

Article

Deep Region of Interest and Feature Extraction Models for Palmprint Verification Using Convolutional Neural Networks Transfer Learning

Mahdieh Izadpanahkakhk ¹ , Seyyed Mohammad Razavi ^{1,*} ,
Mehran Taghipour-Gorjikolaie ¹ , Seyyed Hamid Zahiri ¹  and Aurelio Uncini ² 

¹ Department of Electrical and Computer Engineering, University of Birjand, Birjand 971481151, Iran; Izadpanah1987@birjand.ac.ir (M.I.); mtaghipour@birjand.ac.ir (M.T.-G.); hzahiri@birjand.ac.ir (S.H.Z.)

² Department of Information Engineering, Electronics and Telecommunications of the University of Rome “La Sapienza”, 00185 Rome, Italy; aurelio.uncini@uniroma1.it

* Correspondence: smrazavi@birjand.ac.ir

Received: 15 April 2018 ; Accepted: 2 July 2018; Published: 23 July 2018



Abstract: Palmprint verification is one of the most significant and popular approaches for personal authentication due to its high accuracy and efficiency. Using deep region of interest (ROI) and feature extraction models for palmprint verification, a novel approach is proposed where convolutional neural networks (CNNs) along with transfer learning are exploited. The extracted palmprint ROIs are fed to the final verification system, which is composed of two modules. These modules are (i) a pre-trained CNN architecture as a feature extractor and (ii) a machine learning classifier. In order to evaluate our proposed model, we computed the intersection over union (IoU) metric for ROI extraction along with accuracy, receiver operating characteristic (ROC) curves, and equal error rate (EER) for the verification task. The experiments demonstrated that the ROI extraction module could significantly find the appropriate palmprint ROIs, and the verification results were crucially precise. This was verified by different databases and classification methods employed in our proposed model. In comparison with other existing approaches, our model was competitive with the state-of-the-art approaches that rely on the representation of hand-crafted descriptors. We achieved a IoU score of 93% and EER of 0.0125 using a support vector machine (SVM) classifier for the contact-based Hong Kong Polytechnic University Palmprint (HKPU) database. It is notable that all codes are open-source and can be accessed online.

Keywords: region of interest extraction; palm print verification; deep learning; convolutional neural network; transfer learning; feature extraction

1. Introduction

Biometric-based authentication has been discussed in a wide range of state-of-the-art research in the context of security applications. The increasing amount of industrial and governmental funding on this topic makes it a rapidly growing field. There are several biometric characteristics (e.g., DNA, face, and palmprint) that can be exploited in authentication systems [1]. Each one has its own pros and cons, and therefore there is no single optimal characteristic that satisfies all application requirements [2]. Consequently, based on the constraints and conditions of the operational mode, one or more biometric characteristics can be applied. There are several advantages that make the palmprint authentication more appropriate for real-world applications [3–6]: (i) low-cost image acquisition devices; (ii) applicability on both low and high-resolution images; (iii) discriminative features including ridges and creases; (iv) wide region of interest; (v) unique and reliable properties; (vi) hardly affected by age; and (vii) the high user acceptance of palmprint.

Among the existing approaches for palmprint verification, convolutional neural networks (CNNs) outperform the state-of-the-art techniques. CNNs are a class of deep learning concerned with architectures inspired by the structure and function of the brain, called artificial neural networks. Well-known CNN architectures such as Alexnet [7], VGGNet [8], and ResNet [9] have shown excellent performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competitions [10]. Although CNN-based approaches effectively represent the biometric and perceptual features from the input data [11,12], there are some challenges in applying this concept to palmprint verification. On one hand, the number of samples is a challenging issue in current palmprint databases since the common CNN techniques need a large amount of input data for the training phase. On the other hand, CNN performance strongly depends on the deep architecture. Due to these issues, using a CNN architecture may result in overfitting in the case of small databases. It should be mentioned that data augmentation techniques can hardly reduce overfitting because of small intra-class variability among palmprint images.

Because it is possible to use a pre-trained CNN to solve new problems faster, a paradigm called *transfer learning* has appeared [13]. Transfer learning is useful when one wants to train a CNN on a dataset, but due to insufficient training data, it is not possible to train a full neural network on that database. Transfer learning refers to the process of taking a pre-trained CNN, replacing the fully connected layers (and potentially the last convolutional layer), and training those layers on the pertinent dataset. By freezing the weights of convolutional layers, a CNN can extract discriminative image features such as edges, and fully connected layers can take this information and use it to classify the data in a way that is pertinent to the problem. By using this concept, we can significantly address the challenges of CNN approaches, including high computational complexity in the training phase and overfitting caused by small palmprint databases. The issue that still remains a challenge in palmprint verification using CNN transfer learning is to track and detect the region of interest (ROI), which is commonly a square region in the center of the palmprint.

1.1. Contributions

Motivated by the aforementioned considerations, we address the problem of palmprint verification using a transfer learning approach. Our approach uses low-resolution images, and the main contributions are summarized as follows:

- To the best of our knowledge, this is the first study that extracts palmprint ROIs by convolutional neural networks. Lack of sample images and small intra-class variability are addressed using transfer learning.
- We successively apply a pre-trained CNN network to extract discriminative features along with a machine learning classifier to measure the similarity for one-to-one matching.
- We achieve an intersection over union (IoU) score of 93% for palmprint ROI extraction and equal error rate (EER) of 0.0125 for the verification task using the support vector machine (SVM) classifier for the contact-based Hong Kong Polytechnic University Palmprint (HKPU) database. This can be attributed to the superiority of discriminative deeply learned features over hand-crafted features.

1.2. Paper Organization

The remainder of this paper is organized as follows. Section 2 presents a comprehensive literature review. Section 3 presents the problem definition, related assumptions, and overviews the proposed architecture and its main components. The preliminaries are discussed in Section 3.1, and an outline of the proposed model for deep ROI and feature extraction is provided in Section 3.2. Sections 3.3–3.5 detail the proposed ROI extraction, feature representation, and matching modules, respectively. The obtained results are detailed in Section 4. Finally, Section 5 concludes the paper with some final remarks and outlines open research problems.

2. Related Work

In the following, we briefly discuss the main literature on palmprint verification. We divide the related works into two main categories: (i) ROI extraction and (ii) feature extraction and matching.

2.1. ROI Extraction

The ROI extraction phase plays an important role in palmprint verification, since the accuracy of the whole process is highly dependent on ROI extraction, where several findings have been discussed in previous work [14–25]. In [18], ROI extraction steps consist of (i) skin-color thresholding, (ii) hand valley detection, and (iii) finding palm region. The authors used Chang and Robles's skin-color model in the thresholding phase. In the second and third phases, they proposed the competitive hand valley detection (CHVD) algorithm to locate valley key-points. Thereafter, the output was exploited to locate the ROI. ROI extraction based on skin-color segmentation is used in [19,20]. Michael et al. [19] extended the CHVD algorithm to simultaneously track palmprint and knuckle-print ROIs. The most challenging issue in these approaches is poor segmentation for images with a skin-colored background. To address this problem, statistical segmentation models were employed. In this way, Reference [21] exploited active shape model (ASM), while [22,23] proposed their methods based on the active appearance model (AAM). These approaches suffer from the probability of important features loss due to small extracted ROIs.

Zhang et al. [14] presented an ROI extraction technique including several steps. First, they applied a low-pass filter to the original image. Thereafter, boundaries of the gaps between the fingers were obtained using a boundary tracking algorithm. Finally, based on the coordinate system, a fixed-size sub-image was extracted. Similarly, the methods of Connie [15], Han [16], and Badrinath [17] used the finger gaps to extract the ROI. Although these methods are famous and widely used, they depend on the gaps between the fingers as reference points to determine the coordinate system, which means that all fingers must be spread and the hand should be facing toward the camera. To tackle this problem, Ito et al. [24] proposed a five-step method that can be summarized as follows: (i) binarization of the input image, (ii) combination of binarized image and edge, (iii) key point candidate detection using the radial-distance function, (iv) optimal key-point selection, and (v) palm region extraction. This approach suffers from the probability of important features loss due to small extracted ROIs.

2.2. Feature Extraction and Matching

So far, many approaches have been proposed for feature extraction and matching tasks that can be categorized as hand-crafted and deep learning descriptors. Comprehensive studies of feature extraction approaches are provided in [26–29], which are divided into three main categories: holistic-based, local-based, and hybrid methods.

Among the available local feature-based approaches, some have been shown to perform effectively on low-resolution palmprint images. To extract principal lines (PriLine), Malik et al. [30] applied a Radon filter as an edge detector and provided two levels of authentication in order to increase the accuracy. Moreover, there are some coding-based methods, including competitive code (CompCode) [31], double orientation code (DOC) [32], and extended binary orientation co-occurrence vector (E-BOCV) [33], where a bank of Gabor filters was applied to represent the orientation features of palmprint images. The local line directional patterns (LLDP) [34], as a local texture descriptor, also extract the local line directional features by convolving the line filters with a palmprint image such as local line directional patterns-modified finite radon transform (LLDP-MFRAT) and LLDP-Gabor descriptors.

Although there are numerous studies in this field [35–49], sensitivity to illumination, distortion, translation, and rotation variances still remains as a main challenge [6]. Furthermore, some local feature-based descriptors in hand-crafted techniques highly depend on the input image resolution, leading to dramatic increment in the sensor cost [28].

Considering these issues, instead of designing algorithms for hand-crafted descriptors, which can be very time-consuming, feature extraction can be automatically obtained by deep learning descriptors [50]. This automatic feature mapping results in high-level features which can hardly be obtained via hand-crafted approaches [51]. Another advantage of deep learning descriptors over the hand-crafted ones is that they can be trained with a large number of inputs to become robust against illumination, distortion, translation, and rotation variances. Automatic feature engineering exploits the hierarchical architecture to effectively learn complex models that are highly appropriate for unconstrained conditions (e.g., image background or palm position in the input image) [51].

Motivated by the aforementioned considerations, CNN as the most notable deep learning descriptor has attracted researchers' attention in recent years [5,52–56]. The work of Kumar and Wang [52] explored the matching possibility of left and right palmprint images, and several algorithms were investigated; the best results were obtained by CNN. Although they focus on an interesting topic, their method error rate is especially high for forensic applications. The authors of [5] used a principal component analysis network (PCANet) model [57] in feature extraction and an SVM classifier for palmprint identification. In this way, each spectral band was represented by features extracted by a deep learning technique. Since they used the ROI extraction method described in [14], their approach lacks robustness against illumination, distortion, translation, and rotation variances in input images. For the same reason, the approaches in References [53,54] are not robust against input image deformation. Minaee et al. [55] exploited a scattering network (a convolutional network where the architecture and filters are predefined wavelet transforms) for feature extraction, but no learning was involved in the ROI extraction. In [56], the input images were resized into 28×28 pixel images without applying ROI extraction techniques, which led to missing important features.

3. Proposed Model

Here the proposed model is precisely discussed. Basic preliminary facts about the CNN architecture used in this paper are discussed, followed by the outline of the proposed model.

3.1. Background

Chatfield et al. [58] described a fast CNN architecture inspired by AlexNet [7] model. They have shown that reducing the number of kernels in convolutional layers in their architecture does not impact the output performance in comparison to AlexNet. Figure 1 illustrates the aforementioned fast CNN architecture. This architecture is comprised of eight layers, including five convolutional layers (Conv. Layers) and three fully connected ones (FC Layers). The input image of this architecture is $224 \times 224 \times 3$ and the structure of the five convolutional layers are as follows: (i) 64 kernels of size $11 \times 11 \times 3$ in the first layer (Conv. 1 in Figure 1), (ii) 256 kernels of size $5 \times 5 \times 64$ in the second layer (Conv. 2), and (iii) 256 kernels of size $3 \times 3 \times 256$ in the next three layers (Conv. 3, Conv. 4, and Conv. 5). Both of the first two fully connected layers has 4096 neurons (FC6 and FC7). The last fully connected layer, referred to as the softmax layer, has an output of 1000 dimensions (FC8).

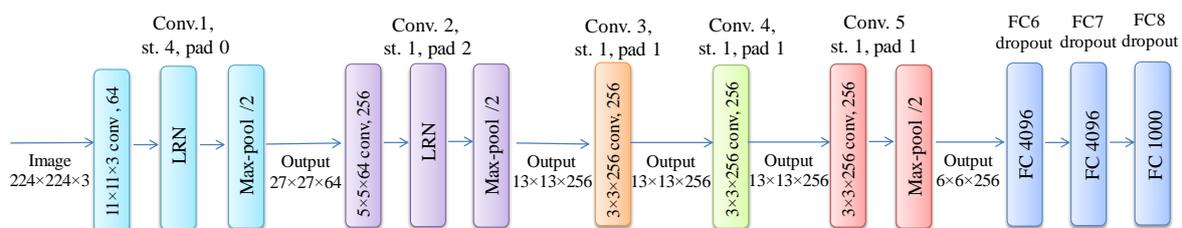


Figure 1. Chatfield's fast convolutional neural network (CNN) architecture. Conv: convolutional layer; FC: fully connected layer; st: stride; pad: padding; LRN: local response normalisation.

3.2. Outline

The outline of the proposed model is illustrated in Figure 2, which is composed of three modules: (i) ROI extraction module (REM), (ii) feature extraction module (FEM), and (iii) matching module (MM). In the first module, palmprint ROIs are extracted using the bounding box concept. A pre-processing procedure is applied to the input images. Thereafter, the location of the boxes on the palmprint images are specified using a CNN transfer learning approach. The output of this module is palmprint ROIs, which are the inputs of the feature extraction module. The FEM employs a pre-trained CNN architecture to represent features. Finally, this feature vector feeds the MM as the input to perform the verification task using a machine learning classifier.

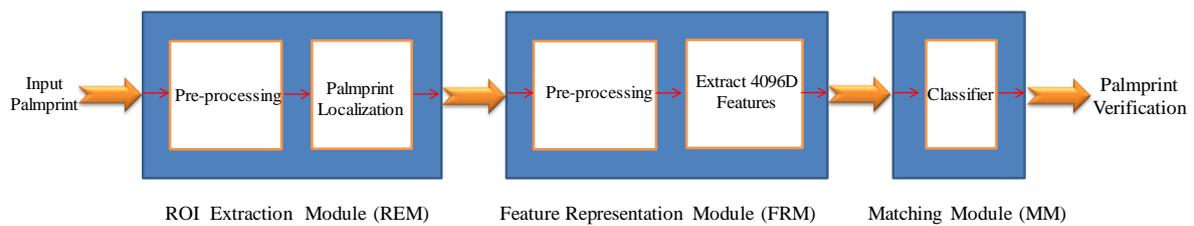


Figure 2. Our proposed model for palmprint verification. ROI: region of interest.

3.3. ROI Extraction Module

We propose an architecture for ROI extraction based on the one discussed in Section 3.1 by replacing the last fully connected layer, FC8, with a four neuron-layer unit (depicted in Figure 3). Since Chatfield’s architecture expects a fixed pixel size, a pre-processing procedure is required for input images. In brief, REM consists of two steps: (i) pre-processing and (ii) palmprint localization. In the following, each of these parts is discussed in detail.

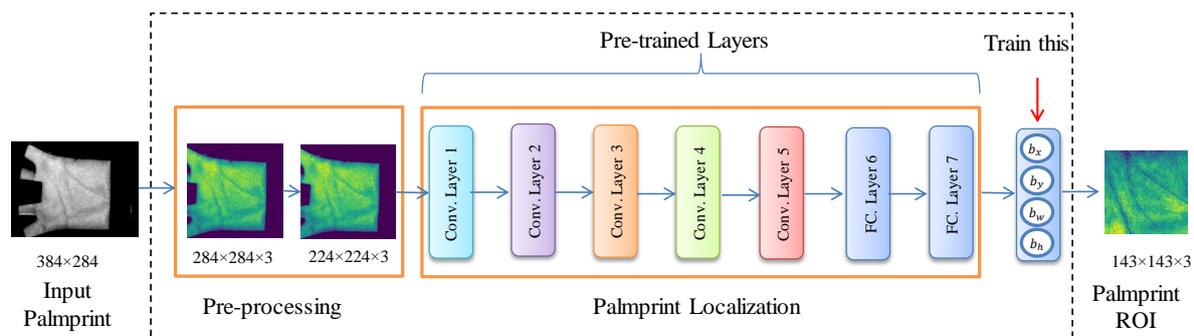


Figure 3. ROI extraction module. B_x and B_y : center point; B_y : width and B_h : high of the box containing the palmprint ROI.

3.3.1. Pre-Processing

As the network needs inputs of $224 \times 224 \times 3$ pixel size, the pre-processing step maps gray-scale palmprint images to suitable ones to be fed into our CNN architecture. To this end, the inputs are rescaled to square images by removing the extra pixels from both left and right sides of the inputs. Then, these gray-scale images are converted to RGB color space by the weighted method and are resized to $224 \times 224 \times 3$ using the nearest neighbour method (nearest-neighbor interpolation, http://en.wikipedia.org/wiki/Nearest-neighbor_interpolation). Figure 3 illustrates an example of the final pre-processed image.

3.3.2. Palmprint Localization

Palmprint localization is the process of finding the location of the ROI in palmprint images by putting a bounding box or drawing a rectangle around the position of the palmprint in the images. Since CNN-based approaches have shown encouraging performance in object localization [59], we propose a CNN architecture for palmprint localization based on Chatfield’s fast CNN (Section 3.1).

As shown in Figure 3, we keep the first seven layers of Chatfield’s architecture and replace the last layer with a four neuron-layer unit that outputs a bounding box. The four numbers parameterize the bounding box, determining the extracted palmprint ROI. In the architecture, the localization can be specified using four parameters (b_x, b_y, b_w , and b_h) representing the center point, width, and height of the box containing the palmprint ROI, respectively.

Based on the fact that there are not enough palmprint images in the available databases, we employ the learning concept. To this end, the network weights of the first seven layers are pre-trained on ImageNet using the Alexnet model and the final layer is trained using our database. In other words, we freeze the parameters in all network layers, except the last one, and then we train and fine-tune pertinent parameters associated with our bounding boxes (i.e., b_x, b_y, b_w , and b_h). The palmprint ROIs can be extracted by applying these parameters to our palmprint images.

Considering ω as the set of learning parameters and $b'_x(\omega), b'_y(\omega), b'_w(\omega), b'_h(\omega)$ as the network output values, we define $C(\omega)$, the cost function, as follows:

$$C(\omega) = \frac{1}{m} \sum_{i=1}^m \left(\left| b_{x_i} - b'_{x_i}(\omega) \right| + \left| b_{y_i} - b'_{y_i}(\omega) \right| + \left| b_{w_i} - b'_{w_i}(\omega) \right| + \left| b_{h_i} - b'_{h_i}(\omega) \right| \right) \quad (1)$$

where m is the number of samples and $b_{x_i}, b_{y_i}, b_{w_i}, b_{h_i}$ are the target values for the i th sample. The cost function specifies the difference between the target values and predicted ones. The objective is to determine a set of ω minimizing $C(\omega)$.

3.4. Feature Extraction Module

The main goal of FEM, as shown in Figure 4, is to represent distinctive features of palmprint ROIs. Because the output of REM is a $143 \times 143 \times 3$ palmprint ROI but the network input should be $224 \times 224 \times 3$ pixel size, a pre-processing step is expected. To do this, the images are resized based on the network requirements.

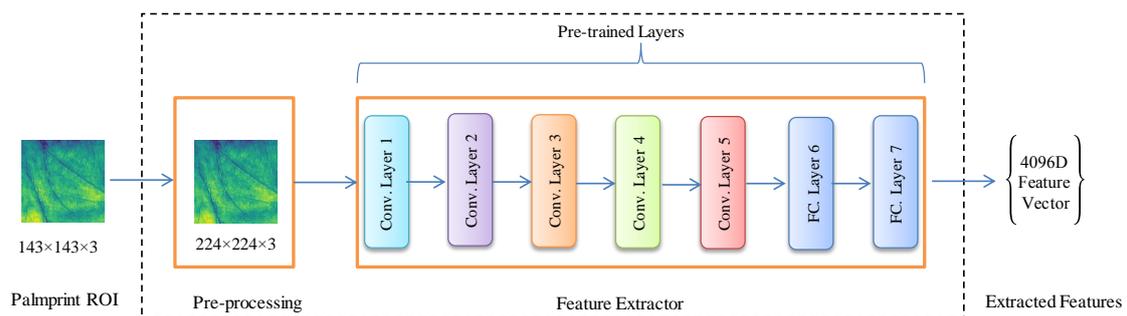


Figure 4. Feature extraction module.

In the next step, a pre-trained network is used for the feature extraction task. As mentioned, before using transfer learning techniques in the feature extractor, the learning process starts based on the weights learned from the ImageNet database. We should fine-tune these pre-learned parameters on our pertinent database using our limited palmprint ROI images. In this way, we use a CNN architecture similar to Figure 1, remove the last layer (FC8), and then initialize it with the learned weights to implement parameters transfer. The advantage of this pre-trained architecture is that

it has the superior capacity of the fully trained network. After that, we fine-tune this pre-trained architecture by passing palmprint ROIs through the pre-trained layers. It should be noted that the fine-tuning process is similar to the training process except that not all weight parameters should be fine-tuned because of the lack of training data. The effective way to fine-tune the pre-trained weights is to optimize only the last fully connected layer and keep the parameters of other layers frozen. To fine-tune these parameters, some superior optimizers are used to adjust the pre-trained weights automatically during training. The Adam optimizer [60] is a gradient-based first-order algorithm used to optimize stochastic objective functions. The optimizer updates exponential moving averages with the learning rate and the exponential decay rates β_1 and β_2 .

In order to evaluate this module and analysis fine-tuning process, the cost function of fine-tuning to minimize the difference between the target output and predicted output is calculated. To prevent overfitting, we use a logistic regression algorithm, L , using the function below:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

So, the cost function can be expressed as below:

$$C(\Omega) = \frac{1}{m} \sum_{i=1}^m \left(\left| L(\Omega'_i, \Omega_i) \right| \right) \quad (3)$$

where Ω is the set of parameters that needs to be optimized during the fine-tuning process, Ω' is the set of predicted parameters, and m is the number of samples. It should be mentioned that the output is a fixed length 4096D vector for the different databases.

3.5. Matching Module

Two approaches are possible for the matching process: combining MM and FEM as a CNN block or using a separate classifier like SVM, K-nearest neighbor (KNN), or random forest (RF). The main differences between MM and REM are the number of existing samples for each class. In the REM, all input images belong to one class (palmprint), and therefore all database images can be used for learning of that class. On the other hand, there are several classes—one per individual—in the matching phase. To explain further, consider n as the number of images in the pertinent database, p as the number of individuals, and x as the number of existing samples per class. There are n samples for each class in REM ($x_{REM} = n$), while it is n/p for MM ($x_{MM} = n/p$). As an example, in the typical contact-based Hong Kong Polytechnic University Palmprint (HKPU) database (<http://www4.comp.polyu.edu.hk/~biometrics/>), $x_{REM} = 7752$ and $x_{MM} \approx 20$. Due to a lack of training samples per individual, combining MM and FEM as a CNN block may lead to overfitting, even by exploiting the transfer learning.

Based on the aforementioned considerations, we chose a separate classifier for the MM. In this manner, three different classifiers were evaluated as the MM: support vector machines (SVMs), K-nearest neighbor (KNN), and random forests (RFs). SVMs are supervised learning models with associated learning algorithms that analyze data used for classification. Given a set of training examples, each marked as belonging to one class, the SVM training algorithm builds a model that assigns new examples to one class, making it a linear classifier. SVMs are the representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. At this point, new examples are then mapped into that same space and predicted to belong to a class based on which side of the gap they fall. In more detail, SVM constructs a set of hyperplanes in a high-dimensional space to classify the input features. The main advantages of SVMs are [61]: (i) effective in high-dimensional spaces, (ii) effective in cases where the number of samples is lower than the number of dimensions, (iii) efficient memory usage (because it

uses a subset of training points in the decision function called support vectors), and (iv) flexibility (different Kernel functions can be used for the decision function).

KNN is a non-parametric technique that classifies new cases based on the majority vote of its k -nearest neighbors, where the votes are considered according to the distances between testing and training samples. RF is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control overfitting.

4. Experimental Results

In this section, we first explain the pertinent databases, the exploited framework, the optimizer, and the parameters set in our experiment. Thereafter, since the output of the REM is a region in the input image, it is evaluated via computing the IoU metric. The FEM is analyzed by the cost function, explained in Section 3.4 during the fine-tuning process. The final output is evaluated via accuracy, receiver operating characteristic (ROC) curves, and EER.

4.1. Settings

In this paper, we used low-resolution palmprint images including both contact-based and contact-less databases. The typical HKPU version 2 was employed as contact-based palmprint images and the 2D contact-free HKPU [62] and Indian Institute of Technology Delhi (IITD) Touchless Palmprint Database [63] were used as the contact-less ones.

The typical contact-based HKPU contains 7752 gray-scale images corresponding to 193 subjects or 386 palms. The subjects were 131 males and 62 females. The minimum number of samples corresponding to each palm was 10, collected at two different time intervals. To have a consistent number of samples assigned to each class, we utilized 10 samples and ignored the extra ones. Although it is mentioned by the database owner that there are approximately 18 samples for each person, it should be noted that the existing samples of the 150th individual are 11.

In 2D contact-free HKPU, each individual contributed five hand images in the first session, followed by another five in the second session. The database currently has 3540 hand images.

The IITD palmprint image database includes hand images from 230 users, all the images are in bitmap (*.bmp) format. Subjects in the database were in the age group 12–57 years. Five to six images from each person's palm exist in the database.

The simulation was performed in Python programming language, and the TensorFlow [64] framework was exploited to implement the CNN methods. Based on the documentation, it is recommended to leave the parameters of networks at their default values, except sometimes for the learning rates. In our experiment, the Adam optimizer was exploited with the good default settings by the tested machine learning problems, for $\beta_1 = 0.9$ and $\beta_2 = 0.999$ in the CNN architecture of REM and FEM. Besides, according to the documentation, the recommended learning rates are 1, 0.01, 0.001, and 0.0001. We used 70% of the palmprint images as the training set and the remaining as the test set for palmprint ROI extraction.

4.2. Evaluation of ROI Extraction Module

As mentioned in Section 3.3.1, a pre-processing step is needed before extracting the bounding boxes. Considering the contact-based HKPU database, the pre-processing step maps gray-scale palmprint images of 384×284 to suitable ones to be fed into our CNN architecture. To this end, the inputs were rescaled to 284×284 square images and thereafter were represented in RGB. Finally, the images were resized to $224 \times 224 \times 3$ as the input of the network for the ROI extraction task.

In REM, we set the learning rate in two modes: a constant value of $L_r = 10^{-4}$ and an adaptive one in which it starts with a predefined value (10^{-3}) and decays by a factor of 0.96 during steps. Experimental results showed the same performance for both values, therefore, we report the results of the constant learning rate.

4.2.1. Training Phase

In order to evaluate this module, we used the cost function described in Section 2.1. In our experiment, the training procedure was considered to be completed after 100 epochs. Figure 5 illustrates the predicted boxes during different epochs in the training procedure.

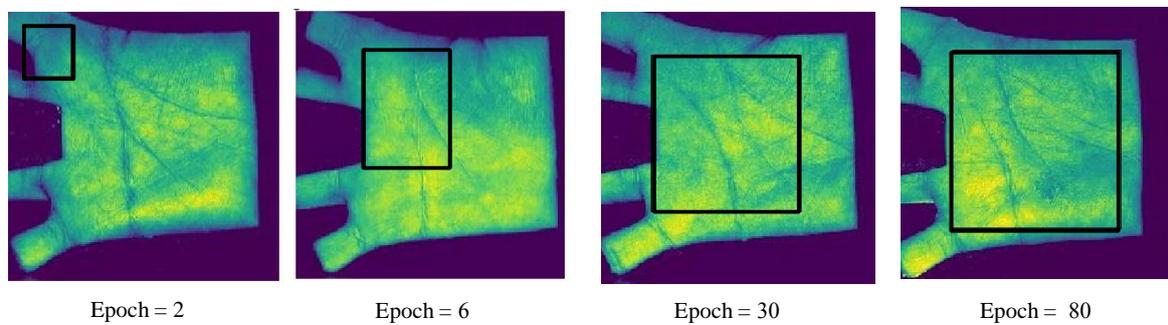


Figure 5. Examples of predicted ROIs for different epochs.

As can be seen, the bounding box fit the intended ROI during epochs. The learning parameters of each epoch were supposed as the inputs of the cost function depicted in Figure 6.

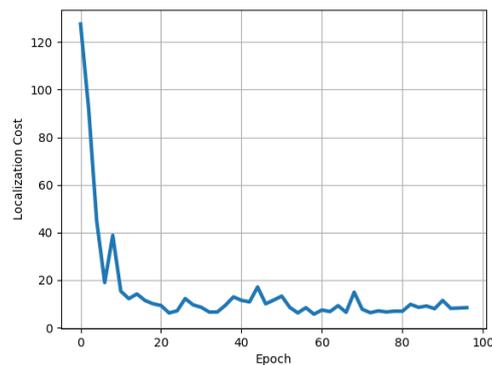


Figure 6. Palmprint localization cost versus epochs.

Based on this plot, the cost of palmprint localization dramatically decreased during the early steps while it started to become steady after approximately 100 epochs.

4.2.2. Testing Phase

In this phase, we applied the remaining images to our network as the test set. The results of the testing phase, illustrated in Figure 7, show that REM performed very well on the test set.

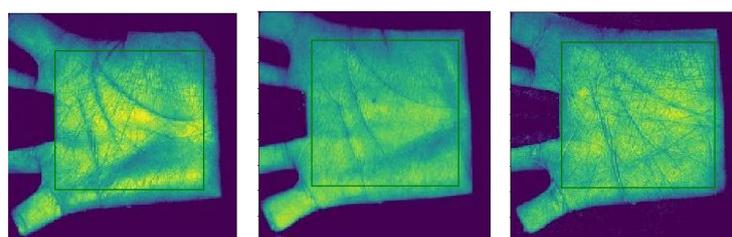


Figure 7. Examples of extracted palmprint ROIs in the testing phase.

To investigate the performance of the proposed REM, we used the IoU metric [65], which measures the overlap between the predicted ROI and ground truth ROI or the target region. Considering ground-truth ROIs as R_g and the corresponding predicted ROIs by our model as R_p , the IoU can be calculated by the Formula (4):

$$IoU = \frac{\text{area}(R_g \cap R_p)}{\text{area}(R_g \cup R_p)} \tag{4}$$

Figure 8 presents a visual sample of the ground-truth ROI versus the predicted ones by our model. It should be noted that ground truth ROIs are the ones firstly produced by bounding boxes in our own works. The final IoU score is obtained by the mean taken over all classes, and was 93% in our experiment.

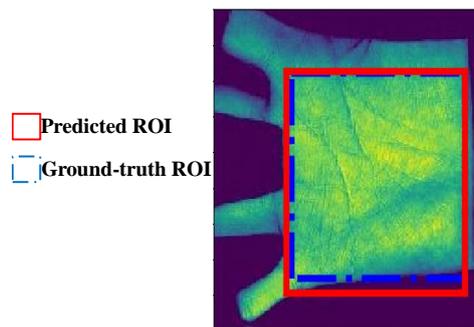


Figure 8. A visual sample of the intersection over union (IoU) metric in our proposed model.

In order to validate the effectiveness of our proposed ROI extraction model, we compared it to the Han model [16], which performs well even under unconstrained scenes.

Figure 9 shows the learned features for the palmprint localization task in our model and the Han method. The Han method selects Haar features in order to represent the shape features of the hand, while our model extracts discriminative features automatically that are easier to learn than the complicated Haar features. In our ROI extraction module, the first convolutional layer extracts basic features like edges and lines, and other convolutional layers represent high-level features automatically.

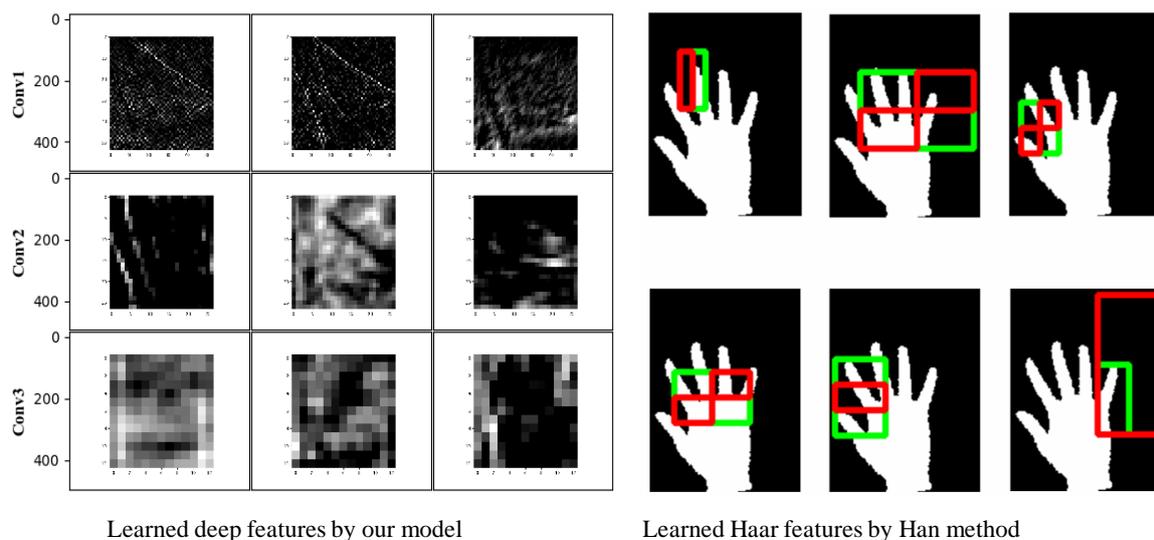


Figure 9. Comparison of learned features in our ROI extraction model with the Han method [16].

As depicted in Figure 10, our proposed model could effectively extract the palmprint ROI using deep features in comparison to the Han method, especially in segmenting the background from the palmprint and keeping the important regions of the palm.

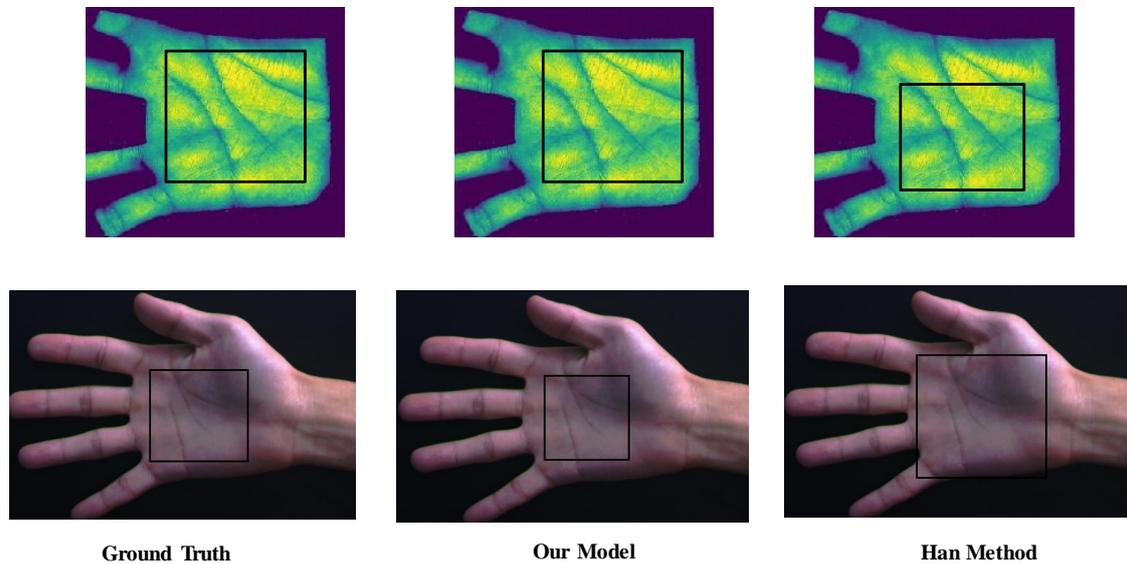


Figure 10. Comparison of extracted palmprint regions with the ground truth ROI.

4.3. Evaluation of Feature Extraction and Matching Modules

To extract palmprint ROI features, we passed palmprint ROI images, obtained by REM, through the network and optimized the final vector of $n \times 4096$ feature matrix, where n is the overall number of images in each database. In other words, the values of each row in this vector correspond to the features of each individual’s image. For example, considering $n = 3480$ for the contact-based HKPU database, the first 18 images of overall 3480 images are related to the first person, the second batch of 18 images is for the second person, and so on. A learning rate of $L_r = 10^{-5}$ was fixed for the FEM network. The cost function of the fine-tuning process in FEM is depicted for three databases in Figure 11.

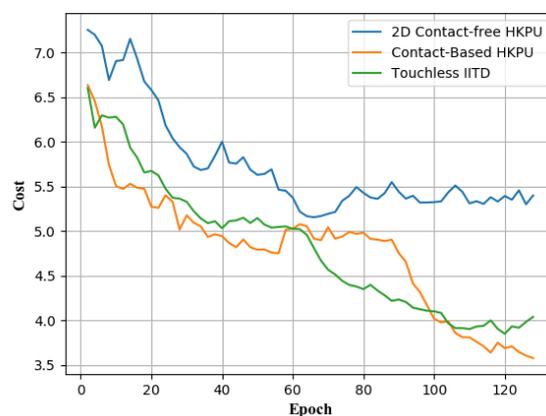


Figure 11. Fine-tuning cost versus epoch in Feature Extraction Module for contact-based Hong Kong Polytechnic University (HKPU), 2D contact-free HKPU, and Indian Institute of Technology Delhi (IITD) Touchless Palmprint Databases.

As we show, the network parameters could be well optimized during the fine-tuning process. Moreover, the cost function had the lowest value for contact-based HKPU because of more discriminative features in the database images. To have a better perspective on the issue, we visualize the feature vector of the FC7 layer in FEM as an image shown in Figure 12. In other words, we reshaped the 4096D matrix into a 64×64 image. These extracted deep features were used as the input of the SVM classifier in the verification task.

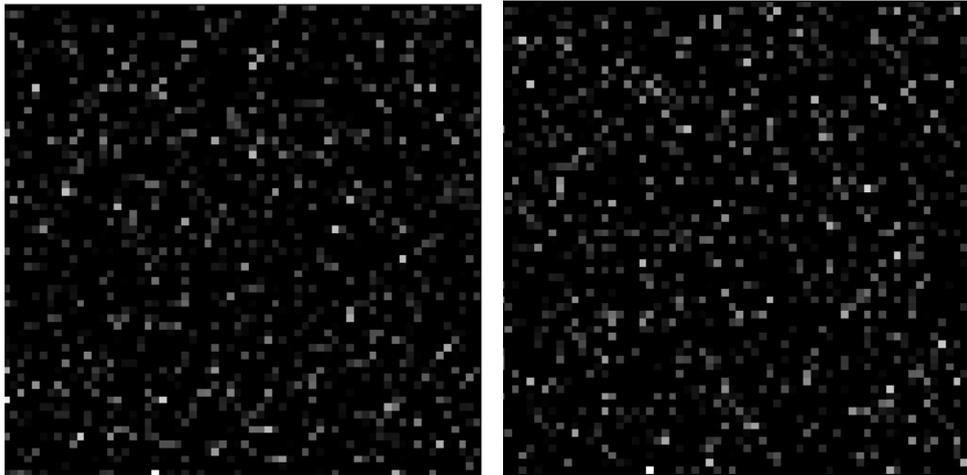


Figure 12. Visualization of the extracted deep features for two individuals.

To evaluate the matching module, we compared the performance of different methods via verification accuracy, ROC curves, and EER using four main concepts: true positives (TPs) (data items correctly predicted as positive), false positives (FPs) (data items incorrectly predicted as positive), true negatives (TNs) (data items correctly predicted as negative), and false negatives (FNs) (data items incorrectly predicted as negative.). Furthermore, the verification performance of each classifier was calculated by matching palmprint ROIs from two sessions. The first classifier was a linear SVM with parameters $C = 1$ and $\text{max iter} = 1000$. We also applied KNN, which classifies feature vectors by the number of neighbors 3 via the Minkowski metric. The last was the RF classifier, for which the number of estimators was 200.

4.3.1. Accuracy

As mentioned, we calculated the accuracy of three classifiers for the assessment of the verification task. The accuracy can be defined by the following formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (5)$$

Although the main focus was on evaluating our proposed model, we also validated the verification performance for different training sample sizes.

The impact of different classification methods is illustrated in Figures 13–15 related to the matching stage. As can be seen in Figure 13, linear SVM outperformed other classification methods and achieved a verification accuracy of 1 for the contact-based HKPU database. Additionally, SVM showed the best performance in terms of verification accuracy for 2D contact-free HKPU and IITD touchless databases.

Tables 1–3 show the accuracy results obtained using SVM, KNN, and RF classifiers with the input of extracted deep features on three databases. The overall trend shows that the larger the training sample size, the higher the accuracy. It can be realized that the classifiers in our proposed model required low training samples to achieve good performance.

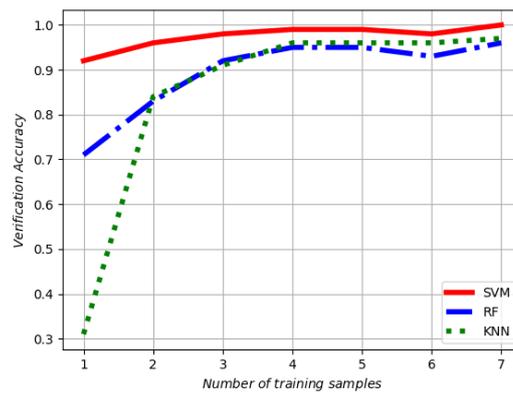


Figure 13. Accuracy curves of support vector machine (SVM), random forest (RF), and K-nearest neighbour (KNN) classifiers in the contact-based Hong Kong Polytechnic University Palmprint (HKPU) database.

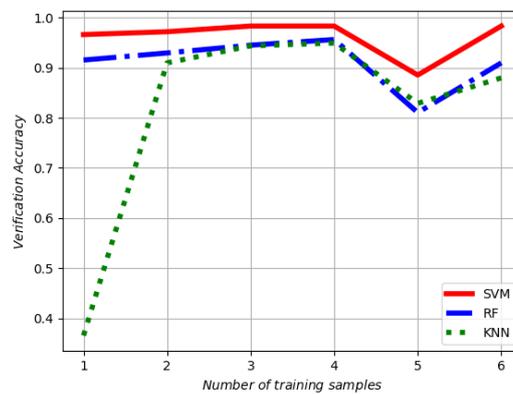


Figure 14. Accuracy curves of SVM, RF, and KNN classifiers in the 2D contact-free HKPU database.

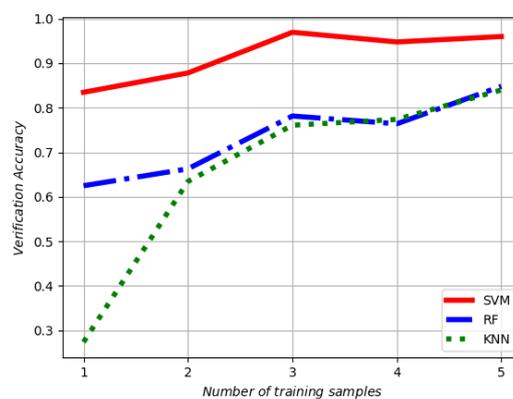


Figure 15. Accuracy curves of SVM, RF, and KNN classifiers in the IITD touchless database.

Table 1. Verification accuracy obtained using different classifiers on the contact-based HKPU database. MM: Matching Module.

Classifier in MM	Number of Training Samples						
	1	2	3	4	5	6	7
SVM	0.927	0.968	0.989	0.994	0.994	0.989	1
KNN	0.310	0.844	0.917	0.963	0.968	0.963	0.974
RF	0.718	0.836	0.924	0.955	0.959	0.931	0.963

Table 2. Verification accuracy obtained using different classifiers on the 2D contact-free HKPU database.

Classifier in MM	Number of Training Samples					
	1	2	3	4	5	6
SVM	0.966	0.971	0.983	0.983	0.885	0.983
KNN	0.365	0.909	0.943	0.949	0.828	0.879
RF	0.915	0.929	0.944	0.956	0.810	0.909

Table 3. Verification accuracy obtained using different classifiers on the IITD touchless database.

Classifier in MM	Number of Training Samples				
	1	2	3	4	5
SVM	0.834	0.878	0.969	0.947	0.969
KNN	0.27	0.63	0.760	0.773	0.84
RF	0.640	0.660	0.783	0.766	0.841

4.3.2. Receiver Operating Characteristic

The receiver operating characteristic (ROC) curve is a graphical plot of trade-off which illustrates genuine acceptance rate (GAR) as the y coordinate versus false acceptance rate (FAR) as the x coordinate. The equations for GAR and FAR are as follows:

$$GAR = \frac{TP}{TP + FN'} \tag{6}$$

$$FAR = \frac{FP}{TP + FN} \tag{7}$$

A robust way to compare the performance of different classifiers is to measure the area under the ROC curves (AUCs). Figures 16–18 show ROC curves and AUC metric values for three databases. As can be seen, SVM had the highest AUC value among the classifiers.

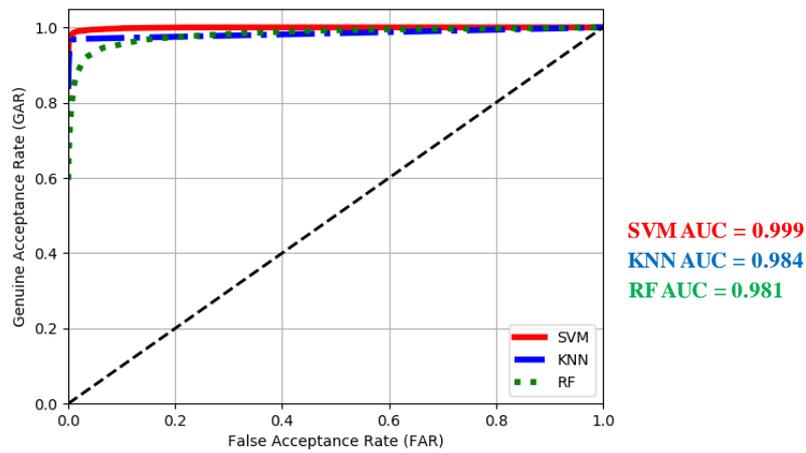


Figure 16. Receiver operating characteristic (ROC) curves of our proposed model for the contact-based HKPU database. AUC: area under the ROC curve.

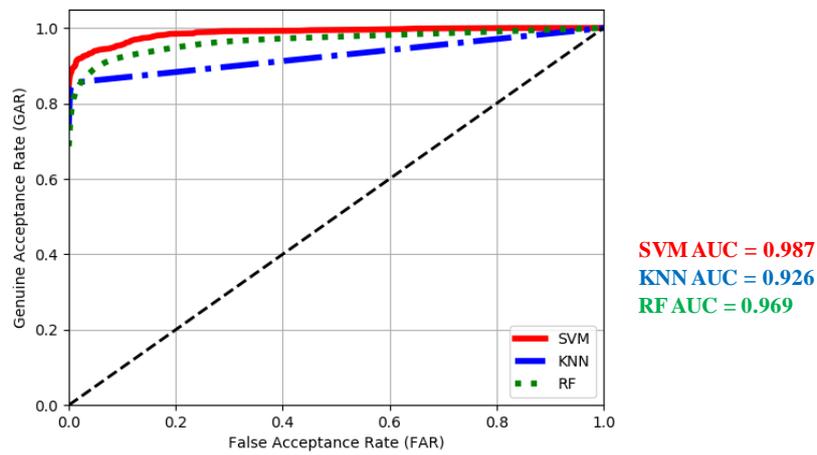


Figure 17. ROC curves of our proposed model for the 2D contact-free HKPU database.

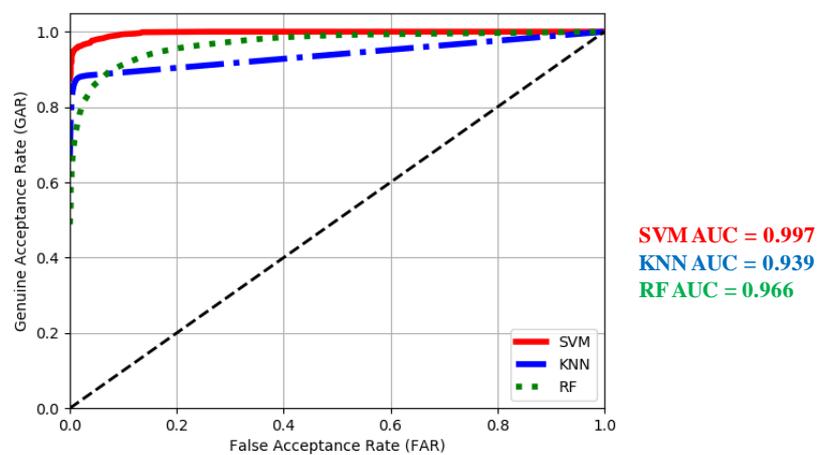


Figure 18. ROC curves of our proposed model for the IITD touchless database.

4.3.3. Equal Error Rate

The equal error rate (EER) corresponds to the error rate for which (1- GAR) and FAR are equal. To verify the effectiveness of our model, we compared it with the state-of-the-art hand-crafted methods in terms of EER as the performance indicator. These methods included the principal line (PriLine) [30], competitive code (CompCode) [31], double orientation code (DOC) [32], extended binary orientation co-occurrence vector (E-BOCV) [33], local line directional patterns-modified finite radon transform (LLDP-MFRAT) [34], and local line directional patterns-Gabor (LLDP-Gabor) [34] described in Section 2.2.

Table 4 presents EER results of the aforementioned methods and our proposed model in the verification task for the contact-based HKPU database. To avoid presenting a large table, we report only more recent state-of-the-art methods demonstrated to outperform old ones in our results. It should be pointed out that the verification performance of each algorithm was obtained by matching palmprints from different sessions. In fact, the images related to the first session were used as the training set and the images related to the second session were considered as the test set.

Table 4. Comparison with state-of-the-art methods in terms of EER value for the contact-based HKPU database. CompCode: competitive code; DOC: double orientation code; E-BOCV: extended binary orientation co-occurrence vector; PriLine: principal line.

Method	PriLine	CompCode	DOC	E-BOCV	Classifier in Our Proposed Model		
					SVM	KNN	RF
EER (%)	0.133	0.026	0.019	0.053	0.0125	0.0156	0.0625

We also performed verification experiments for the IITD touchless database. The EERs obtained using different methods are summarized in Table 5. In the experimental results, the training set included the first four images and the remaining images were utilized as the test set.

Table 5. Comparison with state-of-the-art methods in terms of EER value for the IITD touchless database. LLPD-MRFT: local line directional patterns-modified finite radon transform.

Method	CompCode	DOC	LLPD-MRFT	LLPD-Gabor	Classifier in Our Proposed Model		
					SVM	KNN	RF
EER (%)	0.101	0.062	0.031	0.028	0.0276	0.0328	0.0889

According to Tables 4 and 5, the results corroborate that using the linear SVM classifier in our matching module with the input of extracted deep features achieved the best performance in terms of the EER value among the state-of-the-art methods, which can be attributed to discriminative deeply learned features. In the other words, the CNN features extracted by our model were highly dominating over the only line features in PriLine and the orientation-based features in coding methods, so it can address the shortcomings of hand-crafted features. Moreover, as local feature-based methods, hand-crafted approaches generally demand more high-resolution palmprint images and have slower matching speed than our proposed model. Although LLPD-based methods have shown good performance as palmprint texture descriptors, they suffer from computation complexity and slower matching speed.

Furthermore, we can see from the comparative results that the overall EER rate for contact-based HKPU database was higher than that of the IITD touchless database because illumination, distortion, translation, and rotation variances in contact-less palmprint images led to less-significant features. Since our proposed model was tested on two palmprint image databases, we are fairly confident about the generalizability of our model.

5. Conclusions and Future Work

In this paper, we focused on palmprint verification using deep learning approaches, where we defined three main modules: (i) an ROI extraction module called REM, (ii) a feature extraction module called FEM, and (iii) a matching module called MM which is a machine learning classifier. To the best of our knowledge, this is the first study of extracting palmprint ROIs using CNN transfer learning. For the FEM, we utilized a pre-trained network to obtain a discriminative feature vector. Moreover, to find the best-adopted classification method with our proposed REM and FEM, we investigated different classification methods. Experimental results showed that the linear SVM classifier had the best performance. Additionally, the superiority of our proposed model over the state-of-the-art hand-crafted approaches demonstrates that deeply learned features are more discriminative than hand-crafted features.

In future work, we will collect a mobile palmprint database and apply our proposed model, in addition to evaluating new architectures like generative adversarial networks and capsule networks for online and reliable personal authentication.

Author Contributions: M.I. proposed the idea, conceptualization, formal analysis, calculated and plotted the feasible regions, and compared and classified the background and taxonomy presentation and evaluated their pros and cons. S.M.R., M.T-G. and S.H.Z. technically supported the content, verified the accuracy, writing-original draft preparation, and revised the clarity of the work as well as helping to write and organize the paper. Finally, A.U. helped in the proper preparations, English corrections, and submission.

Funding: Mahdiah Izadpanahkakhk was supported in part by Italian MIUR, GAUChO project, under Grant 2015YPXH4W 004.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Palma, D.; Montessoro, P.L.; Giordano, G.; Blanchini, F. Biometric Palmprint Verification: A Dynamical System Approach. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**. [[CrossRef](#)]
2. Ito, K.; Aoki, T. Recent Advances in Biometric Recognition. *ITE Trans. Media Technol. Appl.* **2018**, *6*, 64–80. [[CrossRef](#)]
3. Latha, Y.M.; Prasad, M.V. A Survey on Palmprint-Based Biometric Recognition System. In *Innovative Research in Attention Modeling and Computer Vision Applications*; IGI Global: Bangalore, India, 2015; p. 304.
4. Kong, A.; Zhang, D.; Kamel, M. A survey of palmprint recognition. *Pattern Recognit.* **2009**, *42*, 1408–1418. [[CrossRef](#)]
5. Meraoumia, A.; Laimeche, L.; Bendjenna, H.; Chitroub, S. Do we have to trust the deep learning methods for palmprints identification? In Proceedings of the Mediterranean Conference on Pattern Recognition and Artificial Intelligence, Tebessa, Algeria, 22–23 November 2016; pp. 85–91.
6. Xu, Y.; Fei, L.; Wen, J.; Zhang, D. Discriminative and robust competitive code for palmprint recognition. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 232–241. [[CrossRef](#)]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [[CrossRef](#)]
8. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
9. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
10. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
11. Bhanu, B.; Kumar, A. *Deep Learning for Biometrics*; Springer: Cham, Switzerland, 2017.
12. Guo, Y.; Liu, Y.; Oerlemans, A.; Lao, S.; Wu, S.; Lew, M.S. Deep learning for visual understanding: A review. *Neurocomputing* **2016**, *187*, 27–48. [[CrossRef](#)]

13. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [[CrossRef](#)]
14. Zhang, D.; Kong, W.K.; You, J.; Wong, M. Online palmprint identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1041–1050. [[CrossRef](#)]
15. Connie, T.; Jin, A.T.B.; Ong, M.G.K.; Ling, D.N.C. An automated palmprint recognition system. *Image Vis. Comput.* **2005**, *23*, 501–515. [[CrossRef](#)]
16. Han, Y.; Sun, Z.; Wang, F.; Tan, T. Palmprint recognition under unconstrained scenes. In Proceedings of the Asian Conference on Computer Vision, Tokyo, Japan, 18–22 November 2007; pp. 1–11.
17. Badrinath, G.; Gupta, P. Palmprint based recognition system using phase-difference information. *Future Gener. Comput. Syst.* **2012**, *28*, 287–305. [[CrossRef](#)]
18. Michael, G.K.O.; Connie, T.; Teoh, A.B.J. Touch-less palm print biometrics: Novel design and implementation. *Image Vis. Comput.* **2008**, *26*, 1551–1560. [[CrossRef](#)]
19. Michael, G.K.O.; Connie, T.; Jin, A.T.B. An innovative contactless palm print and knuckle print recognition system. *Pattern Recognit. Lett.* **2010**, *31*, 1708–1719. [[CrossRef](#)]
20. Choraś, M.; Kozik, R. Contactless palmprint and knuckle biometrics for mobile devices. *Pattern Anal. Appl.* **2012**, *15*, 73–85. [[CrossRef](#)]
21. Doublet, J.; Lepetit, O.; Revenu, M. Contact less hand recognition using shape and texture features. In Proceedings of the 2006 8th International Conference on Signal Processing, Beijing, China, 16–20 November 2006; Volume 3.
22. Aykut, M.; Ekinici, M. Developing a contactless palmprint authentication system by introducing a novel ROI extraction method. *Image Vis. Comput.* **2015**, *40*, 65–74. [[CrossRef](#)]
23. Aykut, M.; Ekinici, M. AAM-based palm segmentation in unrestricted backgrounds and various postures for palmprint recognition. *Pattern Recognit. Lett.* **2013**, *34*, 955–962. [[CrossRef](#)]
24. Ito, K.; Sato, T.; Aoyama, S.; Sakai, S.; Yusa, S.; Aoki, T. Palm region extraction for contactless palmprint recognition. In Proceedings of the 2015 International Conference on Biometrics (ICB), Phuket, Thailand, 19–22 May 2015; pp. 334–340.
25. Harun, N.; Rahman, W.E.Z.W.A.; Abidin, S.Z.Z.; Othman, P.J. New algorithm of extraction of palmprint region of interest (ROI). *J. Phys.* **2017**, *890*, 012024. [[CrossRef](#)]
26. Zhang, D.; Zuo, W.; Yue, F. A comparative study of palmprint recognition algorithms. *ACM Comput. Surv.* **2012**, *44*, 2. [[CrossRef](#)]
27. Jia, W.; Hu, R.X.; Gui, J.; Zhao, Y.; Ren, X.M. Palmprint recognition across different devices. *Sensors* **2012**, *12*, 7938–7964. [[CrossRef](#)] [[PubMed](#)]
28. Raghavendra, R.; Busch, C. Texture based features for robust palmprint recognition: A comparative study. *EURASIP J. Inf. Secur.* **2015**, *2015*, 5. [[CrossRef](#)]
29. Fei, L.; Lu, G.; Jia, W.; Teng, S.; Zhang, D. Feature Extraction Methods for Palmprint Recognition: A Survey and Evaluation. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**. [[CrossRef](#)]
30. Malik, J.; Girdhar, D.; Dahiya, R.; Sainarayanan, G. Accuracy improvement in palmprint authentication system. *Int. J. Image Graph. Signal Process.* **2015**, *7*, 51. [[CrossRef](#)]
31. Kong, A.K.; Zhang, D. Competitive coding scheme for palmprint verification. *Pattern Recognit.* **2004**, *1*, 520–523.
32. Fei, L.; Xu, Y.; Tang, W.; Zhang, D. Double-orientation code and nonlinear matching scheme for palmprint recognition. *Pattern Recognit.* **2016**, *49*, 89–101. [[CrossRef](#)]
33. Zhang, L.; Li, H.; Niu, J. Fragile bits in palmprint recognition. *IEEE Signal Process. Lett.* **2012**, *19*, 663–666. [[CrossRef](#)]
34. Luo, Y.T.; Zhao, L.Y.; Zhang, B.; Jia, W.; Xue, F.; Lu, J.T.; Zhu, Y.H.; Xu, B.Q. Local line directional pattern for palmprint recognition. *Pattern Recognit.* **2016**, *50*, 26–44. [[CrossRef](#)]
35. Dai, J.; Feng, J.; Zhou, J. Robust and efficient ridge-based palmprint matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1618–1632. [[PubMed](#)]
36. Roux, V.; Aoyama, S.; Ito, K.; Aoki, T. Performance improvement of phase-based correspondence matching for palmprint recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014; pp. 70–77.
37. Liu, E.; Jain, A.K.; Tian, J. A coarse to fine minutiae-based latent palmprint matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2307–2322. [[PubMed](#)]

38. Chen, F.; Huang, X.; Zhou, J. Hierarchical minutiae matching for fingerprint and palmprint identification. *IEEE Trans. Image Process.* **2013**, *22*, 4964–4971. [[CrossRef](#)] [[PubMed](#)]
39. Morales, A.; Ferrer, M.A.; Kumar, A. Towards contactless palmprint authentication. *IET Comput. Vis.* **2011**, *5*, 407–416. [[CrossRef](#)]
40. Kumar, A.; Shekhar, S. Personal identification using multibiometrics rank-level fusion. *IEEE Trans. Syst. Man Cybern. Part C* **2011**, *41*, 743–752. [[CrossRef](#)]
41. Li, W.; Zhang, L.; Zhang, D.; Lu, G.; Yan, J. Efficient joint 2D and 3D palmprint matching with alignment refinement. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 795–801.
42. Zhang, D.; Guo, Z.; Gong, Y. An online system of multispectral palmprint verification. In *Multispectral Biometrics*; Springer: Berlin, Germany, 2016; pp. 117–137.
43. Fei, L.; Xu, Y.; Zhang, D. Half-orientation extraction of palmprint features. *Pattern Recognit. Lett.* **2016**, *69*, 35–41. [[CrossRef](#)]
44. Xu, Y.; Zhang, D.; Yang, J.; Yang, J.Y. A two-phase test sample sparse representation method for use with face recognition. *IEEE Trans. Circ. Syst. Video Technol.* **2011**, *21*, 1255–1262.
45. Gui, J.; Jia, W.; Zhu, L.; Wang, S.L.; Huang, D.S. Locality preserving discriminant projections for face and palmprint recognition. *Neurocomputing* **2010**, *73*, 2696–2707. [[CrossRef](#)]
46. Jia, W.; Hu, R.X.; Lei, Y.K.; Zhao, Y.; Gui, J. Histogram of oriented lines for palmprint recognition. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 385–395. [[CrossRef](#)]
47. Xu, Y.; Fei, L.; Zhang, D. Combining left and right palmprint images for more accurate personal identification. *IEEE Trans. Image Process.* **2015**, *24*, 549–559. [[PubMed](#)]
48. Raghavendra, R.; Busch, C. Novel image fusion scheme based on dependency measure for robust multispectral palmprint recognition. *Pattern Recognit.* **2014**, *47*, 2205–2221. [[CrossRef](#)]
49. Dai, J.; Zhou, J. Multifeature-based high-resolution palmprint recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 945–957. [[PubMed](#)]
50. Lu, J.; Hu, J.; Tan, Y.P. Discriminative deep metric learning for face and kinship verification. *IEEE Trans. Image Process.* **2017**, *26*, 4269–4282. [[CrossRef](#)] [[PubMed](#)]
51. Sun, Y.; Wang, X.; Tang, X. Hybrid deep learning for face verification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 1489–1496. [[CrossRef](#)] [[PubMed](#)]
52. Kumar, A.; Wang, K. *Identifying Humans by Matching Their Left Palmprint with Right Palmprint Images Using Convolutional Neural Network*; DLPR Proceedings; The Hong Kong Polytechnic University: Hong Kong, 2016; pp. 1–6.
53. Dian, L.; Dongmei, S. Contactless palmprint recognition based on convolutional neural network. In Proceedings of the 2016 IEEE 13th International Conference on Signal Processing (ICSP), Chengdu, China, 6–10 November 2016; pp. 1363–1367.
54. Svoboda, J.; Masci, J.; Bronstein, M.M. Palmprint recognition via discriminative index learning. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 4232–4237.
55. Minaee, S.; Wang, Y. Palmprint recognition using deep scattering convolutional network. *arXiv* **2016**, arXiv:1603.09027.
56. Jalali, A.; Mallipeddi, R.; Lee, M. Deformation invariant and contactless palmprint recognition using convolutional neural network. In Proceedings of the 3rd International Conference on Human-Agent Interaction, Kyungpook, Korea, 21–24 October 2015; pp. 209–212.
57. Chan, T.H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A simple deep learning baseline for image classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [[CrossRef](#)] [[PubMed](#)]
58. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the devil in the details: Delving deep into convolutional nets. *arXiv* **2014**, arXiv:1405.3531.
59. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
60. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
61. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [[CrossRef](#)]

62. 2D Contact-free HKPU. Available online: http://www4.comp.polyu.edu.hk/~csajaykr/myhome/database_request/3dhand/Hand3D.htm (accessed on 13 May 2018).
63. IITD Touchless Palmprint Database. Available online: http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm (accessed on 13 May 2018).
64. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. *OSDI* **2016**, *16*, 265–283.
65. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).