IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Rotation Invariant Fingerprint Matching based on Gray values using SLFNN

Ravinder Kumar*

Department of CSE, HMR Institute of Technology and Management (Affiliated with GGSIPU), Delhi, India,

*Corresponding Author: ravinder_y@yahoo.com, Tel.: 011 – 27724115

Available online at: www.ijcseonline.org

Accepted: 12/Aug/2018, Published: 31/Aug/2018

***Abstract*—** Fingerprint matching is most widely used mean of person identification or verification since last two decades. The issues related to efficient matching under transformation requires lots of attention of the research community. This paper presents rotational invariant directional features computed directly from gray values of fingerprint images and referred as Local Directional Pattern (LDP). Single hidden Layer Feed Forward Neural Network (SLFNN) is proposed to be used for classification. Network is trained using four different training algorithms to determine the suitability of these algorithms. The results show that these features are very discriminatory under rotation and also the efficiency of SLFNN for matching. It is also evident that Resilient Propagation (RP) algorithm is much faster and gives best performance as compared to other training algorithms.

***Keywords*—** Fingerprint Matching, Image based matching, Region of Interest, Ressiliant Propagation, Rotation Invariant.

I. INTRODUCTION

Fingerprint based human identification systems have been deployed since ancient time in criminal investigations, financial transactions, civilian and forensic domains. Fingerprint individuality has not been proved in court of justice rather it is based on the fact introduced in 1892 by Sir Francis Galton. Sir Francis Galton observed that any similar looking object may be distinguished, if examine at a very finer detail. Based on this theory Galton proposed a technique to distinguish fingerprint patterns [1]. The popularity of fingerprint exists because of high reliability and ease of use. It is also obvious from the literature that fingerprints have been fused in multimodal biometric system to improve the performance and to make usable to a large population [2]. Fingerprint image are characterized by a pattern of interleaved ridges and valleys on the fingertips. The distinction among fingerprint images is characterized by the location of certain abnormal point on ridges called as minutiae points. Fingerprint matching algorithms are based on image processing techniques to compare fingerprint patterns and return degree of similarity or binary decision of match or non-match. These matching approaches can be coarsely classified into two categories (i) *Minutiae based technique*; (ii) *Non-Minutiae or image based technique*. Minutiae based techniques are most widely used and most well-known methods for fingerprint matching [2]. Minutiae matching are based on optimum pairing of minutiae pairs of query and input fingerprint images in order to maximize the minutiae match [3, 4]. Tico and Kuosmanen in [5] proposed a well-known method which utilizes minutiae points and directional information around the minutiae for matching. Yang and Wang in [6] propose a technique by combining global orientation field along the minutia for matching. However, minutiae based techniques still suffering from problems like:

1. Underperform in low quality images.
2. Unable to fully extract detailed information of fingerprint images.
3. Alignment of minutia point is required before matching.

The counterpart of minutia based matching are image based methods which, can directly use the gray value image to extract rich discriminatory information like finger print ridge pattern, local directional patterns and ridge shape and frequency, and other texture and statistical features [7]. These methods can further e classified as: *transform based* and *Filter based* methods. A very few transform based features extraction have been proposed in the literature. These methods include Discrete Cosine Transform (DCT), Wavelet Transform, Wavelet and Fourier Mellin –Transform (WMFT) [10, 11, 12]. On the other hand most of the image based methods uses filter based approach [7]. In this approach, a various Gabor filters are deployed spatial and frequency domain to capture local ridge features [6, 7]. A compact and fast, fixed length feature descriptor are thus obtained and are popularly known as fingercode and remains the most popular among these approaches [8]. Fingercode is obtained by ordered enumeration of local features extracted from the tessellated fingerprint images around core point. The original fingercode is further modified to include directional features and referred as Absolute Difference Deviation (AAD) features [9]. Local Binary Pattern (LBP) is another useful descriptor in this series which is based on multiresolution analysis of fingerprint images to provide the grayscale invariant texture descriptor [10].

Although the performance of image based methods is improved for poor quality images but is not up to the mark to be deployed as standalone fingerprint matcher in the absence of effective alignment step.

Hybrid approaches have also been proposed in the literature to combine both these techniques together in order to improve the matching performance [13, 14]. The first method in these series was proposed by Jain et al., by combining minutiae set with ridge feature map by using a set of Gabor. Jain et al., [15] has also proposed another approach by combining minutia orientation and texture map. The matching accuracy had been enhanced using these but performs very poor in low quality images.

The classifier in biometric system has attracted the considerable amount of interest of scientific community in recent years. Various algorithms have been proposed for classification and most of them uses a relatively smaller feature sets. Most of these methods use Neural Network (NN) for fingerprint image pre-processing, feature extraction, recognition and classification [17, 18, 19, 20, 21, 22]. A nonlinear back-propagation neural-network (BPNN) has been deployed by Yang and Park using invariant moment features for fingerprint matching. Yang and Park [13] had also proposed Linear Vector Quantization (LVQ) network for matching. In recent years, Yang et al., [20] proposed fingerprint classification using extreme learning machine (ELM) on invariant moment features for matching. Latest literature on biometrics based personal authentication is presented in [27-39].

In this paper, a Feed Forward Neural Network with single hidden layer (SLFNN) is used as a classifier to find the match between input and query feature vector. The training of the SLFNN was done using four training algorithms (each is selected from different class of training algorithm) and the performance is compared using following criteria:

1. Training Error Rate
2. Matching accuracy
3. Testing Error Rate
4. Convergence Time

Organization of this paper is as follows: Section II details the proposed methodology including the pre-processing, normalization, the image segmentation and the ridge orientation computation processes. The section III of this article details the ROI extraction process. Feature extraction is explored in the section IV. SLFNN is detailed and explored in section V. Section VI presents the experimental results and the paper is finally concluded in section VII along with future research directions.

II. PROPOSED METHODOLOGY

In this section, the detailed methodology of the proposed system which consists of five main sub-sections as depicted in Figure. 1 is presented.

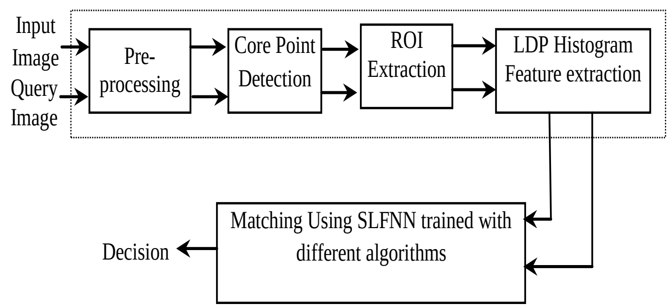


Figure 1. Detailed steps of proposed Fingerprint Matching

A. Pre-processing

Although the proposed LDP features are directly extracted from raw image but, to extract the region of interest (ROI) image need to be pre-processed. Followings steps are used for pre-processing:

a) *Normalization:* To have the gray values of input image is specified ranges an input fingerprint image is normalized using calculated mean and variance.
b) *Ridge Segmentation:* Normalized image is segmented to separate out the background and the foreground.
c) *Ridge Orientation:* The direction of ridges and valleys is extracted to provide ridge orientation and are used to locate reference point.
d) *Core Point detection:* Core point is the reference point in any fingerprint image and is considered as the ridge point with maximum curvature.

a. Normalization

Normalization is the process of elimination of gray values of an image along the valleys and ridges. However, this does not alter the geometrical structure of many fingerprint images. Following equation is used for normalization.

$$N(i,j)=\begin{cases} M_0+\sqrt{\dfrac{V_0\left(I(i,j)-M_I\right)^2}{V_I}} & \text{if } I(i,j)>M_I \\ M_0 \quad \sqrt{\dfrac{V_0\left(I(i,j)-M_I\right)^2}{V_I}} & \text{otherwise} \end{cases} \tag{1}$$

Where $N(i,j)$ represents the normalized image of input image. The gray-value at pixel (i,j) is given by $I(i,j)$, V_I and M_I represents the mean and variance value of input image I respectively. The original and normalized image at a desired variance and mean are shown in Figure 2.

Figure 2. Original images are shown in row 1, whereas second row shows their corresponding normalized images.

b. Image Segmentation

It is the process of separating the ridges from background. Here we perform the segmentation by using variance and mean based method, which is described in the following algorithm:

1. Input image I is decomposed into non-overlapping blocks of size $w \times w$.
2. The mean value of each block is computed using following equation.

$$M(I)=\frac{1}{w^2}\sum_{i=-\frac{w}{2}}^{\frac{w}{2}}\sum_{j=-\frac{w}{2}}^{\frac{w}{2}} I(i,j) \tag{2}$$

3. Obtain the standard deviation $\mathrm{std}(I)$ by using mean value computed in above step as follows:

$$std(I)=\sqrt{\frac{1}{w^2}\sum_{i=-\frac{w}{2}}^{\frac{w}{2}}\sum_{j=-\frac{w}{2}}^{\frac{w}{2}}\left(I(i,j)-M(i,j)\right)^2} \tag{3}$$

4. Empirically select a threshold value. If $std(I) > threshold$ the block is ridge region, while the block is the background.

Ridge segmented image is shown in Figure. 3.

Figure 3. Ridge-segmented fingerprint image

c. Ridge Orientation computation

In this paper we have used gradient-based approach to make fingerprint-matching process more computationally efficient. Orientation provides the direction estimation of valleys and ridges in terms of coordinates. Let φ_{xy} is the angle between fingerprint ridges and horizontal axis representing local ridge orientation at pixel $[x,y]$. For orientation computation, the algorithm described in [8] is used with slight modification. The Gaussian operator as shown in equation (6) have been used to compute the G_x and G_y, the gradient magnitude in x and y directions, respectively.

$$h_g(x,y)=e^{\left(x^2+y^2\right)/\left(2\sigma^2\right)} \tag{4}$$

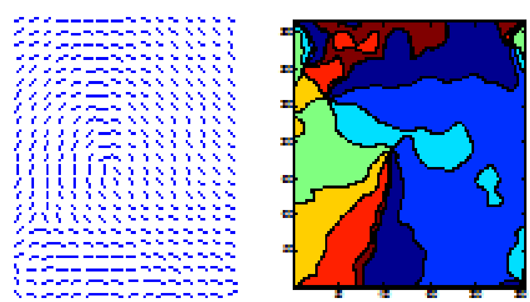
in order to reconstruct the erroneous orientation of local ridges altered by noise and low gray value variance low pass filter is applied. In order to apply low pass filter, the orientation image is converted into a continuous vector field using following equations:

$$\varphi_x = \cos\left(2\theta\left(x,y\right)\right), \tag{5}$$

And

$$\phi_y = \sin\left(2\theta(x,y)\right), \tag{6}$$

where φ_x and φ_y are the x and y components of the vector field, respectively. The Orientation field image is shown in Figure.4.



(a) (b)

Figure 4. Orientation field image (b) Surface view of Orientation image.

d. Core Point detection

The most widely used Poincare index method can usually detect all the true singular points when the index is generated along small region boundaries.

Let (i, j) be the elements position θ_{ij} in the orientation image; then the poincare index $P(i, j)$ enclosed by a digital curve (with N points) can be computed using follows.

$$P(i,j) = \sum_{k=0}^{N} \Delta(k) \quad (9)$$

where

$$\Delta(k) = \begin{cases} \delta(k) & if\ |\delta(k)| < 90^o \\ \delta(k)+180 & if\ \delta(k) \le -90^o \\ \delta(k)-180 & Otherwise \end{cases} \quad (10)$$

Where $\delta(k) = \theta(i_{(k+1)\bmod N}, j_{(k+1)\bmod N}) - \theta(i_k, j_k)$ is the difference between any two neighbouring elements of θ_{ij} in the orientation image Clockwise direction is shown by the sequential ordering in our case from $0\,to\,N-1$, and the size of the closed curve is chosen as 3 i.e. N = 8, and $(k+1)\mathrm{mod}8$ signifies that $d_8 = d_0$. It is well known and is seen that, on closed curves, the poincare index assumes only one out of these discrete values: $0^o, \pm180^o, 360^o$. In case of fingerprint singularities.

$$\boldsymbol{P(i,j)} = \begin{cases} 0^{\boldsymbol{o}} & \boldsymbol{if\ pixel(i,j) does not\ belong\ to any}\ \sin \boldsymbol{gular\ region} \\ 360^{\boldsymbol{o}} & \boldsymbol{if\ pixel(i,j) belongs\ to a whorl\ type}\ \sin \boldsymbol{gular\ region} \\ 180^{\boldsymbol{o}} & \boldsymbol{if\ pixel(i,j) belongs\ to a core\ type}\ \sin \boldsymbol{gular\ region} \\ -180^{\boldsymbol{o}} & \boldsymbol{if\ pixel(i,j) belongs\ to a delta\ type}\ \sin \boldsymbol{gular\ region} \end{cases}$$

Here, we are only interested in generating core points, so considered a region of interest by mask processing of size 2×2 for detection of the core point as shown in Figure 5.

$(i-1, j)$	$(i-1, j+1)$
(i, j)	$(i, j+1)$

Figure 5. The mask for singular point detection.

B. ROI Extraction

To accelerate the recognition process, a square region is considered around the core point as the region of interest (ROI) for feature computing vector. Here, we cropped the fingerprint image around the core point detected of size (101× 100) in earlier step as shown in Figure 6.

Figure 6. ROI image used for feature extraction.

III. Feature extraction

A. Computing LDP Code

The LDP assigns an 8-bit binary code to each pixel of an input image. The LDP pattern is then computed by comparing the relative edge response values of a pixel in eight different directions. Various edge detectors might be used for computing LDP such as: Sobel edge detector, prewitt, and Kirsch. Most of the applications in the literature used Kirsch edge detector due to its high accuracy than others and this also consider edge response from all the eight directions. Kirsch mask is shown in Figure. For a given central pixel, the eight directional response value mk, k = 0, 1, ----, 7 are computed using Kirsh masks.

Higher directional response value represents the presence of corner or an edge in that particular direction. P such prominent directions are used to compute the LDP. The bit response corresponding to these p most prominent directions are set to 1, and remaining 8-p directional bit is set to 0. The whole image is than transformed to LDP map using computed LDP codes.

Local Binary Pattern and its variant is widely used for image representation by micro-level information of edges as well as local features using information of intensity changes around the pixel. LBP have no direct application to the fingerprint, which is supposed to be an oriented pattern rather than texture pattern. Several researchers have used the gradient magnitude instead of pixel intensity [23]. These generally encode the relative changes and do not encode its own directional information or gradient magnitude. Being motivated by these and assuming the fingerprint images as directed pattern, we have employed the various LDP, for computing the edges response values in eight directions and to encode these fingerprint images.

The LDP operator generally operates on non-overlapping blocks of 3×3 size and assigns binary number computed from all eight directions by convolving with eight Kirsch masks to the central pixel of the block. This process is applied to the entire image. Figure. 7 shows the eight directional kirsch masks and the Figure 8. Shows the process of computation of LDP Code.

$$\begin{bmatrix}-3 & -3 & 5\\ -3 & 0 & 5\\ -3 & -3 & 5\end{bmatrix} \quad \begin{bmatrix}-3 & 5 & 5\\ -3 & 0 & 5\\ -3 & -3 & -3\end{bmatrix} \quad \begin{bmatrix}5 & 5 & 5\\ -3 & 0 & -3\\ -3 & -3 & -3\end{bmatrix} \quad \begin{bmatrix}5 & 5 & -3\\ 5 & 0 & -3\\ -3 & -3 & -3\end{bmatrix}$$

East M_0 **North East** M_1 **North** M_2 **North West** M_3

$$\begin{bmatrix}5 & -3 & -3\\ 5 & 0 & -3\\ 5 & -3 & -3\end{bmatrix} \quad \begin{bmatrix}-3 & -3 & -3\\ 5 & 0 & -3\\ 5 & 5 & -3\end{bmatrix} \quad \begin{bmatrix}-3 & -3 & -3\\ -3 & 0 & -3\\ 5 & 5 & 5\end{bmatrix} \quad \begin{bmatrix}-3 & -3 & -3\\ -3 & 0 & 5\\ -3 & 5 & 5\end{bmatrix}$$

West M_4 **South West** M_5 **South** M_6 **South East** M_7

Figure 7. Kirsch Edge response Masks in 8 directions.

126	112	122
122	117	149
120	123	116

(a)

Mask index	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0
Mask Value	134	-98	-50	-26	-90	-90	94	126
Rank	1	3	7	8	5	6	4	2
Code Bit	1	1	0	0	0	0	1	1
LDP Code	195							

(b)

Figure 8. Generating LDP Code.

B. Feature Representing using LDP Code

The LDP texture feature is represented as histogram of LDP coded image generated in the above section. pth bin of the LDP histogram can be defined as

$$T_q = \sum_{x,y} I\{LDP(x,y) = q\}, q = 0,1,\dots,n-1, \quad (11)$$

Where n represent number of histogram bins. Finally the histogram of LDP map is represented as

$$H = (T_0, T_1, \dots, T_{n-1} \quad (12)$$

The final LDP feature vector (f) is obtained by concatenating the histograms obtained for each non-overlapping window of fingerprint image as shown as

$$f = (H^1, H^2, \dots, H^s) \quad (13)$$

Where s is the number of non-overlaped windows in the image.

C. Generating Rotated Datasets

FVC 2002 DB2 database is used for experiments. This database consists of of 100 fingers and 8 impressions per fingers (total 800 impressions). To prove the rotation invariance of the proposed LDP feature vector, five impressions from each of 8 images of a subjects is obtained by rotated five times and thus generate 40 impressions 3 subject (Total size of generated dataset is now 4000). The rotation is performed with randomly selected angle of rotation in the range$[-10°, 10°]$. During matching process, alignment of fingerprint images affects the gray values of the image and thus affects the matching accuracy. In the Table 1 below the computation of Euclidean distance of original image and its aligned image after rotation.

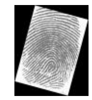
Rotation / angle	5	10	15
Clockwise(A)			
Anti-clockwise (B)			
Eucl. Dist. (C)	7.9088e+003	7.0488e+003	7.6217e+003
Sizes of A	385 × 289	405 × 317	421 × 345
Sizes of B	409 × 325	457 × 385	497× 445

Table 1. Results of rotation of fingerprints with different rotation followed by rotation in opposite direction by same angle and Euclidian Distance is being calculated

IV. CLASSIFICATION USING NEURAL NETWORKS

Single hidden Layer Feed-Forward Neural Network (SLFNN) is used for classification of fingerprint images into two classes match or non-match. In the Literature, numerous pattern classification methods are presented [24-26]. Among them, SLFNNs have been widely used and investigated training algorithm because of its flexible architecture and better generalization capability. This paper presents a comparative analysis of performances of various back propagation algorithms on proposed Fingerprint features. A typical structure of a SLFNNs is shown in Figure 9.

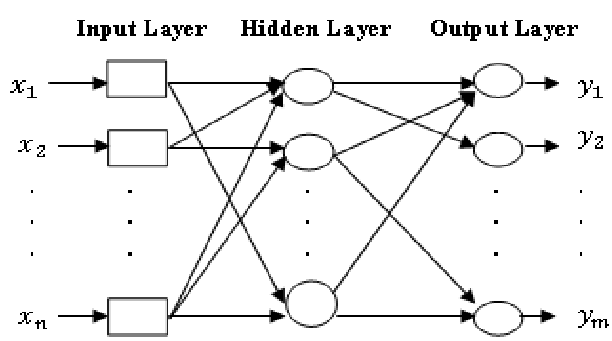


Figure 9. A Typical structure of a SLFNN.

The number nodes at the input layer are determined from the size (number of values) of the input feature vector. The number of nodes at output layer is determined by number of classes used for the fingerprint classification. The numbers of hidden layer nodes are fixed by conducting experiments by trial and error experiments and are specific to the particular problem. General, back propagation is the group of algorithms in which, at the input layer the feature vector is presented and then propagates to the output layer. At output layer a difference between the observed output and the expected output is calculated to represent error and is propagates towards the input layer. In order to minimize the error at output layer the weights and biases are accordingly adjusted. This weights adaptation depends on the optimization techniques used in the different training algorithms: Table 1, below summarizes the weight adaptation strategy used by different back propagation algorithms.

Table 2. Summary of weight adaptation strategy used by different back propagation algorithms

Training Algorithm	Description
traingdx (GDX)	Gradient Decent with adaptive learning rate and momentum forms the basis of weight updation.
traingrp (RP)	resilient backpropagation algorithm (RPROP) forms the basis of weight updation
trainscg (SCG)	Scaled Conjugate Gradient Method forms the basis of Weight updation
trainlm (LM)	Levenberg-Marquardt Optimization forms the basis of Weight updation

V. EXPERIMENTAL RESULTS

Experiments were conducting on Intel Core i5 with 4GB RAM and window 7 using nntoolbox of MATLAB version 7.12.0. Fingerprint Verification Competition 2002 (FVC2002) public dataset was used for experiments. Each dataset contains 110 fingers and 8 impressions per finger (880 fingerprints in all); first 1 to 100 fingers constitute a set A used for verification and fingers 101 to 110 constitute set B to be used by the researcher to tune the parameter of their matching algorithms. These images were scanned using different scanners (including Optical / Capacitive or synthetic fingerprint generator) at a resolution of 500 dpi or higher. This benchmark constitutes 8 impression for each fingers captured at different rotation chosen randomly between $[-10°, 10°]$. A normalized LDP feature vector in the interval $[-0.9, 0.9]$ is thus generated to represent the input feature set. The feature set is then randomly separated into three disjoint sets in the ratio of 15:15:70 to form to be used for training, validation and testing. A *tansig* activation function is used for both output and hidden layers and all the other parameters remain default.

A number of experiments were conducted by changing the values of number of hidden nodes in a step of 5 from 5 to 70. The performance of the proposed algorithm depends on the initialization of the weights at input and output layers. In order to deal with the problem of assigning initial values to the weights, 30 trials have been performed for each experiment. In total $4 \times 14 \times 30 = 1680$ (4 training algorithms, 14 different values of hidden nodes, 30 different trials) experiments were performed to show the validity of the proposed features.

For relative performance of the four different algorithms is analyzed using training time and matching accuracy as performance parameters. Table 2 to 7 shows mean, training error, mean validation error, mean testing error, avg. training time, average number of epochs and matching accuracy at various values of hidden nodes.

The best results in terms of average matching accuracy obtained corresponding to the number of hidden neurons for each experiment are shown in bold print in each table. Figure 10-15 shows the graph for errors and matching rate vs. time and number of hidden nodes for each training algorithm.

Table 3. Experimental results with GDM at various hidden neurons

# of Hidden Nodes	Mean Training Error	Mean Validation Error	Mean Testing Error	Avg. Training Time	Average # of Epoch	Min. Matches	Max. Matches	Avg. Matches	Std
5	0.126	0.149	0.148	31.614	5747	23.21	94.64	68.75	13.18
10	0.042	0.061	0.061	35.44	6000	75	98.21	90.36	6.72
15	0.0328	0.05	0.049	36.035	6000	82.14	100	93.99	5.65
20	0.0319	0.0479	0.0491	36.808	5823	67.86	100	93.45	8.61
25	0.02	0.037	0.0397	38.95	6000	71.43	100	95.48	6.26
30	0.019	0.0379	0.036	40.29	5844	87.5	100	96.13	3.25
35	0.018	0.036	0.034	44.56	6000	76.79	100	96.25	5.35
40	0.013	0.03	0.0288	45.84	6000	92.86	100	98.04	2.11
45	0.0151	0.031	0.033	48.75	6000	85.71	100	96.73	4.22
50	0.0121	0.0283	0.0276	51.78	6000	94.64	100	97.98	1.91
55	0.0111	0.0256	0.0258	53.725	6000	92.85	100	98.22	2.24
60	0.01	0.0268	0.027	55.674	6000	92.85	100	98.1	2.15
65	**0.01**	**0.0261**	**0.0249**	**58.238**	**6000**	**94.64**	**100**	**98.57**	**1.78**
70	0.011	0.0257	0.0282	58.92	5874	78.58	100	97.02	4.55

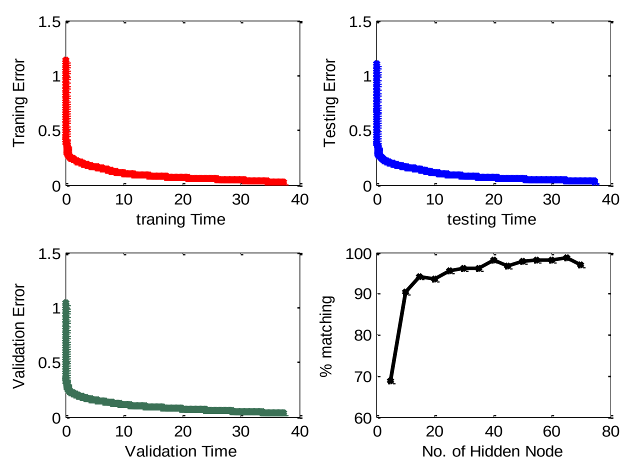


Figure 10. Graph for errors and matching rate Vs. time and number of hidden nodes for RP training algorithm.

Table 4. Experimental results with GDX at various hidden neurons.

# of Hidden Nodes	Mean Training Error	Mean Validation Error	Mean Testing Error	Avg. Training Time	Average # of Epoch	Min. Matches	Max. Matches	Avg. Matches	Std
5	0.0252	0.0263	0.027	0.771	77	5.36	66.071	41.309	17.38
10	0.123	0.137	0.143	0.919	86.7	53.571	83.928	71.964	7.25
15	0.094	0.108	0.108	0.973	86.667	58.928	94.642	82.261	9.307
20	0.09	0.106	0.108	0.999	85.433	60.714	100	82.797	8.66
25	0.0835	0.101	0.103	1.057	85.2	64.285	92.857	82.38	6.425

30	0.081	0.0921	0.0982	1.0431	83.666	58.929	96.429	85.654	6.93
35	0.089	0.104	0.107	1.054	82.4	17.85	100	83.452	14.84
40	0.083	0.101	0.103	1.083	81	58.928	96.428	84.166	8.59
45	0.0864	0.1	0.107	1.114	79.8333	60.714	96.428	80.952	9.35
50	0.081	0.0998	0.102	1.13	78.233	62.5	96.428	84.047	7.61
55	**0.082**	**0.0962**	**0.101**	**1.143**	**77.5**	**69.642**	**100**	**84.821**	**7.2**
60	0.0854	0.0984	0.103	101721	76.7	60.714	98.214	84.761	7.644
65	0.062	0.106	0.104	1.158	74.233	67.857	100	83.035	8.7
70	0.086	0.1	0.102	1.231	76.4333	66.071	100	84.821	7.2

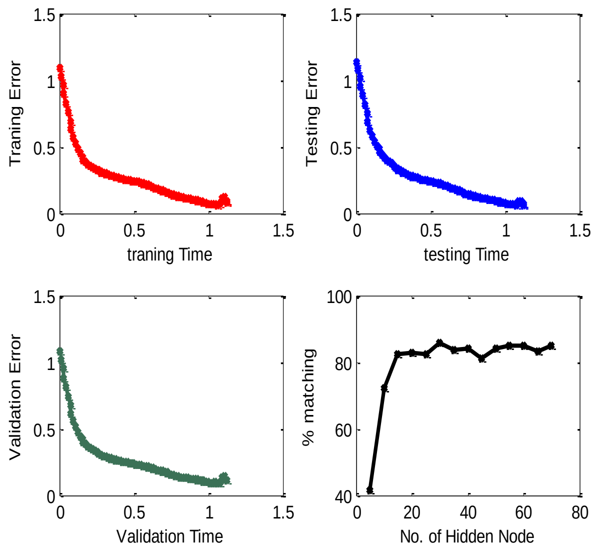


Figure 11. Graph for errors and matching rate Vs. time and number. of hidden nodes for GDX training algorithm.

Table 4. Experimental results with GDA at various hidden neurons.

# of Hidden Nodes	Mean Training Error	Mean Validation Error	Mean Testing Error	Avg. Training Time	Average # of Epoch	Min. Matches	Max. Matches	Avg. Matches	Std
5	0.2711	0.2879	0.2913	0.5769	103.366	0	66.07	40.297	18.9
10	0.08	0.0918	0.095	1.105	191.2	50	100	82.678	13.05
15	0.054	0.067	0.069	1.272	211.4666	64.285	100	88.928	10.594
20	0.027	0.04	0.039	1.893	297.033	67.857	100	95	7.55
25	0.026	0.038	0.039	1.791	272.733	67.857	100	93.928	9.594
30	0.02	0.0291	0.033	1.9	283.033	75	100	95.6	6.356
35	0.0183	0.0319	0.0302	2.234	319.4	73.21	100	96.011	6.69
40	0.0185	0.03	0.0311	2.228	308.7	76.785	100	96.011	6.709
45	0.0182	0.0277	0.0294	2.562	340.7667	80.357	100	96.547	5.818
50	0.0156	0.0256	0.0291	2.338	306.033	76.785	100	95.6	6.66
55	0.022	0.0316	0.0367	2.588	326.8	71.428	100	94.523	8.57
60	**0.022**	**0.0362**	**0.033**	**2.177**	**267.533**	**87.5**	**100**	**97.261**	**3.405**
65	0.0229	0.0348	0.0341	2.629	315.033	80.357	100	95.833	5.515
70	0.0138	0.0215	0.0252	3.456	403.7	76.79	100	96.9	6.16

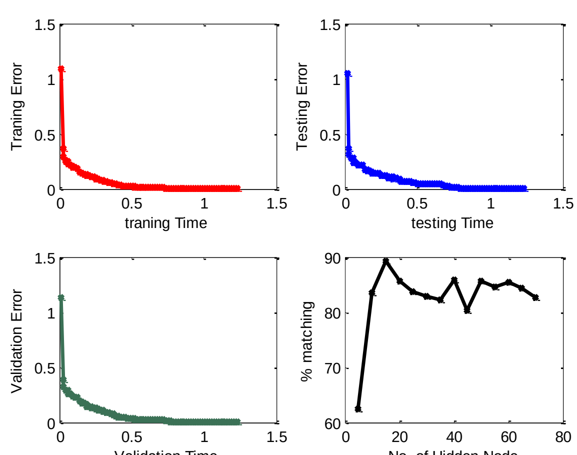


Figure 12. Graph for errors and matching rate Vs. time and number. of hidden nodes for GDA training algorithm.

Table 5 Experimental results with GD at various hidden neurons.

# of Hidden Nodes	Mean Training Error	Mean Validation Error	Mean Testing Error	Avg. Training Time	Average # of Epoch	Min. Matches	Max. Matches	Avg. Matches	Std
5	0.113	0.133	0.139	30.163	5904	48.21	87.5	70.95	9.67
10	0.0459	0.072	0.0687	32.78	6000	69.64	100	88.92	8.485
15	0.0275	0.0467	0.0474	34.868	6000	76.78	100	93.098	5.977
20	0.0251	0.0451	0.0451	36.785	6000	82.142	100	94.524	4.826
25	0.02	0.039	0.037	38.761	5982	80.357	100	95.654	4.159
30	0.0195	0.0391	0.0391	41.559	6000	69.642	100	95.059	6.036
35	0.0155	0.0317	0.0332	43.438	6000	87.5	100	97.083	3.852
40	0.0169	0.0315	0.0329	44.393	5860	87.5	100	97.2	3.74
45	0.011	0.0298	0.028	47.9	6000	91.071	100	97.62	2.666
50	0.011	0.0284	0.029	50.463	6000	91.07	100	97.85	2.62
55	0.011	0.0257	0.0263	56.006	6000	85.71	100	98.035	2.97
60	0.011	0.024	0.026	57.12	6000	85.714	100	97.619	3.192
65	**0.01**	**0.024**	**0.0255**	**58.699**	**6000**	**89.285**	**100**	**98.035**	**2.625**
70	0.01	0.028	0.026	59.46	6000	91.07	100	97.86	2.363

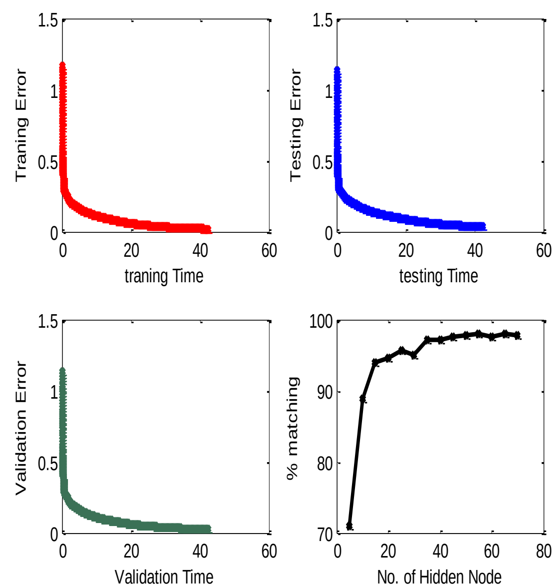


Figure 13. Graph for errors and matching rate Vs. time and number of hidden nodes for GD training algorithm.

Table 6 Experimental results with SCG at various hidden neurons.

# of Hidden Nodes	Mean Training Error	Mean Validation Error	Mean Testing Error	Avg. Training Time	Average # of Epoch	Min. Matches	Max. Matches	Avg. Matches	Std
5	0.125	0.139	0.143	1.02	102.9	5.35	96.428	62.44	22.966
10	0.051	0.056	0.0634	1.071	102.333	10.714	100	83.452	19.7
15	**0.038**	**0.039**	**0.0418**	**1.318**	**115.533**	**19.642**	**100**	**89.1666**	**20.567**
20	0.04	0.047	0.049	1.427	123.166	35.714	100	85.714	20.099
25	0.056	0.057	0.057	1.314	108.466	14.285	100	83.809	26.812
30	0.0565	0.0578	0.06	1.559	123.733	21.428	100	82.976	23.24
35	0.0515	0.0533	0.062	1.5438	119.466	16.0714	100	82.14	26.742
40	0.0479	0.0532	0.0506	1.689	125.333	1.785	100	85.952	24.33

45	0.061	0.0631	0.0667	1.819	131.166	3.571	100	80.238	26.17
50	0.0431	0.046	0.0502	1.927	133.666	26.785	100	85.535	20.937
55	0.0473	0.0521	0.0532	2.023	136.1	12.5	100	84.5238	22.94
60	0.0473	0.0514	0.0497	2.2151	146.866	10.714	100	85.357	24.77
65	0.0497	0.0515	0.0523	2.287	145.366	25	100	84.86	22.021
70	0.056	0.062	0.0591	2.117	129.8	16.071	100	82.55	26.318

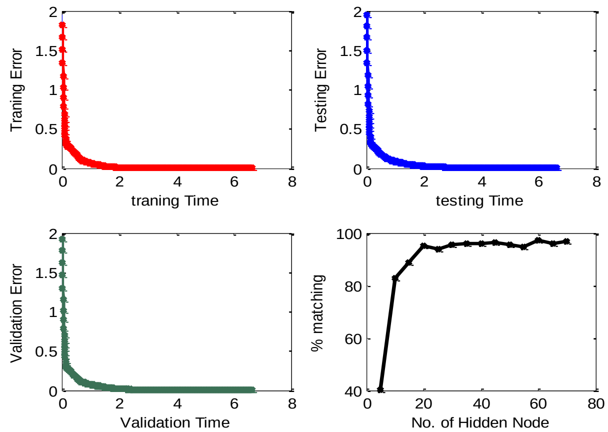


Figure 14. Graph for errors and matching rate Vs. time and number of hidden nodes for SCGtraining algorithm.

Table 7. Experimental results with RP at various hidden neurons.

# of Hidden Nodes	Mean Training Error	Mean Validation Error	Mean Testing Error	Avg. Training Time	Average # of Epoch	Min. Matches	Max. Matches	Avg. Matches	Std
5	0.057	0.0896	0.0972	0.0564	93.6	62.5	96.42	81.726	8.54
10	0.0136	0.0339	0.0326	0.484	76.3	67.857	100	93.809	6.951
15	0.002	0.0137	0.0148	0.0442	68.466	89.285	100	98.333	2.478
20	0.0016	0.0086	0.011	0.384	57.633	91.071	100	98.392	2.1154
25	0.0009	0.0078	0.014	0.388	57.2333	89.285	100	98.035	2.452
30	0.0012	0.008	0.0104	0.336	49.4333	92.857	100	98.93	1.729
35	0.0019	0.0092	0.0115	0.339	48.5666	87.5	100	98.75	2.53
40	0.0010	0.0059	0.0123	0.341	47.166	89.285	100	98.27	2.458
45	0.0014	0.00944	0.0097	0.3515	46.1667	91.07	100	98.571	2.21
50	0.001	0.00568	0.0092	0.4054	50.1667	91.071	100	98.809	2.114
55	0.0019	0.0078	0.0093	0.4065	49.633	92.8571	100	98.869	1.96
60	0.0014	0.00569	0.0107	0.04035	48.1	94.642	100	98.64	1.6
65	0.0015	0.0057	0.0091	0.4285	49.3667	91.071	100	98.69	1.985
70	0.0029	0.00763	0.0094	0.4	45.033	92.857	100	98.809	2.007

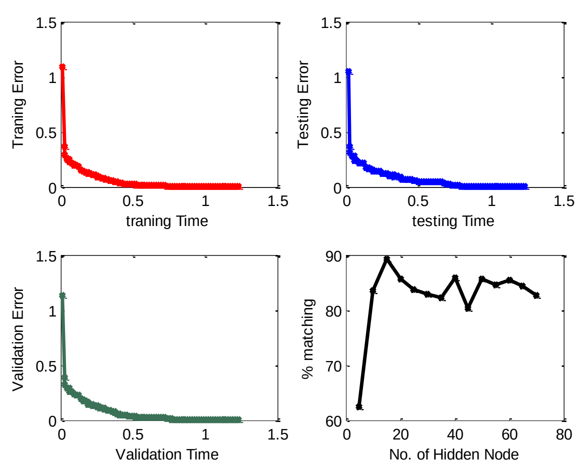


Figure 15. Graph for errors and matching rate Vs. time and number of hidden nodes for RPtraining algorithm.

From the above results, it is clear that GDM performs better in comparison with other gradient decent training algorithms (GD, GDX, and GDA). On the other hand, resilient propagation (RP) performs much better that SCG and LM.

VI. Conclusion

This paper exploits the local directional pattern (LDP) considering only the first four ranks (K=4) and presents a robust fingerprint matching system invariant under rotation. Matching of the extracted features is performed using Single Hidden Layer Feed-forward neural network (SLFNN) trained using three different categories of training algorithms. The results of the experiments performed on FVC 2002fingerprint database shows the superiority of Resilient Propagation (RP) training algorithm in term of convergence time and matching accuracy over other class of training algorithms like, GDM and SCG. The average rate of 98.809 % shows invariance and suitability of proposed features for fingerprint matching. Exploiting the other NN based classifiers for fingerprint matching can be future research direction of this work.

References

[1] D. Maltoni, D. Maio, A. K. Jain and S. Prabhakar, Handbook of Fingerprint Recognition, Springer-Verlag, June 2009.

[2] Berry, John, and David A. Stoney. "The history and development of fingerprinting." Advances in fingerprint Technology Vol.2 pp.13-52 , 2001

[3] Galton, Francis. Fingerprint directories. Macmillan and Co., 1895.

[4] D. Maltoni, D. Maio, A. K. Jain and S. Prabhakar, Handbook of Fingerprint Recognition, Springer-Verlag, June 2009.

[5] Jiang, Xudong, and Wei-Yun Yau. "Fingerprint minutiae matching based on the local and global structures." Pattern recognition, 2000. Proceedings. 15th international conference, IEEE Vol. 2, 2000.

[6] M. Tico and P. Kuosmanen, "Fingerprint matching using and orientation-based minutia descriptor," IEEE Trans. Pattern Anal. Mach. Intell., Vol. 25, No. 8, pp. 1009–1014, 2003.

[7] J. Qi, S. Yang, and Y. Wang, "Fingerprint matching combining the global orientation field with minutia," Pattern Recognition. Lett., Vol. 26, pp. 2424–2430, 2005.

[8] Jain, Anil K., Salil Prabhakar, Lin Hong, and Sharath Pankanti. "Filterbank-based fingerprint matching." Image Processing, IEEE Transactions Vol. 9, No. 5, pp.846-859, 2000

[9] Sha, Lifeng, Feng Zhao, and Xiaoou Tang. "Improved fingercode for filterbank-based fingerprint matching." In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference, IEEE, Vol. 2, pp. 895-898, 2003.

[10] Nanni, Loris, and Alessandra Lumini. "Local binary patterns for a hybrid fingerprint matcher." Pattern recognition Vol.41, No. 11 pp.3461-3466, 2008

[11] Tico, Marius, P. Kuosmanen, and J. Saarinen. "Wavelet domain features for fingerprint recognition." Electronics Letters Vol.37, No. 1, pp. 21-22, 2001

[12] Amornraksa, T., and S. Tachaphetpiboon. "Fingerprint recognition using DCT features." Electronics Letters Vol.42, No. 9 pp. 522-523, 2006
[13] Andrew Teoh Beng, David Ngo Chek Ling, and Ong Thian Song. "An efficient fingerprint verification system using integrated wavelet and Fourier–Mellin invariant transform." Image and Vision Computing Vol.22, No. 6, pp. 503-513, 2004
[14] Ross, Arun, Anil Jain, and James Reisman. "A hybrid fingerprint matcher." Pattern Recognition Vol.36, No. 7 pp. 1661-1673, 2003
[15] Benhammadi, F., M. N. Amirouche, H. Hentous, K. Bey Beghdad, and M. Aissani. "Fingerprint matching from minutiae texture maps." Pattern Recognition, Vol.40, No. 1, pp.189-197, 2007
[16] R. Cappelli, M. Ferrara, D. Maltoni, and M. Tistarelli, "MCC: A baseline algorithm for fingerprint verification in FVC-onGoing," in Proc. 11th Int. Conf. Control, Automation, Robotics and Vision, pp. 19-23, 2010.
[17] L. Hong, A. Jain, "Classification of fingerprint images", in: 11th Scandinavian Conference on Image Analysis, Vol. 2, pp. 665-672, 1999.
[18] Masahiro Kawagoe , Akio Tojo, "Fingerprint pattern classification", Pattern Recognition, Vol 3, pp.295-303, 1984.
[19] Jain, A.K., Prabhakar, S., Hong, L. and Pankanti, S., "Filterbank-based fingerprint matching", IEEE Trans. Image Process. Vol.9(5), pp. 846-859, 2000.
[20] Ju Cheng Yang , Dong Sun Park, "A fingerprint verification algorithm using tessellated invariant moment features", Neurocomputing, Vol.71, No.10-12, pp.1939-1946, 2008.
[21] Jang X, Yau WY "Fingerprint minutiae matching based on the local and global structures." In: Proceedings of international conference on pattern recognition, Vol. 2, pp 1024–1045, 2000.
[22] Kumar, R., Chandra, P., & Hanmandlu, M. "A Robust Fingerprint Matching System Using Orientation Features." Journal of information processing systems, Vol.12(1), pp.83-99, 2016.
[23] Jabid, T., Kabir, M. H., & Chae, O. "Local directional pattern (LDP)–A robust image descriptor for object recognition." In Advanced Video and Signal Based Surveillance (AVSS), Seventh IEEE International Conference pp. 482-487, 2010.
[24] Simon Haykin,"Neural Network and Learning Machines",Third Edition, Prentice Hall India, 2009.
[25] Mahajan, A., Singh, H. P., & Sukavanam, N. "An unsupervised learning based neural network approach for a robotic manipulator." International Journal of Information Technology, Vol.9(1), pp.1-6, 2017.
[26] Sinha, G. R. "Study of assessment of cognitive ability of human brain using deep learning." International Journal of Information Technology, Vol.9(3), pp.321-326, 2017
[27] Kumar, Ravinder, Pravin Chandra, and M. Hanmandlu. "Fingerprint matching based on texture feature." Mobile communication and power engineering. Springer, Berlin, Heidelberg, pp.86-91, 2013
[28] Kumar, Ravinder, Pravin Chandra, and Madasu Hanmandlu. "Rotational invariant fingerprint matching using local directional descriptors." International Journal of Computational Intelligence Studies Vol.3.4, pp.292-319, 2014
[29] Kumar, Ravinder, Pravin Chandra, and Madasu Hanmandlu. "Local directional pattern (LDP) based fingerprint matching using SLFNN." Image Information Processing (ICIIP), Second International Conference on. IEEE, 2013.
[30] Kumar, Ravinder, Pravin Chandra, and M. Hanmandlu. "Fingerprint matching based on orientation feature." Advanced materials research. Vol. 121, pp.83-99, 2012.
[31] Kumar, Ravinder, Pravin Chandra, and Madasu Hanmandlu. "Fingerprint matching using rotational invariant image based descriptor and machine learning techniques." Emerging Trends in Engineering and Technology (ICETET), 6th International Conference on. IEEE, 2013.
[32] Kumar, Ravinder, Madasu Hanmandlu, and Pravin Chandra. "An empirical evaluation of rotation invariance of LDP feature for fingerprint matching using neural networks." International Journal of Computational Vision and Robotics Vol.4.4 pp.330-348, 2014.
[33] Kumar, Ravinder, Pravin Chandra, and M. Hanmandlu. "Fingerprint singular point detection using orientation field reliability." Advanced Materials Research, Trans Tech Publications ,Vol. 403, 2012.
[34] Kumar, Ravinder, Pravin Chandra, and M. Hanmandlu. "Statistical descriptors for fingerprint matching." International Journal of Computer Applications Vol.59.16, 2012.
[35] Kumar, Ravinder, Pravin Chandra, and M. Hanmandlu. "Information Theoretic Approach for Fingerprint Matching." Proc. Int. Conf. on Advances in Computing, Control, and Telecommunication Technologies,(ACT) 2012.
[36] Kumar, Ravinder, and Brajesh Kr Singh. "Empirical analysis of contents based image retrieval using gabor feature extractor." International Journal of Advanced Research in Computer Science Vol. 8, Issue.7, pp.1015 -1020, 2017.
[37] Kumar, Ravinder. "A Robust Biometrics System Using Finger Knuckle Print." Handbook of Research on Network Forensics and Analysis Techniques. IGI Global, pp.416-446, 2018.
[38] Kumar, Ravinder. "Hand Image Biometric Based Personal Authentication System." Intelligent Techniques in Signal Processing for Multimedia Security. Springer, Cham, pp.201-226, 2017.
[39] Ravinder Kumar, Brajesh Kumar Singh, "Performance evaluation of Invariant moment features on Image retrieval", International Journal of Computer Sciences and Engineering, Vol.5, Issue.12, pp.73-78, 2017.
[40] Kumar, Ravinder. "A Review of Non-Minutiae Based Fingerprint Features." International Journal of Computer Vision and Image Processing , Vol.8, No.1, 32-58, 2018.

Authors Profile

Dr. Ravinder Kumar received Ph. D. in IT from GGSIP University, Delhi in 2013 and M. Tech. degree in Computer Science & Engineering in 1998 from GJ University of Science and Technology, Hisar, India. Since 1999, he has been with the University School of ICT, GGSIP University, Delhi. Currently, he is Professor and Head, Department of CSE with HMR Institute of Technology and Management Delhi, India. His research interest is in the image processing and biometrics.He has 18 years of teaching experience and 8 years of Research Experience.

