International Journal of Data Informatics and Intelligent Computing (IJDIIC)
Vol. 3, No. 1, March 2024, pp. 23~35
DOI: 10.59461/ijdiic.v3i1.95

Gaussian Proximal Hough Transformative Regularized Incremental Extreme Learning Machines for Palmprint Detection

N.Kohila[1], T.Ramprabha[2]
[1]PG & Research Department of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women (Autonomous), Tiruchengode, India
[2]Department of Information Technology, School of Computational Sciences, Nehru Arts and Science College, Coimbatore, India

Article Info

Article history:

Received January 12, 2024
Revised February 24, 2024
Accepted March 01, 2024

Keywords:

Palmprint detection
Machines Learning
Feature extractions
Sokal–Sneath similarity index
Outlier robust function

ABSTRACT

Palmprint detection is employed for identifying individuals based on the unique patterns present on the surface of their palms. Palmprint detection aims to be used in biometric authentication systems for security applications such as access control, forensic identification, and identity verification. Several research works have been developed for palmprint detection but have faced relative difficulty achieving higher accuracy. In this paper, Gaussian Proximal Hough Transformative Regularized Incremental Extreme Learning (GPHTRIEL) is developed with higher accuracy. First, palm images are collected from the dataset. Preprocessed images are provided as input to Outlier Regularized Incremental Extreme Learning Machines, consisting of three types of layers. The input layer receives preprocessed palm images. The first hidden layer performs image segmentation. Next, a set of geometric features is extracted in the second hidden layer and sent to the third hidden layer. Finally, feature matching is performed using the Sokal–Sneath similarity index function. With this, the outlier robust function correctly detects palmprints with a minimum error. An experiment is carried out with different factors. The analyzed research results indicate that the GPHTRIEL technique achieves improved performance in 6% accuracy, 11% sensitivity, and 10% specificity and minimizes 14% computation time compared to conventional methods.

Corresponding Author: N.Kohila (e-mail: padmeeshraj@gmail.com)

1. INTRODUCTION

Automatic authentication utilizing biometric characteristics is steadily rising in popularity. Biometrics involves studying techniques for uniquely identifying individuals based on one or more intrinsic physical or behavioural traits, including fingerprints, finger veins, irises, faces, palmprints, etc. Palmprint identification offers several advantages, including robustness, user-friendliness, high accuracy, and cost-effectiveness. Palmprints may contain more helpful information about humans than other behavioural traits. Many techniques have been developed in palmprint detection. However, an accurate and robust palmprint detection algorithm is a critical issue in automatic palmprint authentication systems.

A Joint Constrained Least-Square Regression (JCLSR) framework was introduced in [1] by using deep convolutional neural networks (DCNN) with the aim of palmprint recognition. Though the framework enhances the accuracy of palmprint recognition, the time consumption and error rate were not minimized effectively. The Palmprint Enhancement Network (PEN) was developed in [2] and aimed to achieve robust identification through feature matching by applying a deep learning model. However, it failed to create a robust minutiae matching algorithm to minimize the time-consuming palmprint matching process.

A multimodal palmprint biometric system was introduced in [3], integrating left and right palmprint images to achieve an optimal recognition rate. However, the system failed to consider a more significant

number of images in the dataset to evaluate the algorithm's robustness. A practical unimodal and multimodal biometric approach was introduced in [4] by employing deep learning and feature selection to recognize palmprint images rapidly. However, the processing time for palmprint recognition was high. A new palmprint identification algorithm utilizing Convolutional Sparse Coding was developed in [5] with minimal computational complexity. However, it was not efficient for processing a large number of training samples. A Convolutional Neural Network (CNN) was developed in [6] for hand segmentation to enhance contactless palmprint recognition. However, the performance of sensitivity in palmprint recognition was unaddressed.

The line feature local tri-directional patterns were developed in [7] for palmprint recognition with minimal computational complexity. However, the robustness of the model was not enhanced. A multimodal ultrasound recognition method was developed in [8] by integrating palmprint feature fusion. However, the technique lacks improvement in the accuracy of palmprint recognition. A Joint Pixel and Feature Alignment (JPFA) method was developed in [9] for palmprint recognition. The designed method minimizes the error rate, but the complexity of palmprint recognition was not reduced. An automatic palmprint alignment and classification system was developed [10] to enhance the accuracy significantly in palmprint verification.

The key contributions of the GPHTRIEL technique are listed below,

- The GPHTRIEL technique has been developed to enhance the accuracy of palmprint detection by incorporating several processes: segmentation, feature extraction, and classification into the Outlier Regularized Incremental Extreme Learning Machines.
- To minimize computational time, the Gaussian proximal connectedness graph segmentation is employed in the hidden layer of Outlier Regularized Incremental Extreme Learning Machines to extract the ROI from the image. The Generalized Gradient Hough Transform is applied for principal line extraction, and adaptive weighted harmonic thresholding is used for wrinkle extraction. Log-Gabor filters are employed for ridge extraction, minutiae points extraction, singular points extraction, and texture extraction, which are performed based on correlation.
- To enhance accuracy, a Sokal–Sneath similarity index has been introduced to compare the extracted feature vectors with the pre-stored feature vectors, thereby accurately detecting the palmprints.
- To improve sensitivity and specificity, the GPHTRIEL technique incorporates an outlier-robust model into an incremental extreme learning machine for accurate palmprint detection.
- To evaluate the performance of our GPHTRIEL technique, comprehensive experimentation is conducted and compared using various evaluation metrics.

1.1. Organization of paper

The paper is structured as follows: Section 2 reviews related works. In Section 3, the proposed GPHTRIEL technique is described in detail. Section 4 presents the experimental results along with a comprehensive quantitative analysis. Finally, Section 5 provides the paper's conclusion.

2. LITERATURE REVIEW

A robust features fusion methodology was introduced in [11] for palmprint recognition with minimal time consumption. A unique multistep fusion matcher was introduced in [12] to enhance recognition. However, the accuracy of recognition still faces significant challenges. A keypoint selection algorithm was introduced in [13] for palmprint recognition. However, an efficient machine-learning model was not utilized to enhance the palmprint recognition process. A heuristic palmprint recognition approach was designed in [14] to focus on extracting three types of palmprint features. However, this approach failed to explore additional hand-crafted features to enhance palmprint recognition performance further.

A multimodal biometric system was developed in [15] based on palmprint and fingerknuckle traits for palmprint recognition. A simultaneous heterogeneous palmprint feature learning and encoding model was developed in [16] for heterogeneous palmprint recognition. However, it failed to collect a new heterogeneous palmprint database for further enhancing palmprint recognition.

An integration of an autoencoder (AE) and a convolutional neural network (CNN) model was developed in [17] for palmprint authentication. However, addressing the adoption of deep learning networks at different stages of the palmprint authentication process, from ROI extraction to classification, remained unaddressed. A novel palmprint recognition model was developed in [18] using an adversarial domain approach. However, this model did not effectively reduce the complexity of palmprint recognition.

A deep generative architecture was designed in [19] to represent palmprint identification effectively. However, it failed to include ridge features for effective palmprint identification. An Extended

Binary Orientation Co-occurrence Vector (E-BOCV) was developed in [20] to enable cross-spectral palmprint recognition.

DL-basis framework was introduced in [21] with higher matching precision. Soft-shifted triplet loss was utilized to find contactless palmprint images. However, the palmprint detector's performance was not sufficient. The extracted features are extracted in [22] with the aid of CNN Transfer Learning to achieve maximum efficiency. ML classifier was developed to compute the similarity for one-to-one matching. However, the computational time was higher.

The novel network structure GLGAnet was discussed in [23] to combine CNN and Transformer. Feature selection method was introduced to pick the features. Deep convolutional layers were utilized for extracting local features. Next, a Transformer was employed to extract global features to minimize the dimensionality. However, the accuracy was not enhanced in palmprint recognition. Neural architecture search (NAS) technology was examined in [24] for palmprint vein recognition. However, the sensitivity was not considered. A correntropy-induced discriminative nonnegative sparse coding method was developed in [25] for accurate error detection. However, it failed to segment the palmprint images.

3. PROPOSAL METHODOLOGY

This section introduces a novel GPHTRIEL technique for accurate palmprint detection in biometric authentication with minimal time consumption. The method considers the challenges of precise palmprint detection in the biometric authentication system. The architecture of the proposed GPHTRIEL technique includes a structural framework that facilitates a complete workflow of various processes.

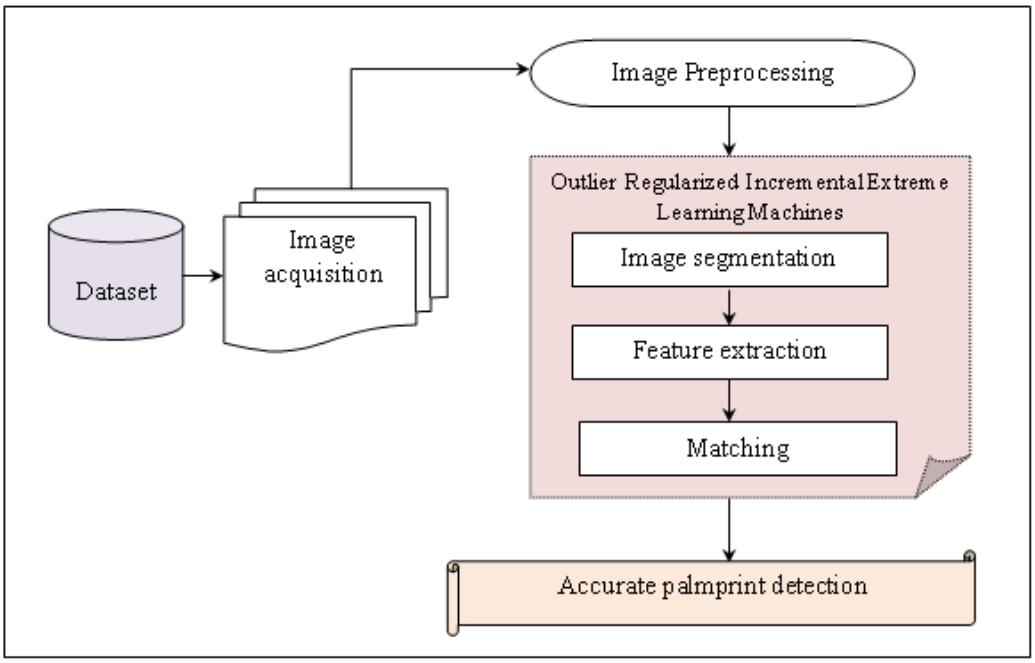

Figure 1. Architecture diagram of the proposed GPHTRIEL technique

Figure 1 depicts the architecture of the proposed GPHTRIEL technique for accurate palmprint detection. As shown in the figure, several palm images PI_1, PI_2, PI_3, …., PI_n are collected from the dataset. After the image collection, the proposed GPHTRIEL technique includes different processes such as image segmentation, extraction, and classification integrated into an outlier regularised incremental extreme learning machine. The preprocessed images are given as input to the extreme learning machine. First, image segmentation is performed to extract the ROI from the input image. Following image segmentation, the different features of the palms, such as principal lines, wrinkles, ridges, minutiae points, singular points, and texture features, are extracted from ROI. Finally, feature matching is performed using the extracted features and pre-stored templates of the given palm images for accurate palmprint detection. These processes of the proposed GPHTRIEL technique are described in the following subsections.

3.1. Image acquisition and preprocessing

The image acquisition and preprocessing are the fundamental steps in the proposed GPHTRIEL technique. Image acquisition involves capturing palm images from the Birjand University Mobile Palmprint

Database (BMPD). This dataset includes 1640 palm images from the left and right hands of 41 Iranian females. After collecting the image, image preprocessing involves improving their quality and enhancing features by removing the noisy artefacts from the input palm images.

3.2. Outlier regularised incremental extreme learning machines

Incremental Extreme Learning Machine is a feed-forward neural network having a single or multiple layers of hidden nodes for classification and feature learning. An incremental approach in this Extreme Learning Machine involves incorporating new hidden nodes when the new input arrives. The Outlier regularisation is proposed to minimize the error in the accurate classification. These models can produce better performance and faster training speed than backpropagation deep learning networks. Therefore, the proposed GPHTRIEL technique uses the Outlier regularised incremental extreme learning machines for accurate palmprint detection with minimum time.

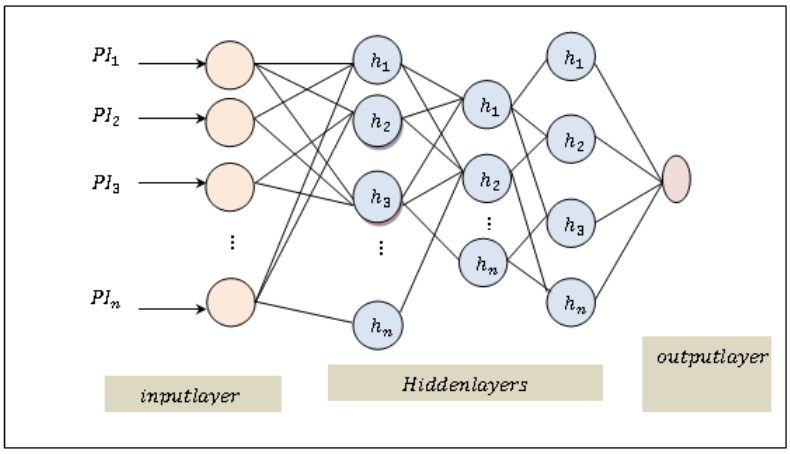

Figure 2. Constructions of Outlier regularised incremental extreme learning machines.

Figure 2 illustrates the constructions of Outlier regularised incremental extreme learning machines with a multiple layer of hidden nodes 'h_n'. The constructions of the Extreme Learning classifier include the input layer, three hidden layers, and an output layer. As shown in the above figure, let us consider that the training set $\{PI, Y\}$ where 'PI' denotes training palm images $\{PI_1, PI_2, PI_3, \ldots., PI_n\}$ and an output 'Y' represents its output of palmprint detection.

The input layer receives the preprocessed palm images but performs no calculations. The number of nodes or neurons in the input layer assigns the weights and the bias and remains fixed during the training phase. The activity of the neuron is represented as follows,

$$Z = \sum_{i=1}^{n}\sum_{j=1}^{m}(PI_i * \tau_j) + b_{ih} \quad (1)$$

Where Z indicates a neuron output, τ_j denotes weights between the input layer and hidden layer, preprocessed palm image 'PI_i'. Here, 'b_{ih}' indicates a bias. The input sample is transferred into the first hidden layer, where image segmentation is carried out by applying a Gaussian proximal connectedness graph segmentation.

It is the process of partitioning the images into different regions to extract meaningful information by connecting the similar pixel intensity of the image. Let us consider the Graph $G = \langle q_i, C\rangle$ where x_i denotes a pixel $q_1, q_2, q_3, \ldots q_m$ and 'C' indicates connectivity or connectedness between the pixels.

Therefore, the degree of connectivity between the pixels is measured by applying the graph segmentation algorithm given below,

$$C = max\{Q\ (q_{i,}q_{i+1}|i = 1,2, \ldots m)\} \quad (2)$$

$$Q = \exp\left(-\frac{dij^2}{2\sigma^2}\right) \quad (3)$$

$$dij = \frac{1}{m}\sum|q_i - q_{i+1}| \quad (4)$$

Where 'C' denotes connectivity or connectedness between the pixel intensity q_i and q_{i+1}, Q denotes a Gaussian function ranging from 0 to 1, dij denotes a Prevosti's distance between the pixels, σ $d$enotes a deviation, m denotes the number of pixels, $max\, m$ denotes a maximum value.

From the above (2)(3)(4), the maximum value of the 'Q' provides better proximal connectivity between the two pixels. In other words, the pixel intensity q_i and q_{i+1} are called adjacent. Therefore, the adjacent pixels are grouped to form a region. In this way, image segmentation is carried out, and the ROI is to minimize the time complexity of the palmprint detection. The extracted ROI is given to the next hidden layer for feature extraction. The second hidden layer extracts principal lines, wrinkles, ridges, minutiae points, singular points, and texture features.

*Principal lines***,** also known as primary or major lines, are essential features observed on the palm surface. It typically comprises three main lines: the heart line, the headline, and the lifeline. Each line represents different aspects of a person's personality and traits. These lines are extracted using the generalized gradient Hough transform.

Let us consider the pixel with coordinates (x_i, y_i). The principle lies are extracted as follows,

$$R = x\cos\theta + y\sin\theta \tag{5}$$

Where R indicates a distance from the origin to the closest point on the straight line, θ is an angle between the 'x' axis and the normal to the line. After calculating the distances, the peaks in the Hough collector array are determined to correspond to potential lines in the image. A Hough collector array 'A' is a two-dimensional array that stores information about the presence of particular line patterns 'R' in an input-segmented image.

$$A = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1n} \\ R_{21} & R_{22} & \dots & R_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ R_{m1} & R_{m2} & \dots & R_{mn} \end{bmatrix} \tag{6}$$

These peaks are detected using the gradient ascent method. The gradient ascent is a mathematical function used for finding a height peak through the local maximum of a function in the vertical direction.

$$G = \arg\max R \tag{7}$$

Where G denotes a gradient ascent function, $\arg\max R$ denotes an argument of a maximum function to find maximum peaks in an array 'A' for line detection. In this way, principles are detected.

Wrinkle: it refers to the fine lines gathered on the skin's surface, which are unique to each individual for identification or verification. The adaptive weighted harmonic thresholding technique is employed for wrinkle extraction.

Let us consider pixels $q_1, q_2, q_3, \dots q_m$ in an image. The local mean intensity of pixels is computed using weighted harmonic mean as follows,

$$\mu_x = \left(\frac{\sum_{i=1}^{m} \varphi_i {q_i}^{-1}}{\sum_{i=1}^{m} \varphi_i}\right)^{-1} \tag{8}$$

Where μ_x denotes a weighted harmonic mean, φ_i denotes a weight assigned to pixels 'q'. Then, define the threshold value and extract the wrinkle region.

$$Z = \begin{cases} \mu_x > T, & 1 \\ \mu_x < T, & 0 \end{cases} \tag{9}$$

Where Z denotes an output of wrinkle region extraction, μ_x denotes the weighted harmonic mean, $and\ T$ denotes a threshold. The intensity of the pixel means 'μ_x' is greater than the local threshold 'T', then the wrinkle region labelled as '1' is extracted. Otherwise, the output Z is labelled as '0'.

Ridges: The Log-Gabor filter is a type of filter commonly used for ridge extraction in palmprint images. The Log-Gabor filter function G(x, y) in the spatial domain is defined as:

$$\mathrm{G(x,y)} = \exp\left(-0.5 * \frac{\left(log\left(\frac{x^2+y^2}{{f_0}^2}\right)\right)^2}{v^2}\right) \tag{10}$$

Where G(x, y) denotes a Log-Gabor filter function, x and y are spatial coordinates of pixels, f_0 indicates a central frequency parameter, and v indicates controls the bandwidth of the filter. To enhance ridge structures in a palmprint image using the Log-Gabor filter, the input image is convolved with the Log-Gabor filter function in the spatial domain as follows,

$$F = G(x, y) * PI\ (x, y) \tag{11}$$

Where F denotes an outcome, $G(x, y)$ denotes a Gabor filter function, and $PI\ (x, y)$ denotes *the* input image.

Minutiae points: Identify candidate minutiae points by locating ridge endings (terminations) and ridge bifurcations. Ridge Endings is a point where ridges terminate immediately. Ridge bifurcations are also points where a ridge splits into two branches.

Let us consider the given ridge pixel at the coordinates (x_1, y_1) and define a local neighbourhood ridge pixel coordinate (x_2, y_2). The average distance between these two pixels is calculated as follows,

$$AV_D = \frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{m} D_{ij} \tag{12}$$

Where AV_D is the average distance between these two ridge pixels, N denotes the total number of neighbourhood ridge pixels, $and\ D_{ij}$ denotes the distance of each pixel to the nearest background pixel. Therefore, the Ridge Ending is determined by setting the threshold value for the average distance,

$$W = \begin{cases} AV_D > \beta, & Ridge\ end \\ AV_D < \beta, & not - Ridge\ end \end{cases} \tag{13}$$

If the average distance is below a predefined threshold 'β', the pixel at coordinates is classified as a ridge ending.

Ridge Bifurcations: If the pixel has more than two ridge neighbours' pixels within a certain distance threshold 'β', classify it as a ridge bifurcation.

$$H = \begin{cases} q_j > 2\ , & ridgebifurcation \\ otherwise, & not - ridgebifurcation \end{cases} \tag{14}$$

Where H denotes an output function of Bifurcations**,** q_j denotes a neighbouring pixel.

Singular point extraction: Singular points, including core points from palm images for recognition. If the pixel has multiple ridges neighbouring pixels within a certain distance threshold, classify it as a core point.

Texture feature: it is measured to provide the spatial correlation between the pixels' intensities.

$$TF = \frac{1}{\delta_i * \delta_j}\sum_i \sum_j (q_i - \mu_i)(q_j - \mu_j) \tag{15}$$

Where'TF' indicates the texture feature, q_i denotes a pixel, q_j denotes a neighbouring pixel, $and\ \mu_i$ and μ_j indicate a mean of the pixels and neighbouring pixels, respectively.

Finally, the feature vector 'SFV' is formed by combining all features in a single vector as follows,

$$SFV = \sum_{k=1}^{M}[F_k] \tag{16}$$

$$F_k = \left[PL_j, L_j, MP_j, SP_j, TF_j\right] \tag{17}$$

Where $PL_j,\ Lj, MP_j,\ SP_j, and\ TF_j$ denote Principal lines, wrinkles, ridges, minutiae points, singular points, and texture features, respectively

Finally, the extracted feature vectors are sent to the third hidden layer, where the classification is performed through the Sokal–Sneath similarity index function. It is a statistical method used to find the similarity between the extracted features and the pre-stored template features vector.

$$SS = \frac{[PFV \cap SFV]}{[PFV \cup SFV] + [PFV \Delta SFV]} \tag{18}$$

Where 'SS' denotes a Sokal–Sneath similarity index, PFV denotes a set of 'extracted feature vectors, SFV indicates stored pre-stored templates features vector, $PFV \cap SFV$ denotes a mutual

dependence between two features vectors, $PFV \Delta SFV$ denotes a variation between the two features vectors, $PFV \cup SFV$ represents the mutual independence between two features vectors. The coefficient (SS) provides the output ranges between 0 and 1.

$$SS = \begin{cases} 1; & feature\ matched \\ 0; & feature\ not\ matched \end{cases} \quad (19)$$

As a result, the palmprint is detected at the final output layer. The output of the hidden layer is a linear combination of different functions, as given below,

$$h = \sum_{i=1}^{n} a(\ PI_i * \tau_{ho} + b_{ho}) \quad (20)$$

Where h' represents the hidden layer output, 'a' indicates a sigmoid activation function of hidden neurons, 'τ_{ho}' denotes the weight between the hidden and output layer neurons, and b_{ho} bias between the hidden and output layers.

$$a = (1 + \exp(SS))^{-1} \quad (21)$$

The outlier robust function aims to minimize the error as follows,

$$OT\ = \arg\min |L^2| + \frac{1}{r}|\tau_o|^2 \quad (22)$$

Where OT denotes an outlier robust function, $arg\ min$ denotes an argument of the minimum function, L indicates an error function,r indicates the regularization parameter,τ_o indicates the weight of the output. The regularization parameter is used in extreme learning machines to control the balance between fitting the training data well and preventing overfitting. Overfitting means that the model failed to fit additional data. By using this strategy, accurate palmprint detection is achieved, resulting in minimized time consumption. The pseudo-code for the proposed GPHTRIEL is described as follows,

Algorithm 1: Gaussian Proximal Hough Transformative Regularized Incremental Extreme Learning

Input: Dataset, Number of preprocessed palm images$PI_1, PI_2, PI_3, \ldots, PI_n$,
Output: Increase accuracy of palmprint detection

Begin
1. **Collect** the number of preprocessed palm images$PI_1, PI_2, PI_3, \ldots, PI_n$, -**input layer**
2. **for each** input palm image—**hidden layer 1**
3. Measure the degree of connectivity using (2) (3) (4)
4. Find Adjacent pixels
5. Segment the ROI from the image
6. **end for**
7. **for each ROI -------hidden layer 2**
8. ApplyGeneralized gradient Hough transform to extract principle lines using (5) (6) (7)
9. Apply adaptive weighted harmonic thresholding to Extract-Wrinkle using (8) (9)
10. Apply Log-Gabor filter to extract Ridges using (10) (11)
11. Extract Minutiae points using (12) (13)
12. Extract Ridge Bifurcations using (14)
13. **if the pixel** has multiple ridge neighbouring pixels
14. Extract **core points**
15. **end if**
16. Extract texture feature using (15)
17. Combine features into vectors using (16) (17)
18. f**or extracted feature vector '**SFV**'-------hidden layer 3**
19. **for pre-stored texture**'PFV'
20. Measure the Sokal–Sneath similarity index function using (18)
21 i**f** ($SS = 1$) **then**
22. features are matched
23. **else**
24. features are not matched
25. end if

26. **for each output**
27: Apply the Outlier robust function using (22) to minimize the error
28: **end for**
29: Obtain the accurate detection results -- **output layer**
End

Algorithm 1 describes the steps involved in Gaussian Proximal Segmentive Outlier Regularized Incremental Extreme Learning Machines for palmprint detection. Initially, the algorithm collects a number of preprocessed palm images and provides them to the input layer. Subsequently, ROI segmentation is performed in the first hidden layer by measuring the degree of connectivity between pixel intensities, resulting in reduced palmprint detection time. Following this, multiple features such as principal lines, wrinkles, ridges, minutiae points, singular points, and texture features are extracted to form a vector. Finally, the feature vectors are correlated with the pre-stored vector by applying the Sokal–Sneath similarity index. Palmprint detection is performed based on the similarity index. The outlier robust function minimizes the error rate, resulting in improved sensitivity and specificity.

4. EXPERIMENTATION

4.1. Experimental setup

In this section, the GPHTRIEL technique, along with two existing methods, JCLSR [1] and PEN [2], is implemented using MATLAB coding. The evaluation is performed using the Birjand University Mobile Palmprint Database (BMPD) taken from https://www.kaggle.com/datasets/mahdieizadpanah/birjand-university-mobile-palmprint-databasebmpd. The Palmprint Database consists of 1640 images obtained from both the left and right hands of 41 Iranian females during two separate sessions, with a two-week interval between sessions. During the initial session, participants were directed to position their hands against a black background. Following this, six images of each palm were captured from a distance of 20 cm in an open environment. For experimental purposes, 100 to 1000 images are selected from the database.

5. COMPARATIVE RESULT AND DISCUSSION

This section presents a comparative analysis of the GPHTRIEL technique, along with two existing methods, JCLSR [1] and PEN [2]. The performance analysis employs metrics such as Palmprint detection accuracy, sensitivity, specificity and computational time. The performance of each technique in terms of these metrics is illustrated through tables and graphical representations.

Palmprint detection accuracy: It refers to the ratio of correctly detected palmprint images to the total number of palmprint images. It is typically calculated using the following formula:

$$PDA = \sum_{i=1}^{n} \left(\frac{Correctly\ detected\ palmprint\ images}{PI_i} \right) * 100 \quad (23)$$

Where PDA indicates a palmprint detection accuracy, PI denotes a palmprint image. Accuracy is measured in terms of percentage (%).

Sensitivity: Also known as the True Positive Rate (TPR) or Recall, measures the proportion of actual positive cases that are correctly identified by a palmprint detection model. Mathematically, sensitivity is calculated using the following formula,

$$SN = \frac{Tp}{Tp+Fn} \quad (24)$$

Where SN denotes a sensitivity, Tp denotes a true positive that the images are correctly detected as matched with the stored template of the same individual, $and\ Fn$ indicates a false negative referring to palmprint images incorrectly detected as not matched but are actually matched.

Specificity: In palmprint detection, specificity refers to the ability of the technique to correctly detect whether palmprint images belong to the same individual or not.

$$SP = \frac{Tn}{Tn+Fp} \quad (25)$$

Where SP denotes a specificity, Tn denotes a true negative that the images that are detected as matched but are actually not matched, and fp indicates false positives, which refer to matched images that are incorrectly detected as not matched.

Computational time: It is measured as the amount of time consumed by the technique for pamprint detection. The overall time consumption is formulated as follows,

$$Comp\_T = \sum_{i=1}^{n} PI_i * T \ \ [DPI] \tag{26}$$

Where $Comp\_T$ indicates the computational time, T indicates a time, and DPI indicates detecting the single palmprint image (DPI). The overall computational time of palmprint detection is measured in terms of milliseconds (ms).

Table 1. Comparison of palmprint detection accuracy

Number of images	Palmprint detection accuracy (%)		
	Proposed GPHTRIEL	Existing JCLSR	Existing PEN
100	95	90	91
200	95.5	87.5	90
300	93.33	85	88.33
400	94.5	88.75	90
500	94	86	88
600	95	90.33	92
700	93.57	87.85	89.71
800	94.75	89.37	91.25
900	94.44	89.44	91.11
1000	93.3	86.5	88.5

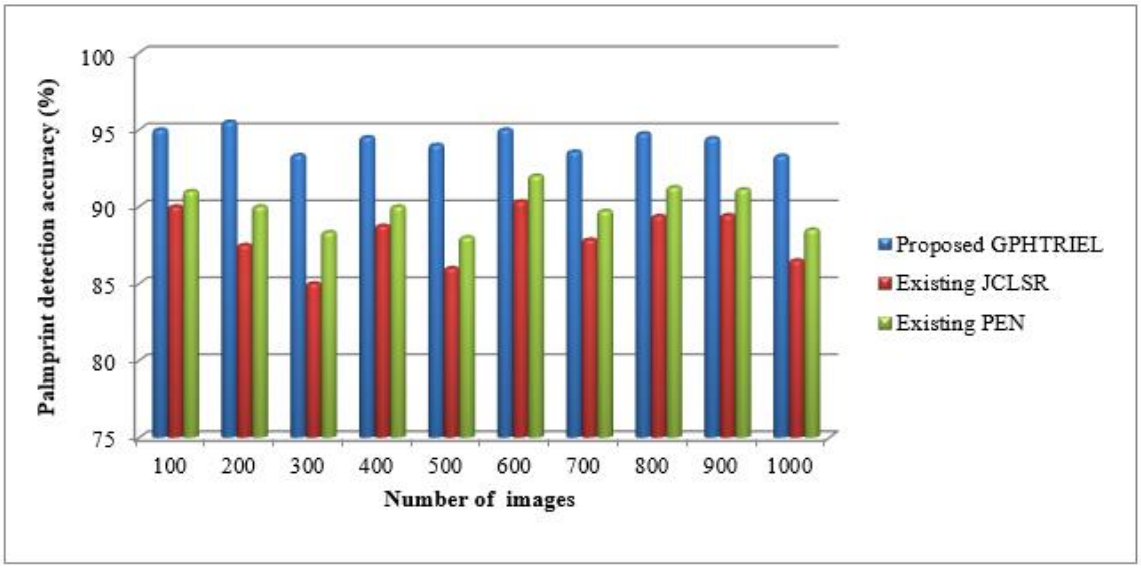

Figure 3. Performance results of accuracy

Figure 3 depicts a graphical illustration of palmprint detection accuracy versus the number of images, ranging from 100 to 1000. The numbers of palm images are taken on the horizontal axis, and the accuracy is observed on the vertical axis. The results graphically illustrate that the proposed GPHTRIEL technique attains higher palmprint detection accuracy compared to existing methods [1] and [2]. For each method, ten various results were observed with varying numbers of input images. The observed results demonstrate that the GPHTRIEL technique outperforms other deep learning models. Considering the first iteration involving 100 images, the palmprint detection accuracy using the GPHTRIEL technique was observed to be 95%. Subsequently, 90% and 91% palmprint detection accuracy were observed by applying [1] and [2], respectively. Multiple iterations were performed for each method, and the overall outcomes were compared. The average of ten comparison results indicates that the GPHTRIEL technique increased the palmprint detection accuracy by 7% when compared to [1] and 5% when compared to [2]. This is because the

GPHTRIEL technique utilizes an Outlier Regularized Incremental Extreme Learning Machines for estimating the features using the Sokal–Sneath similarity index function. Based on the similarity measure, matched palmprints and unmatched ones are correctly detected with the minimum error by applying the outlier robust function.

Table 2. Comparison of Sensitivity

Number of images	Sensitivity		
	Proposed GPHTRIEL	Existing JCLSR	Existing PEN
100	0.931	0.809	0.852
200	0.952	0.820	0.862
300	0.944	0.845	0.874
400	0.936	0.865	0.895
500	0.952	0.852	0.882
600	0.937	0.836	0.863
700	0.942	0.822	0.842
800	0.940	0.836	0.866
900	0.952	0.854	0.885
1000	0.937	0.833	0.855

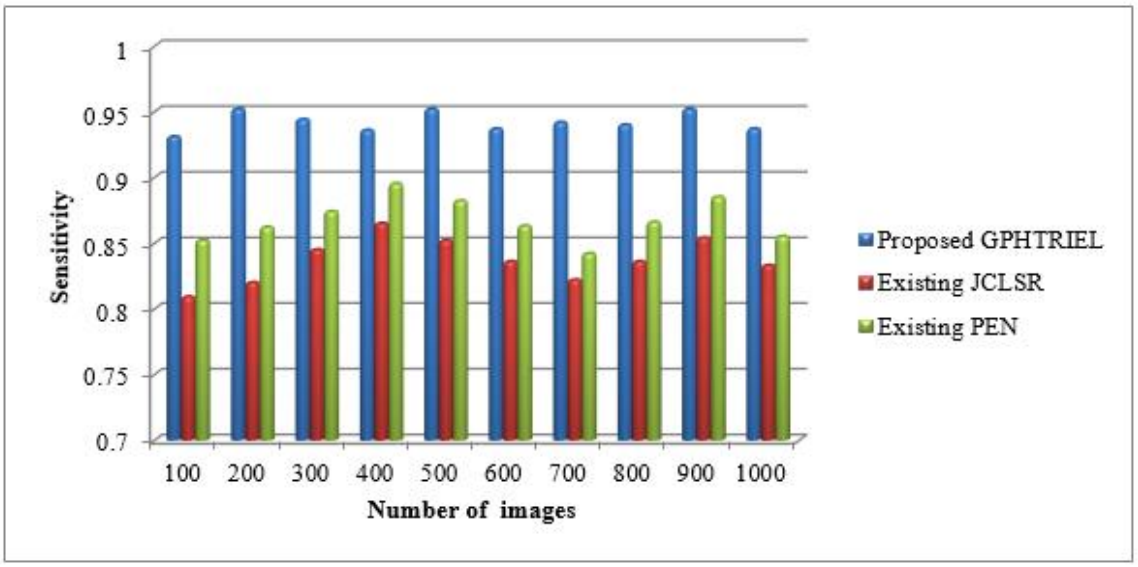

Figure 4. Performance results of sensitivity

Figure 4 depicts a graphical representation of sensitivity versus the number of images, ranging from 100 to 1000. The results illustrate that the proposed GPHTRIEL technique achieves higher sensitivity compared to existing methods [1] and [2]. For each method, ten different results were observed with varying numbers of images. The observed results demonstrate that the GPHTRIEL technique outperforms other methods. From the comparison analysis, the overall performance of sensitivity using the GPHTRIEL technique is improved by 13% and 9% compared to [1] and [2], respectively. This improvement is achieved by incorporating an outlier-robust function into the incremental extreme learning machines. This function minimizes the deviation between the expected and actual palmprint detection outcomes through the regularization parameter, resulting in increased true positives and minimized false negatives.

Table 3. Comparison of Specificity

Number of images	Specificity		
	Proposed GPHTRIEL	Existing JCLSR	Existing PEN
100	0.740	0.675	0.687
200	0.785	0.680	0.725
300	0.823	0.7	0.778
400	0.855	0.695	0.795
500	0.796	0.705	0.752
600	0.822	0.755	0.8
700	0.805	0.698	0.768
800	0.765	0.705	0.744
900	0.804	0.711	0.755
1000	0.799	0.675	0.71

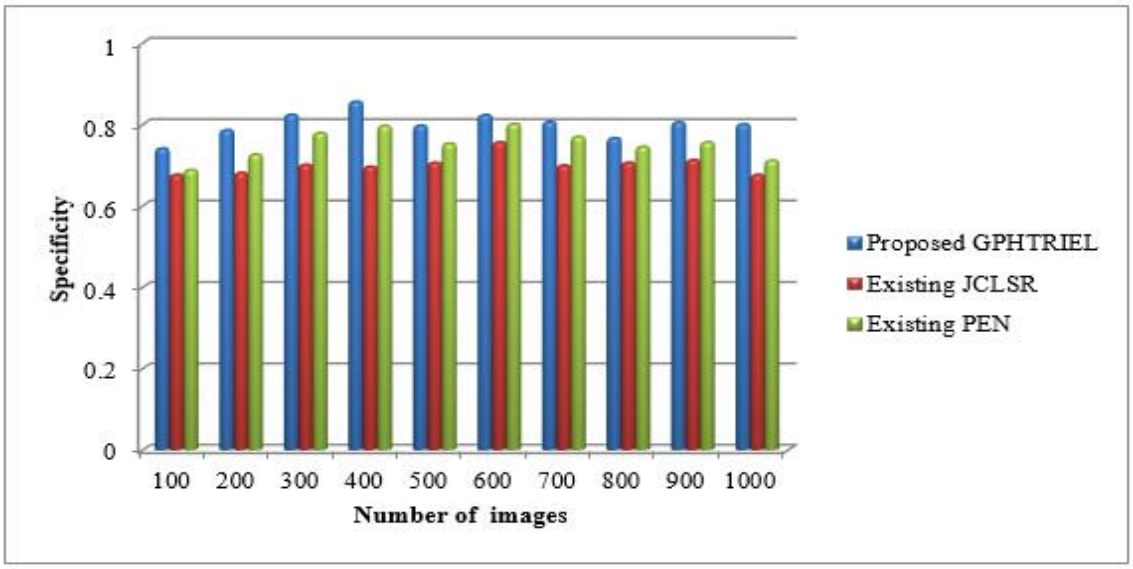

Figure 5. Performance results of specificity

Figure 5 illustrates the performance analysis of specificity by applying three methods, namely the GPHTRIEL technique, JCLSR [1], and PEN [2], versus the number of palm images collected from the dataset. As depicted in Figure 5, the specificity performance is shown to be relatively higher using the GPHTRIEL technique compared to [1] and [2]. The comparison results reveal that the specificity of the proposed GPHTRIEL technique is improved by 14% and 6% compared to [1] and [2], respectively. This improvement is achieved through the application of Outlier Regularized Incremental Extreme Learning Machines, which accurately detect matched and unmatched palm images through feature vector analysis using Sokal–Sneath similarity.

Table 4. Comparison of computation time

Number of images	Computational time (ms)		
	Proposed GPHTRIEL	Existing JCLSR	Existing PEN
100	33	45	42
200	42	52	48
300	48	60	57
400	56	68	62
500	60	70	67.5
600	66	78	75
700	70	80.5	79.1
800	74.4	84	81.6
900	79.2	88.2	86.4
1000	82	92	90

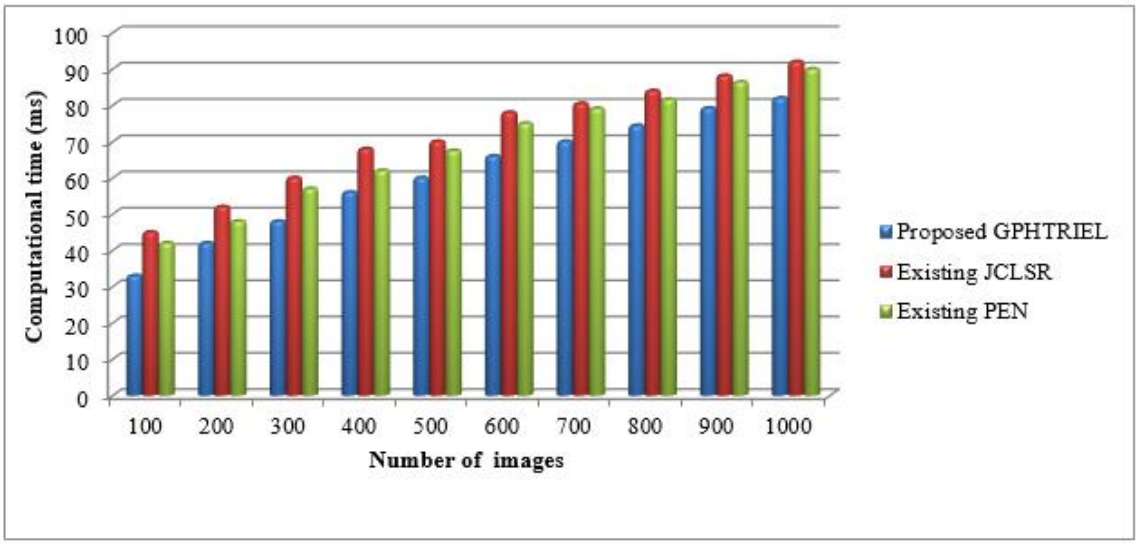

Figure 6. Performance results of Computational time

Figure 6 presents the performance results of computational time for palmprint detection using three methods, namely the GPHTRIEL technique, JCLSR [1], and PEN [2]. The figure indicates that computational time increases as the number of images taken from the datasets rises from 100 to 1000. This increase is due to the larger number of images involved during palmprint detection, resulting in a significant amount of time being consumed. However, experiments with 100 images show that the time consumed for palmprint detection was found to be 33 ms. In comparison, the overall time for [1] and [2] was found to be 45 ms and 42 ms, respectively. This result implies a significant reduction in computational time by 16% and 12% compared to [1] and [2], respectively. The improvement is achieved by performing ROI segmentation and feature extraction. In ROI segmentation, the Gaussian proximal connectedness graph method is employed to measure the degree of connectivity between pixels. After that, similar pixels are grouped to extract the ROI from the input image. Subsequently, a set of geometric features, including principal lines, wrinkles, ridges, minutiae points, singular points, and texture features, are extracted. With these selected features, accurate palmprint detection is achieved with minimal time consumption.

6. CONCLUSION

Palmprint detection involves identifying and analyzing the distinct patterns found on an individual's palm for robust biometric applications. This paper proposes a GPHTRIEL technique for robust palmprint detection with higher accuracy and minimal time consumption. The GPHTRIEL technique collects input images. First, segmentation of the ROI is performed to minimize the palmprint detection time. Following this, geometric methods are employed to extract different features from the palm images. With the extracted feature vector, palmprint detection is performed through feature matching with higher accuracy. Quantitative performance research results indicate that the presented GPHTRIEL technique achieved higher accuracy, sensitivity, and specificity up to 6%, 11%, and 10% in palmprint detection. Additionally, computational time in palmprint detection was minimized by 14% when compared to existing methods. In future work, we will examine the applicability of deep learning models and transfer learning concepts for biometric palmprint recognition. Also, the different parameters such as space complexity and precision are considered for Palmprint Detection.

REFERENCES

[1] S. Zhao and B. Zhang, "Joint Constrained Least-Square Regression With Deep Convolutional Feature for Palmprint Recognition," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 52, no. 1, pp. 511–522, Jan. 2022, doi: 10.1109/TSMC.2020.3003021.

[2] A. B. Mehmood, I. A. Taj, and M. Ghafoor, "Palmprint enhancement network (PEN) for robust identification," *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 14449–14476, Jul. 2023, doi: 10.1007/s11042-023-16043-z.

[3] M. Abdullah, B. Kedir, K. A. Alemayehu, and H. T. Habtemarium, "A Single Objective GA and PSO for the Multimodal Palmprint Recognition System," *Math. Probl. Eng.*, vol. 2023, pp. 1–14, Jan. 2023, doi: 10.1155/2023/7621550.

[4] S. Trabelsi, D. Samai, F. Dornaika, A. Benlamoudi, K. Bensid, and A. Taleb-Ahmed, "Efficient palmprint biometric identification systems using deep learning and feature selection methods," *Neural Comput. Appl.*, vol. 34, no. 14, pp. 12119–12141, Jul. 2022, doi: 10.1007/s00521-022-07098-4.

[5] L. R. Marval-Perez, K. Ito, and T. Aoki, "Phase-Based Palmprint Identification With Convolutional Sparse

Coding," *IEEE Trans. Biometrics, Behav. Identity Sci.*, vol. 4, no. 3, pp. 424–438, Jul. 2022, doi: 10.1109/TBIOM.2022.3183568.
[6] K. Ito *et al.*, "HandSegNet: Hand segmentation using convolutional neural network for contactless palmprint recognition," *IET Biometrics*, vol. 11, no. 2, pp. 109–123, Mar. 2022, doi: 10.1049/bme2.12058.
[7] M. Li, H. Wang, H. Liu, and Q. Meng, "Palmprint recognition based on the line feature local tri-directional patterns," *IET Biometrics*, vol. 11, no. 6, pp. 570–580, Nov. 2022, doi: 10.1049/bme2.12085.
[8] A. Iula and M. Micucci, "Multimodal Biometric Recognition Based on 3D Ultrasound Palmprint-Hand Geometry Fusion," *IEEE Access*, vol. 10, pp. 7914–7925, 2022, doi: 10.1109/ACCESS.2022.3143433.
[9] H. Shao and D. Zhong, "Towards Cross-Dataset Palmprint Recognition Via Joint Pixel and Feature Alignment," *IEEE Trans. Image Process.*, vol. 30, pp. 3764–3777, 2021, doi: 10.1109/TIP.2021.3065220.
[10] D. Brown and K. Bradshaw, "Deep Palmprint Recognition with Alignment and Augmentation of Limited Training Samples," *SN Comput. Sci.*, vol. 3, no. 1, p. 11, Jan. 2022, doi: 10.1007/s42979-021-00859-3.
[11] M. M. Ata, K. M. Elgamily, and M. A. Mohamed, "Robust features fusion utilization for supervised palmprint recognition," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 10, May 2022, doi: 10.1002/cpe.6817.
[12] İ. KILINÇ, Y. O. ARTAN, and E. BAŞESKİ, "A Multistep Fusion Matcher Approach for Large Scale Latent Fingerprint/Palmprint Recognition," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 31, no. 2, pp. 412–430, Mar. 2023, doi: 10.55730/1300-0632.3992.
[13] A. Ignat and I. Păvăloi, "Keypoint Selection Algorithm for Palmprint Recognition with SURF," *Procedia Comput. Sci.*, vol. 192, pp. 270–280, 2021, doi: 10.1016/j.procs.2021.08.028.
[14] L. Wu, Y. Xu, Z. Cui, Y. Zuo, S. Zhao, and L. Fei, "Triple-Type Feature Extraction for Palmprint Recognition," *Sensors*, vol. 21, no. 14, p. 4896, Jul. 2021, doi: 10.3390/s21144896.
[15] G. Jaswal and R. C. Poonia, "Selection of optimized features for fusion of palm print and finger knuckle-based person authentication," *Expert Syst.*, vol. 38, no. 1, Jan. 2021, doi: 10.1111/exsy.12523.
[16] L. Fei, B. Zhang, Y. Xu, C. Tian, I. Rida, and D. Zhang, "Jointly Heterogeneous Palmprint Discriminant Feature Learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 9, pp. 4979–4990, Sep. 2022, doi: 10.1109/TNNLS.2021.3066381.
[17] P. Hemrajani, V. S. Dhaka, G. Rani, P. Shukla, and D. P. Bavirisetti, "Efficient Deep Learning Based Hybrid Model to Detect Obstructive Sleep Apnea," *Sensors*, vol. 23, no. 10, p. 4692, May 2023, doi: 10.3390/s23104692.
[18] H. Shao and D. Zhong, "One-shot cross-dataset palmprint recognition via adversarial domain adaptation," *Neurocomputing*, vol. 432, pp. 288–299, Apr. 2021, doi: 10.1016/j.neucom.2020.12.072.
[19] B. Liu and J. Feng, "Palmprint orientation field recovery via attention-based generative adversarial network," *Neurocomputing*, vol. 438, pp. 1–13, May 2021, doi: 10.1016/j.neucom.2021.01.049.
[20] Y. Ma and Z. Guo, "Palmprint Translation Network for Cross-Spectral Palmprint Recognition," *Electronics*, vol. 11, no. 5, p. 736, Feb. 2022, doi: 10.3390/electronics11050736.
[21] Y. Liu and A. Kumar, "Contactless Palmprint Identification Using Deeply Learned Residual Features," *IEEE Trans. Biometrics, Behav. Identity Sci.*, vol. 2, no. 2, pp. 172–181, Apr. 2020, doi: 10.1109/TBIOM.2020.2967073.
[22] M. Izadpanahkakhk, S. Razavi, M. Taghipour-Gorjikolaie, S. Zahiri, and A. Uncini, "Deep Region of Interest and Feature Extraction Models for Palmprint Verification Using Convolutional Neural Networks Transfer Learning," *Appl. Sci.*, vol. 8, no. 7, p. 1210, Jul. 2018, doi: 10.3390/app8071210.
[23] K. Zhang, G. Xu, Y. K. Jin, G. Qi, X. Yang, and L. Bai, "Palmprint recognition based on gating mechanism and adaptive feature fusion," *Front. Neurorobot.*, vol. 17, May 2023, doi: 10.3389/fnbot.2023.1203962.
[24] W. Jia, W. Xia, Y. Zhao, H. Min, and Y.-X. Chen, "2D and 3D Palmprint and Palm Vein Recognition Based on Neural Architecture Search," *Int. J. Autom. Comput.*, vol. 18, no. 3, pp. 377–409, Jun. 2021, doi: 10.1007/s11633-021-1292-1.
[25] K. Jing, X. Zhang, and G. Song, "Correntropy-Induced Discriminative Nonnegative Sparse Coding for Robust Palmprint Recognition," *Sensors*, vol. 20, no. 15, p. 4250, Jul. 2020, doi: 10.3390/s20154250.

BIOGRAPHIES OF AUTHORS

N. Kohila, M.Sc., M.Phil., M.C.A., (Ph. D)., works as an Asst. Professor from PG and Research Department of Computer Science and Applications, Vivekanandha College of Arts & Sciences for Women(Autonomous), Tiruchengode, Tamil Nadu. She is pursuing her Ph. D at Periyar University, Salem. She has more than 20 years of teaching experience, and her areas of specialization are Image Processing, Network Security, Computer Graphics and Internet of Things. She has published many papers in conferences and journals. She can be contacted at email: padmeeshraj@gmail.com

T. Ramaprabha is working as an associate professor in the Department of Digital Cyber Forensic Science, Nehru Arts and Science College, Coimbatore. She has published 60+ Research articles in reputed International Journals, four books and 8 Book Chapters. She has guided and completed 4 Ph. D Scholars. She took part as a member of several academic bodies and has 26 years of Teaching and 20 Years of research experience. She has been awarded the Best Faculty and Distinguished Professor for her service. She can be contacted at email: nascramapradhacs@nehrucolleges.com