IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 11, 2021, accepted October 20, 2021, date of publication October 26, 2021, date of current version November 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123291

PalmHashNet: Palmprint Hashing Network for Indexing Large Databases to Boost Identification

GEETIKA ARORA[1], SUMIT KALRA[2], ASHUTOSH BHATIA[1], AND KAMLESH TIWARI[1], (Member, IEEE)
[1]Department of CSIS, Birla Institute of Technology and Science Pilani, Pilani Campus, Jhunjhunu, Rajasthan 333031, India
[2]Department of CSE, Indian Institute of Technology Jodhpur, Karwar, Rajasthan 342037, India

Corresponding author: Geetika Arora (p2016406@pilani.bits-pilani.ac.in)

ABSTRACT Palmprint identification aims to establish the identity of a given query sample by comparing it with all the templates in the database and locating the most similar one. It becomes computationally expensive as the size of the database grows. It is because the number of comparisons becomes proportional to the number of templates stored in the database. The process needs to be accelerated to get a response in real-time, especially for large databases. This paper proposes a palmprint database indexing approach called PalmHashNet that generates highly discriminative embeddings to create a fixed-size candidate list for comparison to make identification a constant time operation. Acquired palmprint images are fed to the feature extraction network, which is pre-trained using softmax loss. A margin is added to the softmax loss to minimize the intra-class distance between samples belonging to the same class. It ensures that the features have high intra-class and low inter-class similarity. k-means and locality sensitive hashing (LSH) is investigated for index table creation. In this setting, cluster centers for k-means and hash values in the case of LSH serve as indices. The features are extracted for a given query palmprint and compared with the index values. The candidates lying in the most similar bin are retrieved for identification. The advantage of the proposed approach is that the query palmprint is compared with a small percentage of database instead of the whole. The proposed approach offers probabilistic guarantees for query identification in the selected bin. Experiments are conducted on four widely used palmprint databases *viz.* CASIA, IITD-Touchless, Tongji-Contactless and Hong Kong Polytechnic University Palmprint II (PolyU II). A penetration rate of 0.022%, 1.032%, 4.555%, and 0.39% at 100% hit rate is achieved on these databases, respectively. It makes the identification process approximately 4500, 96, 21, and 256 times faster on the respective databases.

INDEX TERMS Biometrics, palmprint, identification, hashing, indexing, retrieval.

I. INTRODUCTION

Many real-world applications call for the need for access control and security. Contrary to traditional modes of authentication such as PIN, passwords, tokens that can be stolen or forged, biometrics provide a mechanism to authenticate an individual by analyzing their physical (fingerprint, palmprint, iris, face *etc.*) or behavioral (gait, voice, signature *etc.*) traits. Recently, palmprint-based biometric has gained wide popularity due to its non-intrusive nature, easy acquisition, and robust textural features. Palmprint is an impression acquired from the inner part of the hand lying between wrist and fingers. A region of interest (RoI) of palmprint is extracted from the acquired hand images of the individuals. It consists of complex and unique patterns that are utilized as features for human authentication. The features are classified as high resolution and low resolution based on the quality of the acquired image. The low-resolution features include wrinkles, texture, and principal lines, which can be further sub-classified as heart, head, and life line. These are visible to the naked eye as well. On the other hand, ridges, singular points, and minutiae points can only be extracted from high-resolution images [1]. These features are shown in FIGURE 1 (b) and (c)
respectively.

A biometric authentication system involves two stages *viz.* enrolment and recognition. Enrolment, also known as registration, is a phase wherein the palmprint samples are acquired from the population and stored in a database with a unique identifier. Palmprint recognition refers to the process of

The associate editor coordinating the review of this manuscript and approving it for publication was Weizhi Meng.

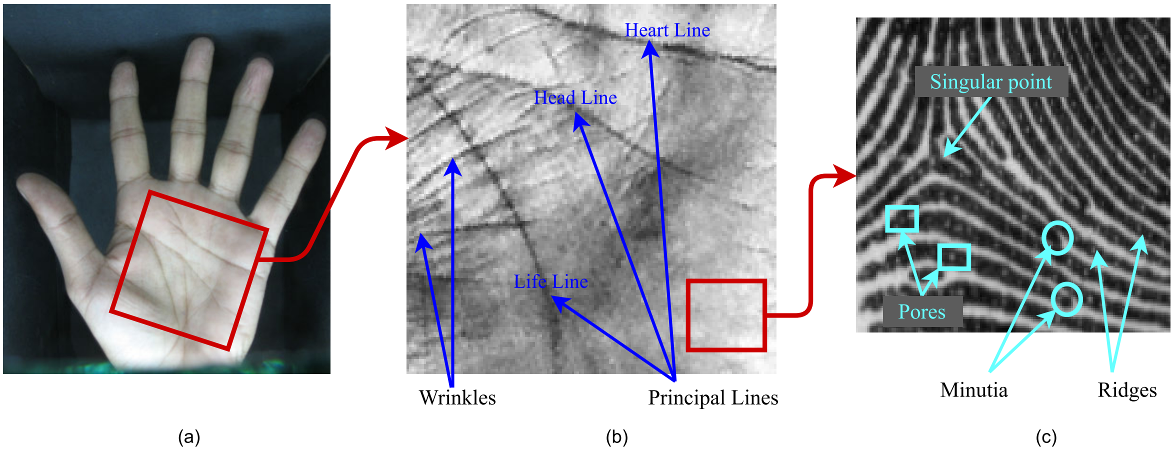


FIGURE 1. Figure showing (a) RoI extraction, marked by the red box, (b) low resolution and (c) high resolution image features respectively.

authenticating a person by utilizing their palmprint sample and comparing it with the palmprint template(s) stored in the database. It is further classified into verification and identification. During palmprint verification, a palmprint sample along with its claimed identity is given to the system. A comparison of the input image and the template stored in the database against the claimed identity is made and the system outputs are either matched or not matched. On the other hand, during identification, the identity of the sample is unknown and needs to be established. To do so, the given sample is compared with all the templates in the database and the most similar one is retrieved as its true match. To sum it up, verification involves only one comparison, whereas the identification involves N comparisons, where N is the number of templates in the database.

With digitization and increase in deployment of biometric authentication systems such as Aadhar (India) [2] and MyKad (Malaysia) [3], there has been an increase in the size of biometric databases. With an increase in the number of enrolments, identification has become a less efficient and computationally expensive task as the number of required comparisons is proportional to the size of the database. The problem can be addressed if the search space can be narrowed down while still outputting the true match of the given sample [4]. This paper addresses the problem of accelerating human identification in large palmprint databases. Research suggests the use of a classification-based approach that splits a biometric database into different categories. When a query sample is provided to the identification system, then its category is determined first. Then all the samples belonging to the category are retrieved as a candidate set for the comparison with the query. An approach discussed in [5] utilizes gender to narrow down the search space of the face database. However, the division of the database is primarily skewed and non-uniform across all the databases. Therefore, it will not always ensure an efficient reduction in the search space. Hence, there is a need to devise a technique to generate a suitable list of candidates for comparison with query palmprint. The process of finding top-t candidates for comparison with the probe image to accelerate the identification process is termed as *Indexing of biometric databases*. Indexing eliminates the need for linear search in the database for the identification of biometric query samples. Instead, the index table returns only a subset of the database that tends to be the most similar to the query image. The advantage of indexing a biometric database in the process of identification is pictorially depicted in FIGURE 2. There are three components of Indexing *viz.* feature extraction, index table generation and retrieval. In the first stage, features are learned for each palmprint image using the proposed network. The learned feature vectors have salient and highly discriminative characteristics of the image. In the second stage, the extracted features that are similar to each other are grouped and are associated with an index. Since there exists high intra-class and low inter-class similarity among the features, the similar feature vectors end up going in the same bin of the index table [6]. The fundamental idea is to find a suitable feature vector and generate an index that could remarkably reduce the number of comparisons. The next stage, retrieval, aims to extract the feature vector for the probe image and find out the most similar index/bin. The candidates lying in the selected bin are fetched out for comparison with the probe sample.

However, there are certain challenges that need to be accounted for while designing an indexing technique for a biometric database [4]. 1) Variable number of features: It is not necessary that the features extracted from a palmprint image at two different time stamps would be same. It may happen that some features could be missing while some false features could also appear. Hence, the features may vary in number. 2) No order: There is no pre-defined order among the biometric features unlike some structural data. Therefore, they may appear in different order which becomes a challenge. 3) Occlusion and illumination: Presence of occlusion, rotation and translation may also affect the feature extraction process. The features may vary in number because of presence of aforementioned variations. 4) Transformation:

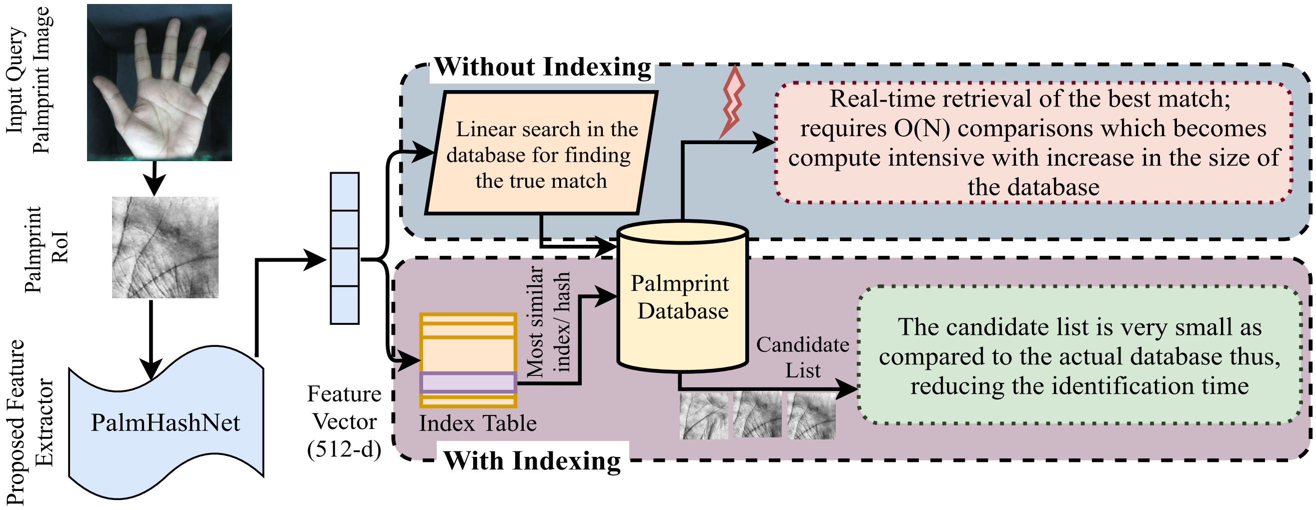

Input Query
Palmprint Image
Palmprint
RoI
Proposed Feature
Extractor
PalmHashNet
Feature
Vector
(512-d)
Without Indexing
Linear search in the database for finding the true match
Real-time retrieval of the best match; requires O(N) comparisons which becomes compute intensive with increase in the size of the database
Palmprint
Database
Most similar
index/ hash
Index Table
Candidate
List
The candidate list is very small as compared to the actual database thus, reducing the identification time
With Indexing

FIGURE 2. Block diagram depicting the advantage of indexing palmprint database in identification.

The acquired image may be rotated, zoomed-in, zoomed-out *etc*. thus, transforming the extracted features. It is therefore, expected of any indexing technique to address all the aforementioned challenges.

The performance of any biometric indexing technique is determined by the quality of extracted features. The features are expected to have low inter-class and high intra-class similarity. After traditional handcrafted approaches, many CNN based methods have been proposed that automatically learn indexing-suitable features for palmprint databases. The feature extraction process is carried out by training the network for a classification problem using softmax cross entropy loss [7]. The features are extracted from the layer connected just before the fully connected layer. It is believed that the learned feature vectors are good if they are able to classify the input image correctly. But, the learned features may not turn out to be optimal if no explicit constraint is applied on feature vector distribution which may lead to a general spread of the learned feature vectors. To address this, metric-based methods have been introduced that uses distance-based criterion to separate feature embeddings from different classes and bring them closer otherwise. It proposes a novel metric-based palmprint feature extraction network that uses a function called 'additive margin loss' [8] to supervise the training process. The softmax loss is capable of maximizing the inter-class distance among samples of different classes but unable to minimize the intra-class dissimilarity among the samples of the same class. Therefore, a margin is added to the loss function to handle the intra-class variation better. The extracted features are indexed using k-means clustering and Locality Sensitive Hashing to boost identification process. A block diagram of the proposed approach is shown in FIGURE 3. Key contributions of the paper are listed below.

Contributions:

1) This paper proposes PalmHashNet, a novel indexing technique that learns compact feature vectors for palmprint images to facilitate faster identification. The proposed approach learns features suitable for indexing that ensure fast and accurate retrieval during identification.
2) Softmax loss with additive margin has been used to train the model for palmprint database indexing and to learn the feature vector embeddings simultaneously. This loss function ensures that the learned feature embeddings have low inter-class along with high intra-class similarity. It ascertains that the index space distribution is regularized to be similar to the uniform distribution. The network outputs a 512-dimensional distinct compact feature embedding corresponding to every palmprint sample and is associated with an index in the index table.
3) Two different techniques *viz*. k-means clustering and locality sensitive hashing (LSH) with k-nearest neighbor search have been explored for index table creation. The generated index is used for the retrieval of top-t matches for identification. The indexing performance of both the considered techniques has been compared.
4) The proposed approach has been evaluated on four publicly available databases *viz*., CASIA [9], IIT Delhi Touchless [10], Tongji Contactless [11] and PolyU II [12] Palmprint Database. All the databases contain palmprint images collected in an unconstrained environment. The results show the efficiency of the learned features in the identification process.

Rest of the paper is organised as follows. Next section discusses related work done in the field of palmprint recognition and palmprint database indexing. Section III describes the proposed approach, *i.e* feature extraction, indexing and retrieval, in detail. Experimental setting and results have been analyzed in the Section IV followed by the conclusion in Section V.

II. RELATED WORK

This section discusses the related work that has been done in the area of palmprint recognition and palmprint database

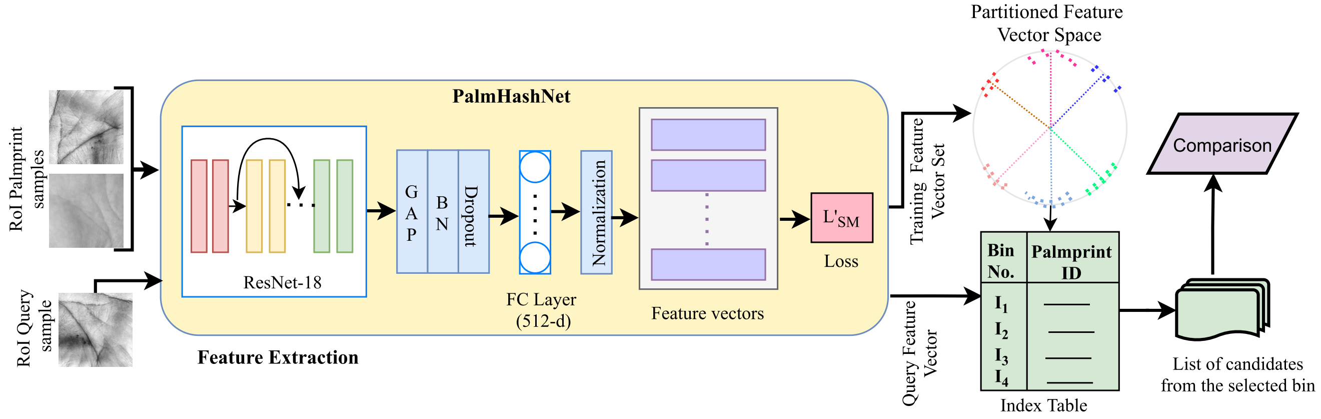


FIGURE 3. **Block diagram of the proposed approach. There are three phases; PalmHashNet is the proposed feature extraction network which is trained using L'_{SM}. The learned feature vectors are given to indexing module for index table creation. The feature of query palmprint sample is extracted using the trained PalmHashNet which is then compared with all the bin numbers and the most similar one is selected for candidate set generation.**

indexing. The quality of the learned feature vectors plays a major role in the effectiveness of the system. This paper compares the recognition performance on the palmprint databases with other state-of-the-art approaches in TABLE 4 to justify the goodness of the learned features through PalmHashNet. Therefore, this section gives an overview of the palmprint recognition and indexing approaches proposed in the literature.

A. PALMPRINT RECOGNITION

Palmprint recognition started with Boles and Chu [13], wherein the authors proposed that palm features such as its shape and lines could be used for human authentication. Lu *et al.* [14] claimed that using principal lines only for matching palmprint samples is not a good idea as some people may have similar patterns of principal lines. Therefore, they utilized a circular Gabor filter to extract features from low-resolution palmprint images. Kong and Zhang [15] proposed a palmprint verification system based on orientation information in palmprint lines. The paper proposes a competitive code by extracting orientation information using 2-D Gabor filters. Angular matching was used to compare the generated codes. An improvised version of competitive code called robust line orientation code (RLOC) was proposed by Jia *et al.* [16]. Feature extraction has been performed using a modified version of the radon transform. The features have been matched using pixel-to-area comparison. Zuo *et al.* [17] extended competitive code and proposed multi-scale orientation palmprint feature extraction called sparse multi-scale competitive code (SMCC). The proposed method is robust to illumination and scale variation. A palmprint verification method combining dominant orientation code and side code (DRCC) has been proposed in [18]. The proposed technique extracted both the codes by applying weights on the Gabor filter responses to improve the results further. A local binary pattern-based feature descriptor (LLDP) to extract local features from palmprint images has been proposed in [19]. Li and Kim [20] proposed a local feature descriptor that takes into account the direction and thickness information, making it robust to translation and rotation. Zhong *et al.* [21] proposed a siamese network utilizing two weights sharing VGG-16 networks to learn discriminative features for palmprint images. A histogram feature descriptor has been proposed in [22]. The paper suggests the use of apparent direction and latent direction computed from the energy map of the apparent direction. These directions are combined to form a single feature descriptor for palmprint images. Zhao and Zhang [23] proposed a CNN-based local feature extraction network in which a palmprint image is divided into five parts, and these parts, along with the complete palmprint image, are given to the proposed network. Zhong and Zhu [24] proposed a palmprint recognition system by combining large margin cosine loss and center loss.

B. PALMPRINT INDEXING:

The first technique for palmprint identification was proposed by You *et al.* [25], where hierarchy of four-level features has been used. The paper uses different matching strategies at different stages while reducing the search space. Li *et al.* [26] proposed a palmprint retrieval technique by using texture features. The searching was performed in two stages; firstly, global features were used to find out a small-sized candidate list and then local features were used to output the final result from the selected candidates. Paliwal *et al.* [27] made use of Vector Approximation (VA+) file database to generate score-based indexing scheme. A ridge features-based indexing technique was proposed by Yang *et al.* [28]. The paper first aligns all the palmprint images in a unified coordinate system and then uses ridge density and orientation information for indexing. Chen *et al.* [29] proposed a technique that outputs a binary feature vector and applied spectral hashing technique to index the feature embeddings. Yue *et al.* [30] proposed two techniques that used different features for hashtable creation. The first one is based on orientation pattern

(OP), which refers to orientation features. While the other one uses principal orientation patterns (POP) *i.e.*, orientation patterns that lie in the region of principal lines. An accelerated and improvised indexing technique that uses features generated from POP was proposed in [31]. A convolutional neural network (CNN) based feature extractor was proposed in [32]. The proposed network outputs a 128-d feature vector, and later, supervised hashing was implemented. A method using the difference of block means has been proposed by Almagtuf *et al.* [33]. In this method, no feature extraction has been performed. Rather, simple operations such as adding and subtracting overlapping blocks are used to compute palmprint code in each direction. Chen *et al.* [34] proposed a double-orientation feature to account for unstable orientation fields. It further used a window-based feature measurement for faster retrieval. A distillation-based loss function has been proposed in [35] that generates binary feature vectors for palmprint images. Zhu *et al.* [36] proposed an adversarial metric learning technique to make the palmprint embeddings uniformly distributed over a hypersphere. This is done by utilizing distance metrics and confusion terms. The paper also introduced a new palmprint database that is collected in an unconstrained environment. Zhao and Zhang [37] proposed a deep convolutional neural network-based technique that extracts discriminative features from the whole palmprint image and its patches separately and combines them to make compact feature. Jia *et al.* [12] evaluated the performance of various hashing techniques for retrieval of palmprint images. The paper considered four supervised, unsupervised and deep hashing methods each and PolyU II, PolyU M_B, HFUT, TJU, and PolyU 3D databases for evaluation. It has been reported that column sampling-based discrete supervised hashing (COSDISH) [38] performs best among other considered hashing techniques. Recently, an end-to-end CNN-based network that learns binary hash values for palmprint images has been proposed in [39]. It uses structural and pixel-level features by adding a similarity measurement module after the last fully connected layer.

III. PROPOSED APPROACH

This section provides details of the proposed approach for indexing a palmprint database. There are three components in this approach *viz.* feature extraction, indexing of the extracted features, and lastly, retrieval of the suitable candidate list for matching with the probe sample. The acquired image from the palm sensor may contain unnecessary backgrounds such as part of the hand other than palm or background clutter. Therefore, the region of interest (RoI) is segregated from the acquired images before feature extraction. This region should contain only the palm region, which lies between the ends of the fingers and the wrist, as shown in FIGURE 1(a). Various techniques have been proposed in literature that aims at extracting palmprint RoI [40]–[51]. The most popular RoI extraction technique has been proposed by Zhang *et al.* [43]. It starts by binarizing the input image to obtain a boundary of the hand that can separate it from the background. The next step is to form a unified coordinate system. A boundary is drawn between the gaps between the little and ring finger and the middle and index finger. The boundary of the gap between the ring and middle finger is not obtained as it is of no use. A tangent is drawn connecting the two boundaries, and two coordinates (x_1, y_1) and (x_2, y_2) are selected that lie on the respective boundaries and the tangent. These two points are joined to get the y-axis of the coordinate system. Origin of the coordinate system is obtained at the intersection of the y-axis and the line perpendicular to the y-axis that passes through the mid-point of the image. A sub-image is extracted from this image for feature extraction.

For feature extraction, we propose PalmHashNet, a metric-based network that learns discriminative feature vectors for input palmprint images. These feature vectors are later indexed using k-means clustering and Locality Sensitive Hashing for index table creation. During retrieval, when a query palmprint needs to be identified, it is passed to the trained PalmHashNet that extracts a feature vector corresponding to the query image. After that, the query feature vector is compared to all the indices (cluster centers or hash values) to find the most similar bin. All the candidates lying in that bin are retrieved for comparison. Hence, instead of matching the query palmprint image with all the samples in the database, it is only matched with the retrieved candidates making it an operation with O(1) time complexity. All the stages of the proposed approach are described in the following subsections.

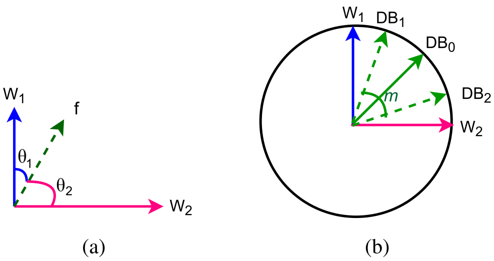
A. FEATURE EXTRACTION

First stage in the identification process is the feature extraction. In this stage salient and discriminative features are extracted from palmprint images. The performance of any biometrics indexing technique is determined by the quality of extracted features. The features are expected to have low inter-class and high intra-class similarity. In other words, the features belonging to same class should be closer to each other than the features belonging to different classes in the feature embedding space. This condition is required to improve accuracy using nearest neighbor. This would hence make the indexing and further, the identification process efficient.

Generally, feature extraction process is carried out by training a classification network using softmax loss [7]. Softmax loss is defined as an amalgamation of softmax function, cross-entropy loss and the last layer of a convolutional neural network [52]. Different subjects or individuals are treated as different classes and the layer connected just before the fully connected layer, serve as the feature embedding layer. Mathematically, softmax loss L_{SM} is defined as:

$$L_{SM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{a_{y_i}}}{\sum_{j=1}^{C}e^{a_j}} \tag{1}$$

where N and C are the total number of samples and number of classes respectively. Activation of j^{th} neuron in the last



(a) (b)

FIGURE 4. **Geometrical Representation of (a) softmax loss and (b) additive margin loss. DB_0 is the decision boundary created by the softmax loss whereas DB_1 and DB_2 are the decision boundaries learned by additive margin loss for class C_1 and C_2 respectively. The boundary becomes a regional margin instead of single vector when additive margin is applied.**

fully connected layer having weight W_j and bias b_j for the i^{th} palmprint sample with feature f_i is given as $a_j = W_j^T * f_i + b_j$. There would be C activations, one corresponding to each class. Let the ground truth for the i^{th} palmprint sample be the class y_i where $i \in \{1, 2, \ldots, C\}$, then the activation of the corresponding neuron can be given as $a_{y_i} = W_{y_i}^T * f_i + b_{y_i}$. Using this, the Eq. 1 can be written as.

$$L_{SM} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{W_{y_i}^T * f_i + b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^T * f_i + b_j}} \tag{2}$$

Considering a binary classifier, the posterior probabilities of a palmprint having the feature vector f_i belonging to the class C_1 or C_2 can be obtained by using the softmax as shown in Eq. 3 and Eq. 4 respectively.

$$p(C_1) = \frac{e^{W_1^T * f_i + b_1}}{e^{W_1^T * f_i + b_1} + e^{W_2^T * f_i + b_2}} \tag{3}$$

$$p(C_2) = \frac{e^{W_2^T * f_i + b_2}}{e^{W_1^T * f_i + b_1} + e^{W_2^T * f_i + b_2}} \tag{4}$$

where (W_1^T, b_1) and (W_2^T, b_2) are the weight and bias corresponding to the class C_1 and C_2. The classifier outputs C_1 as the class of the query palmprint if $p(C_1) > p(C_2)$ and the output is C_2 otherwise. The classification solely depends upon the weight and bias term and uses $W_j^T * f_i + b_j$ for the decision. Element wise multiplication in $W_j^T * f_i$ is equivalent to the dot product therefore the activation can be re-written as $a_j = ||W_j^T||\ ||f_i||\ \cos\theta_j + b_j$, where θ_j is the angle between vectors W_j and f_i. It can be observed that the activation depends upon angle θ_j and the weight vector norm W_j both. If we normalized the weights to unity, the classification would become directly dependent on the angle between f_i and W_j's. Therefore, weight vectors and feature vectors are normalized as shown in Eq 5. Here, f_i and W_j are original feature and weight vector respectively. The values of $||f_i||$ and $||W_j||$ are set to unity using Eq. 5.

$$f_i = \frac{f_i^\star}{\|f_i^\star\|} \quad \text{and} \quad W_j = \frac{W_j^\star}{\left\|W_j^\star\right\|} \tag{5}$$

TABLE 1. **Architecture of the proposed feature extraction network (PalmHashNet) with respect to PolyU II database images as input.**

Layer	# Filters	Output Shape	# of Param
Input	-	128 × 128 × 3	0
Conv2D - BN - PRelu	64	128 × 128 × 64	705
Maxpool 2D - BN	-	64 × 64 × 64	128
(Conv2D - BN - PRelu)*2	64	64 × 64 × 64	73986
Residual Block -BN	-	64 × 64 × 64	128
(Conv2D - BN - PRelu)*2	64	64 × 64 × 64	73986
Residual Block -BN	-	64 × 64 × 64	128
Conv2D - BN - PRelu	64	64 × 64 × 64	36993
Conv2D - BN	128	32 × 32 × 128	73984
Conv2D - BN - PRelu	128	32 × 32 × 128	8349
Residual Block -BN	-	32 × 32 × 128	256
(Conv2D - BN - PRelu)*2	128	32 × 32 × 128	295426
Residual Block -BN	-	32 × 32 × 128	256
Conv2D - BN - PRelu	128	32 × 32 × 128	147713
Conv2D - BN	256	16 × 16 × 256	295424
Conv2D - BN - PRelu	256	16 × 16 × 256	33281
Residual Block -BN	-	16 × 16 × 256	512
(Conv2D - BN - PRelu)*2	256	16 × 16 × 256	1180674
Residual Block -BN	-	16 × 16 × 256	512
Conv2D - BN - PRelu	256	16 × 16 × 256	590337
Conv2D - BN	512	8 × 8 × 512	1180672
Conv2D - BN - PRelu	512	8 × 8 × 512	132097
Residual Block -BN	-	8 × 8 × 512	1024
(Conv2D - BN - PRelu)*2	512	8 × 8 × 512	4720642
Residual Block -BN	-	8 × 8 × 512	1024
Dropout	-	8 × 8 × 512	0
MaxPool 2D	-	4 × 4 × 512	0
GAP - BN	-	512 × 1	0
Dropout - BN	-	512 × 1	0
Fully Connected	-	512 × 1	0

After normalization, the posterior probabilities given in Eq 3 and Eq 4 can be equivalently changed to $p(C_j) = \cos\theta_j$. Therefore, the decision boundary becomes $\cos\theta_1 - \cos\theta_2 = 0$. Normalized features can be plotted on a hypersphere manifold with fixed radius (say 'r') as in shown in FIGURE 4. Softmax loss in the Eq. 2 can now be represented as:

$$L_{SM} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{r\ \cos\theta_{y_i}}}{e^{r\ \cos\theta_{y_i}} + \sum_{j=1, j\neq y_i}^{C} e^{r\ \cos\theta_j}} \tag{6}$$

While training, if the sample belongs to class C_1 then the angle between f_i and W_1 is smaller than the angle between f_i and W_2. However, this kind of decision boundary works for the classification but, it does not enforce higher intra-class similarity as the compact localization of the features of same class is not mandatory. Therefore features obtained from different samples of the same class covering usual variation are scattered around feature space. This is more visible phenomenon if samples contain high intra-class variations such as occlusion, pose, illumination *etc*.

In order to make the decision boundary more stringent, a margin m is added to θ. Consider a sample belonging to class C_1. This would imply that $\cos\theta_1 > \cos\theta_2$. By adding a margin m in the θ_1, the equation that needs to be satisfied becomes, $\cos(\theta_1 - m) > \cos\theta_2$. The expression $\cos(\theta_1 - m)$ is larger than $\cos\theta_1$ which in turn is greater than $cos\theta_2$. The same relationship exists between θ_1 and θ_2. The decision boundary for class C_1 then becomes $cos(\theta_1 - m) = cos(\theta_2)$.

Algorithm 1 Feature Extraction and Indexing

```
Input: Set of palmprint images ($P = P_1, P_2, \ldots, P_N$)
Output: Index table $I$
for each palmprint sample $P_i$ do
    Extract feature $f_{P_i}$ using the trained PalmHashNet.
    Append $f_{P_i}$ to feature set $f_P$ i.e. $f_P \leftarrow f_{P_i}$
end for
Apply k-means clustering or Locality Sensitive Hashing on the feature vector set $f_P$.
for each cluster center $cc_i$ (k-means) or hash value $h_i$ (LSH) do
    Create an entry in index table $I$ with $cc_i$ or $h_i$.
    Put ID and feature vectors of all the candidates in $I$ lying in $cc_i$ or $h_i$.
end for
return I
```

Algorithm 2 Retrieval

```
Input: Index table $I$
Output: Candidate Set $X$
for each query palmprint sample $q_j$ do
    Extract feature $f_{q_j}$ using the trained PalmHashNet.
    for each index $i$ in $I$ do
        Compute cosine similarity between hash value ($h_i$) and $f_{q_j}$.
        Create a table $S$ with $h_i$ and cosine similarity.
    end for
    Find the maximum score value.
    Retrieve IDs stored in the most similar bin to get candidate list $X$ for matching.
end for
return X
```

Similarly, to correctly classify the another feature belonging to class C_2, it is required that $cos(\theta_2 - m) > cos(\theta_1)$ and the decision boundary becomes $cos(\theta_2 - m) = cos(\theta_1)$. Due to the margin, lower bound of $cos\theta_1$ becomes much greater than $cos\theta_2$ thereby, enforcing higher intra-class compactness. The modified softmax function is written as below.

$$L'_{SM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{r\,(\cos\theta_{y_i} - m)}}{e^{r\,(\cos\theta_{y_i} - m)} + \sum_{j=1\,,j\neq y_i}^{C} e^{r(\cos\theta_j)}} \tag{7}$$

The geometrical representation of additive margin loss is shown in FIGURE 4. It shows that the initial decision boundary *i.e.* the one created by softmax is now changed to DB_1 and DB_2 respectively for class C_1 and C_2 respectively. Therefore, we can conclude from the figure that intra-class difference among the samples of same class is minimized by adding this extra marginal region to the angle.

PalmHashNet: This paper proposes a deep convolutional network named PalmHashNet that extracts features from palmprint images using the modified softmax loss function, given in Eq. 7. It has been seen that deeper networks result in loss of information, resulting in stagnation of accuracy, and eventually, it decreases. To address this problem, consider a shallow and a deep network. The deep network consists of a shallow network plus some additional layers that act as an identity function. The deeper network would act like the shallower counterpart in the worst-case scenario when this is done. However, it may happen that the deep network would learn the better features and reduce the error significantly. Residual networks that consist of a series of residual units are widely popular for image classification problems. Residual connections are added to the network to retain information of the previous layers. The proposed approach uses ResNet-18 as the backbone architecture to learn important yet discriminative features from palmprint images for indexing. The feature extraction process initiates by training ResNet-18 with the softmax loss. There are 16 convolution layers, two max-pooling layers and a fully connected layer in ResNet-18 architecture. The filter size of the first convolution layer is 7 × 7 while in other layers, it is 3 × 3. This is followed by a global average pooling layer and a batch normalization layer. The global average pooling (GAP) layer aggregates the input features by taking an average across the channels. This consolidation brings down the requirement of the number of parameters and thus, reduces the chances of over-fitting. The output feature becomes robust to spatial translations of the input images [53]. Activations of the GAP layer output are fed to the batch-normalization (BN) layer [54] which normalizes the input by subtracting it by mini-batch mean and diving by the mini-batch standard deviation. Mini-batch refers to a subset of the training data that is given to the network in one epoch. Batch normalization smoothens the landscape of the loss function by bringing the spread of all the input dimensions to the neurons to the same distribution, resulting in faster training of the model. A dropout layer has been introduced to avoid the over-fitting further. It is followed by a fully connected layer of 512 neurons. The last fully-connected layer serves as the feature embedding. The weights of this layer, along with the feature embeddings, are normalized. This makes the classification process solely dependent on the angle θ formed between the feature vector and weight vector. The whole network (PalmHashNet) is then trained in an end-to-end manner using the modified softmax loss, mentioned in Eq. 7. The architecture is shown in TABLE 1. PalmHashNet learns feature embeddings that have more intra-class and less inter-class similarity. The learned feature vectors are sent to the indexing module for index-table creation, explained in the next sub-section.

B. INDEXING

Identification aims at finding the closest or most similar sample from the database for a query palmprint sample. The naive approach for identification involves comparing the query sample with all the images in the database and sorting the similarity score list to find the most suitable match. However, this process becomes computationally expensive with an increase

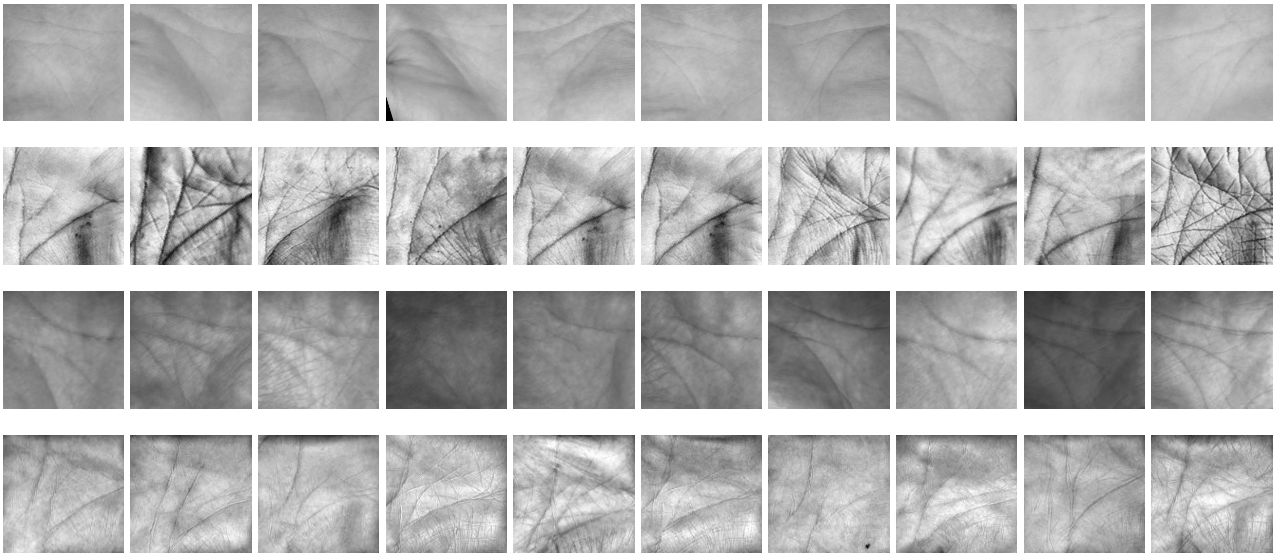


FIGURE 5. Sample palmprint region of interest (RoI) images from the CASIA-Palmprint [9], IIT Delhi Touchless [10], Tongji Contactless [11], and PolyU II Palmprint database [12] (row wise respectively).

in the size of the database. Therefore, there is a need to reduce the number of comparisons by reducing the search space for efficient identification. This is accomplished by indexing the database by associating the extracted features to an index and generating an index table that could be referred to during the retrieval. This paper uses two techniques, namely, 1) k-means Clustering [55] and 2) Locality Sensitive Hashing [56] to explore clustering and hashing as a means for indexing the considered palmprint databases. The algorithm for feature extraction and indexing is given in Algorithm 1.

1) k-MEANS CLUSTERING

The objective of the k-means clustering algorithm is to partition a set of feature vectors into a specified number of disjoint groups. Let there be a feature vector set $F = \{f_{P_1}, f_{P_2}, \ldots, f_{P_N}\}$ where, $P_1, P_2, \ldots, P_N$ represent N palmprint samples. k-means results in splitting F into 'k' disjoint clusters $c_1, c_2, \ldots, c_k$ such that similar feature vectors lie in the same cluster while separating those that are different from each other in the feature vector space. Each cluster has a representative data point that is also known as the mean of the feature vectors lying in that particular cluster. The algorithm starts by initializing 'k' centers using k-means++ initialization. Let the centers of 'k' clusters are represented by $m_1, m_2, \ldots, m_k$. k-means is a distance-based clustering algorithm that computes the euclidean distance between a feature vector f_{P_i} where, $i = \{1, N\}$, and every cluster center. This helps in determining the closest cluster and f_{P_i} is assigned to that cluster. After every iteration, the cluster centers get updated by computing the mean of all the feature vectors assigned to it. The process of assigning feature vectors to a cluster center and updating the cluster centers after each iteration is repeated until no further change in cluster assignment is observed or the maximum number of iterations has been exhausted. The goodness of clustering, which refers to how well k clusters approximate the feature vectors set, is evaluated by computing intra-cluster variance. Intra-cluster variance measures the amount of spread observed among the feature vectors lying in a particular cluster. It is computed using,

$$E_{k-means} = \sum_{j=1}^{k} \sum_{i=1}^{N} ||f_{P_i} - m_j|| \tag{8}$$

where, j and i denote number of clusters and feature vectors respectively. m_j is the center of cluster 'j.' After convergence, the cluster centers are stored as hash values in the index table and the palmprint IDs along with their feature vectors that lie in a particular cluster are stored in the bucket corresponding to it.

2) LOCALITY SENSITIVE HASHING

A Locality Sensitive Hashing (LSH) function maps the feature vectors to a lower-dimensional representation. Similar feature vectors are mapped in the same bucket with a high probability in the lower-dimensional space. The main objective of LSH is to maximize the probability of collision of similar items *i.e*; the probability of two similar feature vectors lying in the same bucket should be high. The hash function for an input feature vector f_{P_i} is computed by using two random values, $\vec{r}$ and x. Here, r is a d-dimensional vector whose entries are randomly chosen from a set of vectors following the Gaussian distribution. The dot product is quantized into a set of hash bins with the objective that all the nearby feature vectors should lie in the same bucket as shown,

$$h^{r,x}(f_{P_i}) = \left\lfloor \frac{\vec{r} \cdot f_{P_i} + x}{w} \right\rfloor \tag{9}$$

In this equation, w is the quantization width, and x is a random variable lying between 0 and w. Quantization width determines the number of entries or candidates that would lie

TABLE 2. Database specification with the number of comparisons during verification.

	CASIA	IITD	PolyU	Tongji
No. of subjects	312	460	193	300
Gallery samples	2496	1841	3860	6000
Probe samples	3006	760	3860	6000
Genuine comparisons	24048	3041	38600	60000
Imposter comparisons	7478928	1396118	14861000	35921592
Total comparisons	7481376	1399159	14899600	35981592

in each bucket of the hash table. Increasing the quantization width results in the compact table as each bucket will have more entries. On the other hand, a lower value of w results in a larger table with fewer candidates in each bucket of the hash table. The search for a true match for the query image is accomplished in a linear manner *i.e.*; the query image is compared with all the candidates lying in the selected bucket of the index table. Therefore, there is a trade-off between the table size and the final number of comparisons.

Two conditions must be satisfied to serve the purpose of reducing number of comparisons for identification of a query palmprint sample. These are,

- The probability of two feature vectors lying in the same bucket of index table should be high if they are close to each other in the feature embedding space. Let there are two feature vectors represented by f_{P_1} and f_{P_2}. Let the euclidean distance between the two feature vectors is $<d_1$. This distance is $\leq d_1$ which is the threshold distance value that determines if the given two feature vectors are close to each other in the feature embedding space. In this case, both f_{P_1} and f_{P_2} will lie in the same bucket. This is mathematically represented as,

$$P[h(f_{P_1}) = h(f_{P_2})] \geq p_1 \quad \text{if } \left\| f_{P_1}, f_{P_2} \right\| = d_1 \leq d \qquad (10)$$

- Contrary to the previous condition, this condition states that the probability of two dis-similar feature vectors, f_{P_1} and f_{P_3}, lying in the same bucket should be low. Let d_1 and d_2 be the euclidean distance between f_{P_1} and f_{P_2} and f_{P_1} and f_{P_3} respectively. Since f_{P_1} and f_{P_3} are dis-similar feature vectors, the distance between them should be greater than d *i.e.*, $d_2 \geq a \times d$, where a is any constant. Therefore, the probability of them lying in the same bucket of the index table will be less. Mathematically, it can be shown as,

$$P[h(f_{P_1}) = h(f_{P_3})] \leq p_1 \quad for \; \left\| f_{P_1}, f_{P_3} \right\| = d_2 \geq a \times d \qquad (11)$$

To further increase or reduce the probability given in the conditions respectively, a hash function of t- bits can be generated by performing t dot products in parallel using Eq. 9. A $t-$bit hash value of a feature vector belonging to palmprint sample P_1 can be computed by concatenating t values determined using the Eq. 9. $h(f^t_{P_1})$ can be written as $=h_1(f_{P_1}), h_2(f_{P_1}), \ldots, h_t(f_{P_1})$. After implementing LSH on the set of feature vectors generated by the trained model, we get a data structure that consists of hash value and the candidate IDs in its corresponding bucket.

$$P[h(f_{P_1}) = h(f_{P_2})] \geq p_1 \quad for \; \left\| f_{P_1}, f_{P_2} \right\| \leq d_1 \qquad (12)$$

$$P[h(f_{P_1}) = h(f_{P_3})] \leq p_2 \quad for \; \left\| f_{P_1}, f_{P_3} \right\| \geq a.d_1 = d_2 \qquad (13)$$

In these equations, f_{P_1}, f_{P_2} and f_{P_3} are feature vectors and d_1 and d_2 is the distance between f_{P_1} and f_{P_2} and f_{P_1} and f_{P_3} respectively. $\|\cdot\|$ is the euclidean distance between two vectors and we have considered that $d_2 > d_1$. After implementing LSH on the set of feature vectors generated by the trained model, we get a data structure that consists of hash value and all its nearby candidates' IDs in one row. To further increase or reduce the probability given in condition (1) and (2) respectively, a hash function of t- bits can be generated by performing t dot products in parallel using Eq. 9. That is, $h(f^t_{P_1})$ can be computed by concatenating t hash bits as given by, $h(f^t_{P_1}) = h_1(f_{P_1}), h_2(f_{P_1}), \ldots, h_t(f_{P_1})$.

TABLE 3. Recognition performance of PalmHashNet on the considered databases. Higher accuracy and DI are better, however EER is better when lower.

	CASIA	IITD	Tongji	PolyU II
Accuracy (%)	99.98	99.62	97.85	99.83
EER (%)	0.031	0.39	0.53	0.011
DI	4.71	3.94	2.82	4.10

C. RETRIEVAL

The objective of the Retrieval stage is to return a list of candidates that could be its probable matches when a query palmprint image is shown to the identification system. It commences by extracting the feature vector of the query image q_i using the proposed PalmHashNet. In the case of the index table generated through k-means, the centers of each cluster act as indices. Therefore, the query feature vector is compared with all the cluster centers or indices of the index table. The one with the highest similarity is selected, and candidates in that cluster are retrieved for identification. On the other hand, in the case of locality sensitive hashing, a $t-$bit hash function corresponding to the query feature vector is computed using Eq. 9. The hash bucket, which is most similar to the calculated hash function, is selected from the generated index table. The process is explained in Algorithm 2. This is followed by the identification that aims at finding a true match of the query sample from the retrieved candidate set. All the retrieved candidate IDs' feature vectors are matched with the query feature vector and their similarity scores are obtained. The score list is sorted in decreasing order, and the rank of the true match is recorded to gauge the performance of the indexing approach. This process makes identification a constant time operation as the size of the retrieved candidate list is fixed.

IV. EXPERIMENTAL RESULTS

This section gives details about experimental setting such as the databases and evaluation parameters that have been considered for evaluating the proposed approach along with

TABLE 4. Table showing Equal Error Rate (EER) comparison of the proposed approach (PalmHashNet) with state-of-the-art palmprint recognition techniques.

Author (s)	Feature	CASIA [9]	IITD [10]	PolyU II [12]	Tongji [11]
Lu et al. [14]	Competitive Code (CC)	0.55	7.72	0.6	-
Jia et al. [16]	Robust line orientation code (RLOC)	0.81	7.44	0.16	-
Zuo et al. [17]	Sparse multiscale CC	0.48	-	0.01	-
Luo et al. [19]	LBP based feature descriptor	-	4.07	0.02	-
Li et al. [20]	Local feature based on direction and thickness	-	0.87	-	-
Zhong et al. [21]	Siamese network	-	-	0.28	-
Zhong et al. [24]	Centralized large margin cosine loss	-	-	0.08	**0.25**
Zhu et al. [36]	Adversarial metric learning	-	1.73	0.86	1.21
Proposed	**PalmHashNet**	**0.03**	**0.39**	**0.01**	0.53

an overview of the training and testing protocol. The results are listed for both verification (recognition) and identification with indexing.

A. DATABASE SPECIFICATIONS

Four publicly available standard databases namely, CASIA Palmprint Image Database, IIT Delhi Touchless Palmprint Database, Tongji Contactless Palmprint database and Hong Kong Polytechnic University Palmprint II Database (PolyU II have been used to evaluate the performance of the proposed approach. The details of the databases are as below. Region of interest (RoI) samples from each database is shown in FIGURE 5.

1) CASIA PALMPRINT IMAGE DATABASE [9]

This database, also referred to as CASIA-Palmprint consists of 5502 palmprint images collected from both left and right hand of 312 subjects. Eight images from each subject were collected in single session and the individuals were not instructed regarding positioning their hands. Therefore, the acquired images have huge pose variance. RoI of size 128 × 128 is segmented from the images.

2) IIT DELHI TOUCHLESS PALMPRINT DATABASE [10]

IITD palmprint database was collected from students and staff of IIT Delhi in June 2006 and July 2007. A total of 2600 images were collected from 460 palms of 230 subjects. All the images are in bitmap format and the RoI has a resolution of 150 × 150 pixels.

3) TONGJI CONTACTLESS PALMPRINT DATABASE [11]

This database is comparatively larger in size than the other two. It consists of palmprint images collected from both hands of 300 subjects. Ten images of each palm per subject were acquired in two separate sessions, making it a database of total 12000 images. The images have a resolution of 128 × 128 pixels.

4) HONG KONG POLYTECHNIC UNIVERSITY PALMPRINT II DATABASE (PolyU II) [12]

This database consists of 7752 palmprint samples collected from 193 individuals. The samples have been collected in two different sessions with a gap of 2 months. Ten palmprint samples from each palm of all the individuals were acquired. The extracted RoI has a size of 128 × 128 pixels.

B. EVALUATION PARAMETERS

The evaluation of the proposed approach is done in two ways. As specified earlier, the performance of indexing and identification would depend on the quality of learned feature vectors. Therefore, before proceeding to the identification, we first evaluate the quality of feature vectors using 1) Accuracy, 2) Equal Error Rate (EER), and 3) Discriminative Index (DI) of the recognition system. Each test sample was matched with each sample in the train partition to calculate genuine and imposter matching scores. A genuine match is when matching is done between samples belonging to the same class or palm. It is called an imposter match if both the samples belong to different palms. The parameters mentioned above are used to evaluate the performance of any biometric verification system. The performance of an identification system is evaluated in terms of correct recognition rate (CRR), hit rate (HR), penetration rate (PR), and Cumulative Match Characteristic (CMC) curve. All these are described below.

1) EQUAL ERROR RATE

False acceptance rate (FAR) is the number of imposter matches that got accepted by the system out of total number of samples shown to the system. Whereas, False Rejection Rate (FRR) is the ratio of the genuine samples that got rejected by the system. Equal error rate (EER) is the point where FAR becomes equal to FRR. It can be written as given in (14), where nFA is the total number of samples that got falsely accepted out of the total number of testing samples that were matched from different class (nIM).

$$EER = \frac{nFA}{nIM} \tag{14}$$

2) DISCRIMINATIVE INDEX

It refers to the separation between genuine and imposter scores. DI is defined as given in the equation below, where $\mu_{(.)}$ and $\sigma_{(.)}$ denotes mean and standard deviation of either

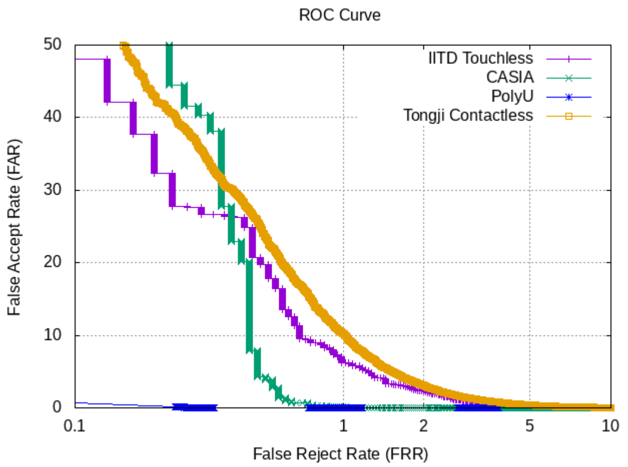


FIGURE 6. ROC of the proposed approach on the four considered databases.

genuine or imposter scores.

$$DI = \frac{\mu_G - \mu_I}{\sigma_G^2 - \sigma_I^2} \tag{15}$$

3) ACCURACY

Accuracy of a recognition system is defined as number of correctly recognized samples out of total queries made to the system. Mathematically it can be expressed as,

$$Accuracy = max(100 - \frac{FAR + FRR}{2}) \tag{16}$$

4) CORRECT RECOGNITION RATE

CRR is defined as the rank-1 recognition rate *i.e*. number of queries that have got their true match at the rank-1. If nCR is the number of correctly recognized queries at rank-1 out of total nQ queries made to the system, then CRR can be defined as given in (17)

$$CRR = \frac{nCR}{nQ} \tag{17}$$

5) HIT RATE

During identification, a query or probe image is referred to as correctly identified if one of the retrieved candidates belongs to the same identity as that of probe image. Therefore, Hit Rate (HR) is defined as the ratio of correctly identified queries ($nCIQ$) with respect to the total number of queries made to the system (nQ), as given in (18).

$$\text{Hit Rate} = \frac{nCIQ}{nQ} \tag{18}$$

6) PENETRATION RATE

It is the percentage of the database that needs to be retrieved for correct identification of the query image. If nQ number of queries are made to the system and for each query i, C_i number of candidates are retrieved from a total of D templates, where D corresponds to the size of the database. Then, penetration rate (PR) can be defined as,

$$\text{Penetration Rate} = \frac{1}{nQ}\sum_{i=1}^{nQ}\frac{C_i}{D}. \tag{19}$$

TABLE 5. Penetration rates (%) for different values of Hit Rate (HR) for the proposed approach when k-means clustering is applied for indexing.

HR(%)	CASIA [9]	IITD [10]	Tongji [11]	PolyU II [12]
90	0.022	0.054	6.55	0.025
95	0.022	2.39	13.15	0.025
96	0.022	3.096	15.633	0.025
97	0.022	4.943	19.166	0.025
98	0.022	7.17	23.089	1.295
99	0.022	17.599	32.866	2.875
100	3.367	34.818	69.983	20.777

TABLE 6. Penetration rates (%) for different values of Hit Rate (HR) for the proposed approach when LSH is applied for indexing.

HR(%)	CASIA [9]	IITD [10]	Tongji [11]	PolyU II [12]
90	0.022	0.054	0.016	0.025
95	0.022	0.054	0.016	0.025
96	0.022	0.054	0.016	0.025
97	0.022	0.054	0.02	0.025
98	0.022	0.054	0.12	0.025
99	0.022	0.162	0.87	0.025
100	0.800	1.032	4.555	0.39

7) CUMULATIVE MATCH CHARACTERISTIC CURVE

CMC curve is a rank-based metric that shows relationship between identification probability at a given rank. That is, what ratio of total queries got correctly identified till a particular rank.

C. TRAINING AND TESTING PROTOCOL

The images contained in the considered databases are not uniform, and there is no standard training and testing protocol associated with the databases. Therefore, this paper uses a training and testing partition that is mostly followed for the mentioned databases. The training partition refers to the gallery images or the samples stored in the database. On the other hand, the testing partition contains the query images for which identification needs to be performed. The CASIA-palmprint database is partitioned into 80%−20% split for training and testing, respectively. In the IIT Delhi palmprint database, each subject has either given five or six images. Hence, we have used four images for training and the remaining 1 or 2 images for testing. For the Tongji contactless palmprint database, the training and testing split is done session-wise *i.e*. ten images from session one are used as gallery images while the remaining ten images collected in session two are used as query images.

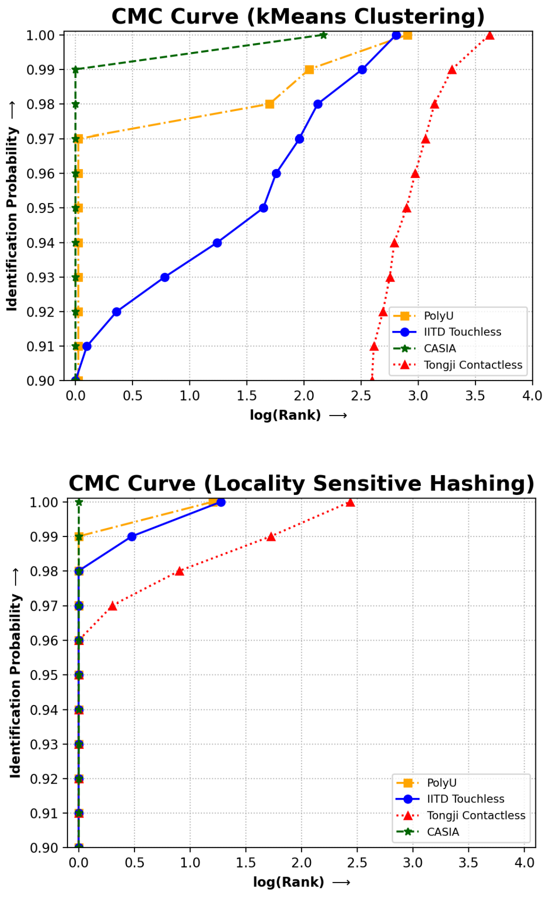


FIGURE 7. CMC curve showing relationship between Identification probability and rank in case of k-means clustering and locality sensitive hashing.

TABLE 7. Query time (in seconds) comparison of the proposed approach against non-indexed process for identification on the four databases.

Database	Without Indexing	With Indexing
CASIA [9]	0.013	**0.0008**
IITD [10]	0.018	**0.006**
Tongji [11]	0.040	**0.020**
PolyU II [12]	0.010	**0.0037**

D. EXPERIMENTAL SETTING

The proposed PalmHashNet is implemented using the Pytorch library of the Python programming language. Lately, the palmprint acquisition has been performed in an unconstrained environment for a better user experience. The acquired images tend to have occlusion, illumination variation *etc*. Data augmentation is applied on the training partition of the databases to make PalmHashNet robust to such variations. It helps make the deep neural network more potent by training it on different variations of the palmprint samples. Python Augmentor library [57] has been used to augment the training partition by applying image transformation operations such as random zoom, distortion, rotation, and illumination. With these four operations applied to each image, four new images were created, increasing the training partition five times. Hence, the size of the training partition became 12480, 9205, 19300, and 30000 for CASIA, IITD-Touchless, PolyU II and Tongji Contactless databases, respectively. The test partition has not been augmented as the test needs to be performed on the original images only. A computer with Xenon (R) processor with 32 GB RAM and 12 GB on-card RAM on NVIDIA Tesla K40C GPU has been used to train the network and evaluate the indexing and retrieval performance. For index table creation, k-means and LSH have been utilized for partitioning the set of learned feature vectors. k-means is used to cluster similar feature vectors based on similarity for index table creation. The number of clusters, denoted by k, for a particular database is determined by using a metric called silhouette coefficient [58]. Silhouette coefficient is a metric that evaluates the goodness of a clustering technique. Mathematically, it is defined as $(q-p)/\max(p, q)$ where p and q refer to the average intra-cluster and inter-cluster distance, respectively. Intra-class distance is the distance between each point within the same cluster. On the other hand, inter-class distance is computed between data points lying in different clusters. The silhouette coefficient value lies between -1 to $+1$ with $+1$ indicating well-separated clusters. The silhouette coefficient for various values of k is computed to determine the appropriate value of k. It was experimentally observed that its value is highest at $k = 60$. This became the initial point for finding the suitable value k for the indexing approach. Further, k-means was implemented for all values in the range interval of [5, 100] with a difference of 5. It was empirically determined that indexing performance achieved the best results when $k = 65$ for the considered databases.

E. RESULTS

Results are computed for both verification and identification system to validate the performance of the proposed approach. Firstly, the quality of the learned features is computed because their discriminative ability determines the performance of the indexing module. Therefore, the recognition results are listed, followed by indexing performance based on two different techniques, namely k-means clustering and Locality Sensitive Hashing (LSH).

1) RECOGNITION PERFORMANCE

To evaluate the verification performance of the proposed approach, we have computed the Accuracy, EER, and DI of the system. Each sample in the test partition is matched with all the samples in the training partition. The detail regarding the number of matchings comprising of genuine and imposter matchings is given in TABLE 2. In the table, gallery images refer to training split as these are the images that are indexed. In contrast, probe samples refer to the testing split as these images are used for querying the identification system. The proposed approach obtained >99% accuracy for CASIA, IITD Touchless, and PolyU II database. However, for Tongji contactless, the reported accuracy is 97.85%. However, the

TABLE 8. Table showing Rank-1 identification rate (I.R.) comparison of the proposed approach (PalmHashNet) with state-of-the-art techniques on various databases.

Author	Feature	CASIA [9]	IITD [10]	Tongji [11]	PolyU [12]
Paliwal et al. [27]	Vector Approximation (VA+) file database	-	-	-	96.01 %
Almaghtuf et al. [33]	Difference of block means	94.17%	99.02%	-	99.40%
Zhu et al. [36]	Adversarial metric learning	-	99.02%	97.71%	99.02%
Zhao et al. [37]	Deep feature by combining local and global features	97.06%	97.25%	-	-
Jia et al. [12]	Column sampling based discrete supervised hashing	-	-	96.38%	99.25%
Chen et al. [34]	Double-orientation feature from orientation fields	-	-	-	99.25%
Liu et al. [39]	Similarity Metric Hashing Network	-	-	97.65%	-
Proposed	**PalmHashNet**	**100%**	**99.42%**	**97.08%**	**100%**

TABLE 9. Table comparing Rank-1 identification rate of model trained using softmax loss and model trained using the proposed methodology.

	CASIA	IITD	Tongji	PolyU II
W/o additive margin	89.71%	84.89%	92.49%	97.25%
Proposed	100%	99.42%	97.08%	100%

obtained EER is $<1\%$ for all the considered databases. Area Under the Curve (AUC) refers to the degree of separability *i.e* it depicts the model's capability in distinguishing between classes. Higher AUC indicates a better model. The obtained values of these parameters for the considered databases are shown in TABLE 3. With low values of EER and high values of accuracy, it is evident that the proposed approach performs quite well. The Receiver Operating Characteristic (ROC) curve depicts the trade-off between false acceptance rate (FAR) and false reject rate (FRR) at different threshold values. It presents the performance of a classification system at various classification threshold values in a comprehensive manner. Area Under the Curve (AUC) refers to the degree of separability by depicting the capability of a model in distinguishing between different classes. ROC curve of the proposed approach on all the considered databases is shown in Figure 6. Here, $FAR = FRR$ line is not straight because FRR is plotted on the log scale. The log scaling brings the focus towards a more meaningful lower side of the curve. The area under the ROC curve depicts error and a system with a lesser area is considered better in general. Different ROC curves shown in Figure 6 compare the classification performance of the proposed feature on the four databases *viz.* CASIA, IITD-Touchless, Tongji Contactless, and PolyU II databases. The verification result of the proposed approach is compared with various palmprint recognition techniques proposed in the literature. It can be seen from TABLE 4 that PalmHashNet performs best in terms of equal error rate when tested on CASIA, IITD Touchless, and PolyU II database. It has the lowest EER for these three databases.

2) INDEXING PERFORMANCE

Indexing performance is evaluated in terms of hit rate and penetration rate. It is expected from a good indexing technique to achieve a lower penetration rate at a high hit rate. TABLE 5 and TABLE 6 show penetration rate required for different values of hit rate on the considered four databases when k-means and LSH are implemented respectively for indexing. The proposed approach achieves a hit rate of 95% at 0.022% and 0.025% penetration rate for CASIA and PolyU II database irrespective of which technique has been applied for indexing. But for the other two databases, the penetration rate is manifold for k-means as compared to LSH. The penetration rate for k-means is increasing very fast while it remains low for LSH at even high hit rates. Therefore, it can be concluded that LSH performs better than k-means clustering for index table creation. Hence by considering LSH, we can conclude that 100%, 98.81%, 97.08%, and 100% hit rate is achieved at Rank-1 for CASIA-Palmprint, IITD Touchless, Tongji Contactless, and PolyU II palmprint databases respectively. The results signify that to obtain a true match in these databases with 100% confidence, it is required to compare the query image with only **0.022%, 1.032%, 4.555%, and 0.39%** of the respective databases. The CMC curve showing the relationship between the probability of identification and rank for k-means and LSH is shown in FIGURE 7.

a: TIME ANALYSIS

The time-based performance of the proposed approach is evaluated in terms of speedup. It refers to how fast the identification process has become by using the proposed approach for indexing compared to the naive approach for identification (without indexing). It is calculated by taking into account the total time (in seconds) to find a suitable candidate set from the created index table for identifying a query image. The proposed approach takes 39.227s, 13.698s, 245.896s, and 39.227s when identification is performed with the traditional approach on CASIA, IITD Touchless, Tongji Contactless, and PolyU II database respectively. However, this time reduced to only 2.5s, 4.992s, 123.623s, and 14.551s, respectively, when the proposed indexing is applied on the palmprint databases. TABLE 7 shows a comparison of the average query time taken for palmprint identification with and without indexing (using LSH) on the considered databases. It can be observed that the average speedup is six times if the proposed approach is used for identification on the considered four databases.

b: COMPARATIVE ANALYSIS

To compare the results with state-of-the-art techniques, we used Rank-1 identification rate as most papers report

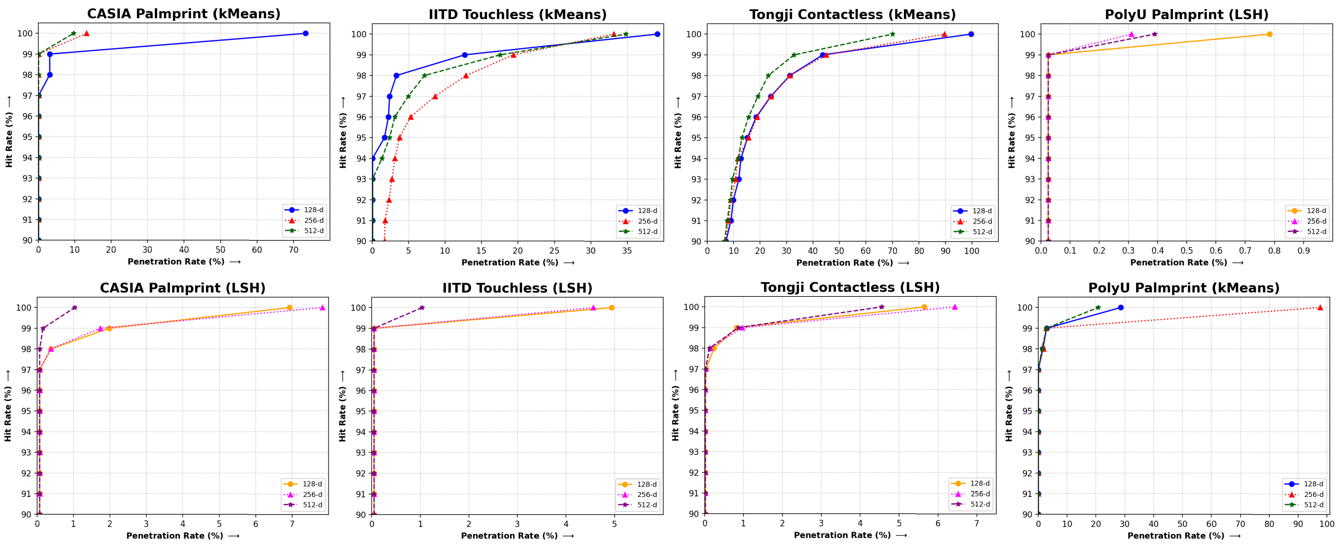


FIGURE 8. **Ablation study for different feature size on the considered databases using kMeans clustering and LSH for index table creation.**

that. The first comparison of the proposed approach is made with [12]. The paper investigated the application of the available supervised, unsupervised and deep hashing techniques on Tongji Contactless and PolyU II databases for retrieval. The study demonstrates which hashing techniques performed better on different palmprint databases concerning rank-1 identification rate. It was reported that rank-1 identification rate for KNNH [59], DSH [60] and ADSH [61] was 96.25%, 91.4% and 96.05% for Tongji Contactless database while the values were 98.30%, 93.35% and 94.86% for PolyU II palmprint database. However, COSDISH [12] achieved best results with 96.38% and 99.25% rank-1 identification rate on these two databases respectively. The proposed approach in this paper achieves a Rank-1 identification rate of 97.08% and 100% on Tongji contactless and PolyU II database, outshining other hashing techniques. Another comparison is made with state-of-the-art palmprint indexing techniques in TABLE 8. The best rank-1 identification rate for CASIA was achieved by Zhang *et al.* [11]. For IITD Touchless and Tongji Contactless databases, the approach proposed by Zhu *et al.* [36] gave the best results till now. However, approach proposed in [33] performed best on PolyU II database. It is clearly evident from the TABLE 8 that our approach achieves best value of rank-1 identification rate for CASIA, IITD touchless and PolyU II databases. Therefore, by examining both the comparisons, we can say that the proposed approach outperforms other techniques proposed in the literature.

F. ABLATION STUDY

The proposed approach solely depends on the feature extraction process. Thus, it was required to find the best features that could perform well for the recognition and identification process. Therefore, two different ablation studies are conducted in this paper. The first objective is to check the effect of adding a margin to the softmax loss for training the feature extraction model. To understand that, the feature extraction model was trained with and without a margin to observe the effect on the quality of the extracted feature vectors. Rank-1 identification rate was computed for the considered databases on the extracted features using only the softmax loss. TABLE 9 shows the comparison with respect to the rank-1 identification rate obtained by the proposed approach and the model trained with the softmax loss. It was observed that the Rank-1 identification rate improved by 10.29%, 14.53%, 4.59%, and 2.75% times on CASIA, IITD-Touchless, Tongji Contactless, and PolyU II databases respectively by introducing additive margin loss in the feature extraction network.

The second study analyzes the effect of different dimensions (sizes) of feature vectors to find the most appropriate size that best represents a palmprint image. Three dimensions of feature vectors such as 128, 256, and 512 are analyzed to find the suitable feature vector size that best represents a palmprint image taken from the considered databases. Indexing is performed on the databases for all the feature dimensions mentioned above using Algorithm 1. It was empirically determined that the system achieves higher accuracy with a 512-dimensional feature vector. The same is shown in TABLE 10. A relationship between the hit rate and the penetration rate is established for all the combinations. A candidate set is retrieved for a query palmprint image from the indexed palmprint database. Hit rate determines the confidence by which a query image can find its true match in the retrieved set of candidates. Penetration rate refers to the percentage of the whole database required to find the true match of a query image. The true identity of a query sample is expected to be established by retrieving only a small percentage of the database (low penetration rate) with high confidence (high hit rate). An efficient biometric indexing approach is expected to achieve a high hit rate at a lower value of penetration rate. The same has been shown in TABLE 10. It can be observed that 512-d feature vector performs best

TABLE 10. **Ablation study: The table showing Penetration Rate (%) at various Hit Rate (%) w.r.t. different feature vector sizes (128-d, 256-d and 512-d) for the considered databases.**

			Hit Rate (%)						
Palmprint Database	**Indexing Scheme**	**Feature Dimension**	**90 (%)**	**95 (%)**	**96%**	**97%**	**98%**	**99%**	**100%**
CASIA [9]	k-means	**128-d**	0.070	0.070	0.070	0.070	3.132	3.132	73.337
		256-d	0.070	0.070	0.070	0.070	0.070	0.070	13.254
		512-d	0.070	0.070	0.070	0.070	0.070	0.070	9.678
	LSH	**128-d**	0.054	0.054	0.054	0.054	0.380	1.980	6.930
		256-d	0.054	0.054	0.054	0.054	0.380	1.730	7.827
		512-d	0.054	0.054	0.054	0.054	0.054	0.162	1.032
IITD [10]	k-means	**128-d**	0.054	1.684	2.227	2.390	3.313	12.710	39.109
		256-d	1.683	3.747	5.268	8.636	12.873	19.391	33.134
		512-d	0.054	2.390	3.096	4.943	7.170	17.599	34.818
	LSH	**128-d**	0.054	0.054	0.054	0.054	0.054	0.054	4.940
		256-d	0.054	0.054	0.054	0.054	0.054	0.054	4.562
		512-d	0.054	0.054	0.054	0.054	0.054	0.162	1.032
Tongji [11]	k-means	**128-d**	7.083	15.233	18.550	24.066	31.266	43.683	99.666
		256-d	7.016	15.680	18.833	24.150	31.266	45.033	89.550
		512-d	6.550	13.150	15.633	19.166	23.083	32.866	69.983
	LSH	**128-d**	0.016	0.016	0.016	0.036	0.238	0.827	5.645
		256-d	0.016	0.016	0.016	0.020	0.140	0.970	6.435
		512-d	0.016	0.016	0.016	0.020	0.120	0.870	4.555
PolyU [12]	k-means	**128-d**	0.025	0.025	0.025	0.025	1.554	2.927	28.626
		256-d	0.025	0.025	0.025	0.025	1.813	2.772	97.642
		512-d	0.025	0.025	0.025	0.025	1.295	2.875	20.777
	LSH	**128-d**	0.025	0.025	0.025	0.025	0.025	0.025	0.783
		256-d	0.025	0.025	0.025	0.025	0.025	0.025	0.310
		512-d	0.025	0.025	0.025	0.025	0.025	0.025	0.390

on mostly all the considered databases. Hence, we have conceded 512-d feature vector in this study. The graphs showing hit rate vs. penetration rate for all the experiments are shown in FIGURE 8.

V. CONCLUSION

This paper proposes a palmprint database indexing approach called PalmHashNet that generates highly discriminative embeddings to create a fixed-size candidate list for comparison to make identification a constant time operation. Softmax loss with additive margin is used to train the model for palmprint database indexing and to learn the feature vector embeddings simultaneously. This loss function ensures that the learned feature embeddings have low inter-class along with high intra-class similarity. It ascertains that the index space distribution is regularized to be similar to the uniform distribution. The learned embeddings are indexed using k-means Clustering and Locality Sensitive Hashing technique to create an index table. Whenever a query image is given to the identification system, the features are extracted using PalmHashNet for query palmprint. The generated feature vector is matched with all the indices of the index table and the candidates lying in the most similar index are retrieved for comparison. Verification and identification experiments are conducted on four publicly available popular palmprint databases *viz.* CASIA, IITD-Touchless, Tongji-Contactless and PolyU II palmprint databases to show a thorough evaluation of the extracted features. The proposed approach achieved >99% accuracy on CASIA, IITD-Touchless, and PolyU II palmprint database while it achieved 97.85% accuracy on Tongji Contactless database. The reported equal error rate for all the databases is less than 1%, which is quite good. During identification, PalmHashNet achieved a penetration rate of 0.022%, 1.032%, 4.555% and 0.39% at 100% hit rate for the respective databases. Therefore, it can be concluded that to find the true match of a query sample with 100% confidence, it is required to look out for <1% of the CASIA and PolyU II database and 1.03% and 4.55% of the IITD-Touchless and Tongji Contactless database respectively. Hence, by using PalmHashNet, we need to search only a small percentage of the database instead of the whole database for identification without compromising accuracy. The proposed approach outperforms other state-of-the-art recognition and indexing techniques proposed in the literature, making identification a constant time operation.

REFERENCES

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

[2] *Aadhar*. Accessed: May 16, 2020. [Online]. Available: http://www.uidai.gov.in

[3] *Malaysia Identity Card*. Accessed: Aug. 20, 2020. [Online]. Available: https://www.malaysia.gov.my/portal/content/30582

[4] P. Gupta, K. Tiwari, and G. Arora, "Fingerprint indexing schemes—A survey," *Neurocomputing*, vol. 335, pp. 352–365, Mar. 2019.

[5] S. Fekri-Ershad, "Gender classification in human face images for smart phone applications based on local texture information and evaluated Kullback-Leibler divergence," *Traitement du Signal*, vol. 36, no. 6, pp. 507–514, Dec. 2019.

[6] X. Tan, B. Bhanu, and Y. Lin, "Fingerprint identification: Classification vs. indexing," in *Proc. IEEE Conf. Adv. Video Signal Based Surveill.*, Jul. 2003, pp. 151–156.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[8] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.

[9] Z. Sun, T. Tan, Y. Wang, and S. Z. Li, "Ordinal palmprint represention for personal identification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 279–284.

[10] A. Kumar, "Incorporating cohort information for reliable palmprint authentication," in *Proc. 6th Indian Conf. Comput. Vis., Graph. Image Process.*, Dec. 2008, pp. 583–590.

[11] L. Zhang, L. Li, A. Yang, Y. Shen, and M. Yang, "Towards contactless palmprint recognition: A novel device, a new benchmark, and a collaborative representation based identification approach," *Pattern Recognit.*, vol. 69, pp. 199–212, Sep. 2017.

[12] W. Jia, B. Wang, Y. Zhao, H. Min, and H. Feng, "A performance evaluation of hashing techniques for 2D and 3D palmprint retrieval and recognition," *IEEE Sensors J.*, vol. 20, no. 20, pp. 11864–11873, Feb. 2020.

[13] W. W. Boles and S. Y. T. Chu, "Personal identification using images of the human palm," in *Proc. Speech Image Technol. Comput. Telecommun.*, vol. 1, 1997, pp. 295–298.

[14] G. Lu, D. Zhang, and K. Wang, "Palmprint recognition using eigenpalms features," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1463–1467, 2003.

[15] A. W.-K. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, Aug. 2004, pp. 520–523.

[16] W. Jia, D.-S. Huang, and D. Zhang, "Palmprint verification based on robust line orientation code," *Pattern Recognit.*, vol. 41, no. 5, pp. 1504–1513, 2008.

[17] W. Zuo, Z. Lin, Z. Guo, and D. Zhang, "The multiscale competitive code via sparse representation for palmprint verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2265–2272.

[18] Y. Xu, L. Fei, J. Wen, and D. Zhang, "Discriminative and robust competitive code for palmprint recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 2, pp. 232–241, Feb. 2016.

[19] Y.-T. Luo, L.-Y. Zhao, B. Zhang, W. Jia, F. Xue, J.-T. Lu, Y.-H. Zhu, and B.-Q. Xu, "Local line directional pattern for palmprint recognition," *Pattern Recognit.*, vol. 50, pp. 26–44, Feb. 2016.

[20] G. Li and J. Kim, "Palmprint recognition with local micro-structure tetra pattern," *Pattern Recognit.*, vol. 61, pp. 29–46, Jan. 2017.

[21] D. Zhong, Y. Yang, and X. Du, "Palmprint recognition using Siamese network," in *Proc. Chin. Conf. Biometric Recognit.*, 2018, pp. 48–55.

[22] L. Fei, B. Zhang, S. Teng, and W. Zhang, "Local apparent and latent direction extraction for palmprint recognition," *Inf. Sci.*, vol. 473, pp. 59–72, Jan. 2019.

[23] S. Zhao and B. Zhang, "Joint constrained least-square regression with deep convolutional feature for palmprint recognition," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jul. 3, 2020, doi: 10.1109/TSMC.2020.3003021.

[24] D. Zhong and J. Zhu, "Centralized large margin cosine loss for open-set deep palmprint recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1559–1568, Jun. 2020.

[25] J. You, W.-K. Kong, D. Zhang, and K. H. Cheung, "On hierarchical palmprint coding with multiple features for personal identification in large databases," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 234–243, Feb. 2004.

[26] W. Li, J. You, and D. Zhang, "Texture-based palmprint retrieval using a layered search scheme for personal identification," *IEEE Trans. Multimedia*, vol. 7, no. 5, pp. 891–898, Oct. 2005.

[27] A. Paliwal, U. Jayaraman, and P. Gupta, "A score based indexing scheme for palmprint databases," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 2377–2380.

[28] X. Yang, J. Feng, and J. Zhou, "Palmprint indexing based on ridge features," in *Proc. Int. Joint Conf. Biometrics (IJCB)*, Oct. 2011, pp. 1–8.

[29] Y.-C. Chen, M.-H. Lim, P.-C. Yuen, and J.-H. Lai, "Discriminant spectral hashing for compact palmprint representation," in *Proc. Chin. Conf. Biometric Recognit.* Cham, Switzerland: Springer, 2013, pp. 225–232.

[30] F. Yue, B. Li, M. Yu, and J. Wang, "Hashing based fast palmprint identification for large-scale databases," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 5, pp. 769–778, May 2013.

[31] F. Yue, H. Ding, B. Li, and X. Chen, "Accelerated POP hashing for fast palmprint identification on large-scale databases," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 11121–11126.

[32] J. Cheng, Q. Sun, J. Zhang, and Q. Zhang, "Supervised hashing with deep convolutional features for palmprint recognition," in *Proc. Chin. Conf. Biometric Recognit.* Cham, Switzerland: Springer, 2017, pp. 259–268.

[33] J. Almaghtuf, F. Khelifi, and A. Bouridane, "Fast and efficient difference of block means code for palmprint recognition," *Mach. Vis. Appl.*, vol. 31, no. 6, pp. 1–10, Sep. 2020.

[34] X. Chen, M. Yu, F. Yue, and B. Li, "Orientation field code hashing: A novel method for fast palmprint identification," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 5, pp. 1038–1051, May 2020.

[35] H. Shao, D. Zhong, and X. Du, "Towards efficient unconstrained palmprint recognition via deep distillation hashing," 2020, *arXiv:2004.03303*.

[36] J. Zhu, D. Zhong, and K. Luo, "Boosting unconstrained palmprint recognition with adversarial metric learning," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 2, no. 4, pp. 388–398, Oct. 2020.

[37] S. Zhao and B. Zhang, "Deep discriminative representation for generic palmprint recognition," *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107071.

[38] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 27–35.

[39] C. Liu, D. Zhong, and H. Shao, "Few-shot palmprint recognition based on similarity metric hashing network," *Neurocomputing*, vol. 456, pp. 540–549, 2021.

[40] A.-S. Ungureanu, S. Salahuddin, and P. Corcoran, "Toward unconstrained palmprint recognition on consumer devices: A literature review," *IEEE Access*, vol. 8, pp. 86130–86148, 2020.

[41] Q. Xiao, J. Lu, W. Jia, and X. Liu, "Extracting palmprint ROI from whole hand image using straight line clusters," *IEEE Access*, vol. 7, pp. 74327–74339, 2019.

[42] S. Aoyama, K. Ito, T. Aoki, and H. Ota, "A contactless palmprint recognition algorithm for mobile phones," in *Proc. Int. Workshop Adv. Image Technol.*, 2013, pp. 409–413.

[43] D. Zhang, W.-K. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.

[44] G. K. O. Michael, T. Connie, and A. B. J. Teoh, "Touch-less palm print biometrics: Novel design and implementation," *Image Vis. Comput.*, vol. 26, no. 12, pp. 1551–1560, 2008.

[45] M. Aykut and M. Ekinci, "Developing a contactless palmprint authentication system by introducing a novel ROI extraction method," *Image Vis. Comput.*, vol. 40, pp. 65–74, Aug. 2015.

[46] M. A. Ferrer, F. Vargas, and A. Morales, "BiSpectral contactless hand based biometric system," in *Proc. CONATEL*, May 2011, pp. 1–6.

[47] J. Doublet, O. Lepetit, and M. Revenu, "Contact less hand recognition using shape and texture features," in *Proc. 8th Int. Conf. Signal Process.*, vol. 3, 2006, pp. 1–5.

[48] H. Shao, D. Zhong, and X. Du, "Efficient deep palmprint recognition via distilled hashing coding," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2019, pp. 714–723.

[49] K. Tiwari, C. J. Hwang, and P. Gupta, "A palmprint based recognition system for smartphone," in *Proc. Future Technol. Conf. (FTC)*, Dec. 2016, pp. 577–586.

[50] F. Gao, L. Leng, and J. Zeng, "Palmprint recognition system with double-assistant-point on iOS mobile devices," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018, pp. 1–8.

[51] L. Leng, F. Gao, Q. Chen, and C. Kim, "Palmprint recognition system on mobile devices with double-line-single-point assistance," *Pers. Ubiquitous Comput.*, vol. 22, no. 1, pp. 93–104, Feb. 2018.

[52] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, vol. 2, 2016, p. 7.

[53] T.-Y. Hsiao, Y.-C. Chang, H.-H. Chou, and C.-T. Chiu, "Filter-based deep-compression with global average pooling for convolutional networks," *J. Syst. Archit.*, vol. 95, pp. 9–18, May 2019.

[54] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization?" in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2488–2498.

[55] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, Feb. 2003.

[56] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors [lecture notes]," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 128–131, Mar. 2008.

[57] M. D. Bloice, C. Stocker, and A. Holzinger, "Augmentor: An image augmentation library for machine learning," 2017, *arXiv:1708.04680*.

[58] S. Aranganayagi and K. Thangavel, "Clustering categorical data using silhouette coefficient as a relocating measure," in *Proc. Int. Conf. Comput. Intell. Multimedia Appl. (ICCIMA)*, vol. 2, Dec. 2007, pp. 13–17.

[59] X. He, P. Wang, and J. Cheng, "K-nearest neighbors hashing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2839–2848.

[60] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2064–2072.

[61] Q. Y. Jiang and W.-J. Li, "Asymmetric deep supervised hashing," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 3342–3349.

GEETIKA ARORA received the M.Tech. degree in computer science and engineering from Banasthali Vidyapith, Rajasthan, India. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Systems, Birla Institute of Technology and Science (BITS), Pilani, India. Her research interests include machine learning, biometrics, and image processing.

SUMIT KALRA received the Ph.D. degree from IIT Kanpur. He is currently an Assistant Professor with the Department of Computer Science and Engineering, IIT Jodhpur. He has a couple of patents and high quality publications to his name. He has also been actively working towards designing and deploying integrated and affordable telemedicine solutions in the recent years. His research interests include exploring the software architectural issues related to quality of the complex software systems across various domains such, as data intensive processing, AI, and IoT-based systems.

ASHUTOSH BHATIA received the Ph.D. degree in computer science from the Indian Institute of Science, Bangalore, India, in 2016. He is currently an Assistant Professor with the Department of Computer Science and Information Systems, BITS Pilani. Previously, he has worked as a Scientist with the Defence Research and Development Organization (DRDO), India, and a Research Engineer at Samsung India Software Operations (SISO). His research interests include building new designs, protocols, algorithms and theories that improve the security, robustness, and performance of networked systems.

KAMLESH TIWARI (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Indian Institute of Technology Kanpur. He is currently an Assistant Professor with the Department of Computer Science and Information Systems, Birla Institute of Technology and Science Pilani, Pilani Campus. He is an APPCAIR Affiliated Faculty and heads the MapMyIndia AI-ML Laboratory. He is a co-in charge of the Advanced Data Analytics and Parallel Technologies Laboratory, BITS Pilani, and an active member of the Multimedia and HCI Laboratory. His research interests include machine learning, deep learning, artificial intelligence, computer vision, multimodal biometric, such as fingerprint, face, palmprint, and knuckleprint, and security. He is a member of the Signal Processing Society (SPS).