Advances in Artificial Intelligence and Machine Learning; Research 4 (1) 1847-1865 Received 25-11-2023; Accepted 23-01-2024; Published 30-01-2024

Advancing Fingerprint Template Generation and Matching With Recast Minutiae Clustering and mRBFN

Diptadip Maiti diptadipmaiti@gmail.com
Department of CSE
Techno India University
Saltlake, Kolkata 700091, India

Madhuchhanda Basak mab.cse@brainwareuniversity.ac.in
Department of CSE
Brainware University
Barasat, Kolkata 700125, India

Debashis Das debashis.d@technoindiaeducation.com
Department of CSE
Techno India University
Saltlake, Kolkata 700091, India

Corresponding Author: Diptadip Maiti

Abstract

Rapid development of automation in the day-to-day life activity marks up the need of securing bio-metric template and the privacy of rightful owner. Minutiae-based matching is the most popular in the fingerprint recognition system, which greatly suffers from non-linear distortion like translation and rotation. To deal with linear distortion most of the technique proposed in the literature depends upon a reference or singular point. The paper proposes a binary template generation technique which applies an unsupervised clustering technique without fixing the number of clusters. Instead of position and orientation of the minutiae points the cardinality of the clusters are stored and converted into binary template. No spatial pattern information about the fingerprint is stored in the template to protect it from spoofing and information leakage. By the help of modified Radial Basis Function Network (mRBFN) with robust and efficient matching technique the generated templates are matched for authentication. We use MCYT dataset for training the mRBFN. The efficiency of the proposed scheme is evaluated on FVC 2000, FVC 2002 and FVC 2004 dataset.

Keywords: Template Generation, Secure Matching, Cardinality, Singular Point, Centroid Clustering, RBFN network.

Citation: Diptadip Maiti, et al. Advancing Fingerprint Template Generation and Matching With Recast Minutiae Clustering and mRBFN. Advances in Artificial Intelligence and Machine Learning. 2024;4(1):107.

1. INTRODUCTION

The Capability to reliably distinguish between an authorized person and an impostor for granting the access privilege with the help of physiological or behavioural characteristics is refers to as Biometric authentication [1]. The features like uniqueness, availability, permanence and easy installation make fingerprint-based authentication most extensively studied and most frequently deployed among the various commercially available biometric techniques like voice, face and iris etc. with a long history in forensic and criminal investigations [2]. With the help of some biometric traits, a Biometric Identity Management System (BIMS) has the capability to identify or to verify the claimed identity of an individual. A BIMS operate in two phases:

- **Enrolment of the Biometric:** Using a specific sensor the biometric is first acquired and specific features are extracted which are arranged in some specified format called biometric template [3]. Along with some extra information, the template is stored in a database for future verification or identification. The entire process is termed as Enrolment of Biometric [4].
- **Verification / Identification of Biometric:** In this process, a template is the first generated from the live biometric sample by following the similar steps as described in the enrolment phase [5]. The estimated template is then matched with the stored template database and the similarity score is reported [6].

Though biometric-based authentication provides usability over traditional password and token-based system, it raises several security and privacy concern [7]. The importance of such concerns is highlighted below:

1. Biometric is authentic but not secret.
2. Biometric cannot be revoked or cancelled.
3. If a biometric is lost once, it is compromised forever.
4. Cross-matching can be used to track individuals without their consent.

To protect biometric template databases from being compromised for supporting widespread use of biometric based authentication is an important research challenge and a critical step in the successful implementation of a BIMS [8]. Every effective BIMS must satisfy some basic security and privacy requirements:

- **Irreversibility:** It should not computationally be feasible to generate the original biometric templates from the supporting data or template stored in the database.
- **Unlinkability / Diversity:** The stored data should not be identical in different databases obtained from the same user.
- **Revocability:** Generation of new template from raw biometric data if the database gets compromised.
- **Matching Performance:** Overall performance and acceptability by a user.

Discriminating power and reliability makes minutiae-based matching most popular approaches in fingerprint matching [9]. It is experienced during such matching that in different regions of fingerprint, minutiae tend to have similar directions due to spatial closeness. Thus, minutiae have a tendency to form clusters [10]. Minutiae in similar regions have large local similarity and can be distinguished using local features [11].

The researchers have tried different techniques like fuzzy vault, fractional Fourier transform, symmetric hashing to implement template generation and matching. These techniques are very complex in nature, which requires large computational capabilities for real life implementation. More over the template generated by these methods is more that 1 KB, which makes them unsuitable for low

memory real life biometric systems. We try to device a approach which is simple in computation and generate a small memory occupying template with out loosing the security and matching accuracy.

In this article, we propose an unsupervised clustering method which divides the minutiae points into different clusters depending on a distance threshold without specifying the number of clusters. We also devise a robust method which to generate different length template from different fingerprints from the clustering information and store them in a database along with specific template labelling. For the effective use of memory and computational speed-up the maximum size of the template is restricted to a fixed bit length. To handle non-linear deformation during the live recognition we train a radial basis function neural network with two hidden layers, to correctly identify the label of the query fingerprint template. With the predicted template the query template is matched with different novel matching parameters and a final matching score is estimated. The proposed method also works efficiently even the raw fingerprints suffer from non-linear deformations.

The manuscript is organized as follows. The introductory note along with the literature survey is presented in Sec-1 and Sec-2 respectively. Sec-3 explains the proposed method in detail. Performance analysis of the proposed method along with relevant discussions are provided in Sec-4. Sec-5 draws the concluding remarks.

2. RELATED STUDIES

Following are some of the popular and important approach and technique proposed in the literature in past decade. Wong, et al. [12], generate a fixed length bit string with the help of kernel principal component analysis and state of the art binarization from unordered and variable size template generated from multi-line code algorithm from minutiae descriptors. Sandhya, et al. [13], proposed two method FS-INCIR and FS-AVGLO on delaunary triangulation net constructed from minutiae to generate a 3D array which will produce a fixed length 1D bit string. Wang, et al. [14], proposed to construct quantized pair minutiae vector from frequency samples of the binary string with a transmission channel finite impulse response to generate the template and blind system for identification. Kirchgasser, et al. [15], studied the impact of ghost fingerprint between a 4-year span separated datasets. Wang, et al. [16], applied partial Hadamard transform for template protection which preserve the stochastic distance after the transformation of template. Wang, et al. [17], used zoned minutiae pair and partial DFT to generate alignment free cancellable template. Sandhya, et al. [18], used on the other hand, fused local and distant structure with DFT to generate cancellable template. Sarkar, et al. [19], proposed a method to generate secret key from fingerprint image and used encryption to use it in two-way communication between sender an and receiver. Roy, et al. [20], proposed a technique to generate a synthetic master-print that can be used to match a large number of target prints. Covariance matrix adaptation evolution strategy, different evolution and particle swarm optimization are uses for synthetic master print generation. Kho, et al. [21], designed fingerprint template based on partial local structure descriptor and per mutated randomized non negative least square. Shahzad, et al. [22], used window shift XOR model and partial discrete wavelet transform to generate alignment free cancellable fingerprint template. Algarni, et al. [23], considered discrete Fourier transform, fractional Fourier transform, Discrete cosine transform and discrete wavelet transform with matrix rotation to generate fingerprint template. Trivedi, et al. [24], used Delaunary trigulation of modified minutiae pairs with user specific key for generation of template. Shukla, et al. [25], considered the maximum presence of five minutiae

points and combinations of ridge ending and bifurcation. For encoding they used Bose choudhuri Hocquenghem codes and SHA-256 hash mapping. Ajish, et al. [26], use modified symmetric hash functions which is a combination of salting and invertible transformation for generation of template. Baghel, et al. [27], applied filters on the genuine vault points from the combination of genuine and chaff points used in fuzzy vault technique. Principal Component analysis is used to align the fingerprints. Bedari, et al. [28], used dynamic random key with minutiae cylinder code generate binary feature vector based on randomly generated keys. The template is further filtered using block-based logic operation.

Over the last decade, there have been several developments in fingerprint template creation targeted at improving security and privacy. Kernel principal component analysis, Delaunay triangulation, and cryptographic key creation are some of the techniques that have been presented. Template protection methods, such as the partial Hadamard transform and zoned minutiae pairs, emphasize attempts to retain stochastic distances and assure cancellability. Addressing the influence of ghost fingerprints over time, developing synthetic master prints for matching, and building cancellable templates utilizing various transformations and optimization approaches are among the challenges. The field also investigates alignment-free approaches, which make use of certain minutiae spots and novel coding systems. The continual development of these approaches demonstrates the ongoing dedication to enhancing the security and reliability of fingerprint template creation in the face of new difficulties.

3. PROPOSED METHOD

The proposed method performs in three consecutive steps- (i) Fingerprint enhancement and minutiae detection, (ii) Template generation and (iii) Fingerprint matching. The first step has been performed by employing existing and reliable state-of-the-art algorithms. The second and third steps have been proposed and designed for providing an efficient matching in low-configured biometric systems.

3.1 Fingerprint Image Enhancement & Minutiae Detection:

The raw fingerprint image may not be in suitable form to feed directly for matching purpose. Hence, it needs a set of preprocessing steps to obtain the enhanced image. We have executed the following steps to enhance the raw image and to identify the minutiae points.

3.1.1 **Normalization**

To reduce the variation in the ridges and valleys of the input image, it is normalized with a pixel wise operation (NI) to a normalized Mean(M) and Variance(V) with the help of the following equation and desired mean(M_0) and variance(V_0) [29]. The effect of normalization is shown in FIGURE 1a and FIGURE 1b as original image and normalized image.

$$\boldsymbol{NI}(i,j) = \begin{cases} M_0 + \sqrt{\frac{V_0(Img(i,j)-M)^2}{V}} & \text{if } Img(i,j) > M \\ M_0 - \sqrt{\frac{V_0(Img(i,j)-M)^2}{V}} & \text{otherwise} \end{cases} \quad (1)$$

3.1.2 **Orientation image estimation**

Normalized image is divided into b x b (16 x 16) blocks and gradient ∂ at each pixel is calculated in direction x and y to find the local orientation of each block using the following formula [29], with FIGURE 1c shows the example:

$$O_x(i,j) = \sum_{p=i-\frac{b}{2}}^{i+\frac{b}{2}} \sum_{q=j-\frac{b}{2}}^{j+\frac{b}{2}} 2\partial_x(p,q)\partial_y(p,q) \tag{2}$$

$$O_y(i,j) = \sum_{p=i-\frac{b}{2}}^{i+\frac{b}{2}} \sum_{q=j-\frac{b}{2}}^{j+\frac{b}{2}} \left(\partial_x^2(p,q)\partial_y^2(p,q)\right) \tag{3}$$

$$\phi(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{O_y(i,j)}{O_x(i,j)}\right) \tag{4}$$

To remove corrupted ridge, valley structure and minutiae, the image is converted into a continuous vector filed to perform low pass filtering on the image with the help of the following equation:

$$\Phi_x(i,j) = \cos(2\phi(i,j)) \tag{5}$$

$$\Phi_y(i,j) = \sin(2\phi(i,j)) \tag{6}$$

The low pass filtering is performed on the vector field as:

$$\Phi'_x(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} F(u,v)\Phi_x(i-uw, j-vw) \tag{7}$$

$$\Phi'_y(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} F(u,v)\Phi_y(i-uw, j-vw), \tag{8}$$

The final local ridge orientation is estimated as:

$$O(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}\right) \tag{9}$$

3.1.3 **Reliability estimation**

The orientation reliability(r_{ij}) is estimated as the coherence of the orientation vectors in the window block. The coherence can be computed using the gradient method using the following formula [29], with FIGURE 1d, represents the example of the same:

$$r_{ij} = \mathrm{coh}(x,y) = \frac{\sqrt{\left(\partial_{xx} - \partial_{yy}\right)^2 + 4\partial_{xy}^2}}{\partial_{xx} + \partial_{yy}} \tag{10}$$

$$\mathrm{O(i,j)} \ -\mathrm{coh}(x,y) = \begin{cases} 255 & \text{if } r_{ij} \le \mathrm{Th} \\ \mathrm{O}\ (i,j) & \text{otherwise} \end{cases} \tag{11}$$

3.1.4 **Ridge frequency image**

Normalized image is divided into 16 x 16 block and ridge frequency is calculated using the following formulas [29], with FIGURE 1e, shows the example of the said operation:

$$X[m] = \frac{1}{W} \sum_{d=0}^{W-1} G(p,q), \quad m = 0, 1, \ldots, l-1 \tag{12}$$

$$p = i + \left(d - \frac{W}{2}\right) \cos O(i,j) + \left(k - \frac{1}{2}\right) \sin O(i,j) \tag{13}$$

$$q = j + \left(d - \frac{W}{2}\right) \sin O(i,j) + \left(\frac{1}{2} - k\right) \cos O(i,j) \tag{14}$$

3.1.5 **Detection of region of interest mask(ROI)**

Each block is classified as recoverable and unrecoverable based on three block features- amplitude, frequency and variance. The classification is done using one nearest neighbour (1NN) classifier [29], with Fig 1f represent the said operation.

3.1.6 **Gabor filtering**

Gabor filter is applied where the orientation of the filter is chosen from the ridge orientation and shape of the filter is chosen from frequency and wavelength of the ridges the image is enhanced. The modulation transfer function (MTF) of the Gabor filter can be represented as [29], with FIGURE 1g, shows the example of the filtering operation:

$$\begin{aligned} H(p,q:\phi,f) = {} & 2\pi\delta_x\delta_y \exp\left\{-\frac{1}{2}\left[\frac{(p_\phi - p_0)^2}{\delta_p^2} + \frac{(q_\phi - q_0)^2}{\delta_q^2}\right]\right\} + \\ & 2\pi\delta_x\delta_y \exp\left\{-\frac{1}{2}\left[\frac{(p_\phi + p_0)^2}{\delta_p^2} + \frac{(q_\phi + q_0)^2}{\delta_q^2}\right]\right\} \end{aligned} \tag{15}$$

where $p_\phi = p\cos\phi + q\sin\phi$, $q_\phi = -p\sin\phi + q\cos\phi$ and $p_0 = \frac{2\pi\cos\phi}{f}$. The final enhanced image is obtained by combining the Gabor filtered image with the region mask and reliability score which is shown in FIGURE 1h.

3.1.7 **Minutiae point detection:**

The enhanced is fed for minutiae detection after performing morphological binarization and skeletonization operations. Each minutiae point is represented as [30]:

$$m_i = (x_i, y_i, \theta_i, t_i)$$

where:x_i, y_i are coordinates of the minutiae point,θ_i is minutiae direction typically obtained from local ridge orientation, t_i is type of the minutiae point (ridge ending or ridge bifurcation). FIGURE 1i

shows the minutiae detected image. The detected Minutiae are classified into six different categories shown in FIGURE 3:

- Ending
- Bifurcation
- Fragment
- Enclosure
- Crossbar
- Other

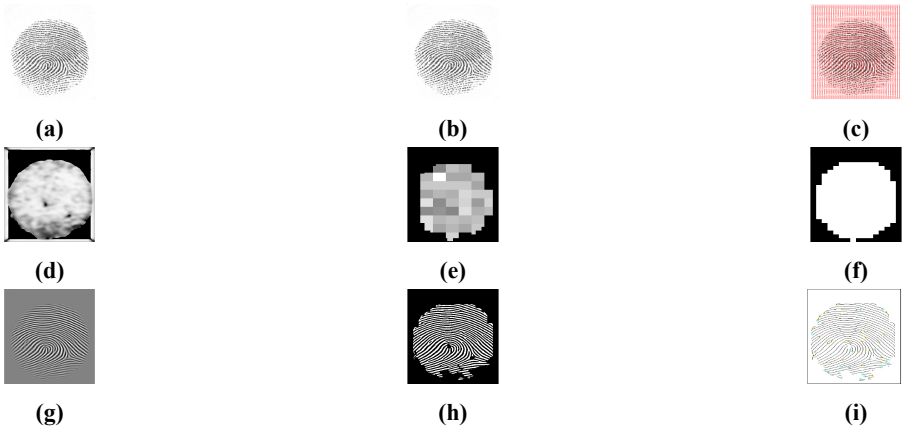


Figure 1: 1a: Original Image, 1b: Normalized Image, 1c: Ridge Orientation, 1d: Ridge Reliability Estimation, 1e: Ridge Frequency Calculation, 1f: Region of Interest Detection, 1g: Gabor Filtering, 1h: Enhanced Fingerprint Image, 1i: Minutiae Detected from Fingerprint Image.

FIGURE 2 shows the effectiveness of the image enhancement and minutiae detection technique over smudge and poor quality image.

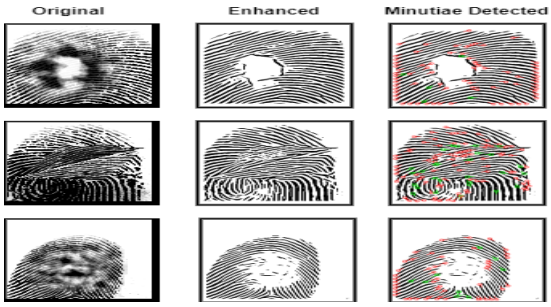


Figure 2: Result of Image Enhancement & Minutiae Detection.

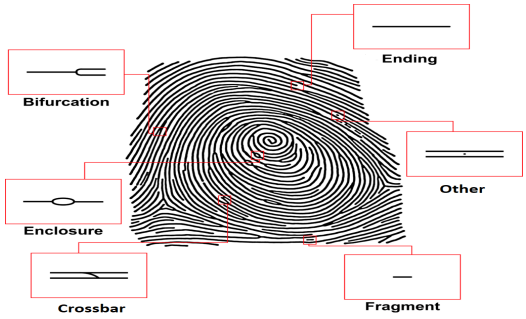


Figure 3: Classification of Different type of Minutiae Point.

3.2 Template Generation:

This step involves the representation of a fingerprint image into a corresponding binary template which assures a minimal storage requirement. The obtained minutiae points are clustered into several classes by applying centroid clustering which is subsequently converted into a binary sequence that contains only the clusters cardinality information instead of any positional or geographical information of minutiae points.

3.2.1 **Centroid clustering**

We have employed centroid clustering as it does not need to assume the cluster number from the beginning. This technique does not include all points in a specific cluster. It resulted with some singular points if its spatial positioning is different from others. The detailed steps are explained below:

Step 1: All the minutiae points are chosen as initial centroid.

Step 2: Two points are merged to form a cluster and a new centroid point is calculated depending on a specific Euclidean distance threshold obtained from experiment. Mathematically,

$$Pair_{k,l} = \{M_k, M_l\} \text{ if Distance } (M_k, M_l) \leq dist_t \tag{16}$$

Initial Centroid (c_x, c_y) estimation is performed as:

$$c_x = \frac{1}{n}\sum_{i=0}^{n-1} x_i \qquad c_y = \frac{1}{n}\sum_{i=0}^{n-1} y_i \tag{17}$$

Step 3: After formation of the initial cluster, the cluster are merged to form larger cluster depending on the distance between two centroids with in a specific range of Euclidean distance threshold.

$$\begin{aligned} \text{Cluster }_i = \{Pair_j, Pair_k\} \\ \text{if Distance } (\text{Cen } (P_j), \text{Cen } (P_k)) \leq d_t \end{aligned} \tag{18}$$

Step 4: Repeat step 3 until no further cluster merging is possible.

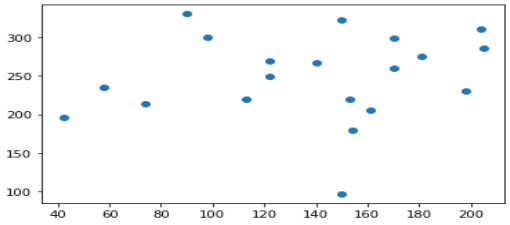


Figure 4: Minutiae point of a Fingerprint Image.

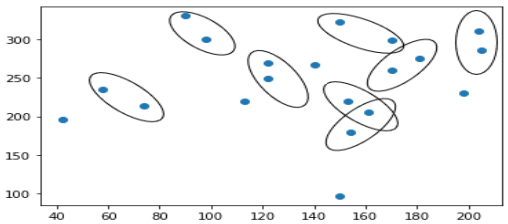


Figure 5: Minutiae Paring.

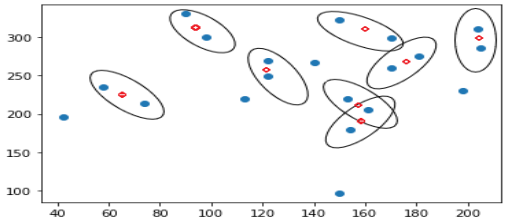


Figure 6: Centroid Calculation.

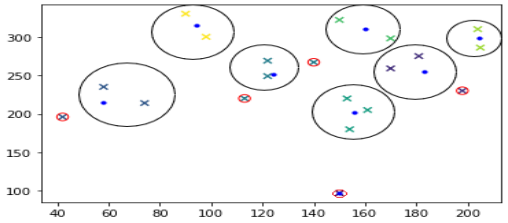


Figure 7: Final Clustering result.

3.2.2 **Binary template generation**

With the help of the clustering information, more specifically the cardinality values of the clusters, we generate an ordered sequence of the available information listed below-

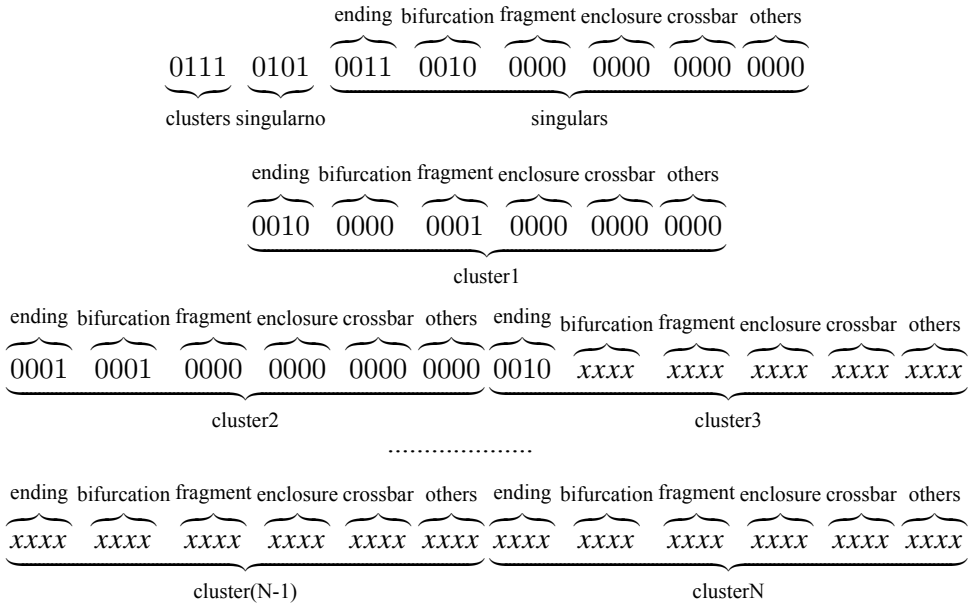
1. Number of Clusters
2. Number of Singular Points
3. Number of Singular Ending
4. Number of Singular Bifurcation
5. Number of Singular Fragment
6. Number of Singular Enclosure
7. Number of Singular Crossbar
8. Number of Singular Others
9. Number of Minutiae in Each Cluster
10. Number of Ending in Each Cluster
11. Number of Bifurcation in Each Cluster
12. Number of Fragment in Each Cluster
13. Number of Enclosure in Each Cluster
14. Number of Crossbar in Each Cluster
15. Number of Others in Each Cluster

Hence, The template is formed as below: $[\ll totalcluster \gg, \ll totalsingularpoints \gg, \ll singularending \gg, \ll singularbifurcation \gg, \ll singularfragment \gg, \ll singularenclosure \gg, \ll singularcrossbar \gg, \ll singularother \gg, \ll minutiaeno^1 \gg, \ll ending^1 \gg, \ll bifurcation^1 \gg, \ll fragment^1 \gg, \ll enclosure^1 \gg, \ll crossbar^1 \gg, \ll others^1 \gg, \ll minutiaeno^2 \gg, \ll ending^2 \gg, \ll bifurcation^2 \gg, \ll fragment^2 \gg, \ll enclosure^2 \gg, \ll crossbar^2 \gg, \ll others^2 \gg, ..., \ll minutiaeno^n \gg, \ll ending^n \gg, \ll bifurcation^n \gg, \ll fragment^n \gg, \ll enclosure^n \gg, \ll crossbar^n \gg, \ll others^n \gg]$

The 4-bit unsigned binary representation for each of the information is merged to generate a continuous bit stream results the binary template for a particular fingerprint.

Fingerprint Binary Template Cardinality=
[7, 5, 3, 2, 0, 0, 0, 0, 2, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 2, ...]

Fingerprint Binary Template=



The proposed method generates binary template that does not contain any spatial information like minutiae point location or minutiae point orientation. This approach stores sufficient information to perform matching using similarity index between two fingerprint templates. It is almost impossible to generate forge fingerprint from the information available in the binary fingerprint template.

3.3 Template Matching:

When a query fingerprint image is submitted to the system for authentication, it goes through all the steps for binary template generation. The matching process is further performed in two steps. In the first step a modified Radial Basis Function Network(mRBFN) is used to predict the closest level of the query fingerprint. In the next step the predicted template is matched with the query template using similarity measures to finally accept or reject the user. The main objective of using mRBFN is to handle the different linear distortions that is introduced during the image acquisition process of the query fingerprint.

3.3.1 Radial basis function

Radial basis functions are meant for approximate multivariate functions by linear combinations of terms base on a single variable function. The main advantage of this method is its applicability in almost any dimension without very little restriction on the prescribe data. It has a high accuracy or fast convergence to the approximated target functions when the data become dense. A Gaussian radial basis function can be expressed as

$$\varphi(r) = e^{-(\varepsilon r)^2} \tag{19}$$

3.3.2 Radial basis function network

A RBFN is a neural network that uses radial basis function as an activation. the network outputs the linear combination of radial basis function and neuron parameters. The output of the Network is a scalar function of the input vectors. The example of RBFN is shown in FIGURE 8. The output function can be mathematically represented as:

$$\varphi(\mathbf{x}) = \sum_{i=1}^{N} a_i \rho\left(\|\mathbf{x} - \mathbf{c}_i\|\right) \tag{20}$$

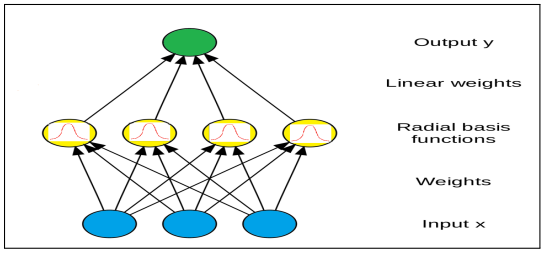


Figure 8: Traditional Radial Basis Function network.

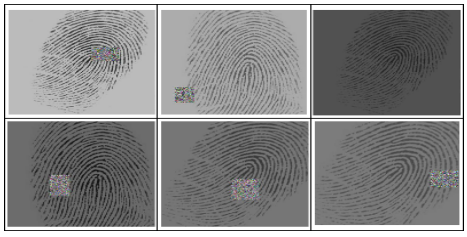


Figure 9: Training Image used in mRBFN.

3.3.3 **Modified radial basis function network**

To achieve higher accuracy, we have modified the traditional radial basis function network for handling linear deformation occurs in the acquisition process of the query image. We introduced two fully connected dense layers after radial basis function layer with random drop-out to prevent the model from over fitting. The structure of the model is shown below in the FIGURE 10. We have used rectifier linear unit(RELU) as the activation function in the hidden layer and categorical Softmax in the output layer. We have applied 20% random drop out in the Model.

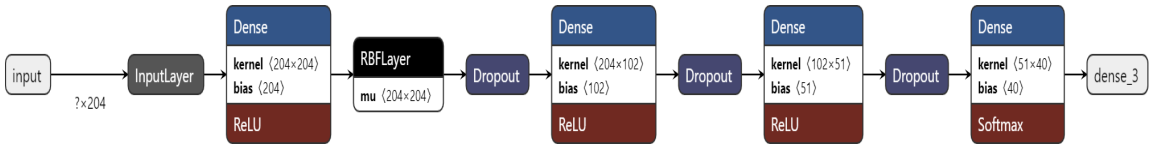


Figure 10: Proposed Modified Radial Basis Function network.

3.3.4 **Training and accuracy measurement of mRBFN**

To handle linear deformation during image acquisition process we have used image augmentation. We have generated a dataset from MCYT [31], with 20 degree of random rotation, random contrast, random zooming with random noise. Some samples of the training images are shown in FIGURE 9. The model accuracy and loss are shown the below FIGURE 11, and FIGURE 12 respectively. Average ROC curve and average precision and recall curve of the proposed model are also shown in the FIGURE 13, and FIGURE 14 respectively.

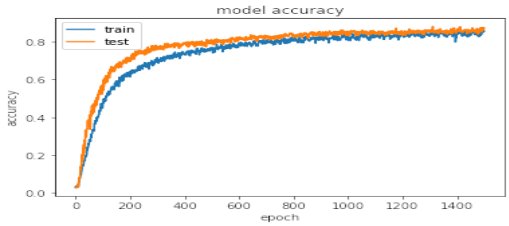


Figure 11: mRBFN Model Accuracy.

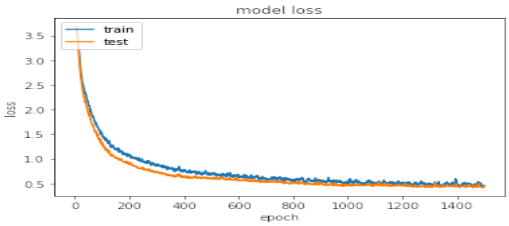


Figure 12: mRBFN Model Loss.

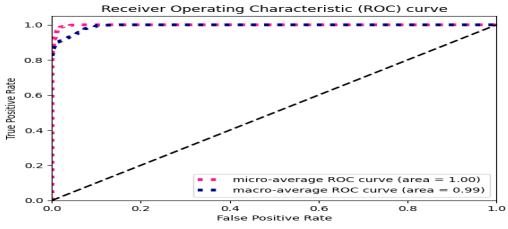


Figure 13: mRBFN Average ROC Curve.

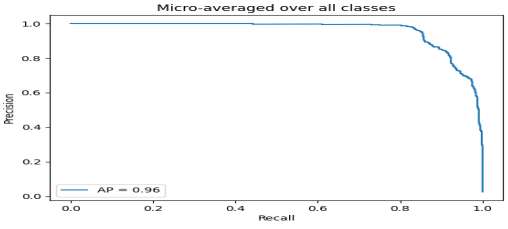


Figure 14: mRBFN Average Precision and Recall Curve.

The study compared classic Radial Basis Function Networks (RBFN) to a modified version, with significant variations in accuracy identified. The original RBFN had an accuracy rate of 82%, which represented its baseline performance. Following that, a modified RBFN was used, which resulted

in a huge improvement with an astounding accuracy of 96%. The improvements made to the classic RBFN design or training approach all led to the significant accuracy increase. TABLE 1 represents the comparative analysis of traditional RBFN with mRBFN.

Table 1: Comparison of Traditional RBFN with mRBFN.

Model	Accuracy	Macro average ROC	Micro average Precession-Recall
Traditional RBFN	87%	0.93	0.91
Modified RBFN	96%	0.99	0.96

3.3.5 **Matching score calculation**

The predicted match, obtained from mRBFN is now verified through the next round of matching. The query fingerprint template is matched further with the predicted stored template information. Three penalty scores are estimated by the following equations which are finally combined to take the decision.

* **Non-Match Penalty 1:** It is the normalized difference between total number of clusters between the query and stored template.

$$P_1 = \frac{\left|CN_{query} - CN_{store}\right|}{MAX\left(CN_{query}, CN_{store}\right)} \tag{21}$$

where CN is the cluster no and MAX find the maximum value between two values.

* **Non-Match Penalty 2:** It is computed as intra cluster variation between the query and store template. The Mathematical formulation is as below:

$$P_2 = \frac{1}{K}\sum_{i=0}^{K-1} C_q^i \tag{22}$$

where K is the number of clusters in the query image and C_q^i is the average minutiae difference with the stored template cluster which can be further defined as

$$C_q^i = \frac{1}{L}\sum_{j=0}^{L} \frac{\left|N_{query}^i - N_{store}^j\right|}{MAX\left(N_{query}^i, N_{store}^j\right)} \tag{23}$$

where N represent the cardinality of the cluster and L represent the number of clusters in stored fingerprint template.

* **Non-Match Penalty 3:** It computes the non-cluster singular points between stored and query template as follows:

$$P_3 = \frac{\begin{array}{c}\left|B_q - B_s\right| + \left|E_q - E_s\right| + \left|F_q - F_s\right| + \\ \left|En_q - En_s\right| + \left|C_q - C_s\right| + \left|O_q - O_s\right|\end{array}}{\begin{array}{c}MAX((B_q + E_q + F_q + En_q + C_q + O_q) \\ , (B_s + E_s + F_s + En_s + C_s + O_s))\end{array}} \tag{24}$$

where

$$B_q, E_q, F_q, En_q, C_q, O_q$$

and

$$B_s, E_s, F_s, En_s, C_s, O_s$$

are non-clustered bifurcation, ending, fragment, enclosure, crossbar and others type minutiae count of the query and stored fingerprint template respectively.

The final Matching Score is calculated through the following mathematical formulation:

$$Matching_{Score} = 1 - [(P_1 + P_2 + P_3)/3] \tag{25}$$

Now, if the matching score secures a predefined threshold, then it is accepted otherwise it is rejected.

$$\text{Decision} = \begin{cases} \text{accept if} \\ \text{reject otherwise} \end{cases} Matching_{Score} > Th \tag{26}$$

FIGURE 15, depicts the entire workflow of the proposed method.

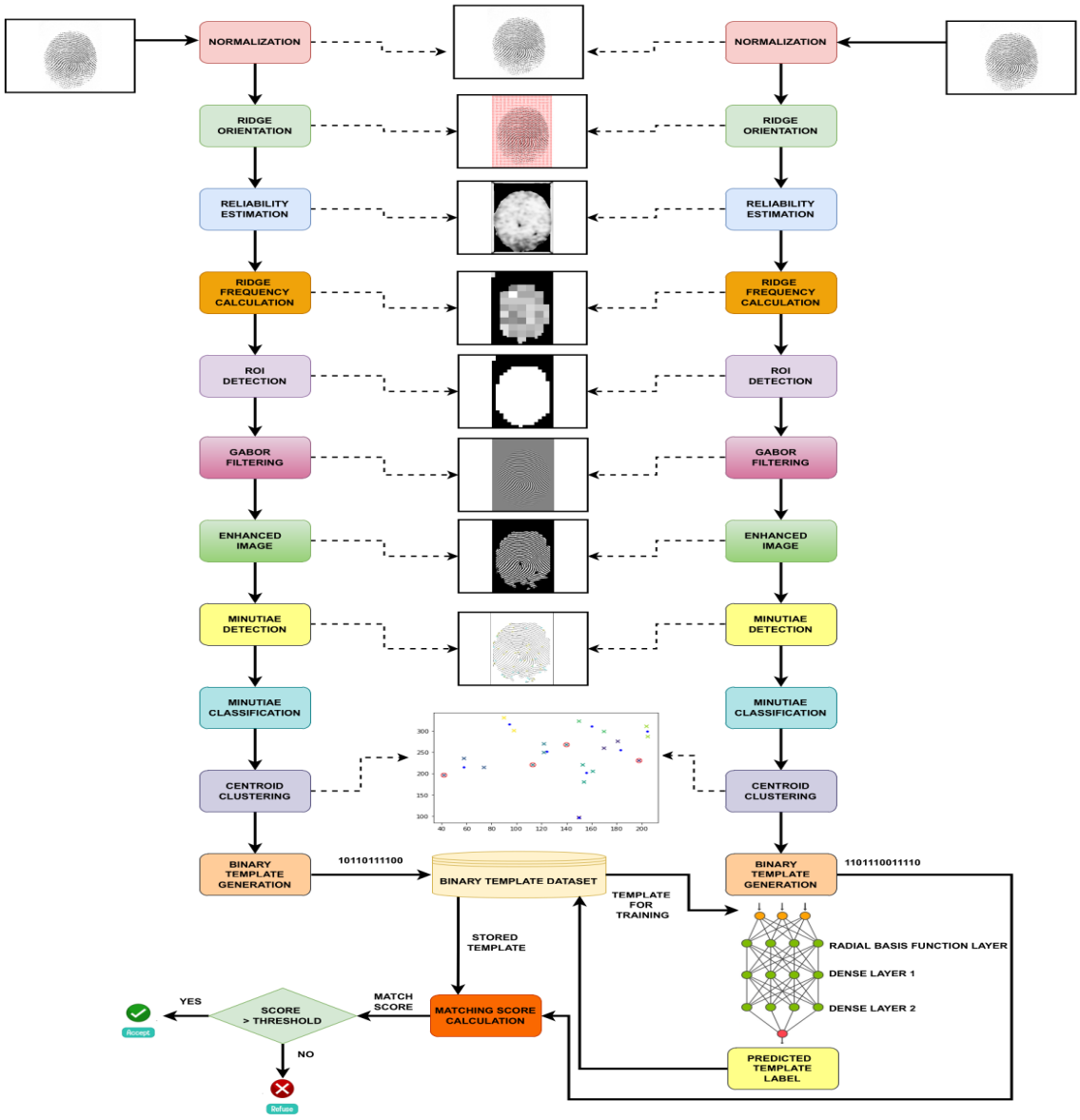


Figure 15: Template Generation and Matching Algorithm.

4. PERFORMANCE ANALYSIS AND DISCUSSION

The proposed method has been designed, implemented and tested on standard fingerprint databases namely-FVC2000 [32], FVC2002 [33], FVC2004 [34], and FVC2006 [35]. We have additionally applied different types of non-linear deformations on randomly selected fingerprints in the pre-processing phase. The modified datasets have been fed to the mRBFN network for training. Such operation helps to deal with the input fingerprint having several types of deformations during testing

phase. To evaluate the performance of the proposed algorithm, we have estimated Rank 1 accuracy and Equal Error Rate (EER). The experimented outcomes are tabulated in TABLE 2.

Table 2: Rank 1 Accuracy & EER of the proposed Algorithm on FVC2000, FVC2002, FVC2004, & FVC 2006 databases.

Database	Data Set	Rank 1 Accuracy	EER (%)
FVC2000	DB I	88.75	13.75
	DB2	93.75	11.25
	DB3	93.75	09.37
FVC2002	DB I	91.25	14.25
	DB2	88.75	6.06
	DB3	89.75	14.60
FVC2004	DB I	95.25	6.00
	DB2	95.25	5.37
	DB3	98.75	3.75
FVC2006	DB2	97.50	16.65
	DB3	94.16	3.60
	DB4	88.83	1248

Table 3: Computational Time of the Proposed Method for FVC2000, FVC2002, FVC2004, and FVC2006 databases.

Database	Dataset	Samples	Temp. Gen (Sec)	Temp. Matching (Sec)	Total (Sec)
FVC2000	DB1	80	96	8.39	104.39
	DB2	80	108	8.07	116.07
	DB3	80	196	8.85	204.85
FVC2002	DB1	80	100	8.21	108.21
	DB2	80	130	9.86	139.86
	DB3	80	61	8.41	69.41
FVC2004	DB1	80	109	8.35	117.35
	DB2	80	86	8.22	94.22
	DB3	80	199	8.42	207.42
FVC2006	DB2	120	269	19	288
	DB3	120	179	19.4	198.4
	DB4	120	69	17.6	86.6

The highlighted values are signifying the best result for any particular dataset. To show the response profile of the proposed algorithm, we have measured the computational time that requires to generate the template and for matching. The entire experiment has been executed on google colab with Intel Xeon processor having a clock speed of 2.30 GHz with a GPU having 0.82 GHz clock speed and 12 GB of RAM. The obtained responses are recorded in TABLE 3.

We have also made a comparison on FVC2004 dataset amongst the proposed method and some existing state-of-the-art techniques. The comparison is provided in TABLE 4. It is evident from TABLE 4, that the proposed method performs significantly well than few of the competitive methods. However method in [16], is producing better result compared to the proposed algorithm as it works in vector domain where all others methods including the proposed method works in spatial domain. Along with the computational efficiency in terms of performance and execution time, we have analysed the memory requirement to store the templates in the biometric device. It is experienced from the literature that minimum memory requirement for storing any particular fingerprint template is more than 1 Kb. However, the proposed method requires not more than 0.5 Kb to store a template

which is a significant factor for installing low memory biometric system. The memory requirement for the proposed algorithm can be estimated as:

$$\mathrm{Max}_{memory} = (32 + 24n) \tag{27}$$

where Max_{memory} is the maximum memory needed in bits and n is the number of clusters. The first 32 bits represent the total number of clusters (4 bit), total number of non-cluster minutiae (4 bit), number of non-clusters ending, bifurcation, fragment, enclosure, crossbar and other with 4 bits each. 24n represent the number of bits for cluster minutiae points. The total memory requirement now depends on the value of 'n' which should be approximated by considering the worst-case possibility. It is found from the experiments, we have done, that maximum number of clusters are being produced is 8. Hence 3 bits are sufficient to represent the number of cluster value. However, we have considered 4 bits for representing the number of clusters which is able to handle all extreme cases. Hence, maximum value for n should be $2^4 = 16$. Therefore, the maximum bit required to represent any fingerprint will be $(32 + 24 * 16) = 416$ bits which is less than $\frac{1}{2}$ kilobits.

Table 4: A Comparison of Proposed Method with Some Existing Methods on FVC2004 database.

METHODS	**EER (%)** (DB1)	**EER (%)** (DB2)	**EER (%)** (DB3)
Method in [36]	7.66	8.18	5.89
Method in [13]	11.89	12.71	17.60
Method in [16]	1.95	6.78	1.35
Method in [17]	-	9.01	-
Method in [21]	-	7.44	-
Method in [28]	8.89	7.63	-
Proposed	**6.00**	**5.37**	**3.75**

5. CONCLUSION

In the proposed algorithm, we have introduced a novel idea for fast, secure and robust fingerprint template matching which utilizes storage memory very efficiently. The template generation strategy is designed in such a way that it does not hold any spatial information like the position and orientation, category of the minutiae points. Such technique protects the template from spoofing. We have also considered the fingerprint matching challenges that occurs due to non-linear deformations. We have defined a modified radial basis function network that takes care of the problem arises due to translation, shifting, rotation and noise during image acquisition process. We have formulated a novel template matching technique along with a temporal threshold value which is able to take intelligent decision of fingerprint acceptance or rejection. FVC 2000, FVC 2002, FVC 2004 and FVC 2006 dataset are used to establish the computational efficiency and matching accuracy of the proposed technique.

The proposed solution is applicable in real-world biometric systems with limited storage space. It's interoperability with existing approaches opens the door to greater accuracy and security in biometric applications. To assess its robustness, the method's modified Radial Basis Function Network (mRBFN) may be extensively validated using templates created by proven methods. Furthermore, the suggested template generation approach shows potential for synergistic integration with other machine learning algorithms, allowing for a thorough assessment of its performance and adaptability in a variety of contexts. Essentially, this technique not only stands alone as a viable solution, but it also has the potential to improve the capabilities of larger, integrated biometric systems.

References

[1] Jain A, Uludag U. Hiding Biometric Data. IEEE Trans Pattern Anal Mach Intell. 2003;25:1494-1498.

[2] Zhu E, Yin J, Zhang G. Fingerprint Matching Based on Global Alignment of Multiple Reference Minutiae. Pattern Recognit. 2005;38:1685-1694.

[3] Feng Y, Feng J, Chen X, Song Z. A Novel Fingerprint Matching Scheme Based on Local Structure Compatibility. 18th International Conference On Pattern Recognition (ICPR'06). 2006;4:374-377.

[4] Marana AN, Jain AK. Ridge-Based Fingerprint Matching Using Hough Transform. XVIII Brazilian Symposium On Computer Graphics And Image Processing (SIBGRAPI'05). 2005:112-119.

[5] He Y, Tian J, Li L, Chen H, Yang X, et al. Fingerprint Matching Based on Global Comprehen-Sive Similarity. IEEE Trans Pattern Anal Mach Intell. 2006;28:850-862.

[6] Liang X, Asano T. Fingerprint Matching Using Minutia Polygons. 18th International Conference on Pattern Recognition (ICPR'06). 2006:1046-1049.

[7] Nandakumar K, Jain AK, Pankanti S. Fingerprint-Based Fuzzy Vault: Implementation and Performance. IEEE Trans Inf Forensics Sec. 2007;2:744-757.

[8] Ross A, Shah J, Jain AK. From Template to Image: Reconstructing Fingerprints From Minutiae Points. IEEE Trans Pattern Anal Mach Intell. 2007;29:544-560.

[9] Ratha NK, Chikkerur S, Connell JH, Bolle RM. Generating Cancelable Fingerprint Templates. IEEE Trans Pattern Anal Mach Intell. 2007;29:561-572.

[10] Tulyakov S, Farooq F, Mansukhani P, Govindaraju V. Symmetric Hash Functions for Secure Fingerprint Biometric Systems. Pattern Recognit Lett. 2007;28:2427-2436.

[11] Li Q, Guo M, Chang E. Fuzzy Extractors for Asymmetric Biometric Representations. IEEE Computer Society Conference On Computer Vision And Pattern Recognition Workshops. 2008: 1-6.

[12] Wong WJ, Teoh ABJ, Kho YH, Wong MLD. Kernel PCA Enabled Bit-String Representation for Minutiae-Based Cancellable Fingerprint Template. Pattern Recognit. 2016;51:197-208.

[13] Sandhya M, Prasad MVNK, Chillarige RR. Generating Cancellable Fingerprint Templates Based on Delaunay Triangle Feature Set Construction. IET Biom. 2016;5:131-139.

[14] Wang S, Hu J. A Blind System Identification Approach to Cancelable Fingerprint Templates. Pattern Recognit. 2016;54:14-22.

[15] Kirchgasser S, Uhl A. Fingerprint Template Ageing vs. Template Changes Revisited. International Conference Of The Biometrics Special Interest Group (BIOSIG). 2017:1-7.

[16] Wang S, Deng G, Hu J. A Partial Hadamard Transform Approach to the Design of Cancelable Fingerprint Templates Containing Binary Biometric Representations. Pattern Recognit. 2017;61:447-458.

[17] Wang S, Yang W, Hu J. Design of Alignment-Free Cancelable Fingerprint Templates With Zoned Minutia Pairs. Pattern Recognit. 2017;66:295-301.

[18] Sandhya M, Prasad MVNK. Securing Fingerprint Templates Using Fused Structures. IET Biomrics. 2017;6:173-182.

[19] Sarkar A, Singh BK. Cryptographic Key Generation From Cancelable Fingerprint Templates. 4th International Conference On Recent Advances In Information Technology (RAIT). 2018:1-6.

[20] Roy A, Memon N, Togelius J, Ross A. Evolutionary Methods for Generating Synthetic Masterprint Templates: Dictionary Attack in Fingerprint Recognition. International Conference On Biometrics. 2018:39-46.

[21] Kho JB, Kim J, Kim I, Teoh ABJ. Cancelable Fingerprint Template Design With Randomized Non-negative Least Squares. Pattern Recognit. 2019;91:245-260.

[22] Shahzad M, Wang S, Deng G, Yang W. Alignment-Free Cancelable Fingerprint Templates With Dual Protection. Pattern Recognit. 2021;111:107735.

[23] Algarni AD, El Banby G, Ismail S, El-Shafai W, El-Samie FEA, et al. Discrete Transforms and Matrix Rotation Based Cancelable Face and Fingerprint Recognition for Biometric Security Applications. Entropy (Basel). 2020;22:1361.

[24] Trivedi A, Thounaojam D, Pal S. Non-invertible Cancellable Fingerprint Template for Fingerprint Biometric. Comput Sec. 2020;90:101690.

[25] Shukla S, Patel SJ. Securing Fingerprint Templates by Enhanced Minutiae-Based Encoding Scheme in Fuzzy Commitment. IET Inf Sec. 2021;15:256-266.

[26] Ajish S, Kumar K. Security and Performance Enhancement of Fingerprint Biometric Template Using Symmetric Hashing. Comput Sec. 2020; 90:101714.

[27] Baghel V, Prakash S, Agrawal I. An Enhanced Fuzzy Vault to Secure the Fingerprint Templates. Multimedia Tool Appl. 2021;80:33055-33073.

[28] Bedari A, Wang S, Yang W. Design of Cancelable MCC-Based Fingerprint Templates Using Dyno-Key Model. Pattern Recognit. 2021;119:108074.

[29] Hong L, Wan Y, Jain A. Fingerprint Image Enhancement: Algorithm and Performance Evaluation. IEEE Trans Pattern Anal Mach Intell. 1998;20:777-789.

[30] Nguyen D, Cao K, Jain A. Robust Minutiae Extractor: Integrating Deep Networks and Fingerprint Domain Knowledge. International Conference On Biometrics (ICB). 2018:9-16.

[31] Ortega-Garcia J, Fierrez-Aguilar J, Simon D, Gonzalez J, Faundez-Zanuy M, et al. McYt Baseline Corpus: A Bimodal Biometric Database. IEE Proc Vis Image Signal Process. 2003;150:395-401.

[32] Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK, et al. FVC2000: Fingerprint Verification Competition. IEEE Trans Pattern Anal Mach Intell. 2002;24:402-12.

[33] Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK, et al. FVC2002: Second Fingerprint Verification Competition. International Conference on Pattern Recognition. 2002;3:811-814.

[34] Maio D, Maltoni D, Cappelli R, Wayman J, Jain A, et al. FVC2004: Third Fingerprint Verification Competition. Biometric authentication: First International Conference, ICBA 2004,Hong Kong, China, Jul 15-17, 2004. Proceedings. 2004:1-7.

[35] Cappelli R, Ferrara M, Franco A, Maltoni D. Fingerprint Verification Competition 2006. Biom Technol Today. 2007;15:7-9.

[36] Liu C, Cao J, Gao X, Fu X, Feng J, et al. A Novel Fingerprint Matching Algorithm Using Minutiae Phase Difference Feature. 18th IEEE International Conference On Image Processing. 2011: 3201-3204.