL	∼568K	0.00142s	0.0710s	128×128
RFN +Triplet loss	∼5.2M	0.0062s	0.00040s	32×32
Proposed	**∼5.2M**	**0.0062s**	**0.049s**	32×32

3.4 Additional CrossDB Experimental Results

It is important to note that database used from 600 different subjects is more challenging, largely due to the image variations resulting from the use of mobile camera under outdoor and ambient illumination. Therefore, this database was judiciously selected as the main dataset to evaluate *CrossDB* performance using the proposed approach.

Another contactless palmprint database made available recently in public domain has been acquired from 300 different subjects and is accessible from [18]. This is two-session database and also provides segmented palmprint images which were used in our additional experiments. We used the same-trained network as used for *CrossDB* performance evaluation for results on 600 subjects and used exactly same protocols as in [18] to ensure fair comparison. The ROC and CMC for the *CrossDB* performance using this database is presented in the following figures.

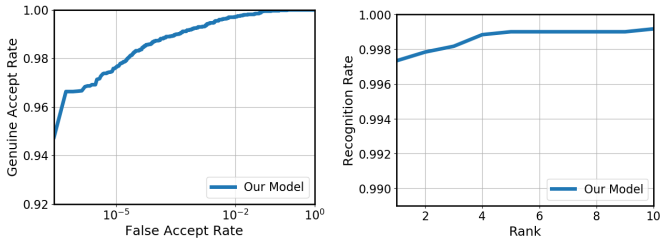


Fig. 4. The ROC curves (**Left**) and CMC curves (**Right**) plots for additional CrossDB performance. The EER is 0.433% while average rank-one recognition accuracy is 99.93%.

It may be useful to note that the reference [18] does not provide verification performance but presents identification performance. The best of the methods presented in [18] for this database which also uses extensive tuning of parameters, under the same matching protocols, present average recognition accuracy of 98.78%. As can be observed from results in Figure 4, our *CrossDB* evaluation achieves average recognition accuracy of 99.73%. Therefore above additional results on the database from [18] also achieve outperforming results and illustrate high generalization capability of our approach using residual features.

3.5 Sample Failure Cases and Analysis

Figure 6 illustrates sample contactless palmprint images from our experiments that falsely matched using the proposed model. It can be observed that some of the mismatch pairs in this figure have high similarity in major palm lines (e,g. in a or b) which could can results in false match from their closer feature map.

The feature map (Figure 5) generated from some palmprint images mismatched pairs are illustrated in Figure 6. This figure presents image samples from right hand images in IITD [18] database and the feature maps are resized from original (32×32) for ease in visualization. A possible reason for such mismatch can be observed the from these feature maps as their shifted distances are relatively smaller. i.e., their hot point from corresponding feature maps are likely to be aligned by shifting horizontally and vertically. Therefore, there appears to be a tradeoff in the generation of feature map using the shifted loss. On the one hand, such loss provides a much softer way to align the feature

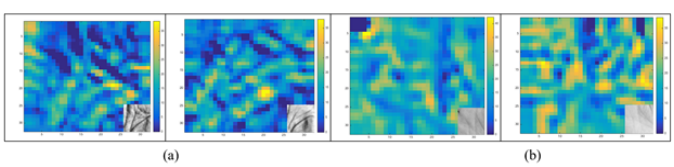


Fig. 5. Sample resized feature maps from two mismatched palmprint image pairs from different subjects.

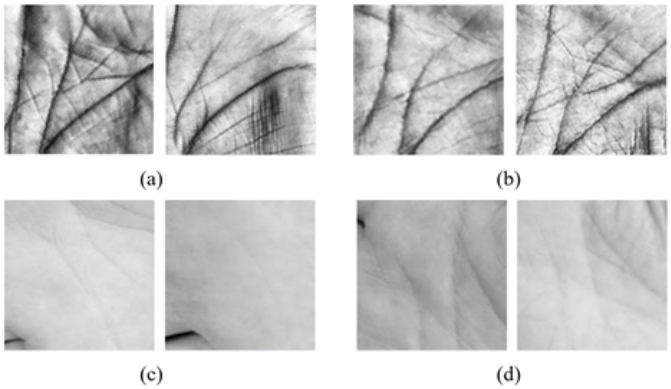


Fig. 6. Sample images pairs from different subject that falsely matched using our network trained from IITD left hand palmprints. Image pairs in (a)-(b) are image samples from IITD right hand while image pairs in (c)-(d) are left hand palmprint images from database in [**?**].

maps which is much needed due to nature of imaging for the contactless palmprint images. On the other hand, such shifts can result false matches from the palmprint images of different subjects as such shifts reduce the similarity between the feature maps with higher similarity in major palmprint lines or creases. Further extension of this work should therefore focus on learning more robust feature map information, possibly through deeper networks to reduce adverse impact from such mismatches.

4 Online Palmprint Identification

In earlier sections, we discussed on our approach for the development of trained model to match contactless palmprint images. The performance evaluation presented in section 3 used automatically segmented contactless palmprint images in respective public databases. The success of this matcher for online palmprint identification also requires a deep learning based palm detector that can automatically detect palmprint, or the region of interest, from the presented hands under complex backgrounds. Currently employed palmprint detectors in the literature use pixel-wise operators to recover the palm region using key points and are suitable for the hand images acquired under relatively fixed backgrounds or for the images used in previous section. However these methods are not suitable for the palmprint images acquired under complex and dynamic backgrounds, largely due to their failure in the detection of reference points. Such challenges are also underlined in more recent references, e.g. [20] use manually labelled 14 key points for the region of interest extraction. Therefore we also developed a palm detector that can detect palmprints under complex and dynamic backgrounds, which is also considered as a challenge using the conventional methods of palmprint detection available in the literature.

4.1 Palmprint Detection

The online palmprint detector developed in our work is based on the Faster R-CNN introduced in [5]. It is composed of two modules. The first module is a deep fully convolutional network (CNN) that proposes possible object regions. The second module is a Fast R-CNN detector [6] that uses these proposed regions to classify the palmprint

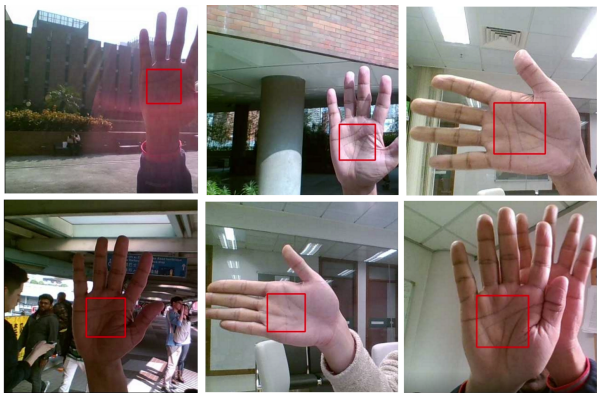


Fig. 7. Sample images from our online system, running on a mobile laptop, depicting palmprint detection from hand images under complex backgrounds.

ones. The entire system is a single, unified network for the palm detection, which employs the popular terminology of neural networks with "attention" mechanisms. The regional proposal network (RPN) modules in the system updates Fast R-CNN module on the specific regions of interest to detect palm region. Our Tensorflow based implementation incorporated [23] for the palmprint data augmentation.

4.2 Palmprint Dataset for Training and Detection

We firstly acquired a set of videos under indoor and outdoor environment for developing the detector. These videos were acquired under 11 different environments with various postures and illuminations. The videos were then segmented at the interval of every 10 frames which resulted in a dataset of 3K raw palmprint images under varying backgrounds. This raw data is then augmented 10 times which resulted in a total of 30K palmprint images that were employed to train the palmprint detector.

The palmprint detector development requires training from ground truth data. The training videos/data acquired under different environment was used to generate ground truth labels (coordinates of the desired bounding box) using semi-automated process. The image frames from the training data were firstly marked manually marked with locations of two key points, i.e., (i) index and middle finger joint, (ii) little and ring finger joint. Each of these two key points were used to automatically generate four coordinates of the bounding boxes representing palmprint region. The method of generating the coordinates corners representing such bounding box is similar to as in [18] or [15]. Each of the image frame, with ground truth locations of bounding box location, is further enriched for the training phase using (automated) data augmentation. Several data augmentation methods are available in the literature, e.g. Gaussian blur, random addition, multiplication on three color image channels, contrast normalization, additive Gaussian noise, etc. and reference [23] provides more details on these augmentation methods. Figure 8 illustrates a typical sample from our database and resulting images from the augmentation.

The images in the Figure 8 use following augmentation, addition (-20, 20) in first row, contrast normalization (0.5 - 1.75) in second row, multiplication (0.8 - 1.2) in third row while scale and aspect ratio augmentation in the last row.

Fig. 8. A typical image sample from training database and resulting images from the augmentation.

Our work also employed scale and aspect ratio augmentation to enhance robustness in the detection of palmprint. There are evidences in the literature that indicate that when deeper neural network is incorporated, this augmentation method can offer better performance. More details on the parameters used in our experiments for the augmentation appear the following.

- Random area ratio ($a = [0.08, 1]$).
- Random aspect ratio ($s = [3/4, 4/3]$).
- Crop size: $W = sqrt(W * H * a * s); H = sqrt(W * H * a/s)$.
- Random offset to select crop center, then crop and resize

4.3 Performance Evaluation

We trained the palmprint detection model with 20K epochs which required about 7 hours for convergence on a single NVIDIA Quadro M6000. The test phase of the trained model requires an average of 0.101 seconds to generate the proposal bounding box with 300 RPN outputs.

We also performed experiments to ascertain the performance during the test phase. These experiments are organized in two categories using the strategy and parameters: (a) Separate the dataset randomly in 0.9:0.1 ratio where 0.9 represents the fraction of data for training while and 0.1 represents remaining data for test/evaluation; (b) Separate the dataset by backgrounds, where 10 different background are mixed together to form the training data and the remaining background dataset is used for the test/evaluation. The first strategy tests randomly separate the dataset into 0.9:0.1 ratio while the second strategy selects one of different backgrounds as the test set to ascertain performance. Table 3 shows the values obtained for mean average precision(mAP), and recall for the all experiments performed. One can observed that the network generates higher accuracy even up-to 0.5 to 0.6 overlap of IOU [36] threshold. Slight degradation in accuracy is observed when overlap IOU threshold is more than 0.6. The exact sample size for tests using strategy (a) and (b) is 3517 and 4770 respectively. Our palmprint detection is least affected by the rotation since we incorporate rotation as a part of data augmentation

TABLE 3
The mAP and recall value at different (IOU) threshold.

Experiments	mAP			recall		
	Overlap IOU threshold			Overlap IOU threshold		
	0.35	0.5	0.6	0.35	0.5	0.6
strategy(a)	100.0	99.89	98.20	100.0	99.84	98.97
strategy(b)	100.0	98.44	86.45	100.0	98.78	90.50

strategy during the network training process. A video file *attached* with this paper, along with samples in Figure 7, provides successful examples of online palmprint detection using unknown test samples, i.e., none of these samples were used for training. In the best of our knowledge, this is first successful attempt to detect palmprint images under complex imaging backgrounds.

Traditional or more successful methods for palmprint matching, e.g. [2], [21], incorporate multiple templates generated from different rotation while the deep learning-based architecture can incorporate augmented triplet pairs, as also in our work, during the network training. We also evaluated the online performance for the palmprint recognition using the developed detector and acquired a two-session video dataset under complex backgrounds and achieved very high accuracy. However, such video palmprint dataset was acquired from eight different subjects palmprints, or volunteers in our lab, and quite small to ascertain any reliable performance estimate expected during the real deployments.

5 Conclusions and Further Work

This paper has developed a novel deep learning based contactless palmprint feature representation model, which can offer superior matching accuracy and high generalization capability for matching contactless palmprint images. We designed a soft-shifted triplet function to enable effective supervision, in learning comprehensive and spatially corresponding residual features, using fully convolutional network. This paper also developed a robust palmprint detector that can detect contactless palmprint images from the hands presented under complex and dynamic backgrounds. Our experimental results presented in section 3 using this detector are quite encouraging and indicate promises for its usage in identifying palmprints under complex backgrounds. These results should however be considered preliminary and more work is required to further improve the palmprint detector performance, e.g. changing the detector backbone from Faster RCNN to other detector [3] which can help to reduce dislocation especially when IOU is not high. Further extension of this work should focus on jointly evaluating large-scale contactless palmprint detection and identification performance. Such evaluation requires a large-scale video dataset using palmprints acquired under complex backgrounds, with palm-pose deformations similar to recently introduced in [16] for finger knuckle, and is also part of further work in this area.

References

[1] Ashbaugh, D.R.: Palmar flexion crease identification. J. For. Ident **41**(4) (1991) 255–273

[2] Fei, L., Lu, G., Jia, W., Teng, S., Zhang, D.: Feature extraction methods for palmprint recognition: A survey and evaluation. IEEE Transactions on Systems, Man, and Cybernetics: Systems **49**(2) (2019) 343–363 2

[3] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 779–788

[4] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 770–778

[5] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis & Machine Intelligence **39**(6) (2017) 1137

[6] Girshick, R.: Fast r-cnn. In: IEEE International Conference on Computer Vision. (2015) 1440–1448

[7] Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015) 815–823

[8] Zhao, Z., Kumar, A.: Towards more accurate iris recognition using deeply learned spatially corresponding features. In: Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy. (2017) 22–29

[9] Li, Y., Zhang, X., Chen, D.: CSRnet: Dilated convolutional neural networks for understanding the highly congested scenes. arXiv preprint arXiv:1802.10062 (2018)

[10] Dai, J., Zhou, J.: Multifeature-based high-resolution palmprint recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5) (2011) 945–957

[11] Michael, G.K.O., Connie, T., Teoh, A.B.J.: Touch-less palm print biometrics: Novel design and implementation. Image and Vision Computing **26**(12) (2008) 1551–1560

[12] Ribaric, S., Fratric, I.: A biometric identification system based on eigenpalm and eigenfinger features. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(11) (2005) 1698–1709

[13] Morales, A., Ferrer, M.A., Kumar, A.: Towards contactless palmprint authentication. IET Computer Vision **5**(6) (2011) 407–416

[14] Zheng, Q., Kumar, A., Pan, G.: A 3d feature descriptor recovered from a single 2d palmprint image. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(6) (2016) 1272–1279

[15] Zheng, Q., Kumar, A., Pan, G.: Suspecting less and achieving more: new insights on palmprint identification for faster and more accurate matching IEEE Transactions on Information Forensics and Security **11**(3) (2016) 633–641

[16] Kumar, A.: Toward pose invariant and completely contactless finger knuckle recognition IEEE Transactions on Biometrics, Behavior, and Identity Science **1**(3) (2019) 201–209

[17] Sun, Z., Tan, T., Wang, Y., Li, S.Z.: Ordinal palmprint representation for personal identification. 2005. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition

[18] Zhang, L., Li, L., Yang, A., Shen, Y., Yang, M.: Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach. Pattern Recognition **69** (2017) 199–212

[19] Kong, A.W.K., Zhang, D.: Competitive coding scheme for palmprint verification. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on., IEEE (2004) 520–523

[20] Shao, H., Zhong S., Li, Y..: Palmgen for cross domain palmprint recognition. In: ICME 2019. Proceedings of International Conference on Multimedia and Expo, IEEE (July 2019) 1390–1395

[21] Jia, W., Huang, D.S., Zhang, D.: Palmprint verification based on robust line orientation code. Pattern Recognition **41**(5) (2008) 1504–1513

[22] IITD Touchless Palmprint Database (ver 1.0). http://www.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm Jan. 2014.

[23] Data augmentation for machine learning experiments. https://github.com/aleju/imgaug Jan. 2018.

[24] CASIA Palmprint Database. http://biometrics.idealtest.org/ 2016.

[25] PolyU-IITD Contactless Palmprint Images Database (version 3.0). http://www4.comp.polyu.edu.hk/~csajaykr/palmprint3.htm 2019.

[26] Wang, Y., Peng, L., Wang, S., Ding, X.: Contactless palm landmark detection and localization on mobile devices. Electronic Imaging **2016**(7) (2016) 1–6

[27] Kumar, A., Wang, K.: Identifying humans by matching their left palmprint with right palmprint images using convolutional neural network. Proc. DLPR (2016)
[28] Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1891–1898
[29] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. (2015) 448–456
[30] Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. CoRR, abs/1703.06868 (2017)
[31] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, Springer (2014) 818–833
[32] Wu, X., Zhao, Q.: Deformed palmprint matching based on stable regions. IEEE Transactions on Image Processing **24**(12) (2015) 4978–4989
[33] Web link to download codes for reproducibility of the approach detailed in this paper. http://www.comp.polyu.edu.hk/ csajaykr/RFN.rar 2018.
[34] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al.: Going deeper with convolutions, CVPR (2015)
[35] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
[36] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4) (1989) 541–551
[37] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088) (1986) 533
[38] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
[39] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
[40] Liu, Y., Kumar, A.: A deep learning based framework to detect and recognize humans using contactless palmprints in the wild. arXiv preprint arXiv:1812.11319 (2018)
[41] Hosang, J., Benenson, R., Dollar, P., Schiele, B: What Makes for Effective Detection Proposals? IEEE Transactions on Pattern Analysis & Machine Intelligence **38**(4) (2016) 814-830