A novel fingerprint classification system using BPNN with local binary pattern and weighted PCA

K. Sasirekha* and K. Thangavel

Department of Computer Science,
Periyar University,
Salem, Tamilnadu, India
Email: ksasirekha7@gmail.com
Email: drktvelu@yahoo.com
*Corresponding author

Abstract: Fingerprint classification is an important indexing scheme to reduce fingerprint matching time. In this paper, a novel approach to classify fingerprint images is proposed. It involves four main parts: denoising, feature extraction, dimensionality reduction and classification. Initially, the fingerprint is denoised using undecimated wavelet transform. Then short time Fourier transform (STFT) is used to enhance the denoised fingerprints. A set of local binary pattern (LBP) features are extracted to overcome the difficulty associated with singular point detection. To reduce the dimensionality of the feature space, quick reduct (QR), principal component analysis (PCA) and weighted PCA have been investigated. Finally, the fingerprint images are classified using back propagation neural network (BPNN). In this research, experiments have been conducted on real-time fingerprint images collected from 150 subjects and also on the NIST-4 dataset. The proposed method has been compared with support vector machine (SVM), K-nearest neighbor (K-NN), and multi-layer perceptron (MLP).

Keywords: fingerprint; classification; local binary pattern; LBP; back propagation neural network; BPNN; quick reduct; weighted PCA.


Biographical notes: K. Sasirekha received her Master’s of Computer Applications from the Anna University, Coimbatore in 2010. She obtained her Master’s of Philosophy from the Periyar University, Salem, India in 2012. She is currently pursuing her PhD in Computer Science at the Periyar University under the guidance of Dr. K. Thangavel. Her area of interests includes image processing, biometrics, data mining, rough set, optimisation and neural network.

K. Thangavel received his Masters of Science in Mathematics from the Bharathidasan University in 1986 and his Master of Computer Applications from the Madurai Kamaraj University, India in 2001. He obtained his PhD degree from the Department of Mathematics, Gandhigram Rural Institute-Deemed University, Gandhigram, India in 1999. Currently he is working as a Professor and Head, Department of Computer Science, Periyar University, Salem. He is a recipient of Tamilnadu Scientist award for 2009. His area of interests includes medical image processing, bioinformatics, artificial intelligence, neural network, fuzzy logic, data mining and rough set.
1 Introduction

As our society has become connected electronically and more mobile, surrogate representations of identity such as passwords and cards cannot be trusted to ascertain a person’s identity. Cards can be lost or stolen and passwords or PIN can be guessed easily. Furthermore, passwords and cards can be easily shared so they do not provide non-repudiation. Biometric methods provide a higher level of security and are more convenient for the user than the conventional methods of personal authentication (Jain et al., 2004). Biometric authentication refers to identifying persons based on their physiological and behavioural characteristics (Yager and Amin, 2004). Among all the biometrics, the fingerprint is a great source for identification of individuals. Fingerprint has been widely used for personal authentication in forensic and civilian applications because of its distinctiveness, immutability and low cost (Sasirekha and Thangavel, 2014; Anoop and Mini, 2015).

A good database of sample patterns is very much essential for any pattern recognition or classification system. Therefore, image acquisition is the first step in any fingerprint classification system. It is concerned with capturing the fingerprint image using a camera or scanner. Many works (Maev and Severin, 2011) in the literature have used sensors for the acquisition of fingerprint images. In this paper, the fingerprint images are collected using eSSL ZK7500 fingerprint sensor. This device utilises optical fingerprint scanning technology for superior quality and reliability. The glass plate is illuminated by light emitting diode (LED). When the input finger is placed on the glass plate, light falling on the ridges is reflected and is captured by a charge coupled (CCD) camera. The acquired fingerprint images are contaminated with noise during its acquisition and transmission (Kwan and Kwan, 2011). So far several techniques have been developed for reducing the noises in fingerprint images in spatial, frequency and wavelet domain (Porwik and Wieclaw, 2008). Denoising images by thresholding in the wavelet domain have been developed principally by Donoho (1994). Enhancement of the denoised image is essential to improve the quality of the fingerprint. The most popular and widely used enhancement technique is short time Fourier transform (STFT) (Chikkerur et al., 2007). Large volumes of fingerprints are collected and stored every day in a wide range of applications, particularly in forensics and government applications. An automatic recognition of people based on fingerprint image requires matching of an input fingerprint with a huge number of fingerprints stored in a database. An accurate and consistent classification can significantly reduce fingerprint matching time for a large database. The first attempts to classify the fingerprints were made by Henry (1900).

The Henry system classifies fingerprints into three broad categories:

1 arch
2 whorl
3 loop.

The loop is further divided into left and right. Loops constitute between 60% and 70% of the total fingerprint patterns encountered; whorls make up between 25% and 35%, and arches account for the remaining percent (Sam et al., 2015). Figure 1 shows the classification of fingerprints.
A novel fingerprint classification system using BPNN

Figure 1  Three major fingerprint classes, (a) arch (b) whorl (c) loop

Fingerprint classification remains a tough problem for both human experts and automatic systems because of large variations in fingerprint configurations. Moreover, the acquired fingerprint images often contain noise, which makes the classification task even more complicated. A substantial amount of experience is required for a forensic expert to reach a satisfactory level of performance in fingerprint classification. The accuracy of the fingerprint classification system is significantly influenced by the quality of features extracted from the enhanced fingerprint image. Most of the information about a fingerprint category is contained in the central part of the fingerprint, called the pattern area (Wang and Xie, 2004). The knowledge-based techniques which use both the core and delta points for classification require that these points be present in the image. The drawbacks of these methods are that they may not utilise the rich discriminatory information found in the fingerprints and may have a high computational complexity.

Early methods for texture classification focus on the statistical analysis of texture images. The representative methods include the co-occurrence matrix (Haralik et al., 1973) and filtering based approaches such as Gabor filtering (Randen and Husy, 1999), wavelet transforms (Laine and Fan, 1993). In general, their classification accuracy is good as long as the training and test samples have identical orientations. However, the rotations of real-world textures will vary arbitrarily, severely affecting the performance of the statistical methods and signifying the need for rotation invariant methods of texture classification. To overcome the shortcomings of the above methods, it is proposed to extract the local binary pattern (LBP) features from the enhanced fingerprint which is rotation invariant (Guo et al., 2010). An efficient method for the classification of fingerprint images is proposed in this paper. Initially, the noise is removed from the acquired fingerprint image using our previously proposed modified universal threshold in wavelet domain (Sasirekha and Thangavel, 2014). Then the denoised image is enhanced using STFT. A vector of LBP features is extracted from the enhanced fingerprint image. The dimensionality of the feature space is reduced using quick reduct (QR), principal component analysis (PCA), and weighted PCA. Finally, the reduced features are used to train a back propagation neural network (BPNN) for classifying fingerprints. Figure 2 shows the methodology of the proposed fingerprint classification system.
Figure 2  Methodology of proposed fingerprint classification system

The paper is organised as follows. In Section 2, the fingerprint image is denoised and enhanced. In Section 3, the feature extraction using LBP is presented. In Section 4, the dimensionality of the feature space is reduced using QR, PCA and proposed weighted PCA. Fingerprint images are classified using BPNN in Section 5. Section 6 provides experimental results of the proposed method on fingerprint images. Finally, this paper concludes with some perspectives in Section 7.

2 Fingerprint image denoising and enhancement

2.1 Denoising using modified universal threshold

Denoising of the fingerprint image is indispensable to get a noise-free fingerprint image. The stationary wavelet transform (SWT) is studied over the conventional discrete wavelet transform (DWT) as it is not a time-invariant transform (Sasirekha and Thangavel, 2014). SWT is also known as undecimated wavelet transform. Figure 3 shows the process of fingerprint denoising.
Figure 3  Block diagram for denoising in wavelet domain

The steps in wavelet-based fingerprint image denoising are as follows:

1. decompose the noisy image using SWT
2. threshold the wavelet coefficients using the selected threshold method,
3. reconstruct the image using ISWT to get the noise-free image.

Visushrink was introduced by Donoho (Sasirekha and Thangavel, 2014). It is also referred to as universal threshold \( T \) and it is defined as

\[
T = \sigma \left( \sqrt{2 + \log(N)} \right)
\]

where \( N \) is the number of elements or pixels in the image and \( \sigma \) is the noise variance in that image. The universal threshold is modified based on the golden ratio and the weighted median as,

\[
T_{\text{new}} = \sigma \left( \sqrt{1.618 \log(N)} \right)
\]

where ‘1.618’ is the golden ratio. The weighted median is used to compute the median of the high pass portion of the image instead of the conventional method. The following classical weight function \( W(x, y) \) (Wang et al., 2010) is adopted for computing the weighted coefficient of the diagonal subband \( HH \) which is given by (3),

\[
W(x, y) = \frac{1}{\left[\log_{10}(\|x, y\|)\right]}
\]

where \( x \) and \( y \) are the coordinates in the \( HH \) subband. The weight \( W \) will be multiplied with \( HH \) to get weighted diagonal subband as given in (4),

\[
HH_1(x, y) = W(x, y) \times HH(x, y)
\]

The noise variance is then calculated from the weighted diagonal subband \( HH_1 \) as given in (5),

\[
\sigma = \frac{\text{Median}(HH_1)}{0.6745}
\]
After computing the noise variance, the modified universal threshold is applied to the noisy fingerprint image and reconstructs using inverse SWT to get noise reduced image as in Algorithm 1 (Sasirekha and Thangavel, 2014). The resultant image after applying the proposed denoising method is shown in Figure 4.

**Figure 4** Fingerprint denoising, (a) input fingerprint image (b) Gaussian noise added fingerprint image (c) denoising using the proposed method

![Fingerprint images](image)

**Algorithm 1** Denoising procedure of modified universal threshold using golden ratio and weighted median

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add noise to the input image (Gaussian).</td>
</tr>
<tr>
<td>2</td>
<td>Decompose the noisy image using forward SWT.</td>
</tr>
<tr>
<td>3</td>
<td>Compute the noise variance ((\sigma)) from the Diagonal subband based on the weighted median.</td>
</tr>
<tr>
<td>4</td>
<td>Calculate weights for the coefficients in the diagonal subband using the classical weight function</td>
</tr>
<tr>
<td></td>
<td>[ W(x, y) = \frac{1}{\text{Median}(HH_1)} ]</td>
</tr>
<tr>
<td>5</td>
<td>Multiply (W(x, y)) with (HH(x, y)) to get weighted diagonal subband (HH_1(x, y)).</td>
</tr>
<tr>
<td>6</td>
<td>Compute noise variance from (HH_1(x, y)) as</td>
</tr>
<tr>
<td></td>
<td>(\sigma = \frac{\text{Median}(HH_1)}{0.6745} )</td>
</tr>
<tr>
<td>7</td>
<td>Threshold the wavelet coefficients in detail subbands using modified universal threshold with golden ratio as</td>
</tr>
<tr>
<td></td>
<td>(T_{\text{new}} = \sigma(\sqrt{1.618 \times \log(N)}))</td>
</tr>
</tbody>
</table>

Reconstruct using ISWT to get the noise-free image. Evaluate the performance using image quality metrics such as MSE, RMSE, PSNR, and SNR.
2.2 Fingerprint image enhancement

2.2.1 Image enhancement using STFT

The denoised fingerprint image is enhanced using contextual filtering through Fourier domain (Sasirekha and Thangavel, 2016). The intrinsic images such as ridge orientation, ridge frequency, and region mask are simultaneously obtained using short-time Fourier analysis. The algorithm for fingerprint image enhancement consists of two stages as described below:

Algorithm 2 Image enhancement using STFT

---

**Stage 1: STFT analysis**

1. For each overlapping block $B(x, y)$ in an image, generate and reconstruct a ridge orientation image $O(x, y)$ by computing gradients of pixels in a block, a ridge frequency image $F(x, y)$ through obtaining the fast Fourier transform (FFT) value of the block, and an energy image $E(x, y)$ by summing the power of FFT value.
2. The orientation image is smoothed using vector average to yield a smoothed orientation image, and a coherence image $C(x, y)$ is generated using the smoothed orientation image.
3. The frequency image is smoothed using isotropic diffusion on frequency map.
4. The region mask $R(x, y)$ is generated by thresholding the energy image using Otsu’s threshold.

**Stage 2: enhancement**

5. For each overlapping block $B(x, y)$ in the image,
   a. Compute the angular filter $F_A$ centred around $O(x, y)$ and with bandwidth inversely proportional to $C(x, y)$.
   b. Compute radial filter $F_R$ centred around the frequency $F(x, y)$.
   c. Filter a block in the FFT domain, $F = F \times F_A \times F_R$.
   d. Compute the enhanced block by inverse Fourier transform $\text{IFFT}(F)$.
6. Reconstruct the enhanced image by composing enhanced blocks and yield the final enhanced image with the region mask as in Figure 5.

---

**Figure 5** Fingerprint enhancement using STFT, (a) original image (b) enhanced image
3 Feature extraction using LBP

The most essential for texture analysis is to describe the spatial behaviour of intensity values using neighbourhoods. LBP is a simple and efficient feature extractor which labels the pixels of a fingerprint image by thresholding the neighbourhood of each pixel and considers the result as a binary number (Pietikainen et al., 2011). The LBP has become a popular approach in various applications due to its discriminating power and computational simplicity. The essential property of the LBP operator in real-world applications is its robustness to monotonic gray-scale changes caused illumination variations. Another important property is its computational simplicity, which makes it feasible to analyse images in challenging real-time settings (Fathi and Naghsh-Nilchi, 2014).

Given a central pixel in a fingerprint image \( I \), the \( LBP_{P,R} \) operator is defined as

\[
LBP_{P,R} = \sum^{P-1}_{p=0} s(I_p - I_c)2^p
\]

where \( I_c \) is the gray value of the central pixel, \( I_p \) is the value of its neighbours, \( P \) is the number of neighbours and \( R \) is the radius of the neighbourhood.

Suppose the image is \( M \times N \). After computing the \( LBP \) pattern of each pixel \((i, j)\), the whole fingerprint image is represented by building a histogram.

\[
H(k) = \sum_{i=1}^{M} \sum_{j=1}^{N} f(LBP_{P,R}(i, j), k), k \in [0, K]
\]

where \( K \) is the maximal \( LBP \) value in the fingerprint image.

Algorithm 3  \( LBP \)

1. Divide the fingerprint image into non-overlapping windows.
2. For each pixel in a window, compare the pixel to each of its eight neighbours in clockwise.
3. Given a central pixel \( I_c \) in a fingerprint image \( I \), the \( LBP_{P,R} \) operator is defined as

\[
LBP_{P,R} = \sum^{P-1}_{p=0} s(I_p - I_c)2^p
\]

\[
s(x) = \begin{cases} 
1, & x \geq 0 \\
0, & x < 0 
\end{cases}
\]

4. After computing the \( LBP \) pattern of each pixel \((i, j)\), the whole fingerprint image is represented by building a histogram

\[
H(k) = \sum_{i=1}^{M} \sum_{j=1}^{N} f(LBP_{P,R}(i, j), k), k \in [0, K]
\]

\[
f(x, y) = \begin{cases} 
1, & x = y \\
0, & otherwise
\end{cases}
\]
A novel fingerprint classification system using BPNN

\( H \) represents a feature vector for the entire fingerprint image. By using Algorithm 3, a set of 59 rotations invariant features are extracted from the preprocessed fingerprint image for classification.

4 Dimensionality reduction

Dimensionality reduction (DR) is the process of reducing the length of features into a smaller one. In some occasions, it is useful or even necessary to first reduce the dimensionality of the data to a convenient size, keeping as much of the original information as possible, and then feed the reduced dimension data into the system (Pasupa, 2013).

4.1 DR using QR

Rough set theory is a formal mathematical tool that can be applied to reducing the dimensionality of datasets. The rough set attribute reduction removes redundant input attributes from datasets of discrete values without losing its information. The approach is fast and efficient; make use of standard operations from conventional set theory.

The reduction of attributes is achieved by comparing the equivalence relations generated by sets of attributes. Redundant attributes are removed so that the reduced set provides the same predictive capability of the decision feature as the original. A reduct can be defined as a subset of minimal cardinality \( R_{\text{min}} \) of the conditional attribute set \( C \). Here, \( C \) is used to denote the condition attributes, \( D \) for decision attributes. A dataset may have many attribute reduct sets, the set \( R \) of all reducts is defined as:

\[
R = \{ X : X \subseteq C; \gamma_x(D) = \gamma_x(D) \} \tag{10}
\]

The core is the intersection of all the sets in \( R \), the elements of the core are those attributes in the dataset that cannot be eliminated without introducing more contradictions. In this method, a subset with minimum cardinality is searched for a single element of the minimal reduct set \( R_{\text{min}} \subseteq R \).

\[
R_{\text{min}} = \{ X : X \in R; \forall Y \in R; |X| \leq |Y| \} \tag{11}
\]

Algorithm 4 attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those attributes that result in the maximum increase in the rough set dependency metric until this produces its maximum possible value for the dataset (Sathishkumar et al., 2013). According to the algorithm, the dependency of each attribute is calculated, and the best candidate is chosen. The next best feature is added to the set until the dependency of the dataset. Initially, an attribute \( d \) is chosen as its corresponding degree of dependency is the highest. Finally, a reduct has been obtained by eliminating those conditional attributes that are not appearing in the set.
Algorithm 4  Quick reduct

1. Input the conditional \((C)\) and decision attribute \((D)\).
2. Initialise reduct \(R \leftarrow \{\}\)
3. do
4. \(T \leftarrow R\)
5. \(x \ (C - R)\)
6. if \(\gamma_{RU}({x}) > \gamma_{T}(D)\), where \(\gamma_{T}(D) = \text{card}(\text{POS}_R(D)) / \text{card}(U)\)
7. \(T \leftarrow RU(x)\)
8. \(R \leftarrow T\)
9. until \(\gamma_T(D) = = \gamma_c(D)\)
10. Return \(R\)

4.2 DR using PCA

PCA is one of the well-known methods for dimension reduction (Sabnis and Khare, 2012). It is an unsupervised DR method. The purpose of PCA is to decrease the large dimensionality of the data space to the smaller intrinsic dimensionality of feature space. Algorithm 5 shows the DR using PCA.

Algorithm 5  DR using PCA

1. Initialise the input data : \(X = x_1, x_2, \ldots, x_m\) represents the extracted LBP features from the fingerprint images [one row for each image].
2. Compute the average vector:
   \[
   \bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i
   \]
3. Subtract the mean:
   \[
   \Phi_j = x_j - \bar{x}
   \]
4. Calculate the covariance matrix:
   \[
   C = \frac{1}{m} \sum_{j=1}^{m} \Phi_j \Phi_j^T
   \]
5. Compute eigenvalues and eigenvectors of the obtained covariance matrix using singular value decomposition
   Eigenvalues = \(\lambda_1 > \lambda_2 > \ldots > \lambda_n\)
   Eigenvectors = \(u_1, u_2, \ldots, u_n\)
6. Order the eigenvectors by eigenvalue, from highest to lowest. This gives the components in order of significance. The eigenvector with the highest eigenvalue represents the principal component of the input image.
7. Output: a reduced feature vector is formed by choosing the highest eigenvalue.

4.3 DR using the proposed weighted PCA

The traditional PCA, which is widely employed in the analysis of high-dimensional data, is not suggested in certain types of data due to the presence of noise. In this work, a new
A novel fingerprint classification system using BPNN

method of weighted PCA (WPCA) is applied to the extracted LBP features. PCA is modified to improve the performance of the DR method. The proposed weighted PCA algorithm is given in Algorithm 6.

Algorithm 6  DR using weighted PCA

1. Initialise the input data: \( X = x_1, x_2, \ldots, x_m \) represents the extracted LBP features from the fingerprint images [one row for each image].
2. The following weight function is multiplied with the input data to get the weighted data matrix.
   \[
   W = \frac{1}{e^{|x|}}
   \]
3. The weight \( W \) will be multiplied with the input data \( X \) to get weighted data matrix 
   \[
   X_{\text{new}} = W \times X
   \]
4. Compute the average from the weighted data matrix
   \[
   \bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_{\text{new}}^i
   \]
5. Subtract the mean :
   \[
   \Phi = x_i - \bar{x}
   \]
6. Calculate the covariance matrix:
   \[
   C = \frac{1}{m} \sum_{i=1}^{m} \Phi_i \Phi_i^T
   \]
7. Compute eigenvalues and eigenvectors of the obtained covariance matrix using Singular Value Decomposition
   Eigenvalues = \( \lambda_1 > \lambda_2 > \ldots > \lambda_n \)
   Eigenvectors = \( u_1, u_2, \ldots, u_n \)
8. Order the eigenvectors by eigenvalue, from highest to lowest. This gives the components in order of significance. The eigenvector with the highest eigenvalue represents the principal component of the input image.
9. Output: a reduced feature vector is formed by choosing the highest eigenvalue.

5  Fingerprint classification

5.1  Neural network classifier

A neural network is a powerful tool to capture and represents complex input/output relationships (Rajakeerthana et al., 2014). The power and effectiveness of neural networks have been demonstrated in several applications in the past including speech synthesis and analysis, diagnostic problems, medicine, business and finance, robotic control, signal processing, image processing and many other problems that fall under the category of pattern recognition. Features extracted and selected in the earlier sections are given as input to the BPNN to perform the classification of fingerprint images.
5.2 Artificial neural networks

The BPNN optimises the net for correct responses to the training input data set. Figure 6 shows the architecture of the BPNN network for fingerprint classification, and it allows for learning of the network using highly parallel series of simple units and is suitable for data that is noisy and vector based. Here $X_1$ to $X_{59}$ represents the features extracted from fingerprint image using LBP. $Y_1$ to $Y_3$ represents class labels for the fingerprint.

**Figure 6**  BPNN for fingerprint classification

5.3 Back propagation algorithm

The back propagation algorithm can be implemented in two different modes: online mode and batch mode. In the online mode, the error function is calculated after the presentation of each input fingerprint image features and the error signal is propagated back through the network, modifying the weights before the presentation of the next image features (Salem et al., 2016). This error function is the mean square error (MSE) of the difference between the desired and the actual responses of the network.

A new fingerprint image features are presented to the network and this process continues until all the image features have been presented to the network (Haykin, 2005). The presentation of all the image features is usually called one epoch. In practice, many epochs are needed to perform before the error becomes tolerably small. In the batch mode, the error signal is calculated for each input fingerprint image features and the weights are modified every time the input image features have been presented to the network. Finally, the error function is calculated as the sum of the individual MSE for each image feature, and the weights are accordingly modified before the next iteration.

5.4 BPNN classifier

The reduced feature set obtained from the feature selection algorithms are normalised between zero and one. Assign these normalised values to the input neurons (Haykin, 2005).
A novel fingerprint classification system using BPNN

The number of hidden neurons is greater than or equal to the number of input neurons. Initial weights are assigned randomly. The output from each hidden neuron is calculated using the sigmoid function as

\[ S_i = \frac{1}{1 + e^{-\lambda x}} \text{ where } \lambda = 1 \text{ and } x = \sum_k W_{ih}k_i \]

(12)

where \( W_{ih} \) is the weight assigned to the input and hidden layer and \( k \) is the input value of the network. The output is calculated using the sigmoid transfer function.

\[ S_i = \frac{1}{1 + e^{-\lambda x}} \text{ where } \lambda = 1 \text{ and } x = \sum_i W_{hi}S_i \]

(13)

where \( W_{hi} \) is the weight assigned between hidden and output layer and \( S_i \) is the output value from hidden neurons. \( S_2 \) is subtracted from the desired output. Using this error value \((E = \text{Desired} - \text{Actual})\), the updating of weight is performed as:

\[ \delta = ES_2(1 - S_2) \]

(14)

The weights assigned between input and hidden layer, the hidden and output layer are updated as:

\[ W_{ho} = W_{ho} + (n.\delta.S_i) \]

(15)

\[ W_{ih} = W_{ih} + (n.\delta.K_i) \]

(16)

where \( n \) is the learning rate, \( \delta \) is the weight and \( k \) is the input value to the network. Again the output is calculated for the hidden and output neurons. Then the error value \((E)\) is checked, and the weights are updated. The above procedure is repeated till the target output is equal to the desired output. The algorithm of back propagation classifier for classification of the fingerprint image is given in Algorithm 7.

Algorithm 7  Back propagation classifier

1. Normalise the feature between 0 and 1 and assigned to input neurons.
2. Initialise random weight
3. Compute each hidden neuron,
   \[ S_i = \frac{1}{1 + e^{-\lambda x}} \text{ where } \lambda = 1 \text{ and } x = \sum_k W_{ih}k_i \]
4. Compute each output layer,
   \[ S_i = \frac{1}{1 + e^{-\lambda x}} \text{ where } \lambda = 1 \text{ and } x = \sum_i W_{hi}S_i \]
5. Subtract \( S_2 \) from the desired output. Using error \((E)\) value, compute the weight as:
   \[ \delta = ES_2(1 - S_2) \]
6. Update the weights using this \( \delta \) value.
   \[ W_{ho} = W_{ho} + (n.\delta.S_i) \]
   \[ W_{ih} = W_{ih} + (n.\delta.K_i) \]
7. Perform steps (3) to (6) with the updated weights, till the target output is equal to the desired output.
6 Experimental results and discussions

6.1 Dataset

Fingerprint dataset has been created from 150 subjects each with ten samples. The following figure shows example fingerprint samples from the database. The fingerprint scanner used to capture the image is eSSL ZK7500. All fingerprint images are 8 bit gray-level BMP files, and the image resolution is 280 × 360 as shown in Figure 7. The fingerprint database contains 102 subjects with Loop, ten subjects with arch (plain and tented) and 38 subjects with whorl classes. The proposed method is also validated on NIST-4 dataset (https://www.nist.gov/srd/nist-special-database-4) which contains 4,000 fingerprint images of 800 images for each class.

Figure 7  Fingerprint dataset

6.2 Evaluation measures

6.2.1 Fingerprint image denoising and enhancement

The image quality metrics such as MSE, Root MSE (RMSE), signal to noise ratio (SNR) and peak signal to noise ratio (PSNR) are used to evaluate the performance of denoising (Sasirekha and Thangavel, 2014). The metrics are shown in Table 1.

Where $I_1$ is the input fingerprint image, $I_2$ is the denoised image, $m$ is the number of rows and $n$ is the number of columns in the image, and $R$ is the maximum fluctuation in the input fingerprint image. Here, $R$ is set as 255, since the fingerprint image data type is an 8-bit unsigned integer.

Tables 2, 3 and 4 show the mean value of MSE, RMSE, PSNR, and SNR of fingerprint database. The performance results in the spatial domain using median filter and wiener filter for the Gaussian noise level as 0.003 is given in Table 2. The MSE, RMSE, PSNR, and SNR of wiener filter are better than the median filter.
A novel fingerprint classification system using BPNN

Table 1
Quantitative metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>[ \sum_{m,n} \left[ I_1(m, n) - I_2(m, n) \right]^2 ]</td>
</tr>
<tr>
<td>RMSE</td>
<td>[ \sqrt{ \sum_{m,n} \left[ I_1(m, n) - I_2(m, n) \right]^2 } ]</td>
</tr>
<tr>
<td>SNR</td>
<td>[ 10 \log_{10} \left( \frac{\text{Var}(I_1)}{\text{Var}(I_2)} \right) ]</td>
</tr>
<tr>
<td>PSNR</td>
<td>[ 10 \log_{10} \left( \frac{R^2}{MSE} \right) ]</td>
</tr>
</tbody>
</table>

Table 2
Median and wiener filter (0.003)

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Filter</th>
<th>MSE</th>
<th>RMSE</th>
<th>PSNR</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Median</td>
<td>297.6306</td>
<td>17.2520</td>
<td>23.4280</td>
<td>2.9774</td>
</tr>
<tr>
<td>2</td>
<td>Wiener</td>
<td>230.1626</td>
<td>15.1711</td>
<td>24.5445</td>
<td>0.8587</td>
</tr>
</tbody>
</table>

Table 3
Visushrink with hard thresholding (0.003)

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Filter</th>
<th>MSE</th>
<th>RMSE</th>
<th>PSNR</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>coif2</td>
<td>80.8178</td>
<td>8.9899</td>
<td>29.0897</td>
<td>0.48191</td>
</tr>
<tr>
<td>2</td>
<td>coif4</td>
<td>93.5269</td>
<td>9.6709</td>
<td>28.4554</td>
<td>0.48891</td>
</tr>
<tr>
<td>3</td>
<td>sym2</td>
<td>79.9228</td>
<td>8.9400</td>
<td>29.1381</td>
<td>0.49661</td>
</tr>
<tr>
<td>4</td>
<td>sym4</td>
<td>90.1331</td>
<td>9.4938</td>
<td>28.6160</td>
<td>0.47804</td>
</tr>
<tr>
<td>5</td>
<td>db1</td>
<td>72.7601</td>
<td>8.5300</td>
<td>29.5459</td>
<td>0.48956</td>
</tr>
<tr>
<td>6</td>
<td>db2</td>
<td>80.6525</td>
<td>8.9807</td>
<td>29.0986</td>
<td>0.49628</td>
</tr>
</tbody>
</table>

Table 4
Proposed modified universal threshold based on GR and weighted median with hard thresholding (0.003)

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Filter</th>
<th>MSE</th>
<th>RMSE</th>
<th>PSNR</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>coif2</td>
<td>7.4912</td>
<td>2.737</td>
<td>39.4193</td>
<td>0.01168</td>
</tr>
<tr>
<td>2</td>
<td>coif4</td>
<td>6.9040</td>
<td>2.6275</td>
<td>39.7738</td>
<td>0.00551</td>
</tr>
<tr>
<td>3</td>
<td>sym2</td>
<td>3.9827</td>
<td>1.9957</td>
<td>42.1630</td>
<td>0.01097</td>
</tr>
<tr>
<td>4</td>
<td>sym4</td>
<td>4.0231</td>
<td>2.0058</td>
<td>41.1192</td>
<td>0.01683</td>
</tr>
<tr>
<td>5</td>
<td>db1</td>
<td>3.7820</td>
<td>1.9447</td>
<td>42.4876</td>
<td>0.01631</td>
</tr>
<tr>
<td>6</td>
<td>db2</td>
<td>3.8535</td>
<td>1.9630</td>
<td>42.3062</td>
<td>0.00983</td>
</tr>
</tbody>
</table>

The performance results of traditional universal threshold (Visushrink) with the Gaussian noise level as 0.003 based on hard thresholding is given in Table 3. The performance results of the proposed modified universal threshold based on GR and weighted median with the Gaussian noise level as 0.003 based on hard thresholding are given in Table 4. The MSE, RMSE, PSNR, and SNR of the proposed modified universal threshold are
better than the traditional universal threshold. Among the filters db1 outperforms. Figure 6 depicts the enhanced image using STFT.

6.2.2 Fingerprint image classification

6.2.2.1 Cross-validation

A cross-validation is a statistical method that divides data into a fixed n folds, or partitions, of approximately the same size (Han et al., 2011), for example, three-fold search containing one-third of the data. The first fold is then used for classifier testing, while the remaining \( n - 1 \) folds are used for classifier training. The process is repeated so that each of the \( n \) folds is used for testing. The resulting process is known as \( n \) fold cross-validation. This process can be combined with stratification, or random sampling to create the folds so as to guarantee that each class is represented evenly in both the testing and training sets, which results in approximately the same class distribution across folds. The result is called stratified \( n \) fold cross-validation.

Ten folds stratified cross-validation is a standard way of measuring the error rate or accuracy of a learning scheme on a particular dataset. In this form of cross-validation, the data is divided randomly into ten subsets of approximately the same size. Among the partitions, nine of them were used as training set, and the remaining one is used as a test set. The classifier is run ten times. Each run, the classifier uses a different subset as the testing set while the other subsets are combined to use as the training set. These results in ten different accuracy estimates for the learner can be averaged to provide an overall accuracy.

6.2.2.2 Confusion matrix

A confusion matrix contains information about actual and predicted classifications done by a classification system. The performance of such systems is commonly evaluated using the data in the matrix (Mahesh and Raj, 2015).

Table 5 shows a confusion matrix for a two-class problem with positive and negative class values. From such a matrix, it is possible to extract a number of widely used metrics to measure the performance of a classifier and is given in Table 6.

The confusion matrix of the first fold for fingerprint classification using BPNN, QR-BPNN, PCA-BPNN and weighted PCA-BPNN is shown in Tables 7–10 respectively.

<table>
<thead>
<tr>
<th>Positive prediction</th>
<th>Negative prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive class</td>
<td>True positive (TP)</td>
</tr>
<tr>
<td>Negative class</td>
<td>False positive (FP)</td>
</tr>
</tbody>
</table>

The validation performance of fingerprint classification using BPNN, QR-BPNN, PCA-BPNN and weighted PCA-BPNN are evaluated and demonstrated in Figure 8. It is observed that the performance of BPNN classifier with weighted PCA outperforms the performance of BPNN classifier with QR and traditional PCA. The gradient and error histogram is shown in Figures 9 and 10 respectively for fingerprint classification.
A novel fingerprint classification system using BPNN

Table 6  Classifier evaluation metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>( \frac{TP + TN}{(TP + FN + TN + FP)} )</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>( \frac{TP}{TP + FN} )</td>
</tr>
<tr>
<td>Precision</td>
<td>( \frac{TP}{TP + FP} )</td>
</tr>
<tr>
<td>Recall</td>
<td>( \frac{TP}{TP + FN} )</td>
</tr>
<tr>
<td>Error rate</td>
<td>( \frac{FP + FN}{(TP + FN + TN + FP)} )</td>
</tr>
</tbody>
</table>

Table 7  Confusion matrix for BPNN of real-time dataset

<table>
<thead>
<tr>
<th>Target class</th>
<th>Loop</th>
<th>Arch</th>
<th>Whorl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output class</td>
<td>Loop</td>
<td>1005</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>Arch</td>
<td>3</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>Whorl</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 8  Confusion matrix for QR-BPNN of real-time dataset

<table>
<thead>
<tr>
<th>Target class</th>
<th>Loop</th>
<th>Arch</th>
<th>Whorl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output class</td>
<td>Loop</td>
<td>997</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Arch</td>
<td>14</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>Whorl</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 9  Confusion matrix for PCA-BPNN of real-time dataset

<table>
<thead>
<tr>
<th>Target class</th>
<th>Loop</th>
<th>Arch</th>
<th>Whorl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output class</td>
<td>Loop</td>
<td>1,010</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Arch</td>
<td>2</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>Whorl</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 10  Confusion matrix for WPCA-BPNN of real-time dataset

<table>
<thead>
<tr>
<th>Target class</th>
<th>Loop</th>
<th>Arch</th>
<th>Whorl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output class</td>
<td>Loop</td>
<td>1,013</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Arch</td>
<td>3</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>Whorl</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 8 Validation performance of real-time dataset, (a) BPNN (b) QR-BPNN (c) PCA-BPNN (d) WPCA-BPNN (see online version for colours)
Figure 8  Validation performance of real-time dataset, (a) BPNN (b) QR-BPNN (c) PCA-BPNN (d) WPCA-BPNN (continued) (see online version for colours)
Figure 9  Gradient and validation of real-time dataset, (a) BPNN (b) QR-BPNN (c) PCA-BPNN (d) WPCA-BPNN (see online version for colours)
Figure 9  Gradient and validation of real-time dataset, (a) BPNN (b) QR-BPNN (c) PCA-BPNN (d) WPCA-BPNN (continued) (see online version for colours)
Figure 10  Error histogram of real-time dataset, (a) BPNN (b) QR-BPNN (c) PCA-BPNN (d) WPCA-BPNN (see online version for colours)
**Figure 10** Error histogram of real-time dataset, (a) BPNN (b) QR-BPNN (c) PCA-BPNN (d) WPCA-BPNN (continued) (see online version for colours)
The efficacy of the proposed algorithm has been compared with the standard benchmark classifiers such as SVM, K-NN, and MLP. Generally, the classifiers take the reduced feature set obtained from QR, PCA, and WPCA. The performance of the fingerprint classification for the real-time and NIST-2 dataset is shown in Tables 11 and 12 respectively.

**Table 11** Performance of fingerprint classification for real-time dataset

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>SVM</th>
<th>K-NN</th>
<th>MLP</th>
<th>BPNN</th>
<th>QR-BPNN</th>
<th>PCA-BPNN</th>
<th>WPCA-BPNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>87.9</td>
<td>93.6</td>
<td>94.0</td>
<td>94.6</td>
<td>91.7</td>
<td>97.5</td>
<td>98.3</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>87.9</td>
<td>94.3</td>
<td>93.7</td>
<td>90.6</td>
<td>93.1</td>
<td>97.1</td>
<td>98.9</td>
</tr>
<tr>
<td>Precision</td>
<td>87.7</td>
<td>93.2</td>
<td>91.8</td>
<td>93.3</td>
<td>91.3</td>
<td>92.3</td>
<td>95.3</td>
</tr>
<tr>
<td>Recall</td>
<td>87.9</td>
<td>94.3</td>
<td>93.7</td>
<td>90.6</td>
<td>93.1</td>
<td>97.1</td>
<td>98.9</td>
</tr>
<tr>
<td>ERR</td>
<td>12.1</td>
<td>06.4</td>
<td>06.0</td>
<td>05.4</td>
<td>08.3</td>
<td>02.5</td>
<td>01.7</td>
</tr>
</tbody>
</table>

**Table 12** Performance of fingerprint classification for NIST-4 dataset

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>SVM</th>
<th>K-NN</th>
<th>MLP</th>
<th>BPNN</th>
<th>QR-BPNN</th>
<th>PCA-BPNN</th>
<th>WPCA-BPNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>89.31</td>
<td>94.0</td>
<td>92.3</td>
<td>94.1</td>
<td>91.5</td>
<td>95.2</td>
<td>96.8</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>87.3</td>
<td>93.2</td>
<td>93.3</td>
<td>94.1</td>
<td>90.4</td>
<td>94.3</td>
<td>96.3</td>
</tr>
<tr>
<td>Precision</td>
<td>86.2</td>
<td>92.5</td>
<td>91.6</td>
<td>94</td>
<td>91.4</td>
<td>95.1</td>
<td>95.9</td>
</tr>
<tr>
<td>Recall</td>
<td>87.3</td>
<td>93.2</td>
<td>93.3</td>
<td>94.1</td>
<td>90.4</td>
<td>94.3</td>
<td>96.3</td>
</tr>
<tr>
<td>ERR</td>
<td>10.69</td>
<td>06.0</td>
<td>07.7</td>
<td>05.9</td>
<td>08.5</td>
<td>04.8</td>
<td>03.2</td>
</tr>
</tbody>
</table>

**Figure 11** Relative performance measures of fingerprint classification algorithms of real-time dataset (see online version for colours)
A novel fingerprint classification system using BPNN

Figure 12  Relative performance measures of fingerprint classification algorithms of NIST-4 dataset (see online version for colours)

![Relative performance measures of fingerprint classification algorithms of NIST-4 dataset](image)

Figure 13  Average performance analysis of accuracy of classifiers (see online version for colours)

![Average performance analysis of accuracy of classifiers](image)

The quantitative results of the proposed weighted PCA with BPNN classifier have been compared with standard pattern recognition algorithms. The measures such as precision, recall, f-measure, accuracy and error rate are utilised to validate the classifiers discussed. From Tables 11 and 12, it is clearly understood that the proposed method outperforms the existing classifiers. It is observed that the highest classification accuracy is 96.80% with the proposed classifier. The relative quantitative measures are given in Figures 11 and 12 respectively. The average performance analysis of the accuracy of classifiers is shown in Figure 13. Similarly, the average performance analysis of error rate of classifiers is shown in Figure 14.
The proposed weighted PCA with BPNN classifier exhibits more accuracy of 96.80% and less error rate of 3.2% than the classifiers such as SVM, K-NN, MLP, BPNN, QR-BPNN, and PCA-BPNN which exhibit accuracies of 87.9%, 93.6%, 94.0%, and 91.7% respectively. Similarly, the error rates of SVM, K-NN, MLP, BPNN, QR-BPNN, and PCA-BPNN are 12.1%, 6.4%, 5.4%, 8.3%, and 2.5% respectively for the real-time dataset. For NIST-2 dataset, the classifiers such as SVM, K-NN, MLP, BPNN, QR-BPNN, and PCA-BPNN exhibits accuracies of 89.31%, 94.0%, 92.3%, 94.1%, 91.5%, and 95.2% which is lesser than the proposed method of 96.8%. Similarly, the error rates of SVM, K-NN, MLP, BPNN, QR-BPNN, and PCA-BPNN are 10.6%, 6.0%, 7.7%, 5.9%, and 4.8% respectively which is higher than the proposed method of 3.2%.

7 Conclusions

A novel method for classifying fingerprint images to reduce fingerprint matching time for a large database is proposed in this paper. Initially, the fingerprint image is denoised using undecimated wavelet transform with a threshold based on the golden ratio and weighted median. Then the denoised fingerprint image is enhanced using STFT. A set of rotation and illumination invariant LBP features are extracted from the enhanced fingerprint to overcome the difficulty associated with singular point detection. The dimensionality of the feature space is reduced using QR, PCA and proposed weighted PCA. Finally, the fingerprint images are classified using BPNN on real-time fingerprint images collected from 150 subjects each with ten samples using eSSL ZK7500 scanner and also with the NIST-4 dataset. The quantitative measures such as accuracy, sensitivity, precision, recall and error rate demonstrate that the performance of proposed weighted PCA with BPNN classifier is better than conventional BPNN, PCA with BPNN, QR with BPNN, SVM, K-NN, and MLP for fingerprint image classification. The proposed method is simple in computation, and yet it is effective in classification.
Acknowledgements

Authors would like to thank UGC, New Delhi for the financial support received under UGC Major Research Project No. 43-274/2014(SR).

References


https://www.nist.gov/srd/nist-special-database-4


