Bee Swarm based Feature Selection for Fake and Real Fingerprint Classification using Neural Network Classifiers

V. Sasikala and V. Lakshmi prabha

Abstract— **With the emergent exercise of biometric authentication systems, fake and real fingerprint classification has become an attractive research area in the last decade. A number of research works have been carried out to classify fake and real fingerprints. But, most of the existing techniques did not utilize swarm intelligence techniques in their fingerprint classification system. Swarm intelligence has been widely used in various applications due to its robustness and potential in solving a complex optimization problem. The main aim of this paper is to develop a new and efficient fingerprint classification approach based on swarm intelligence with fuzzy based neural network techniques to overcome the limitations of the these classification approaches. The proposed classification methodology comprises of four steps, image preprocessing, feature extraction, feature selection and classification. This work uses efficient min-max normalization and median filtering for preprocessing, and multiple static features are extracted from Gabor filtering. Then, from the multiple static features obtained from 2D Gabor filtering, best features are selected using Artificial Bee Colony (ABC) optimization based on its searching capability. This optimization based feature selection selects only the optimal set of features which is used for classification. This would lessen the complexity and the time taken by the classifier. This approach uses Fuzzy Feed Forward Neural Network (FFFNN) for classification and its performance is compared with the SVM classifier. The performance and evaluations are performed using fingerprint images collected from FVC2000 and synthetically generated database using SFinGE. It shows that proposed work provides better results in terms of sensitivity, precision, specificity and classification accuracy.**

Index Terms— **Fake and real Fingerprint classification, multiple static features, normalization, median filtering, Gabor filtering, Artificial Bee Colony (ABC) optimization, Fuzzy Feed Forward Neural network (FFFNN)**

I. INTRODUCTION

BIOMETRIC systems have been widely used in various applications such as access control, law enforcement systems and border management systems for human identification based on biological traits such as face, iris, retina, etc [1]. Nowadays, numerous approaches have been developed in order to fulfill the growing demand for security. Among all the biometric traits aside, fingerprints are being extensively used in various applications. They are highly distinguished and unique, even for identical twins, and are publicly accepted as reliable traits. The ridges and valleys are the main reasons for the distinguishing shapes of the fingerprint. The singular regions namely loop and delta produced by the ridges are the main factors used in fingerprint classification. The ridges would also represent the global attributes of the fingerprint through their unique orientation and frequency.

Recent investigations have [2, 3] showed that biometric systems are being subjected to various threats. The main issue and challenge is to classify whether the biometric fingerprint is real or fake. In fact, it is difficult to make a fake fingerprint image having the better image quality than that of the original. In general, several approaches have been developed by various researchers for the classification of fake and real fingerprints. Among the existing approaches, Thaiyalnayaki et al [4] detected the liveness of a fingerprint by computing the standard deviation of the fingerprint image through the wavelet transform. This work has contributed an essential technique that can detect the liveness by observing the image quality.

The fingerprint dummies can be fabricated through typical materials like gelatin, silicone or latex. These fake fingerprints are created by the intruders to get falsely accepted by the biometric system. Thus, fingerprint classification has been an attractive research area in the last decade. Generally, classification techniques consist of four steps, namely preprocessing, feature extraction, feature selection and classification. Image preprocessing becomes one of the essential steps in biometric systems to eliminate noise from fingerprint images and fake images. Wang and Bhattacharjee proposed an adaptive image preprocessing technique based on their noise level and contrast stretching capability through their power-law transformation and Gabor filter [5]. However, none of the above approaches utilize a normalization approach to eliminate noise in the images.

In this paper, min-max normalization and median filtering approaches are used as image preprocessing steps to eliminate noises from fake and real fingerprint images. The static technique is useful in extracting features, but the major limitation of this approach is that it makes a decision based on only a single image [4]. This would result in the degradation of the classification performance. In order to overcome these issues, this work extracts multiple statistical features such as power spectrum, directional contrast, ridge thickness, ridge signal, and first order histogram, of the fingerprint images using Gabor filtering. After efficient multiple statistical features are extracted, ABC optimization technique is used which selects the best features from the extracted features and then classification is carried out through FFFNN classifier which classifies the real and fake fingerprint. Thus, this research work mostly focuses on developing an efficient fingerprint classification approach with lesser complexity and higher accuracy.

Manuscript received September 22, 2014; revised August 26, 2015.
V. Sasikala is Associate Professor, SNS College of Engineering, Coimbatore, India; (e-mail: v_sasikala@yahoo.com).
V. Lakshmi prabha is Principal, Government College of Technology, Coimbatore, India (e-mail: vlakshmiprabha@yahoo.com).

II. Related Work

A number of existing approaches can be partitioned into hardware-based approaches and software-based approaches [6]. Hardware based approaches focus on detecting the fake fingerprint through additional hardware tools and ability to measure physiological signs. The software-based systems are observed to be inexpensive and less conspicuous. These approaches use feature vectors obtained from one or multiple impressions of the same finger to distinguish real and fake fingers [7].

Wavelet-based techniques were initially used in fake fingerprint detection [8-9], but recently new approaches based on the wavelet transform is presented which can detect the perspiration event using only a single image. Statistical features are extracted for multiresolution scales to distinguish between real and fake fingers. Classification has been carried out using classification trees and neural networks with accuracy of 90.9% on different datasets. In [11], a new approach for discriminating fake fingerprints from real ones has been proposed, based on the study of the distortion effects in fingerprint matching process using spatial characteristics [12]. New techniques for extracting, encoding and comparing skin distortion information are formally defined and measured over a set of real and fake fingers.

Fake fingerprints are used in attempts to get falsely accepted by the biometric system. The fingerprint dummies are fabricated using typical materials like gelatin, silicone or latex. The weakness of the fingerprint based biometric liveness detection technologies were introduced based on skin odor [13]. A new method based on the distribution of minutiae and the orientation field was proposed in [14] [15]. The minutiae are almost uniformly spread in the natural fingerprint area while in the altered fingerprint area they appear in an excessive number, many of them being spurious. The method was tested at finger level and at subject level, on a real altered fingerprints database [16] and compared with the finger print image criterion [17]. It was proven that fingerprint quality estimation methods are not sufficient to detect fingerprints alteration. Fingerprint pattern recognition system using huffman coding [18] introduced new features that can be used for matching.

III. Proposed Fingerprint Classification Methodology

This paper proposes an efficient fingerprint classification method to classify the fingerprint images as fake and real fingerprint image in an efficient manner. Photographs and grayscale figures should be prepared with 300 dpi resolution and saved without compressing, 8 bits per pixel (grayscale). This work initially removes irrelevant noise from original and fake fingerprint image samples to increase both classification accuracy and interpretability of the digital data during the image pre-processing stage. The major steps of pre-processing are image enrichment, binarization, distance transform, segmentation and filtering. In this work, normalization is used as preprocessing step to perform image contrast enhancement and median filtering methods are used to remove the noises from samples. Then, multiple static features extraction are carried out for images which uses Gabor filtering method and thus fulfilling user requirements such as expediency, time complexity and accuracy. From the extracted features, best features are obtained from the Artificial Bee Colony (ABC) optimization algorithm based on the fitness function and then a Fuzzy Feed Forward Neural Network (FFFNN) is used as a classifier. These four steps are explained in detail in the following sections. The entire architecture of the proposed fingerprint Classification method is shown in Fig 1.

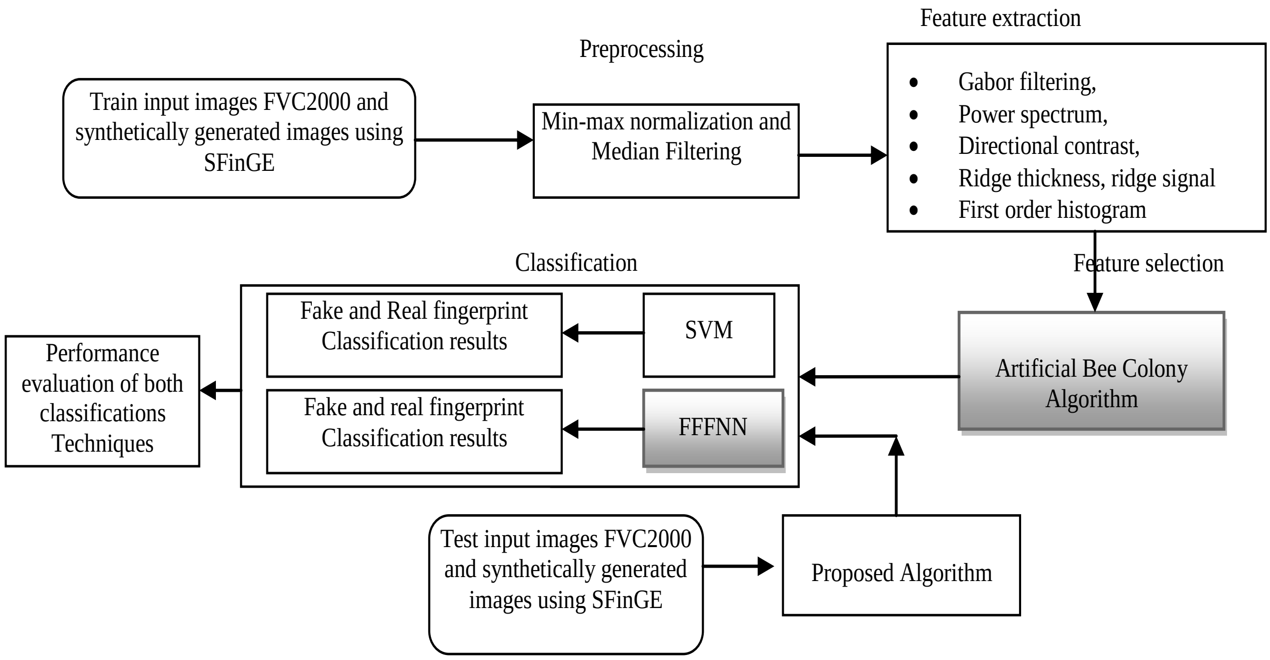


Fig 1. Proposed Fake and Real Fingerprint Classification Methodology

A. Min-max normalization for contrast Enhancement

In image processing, normalization is a process that adjusts the series of pixel intensity values, particularly when the contrast level of the images is low due to clarity. In this work, a Min–Max normalization method that adjusts the range of pixel intensity values for better clearness is used. It carries out a linear transformation function of the original input image. It is measured that min_A and max_A are the minimum and maximum range of pixel intensity values in the input image range $(\mathrm{min}_A, \mathrm{max}_A)$, into a new image range v' in the range $(\mathrm{new\_min}_A, \mathrm{new\_max}_A)$, y calculating,

$$v' = \frac{v - \mathrm{min}_A}{\mathrm{max}_A - \mathrm{min}_A} \quad (1)$$

Min-max normalization method should maintain the same pixel intensity values for original images. If the intensity values of original input image values (A) are changed, it will be encountered as out of range error for future prediction of normalization process. Thus, if the intensity range of the given image is between 30 and 150 and the required image intensity range is between 0 and 255, the normalization process starts with subtracting 30 from each given image of pixel intensity, making the range between 0 and 120.

B. Median filtering method for noise removal

Median filter is one of the commonly used non linear filtering methods used to reduce noise from image samples. Such noise reduction technique is a classic pre-processing step to enhance the results of processing. The sliding median filter of a pre-specified image window size $W \times W$ centered at image pixels $i = (i_1, i_2)$ progress consistently over the noisy image, *g* and selects median μ of the pixels within a specified range of pixels for spatial domain Ω_i^W roughly i to have $g(i)$ and noisy image $g(i)$ is replaced by μ . For the set of pixels within a square window $W_D \times W_D$,centred at $i = (i_1, i_2)$and defined specified range of pixels for spatial domain Ω_i^W approximately by equation, the median, $u(i)$ the pixels in spatial domain Ω_i^W is

$$u(i) = \mu_i = \mathrm{median}\left\{\frac{g(j)}{j} \in \Omega_i^W\right\} \quad (2)$$

Thus, the output of the median filter is defined as θ which produces lesser error rate results with the entire pixels in the local neighborhood defined by the mask. The output of the median filter at spatial location i can also be specified as $\mu_i = \underset{\theta}{\mathrm{argmin}} \sum_{r \in \Omega^W} |g(r) - \theta|$.

Fig 2 shows both the real and fake fingerprint image samples obtained from the Fingerprint Verification Competition (FVC2000) database and the remaining two images are the fake fingerprint images generated from and SFinGE software tool [19].

Fig 3 shows the image samples after the Gaussian noise are added to images samples for both real and fake fingerprint images samples.

Fig 2. Input image samples

Fig 3. Gaussian Noise Incorporated Image

C. Gabor filtering for multiple static feature extraction

In order to enhance classification performance for images, it is not sufficient to extract single static features from images. Since the characteristics of input fingerprint samples vary according to their categories of sensor and characteristics of fake and synthetically generated image samples which are based on conditions such as user skin, surrounding, fabrication materials, etc. In order to obtain better classification performance, it is desirable to extract specific static features. A Gabor filter-based multiple static feature extraction is proposed in this section. In this work, certain important multiple static features such as power spectrum, histogram, directional contrast, ridge thickness, and ridge signal are extracted from each and every fingerprint image and it provides the best description of the visual substance of fingerprint images. Based on this motivation, two-dimensional Gabor filtering is being selected for feature extraction in this approach. Thus, a bidimensional Gabor filter represents a complex sine wave plane of specific frequency and ridge orientation levels, it is transformed by a Gaussian envelope. It achieves an optimal resolution in both spatial and frequency domains.

$$G_{\theta_k, f_i, \sigma_x, \sigma_y}(x,y) = \exp\left(-\left[\frac{x_{\theta_k}^2}{\sigma_x^2} + \frac{y_{\theta_k}^2}{\sigma_y^2}\right]\right).\cos(2\pi f_i x_{\theta_k} + \varphi) \tag{3}$$

Where $x_{\theta_k} = x\cos\theta_k + y\sin\theta_k$, $y_{\theta_k} = y\cos\theta_k - x\sin\theta_k$, f_i present the central frequency of the sine wave at an angle θ_k with the x-axis, σ_x, σ_y represents the standard deviations of ridges together with the axes x and y match to image size. Set the phase $\varphi = \frac{\pi}{2}$ and compute each and every ridge orientation as $\theta_k = \frac{k\pi}{n}$ where $k = \{1, \ldots, n\}$.Thus, certain proper variance values are considered which are a set of radial frequencies of the ridges in the image and a sequence of orientations. Consequently, the filter's parameters are considered as$\sigma_x = 2\sigma_y = 1$, $f_i \in \{0.75, 1.5\}$ is represented as the frequency differentiation of the features and n=5, $\theta_k \in \left\{\frac{\pi}{5}, \frac{2\pi}{5}, \frac{3\pi}{5}, \frac{4\pi}{5}, \pi\right\}$is applied to fingerprint images. The resulted Gabor filter is then grouped into a three-dimensional feature vector. If $'I'$ characterizes such a fingerprint image, then a $[X \times Y]$ size is included, and its feature extraction can be specified as follows:

$$V(I)[x,y,z] = V_{\theta(z), f(z), \sigma_x, \sigma_y}(I)[x,y] \tag{4}$$

Where $x \in [1, X], y \in [1, Y]$ and

$$\theta(z) = \begin{cases} \theta_z, & z \in [1,n] \\ \theta_{z-n}, & z \in [n+1, 2n] \end{cases}, f(z) = \begin{cases} f_1 z \in [1,n] \\ f_2, z \in [n+1, 2n] \end{cases} \tag{5}$$

and

$$V_{\theta(z), f(z), \sigma_x, \sigma_y,}(I)[x,y] = I(x,y) \otimes G_{\theta(z), f(z), \sigma_x, \sigma_y}(x,y) \tag{6}$$

An efficient 2D Gabor filtering method can be performed using Fast Fourier Transform; consequently it is equivalent with the following relation

$$V_{\theta(z), f(z), \sigma_x, \sigma_y,}(I) = FFT^{-1}\left[FFT(I).FFT\left(G_{\theta(z), f(z), \sigma_x, \sigma_y}\right)\right] \tag{7}$$

After the features are extracted, these features are given to the feature selection algorithm in order to select the best features. This work uses ABC algorithm for best feature selection. Fig 4 shows the feature extraction results of real and fake fingerprint image samples with Gabor orientation. Fig 5 shows the Gabor images for real and fake fingerprint images after the Gabor filtering is applied.

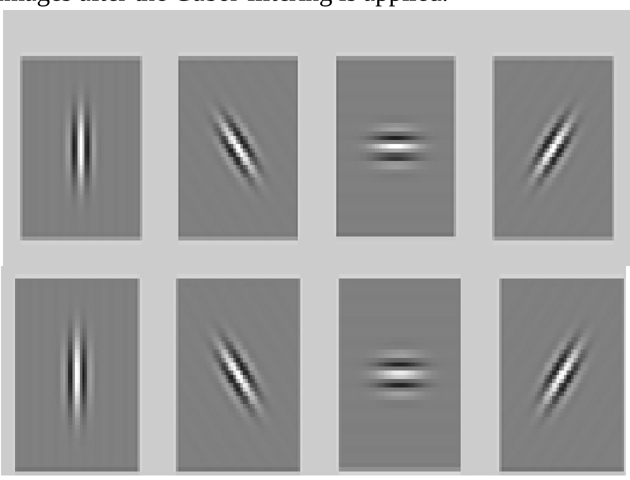


Fig 4. Gabor Orientation Images

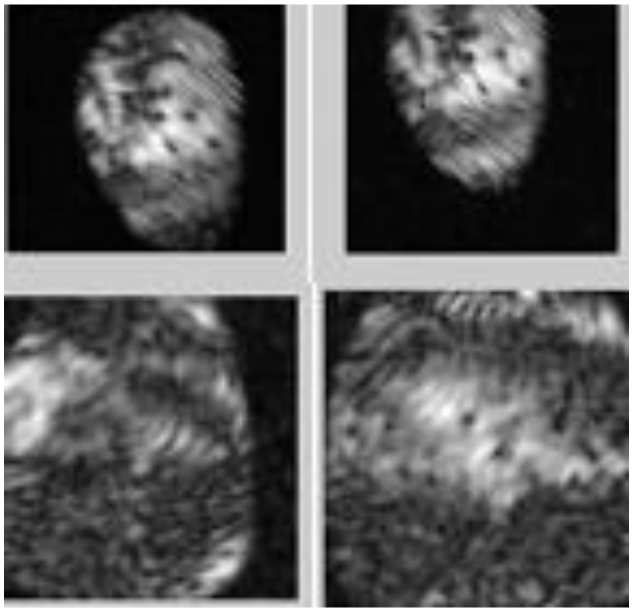


Fig 5. Gabor Images

D. Artificial Bee Colony (ABC) optimization for multiple static feature selection

Artificial Bee Colony (ABC) is one of most important swarm intelligence based optimization algorithms. It has been successfully used for feature selection optimization [20] as it is easy to develop and solve many optimization problems with only a few controls of parameters [21]. ABC suggests the intellectual searching behavior of a honey bee swarm. In ABC, the dependency of artificial bees contains three major groups of bees namely employed bees, onlookers and scouts. As an initial step, initial populations of size SN is randomly generated, where SN (total number of input fingerprint samples with feature extracted results) denotes the size of the population. Each feature selection, solution $x_i, (i = 1, \ldots, SN)$ is a D-dimensional vector. Here, D is the number of optimized parameters. After initialization of features, each population has a number of features positions which is subjected to a maximum number of cycles, $C = 1, \ldots, MCN$, to complete feature selection search process of the bees.

Employed bees perform the local investigation of best feature selection and try to exploit the neighboring locations of features such as power spectrum, directional contrast, ridge thickness, ridge signal, and first-order histogram, food source and search the best places of features food source in the nearby areas of the present value. The bees waiting in the nest area to choose important feature are termed as onlooker bees. The decision of feature selection is made on information about multiple static features by employing bees. Onlooker bees perform global investigation for discovering new multiple static feature selection results and update global optimum multiple static feature selection results. Scout bees randomly search for each multiple static feature selection. Scout bees discover the new features selection areas which are not focused by the employed bees, these bees are completely random in their operation of search. Scout bees avoid the search process to get trapped in local minima. These three steps are continued until a termination criterion is satisfied.

The position of each multiple static feature solution represents a probable solution to the best feature selection and the nectar amount of a feature solution corresponds to the quality of best multiple static features that associates with each one of the features.

$$\mathrm{fit_i} = \frac{1}{1+\mathrm{fit_i}} \quad (8)$$

The fitness of each of multiple static features is assigned randomly based on the importance of the multiple static features. The importance of each multiple static feature is separately estimated. The fitness of each static feature value is described in table 1. The power spectrum values depend on ridge-ridge distance level. The histogram features are selected based on entropy measures. If the corresponding image feature is greater than the entropy value, then the feature is chosen else eliminated. The ridge thickness is estimated based on gray level values of every block in a way usual to the ridge orientation. When ridge thickness gray level values reach the threshold value, it is selected, else it is not selected. The individual fitness condition for each static feature is mentioned in table I.An artificial onlooker bee selects best static features based on the probability value associated with that feature space $\mathrm{p_i}$, calculated by the following expression,

$$\mathrm{p_i} = \frac{\mathrm{fit_i}}{\sum_{n=1}^{SN} \mathrm{fit_n}} \quad (9)$$

Where $\mathrm{fit_i}$ represents the fitness value of the feature solution i in the location and SN is the number of extracted feature results for images, which is equivalent to the number of employed bees.

In order to generate a candidate feature selection position from the earlier feature selection result, ABC uses the following equation (8) and update its location,

$$\mathrm{v_{ij}} = \chi_{ij} + \phi_{ij}\left(\chi_{ij} - \chi_{kj}\right) \quad (10)$$

Where $k \in \{1,.,SN\}$ and $j \in \{1,.,D\}$.are randomly selected feature samples, $\phi_{ij} \in [-1,1]$ it is used to control the production of nearest optimal feature selection sources approximately x_{ij} and represents the evaluation of two optimal feature selection locations visible to a bee. As it is observed from the equation (8), the difference among two different features extracted from image samples of $\chi_{i,j}$ and $\chi_{k,j}$ decreases, the rest of the feature selection position χ_{ij} decreases. So that, the step length adaptation for the optimum feature selection solution in the searching process is reduced. From this result, the parameter value of χ_{ij} exceeds its threshold value and the result of feature selection is chosen as best features, else it is not considered and is replaced with a new feature selection result by the scouts bees which is defined in equation (9). In ABC, these static feature selection operations are replicated by producing a new feature selection position of randomly selected static features and changing it with the discarded one. In ABC, if a current feature selection position does not improve the result within a pre-specified number of iterations, then the current features selection position is assumed to be neglected.

TABLE I
FITNESS CONDITION FOR STATIC FEATURES

Features	Fitness condition
Power spectrum	Ridge-to-ridge distance (500 dot/in)
Histogram and Contrast	Entropy
Ridge thickness, and Ridge signal	Best gray level values

In equation (10), $\phi_{ij} \in [-1,1]$ is randomly generated which in turn decreases the result of the feature selection .In order to overcome this problem, Gaussian distribution function with zero mean and standard deviation value of the current feature samples is introduced in this work which is represented as follows

$$\mathrm{v_{ij}} = \chi_{ij} + G(0,\sigma^2) \quad (11)$$

Where $G(0,\sigma^2)$ be the Gaussian distribution with zero mean and standard deviation of the current feature samples.

$$\chi_i^j = \chi_{min}^j + \mathrm{rand}(0,1)\left(\chi_{max}^j - \chi_{min}^j\right) \quad (12)$$

Then, feature selection sample position $\mathrm{v_{ij}}$ is estimated and its performance is compared with each one of the previous features selection results. If the new feature selection result is better than previous selected feature results, it is replaced with the old feature selection results in the memory. Or else, old feature selection result is kept as such. In other words, a greedy selection system works for the selection of feature operation between newly selected features and subset features. ABC algorithm employs four different selection processes which are explained below.

A global probabilistic selection process for each multiple static features such as power spectrum, directional contrast, ridge thickness, ridge signal, and first-order histogram, in which the probability value is calculated by equation (7) used by the onlooker bees for discovering promising multiple static feature regions.

A local probabilistic multiple static feature selection process for fake and real fingerprint images is carried out in a region by the employed bees and the onlooker bees based on the visual information of features and is named as greedy selection, if quality feature selection results are not achieved, the current feature selection results are not considered and memorizes the candidate source produced by the equation (8).

Bees keep the current multiple feature selection results for the quality feature selection results.

Multiple static features are randomly selected and it is done through scout bees as defined in equation (9).

All the above mentioned steps majorly depend on three parameters which restrict the operation of multiple static feature selection: The number of food sources which is equal to the number of image samples from feature extraction results (SN), maximum and minimum number of iterations to complete multiple feature selection process(MNC).

Algorithm 1: Artificial Bee Colony (ABC) optimization for multiple statistical feature selection

Step 1: Initialize the population of solutions $x_i, i = 1, \ldots\ldots SN$,each population as a number of features $x_1 = \{$Power spectrum, ridge thickness directional contrast, ridge signal, and first -order histogram$\}$

Step 2: Evaluate the population with features

Step 3: Set cycle = 1

Step 4: Repeat

Step 5: Produce new feature selection solutions v_i for the employed bees (features) by using (8) and evaluate them best feature

Step 6: Apply the greedy selection process for the employed bees are considered as features

Step 7: Calculate the probability values P_i for the feature solutions x_i by (7)

Step 8: Produce the new feature solutions v_i for the onlookers from the solutions X_i selected depending on P_i and evaluate them

Step 9: Apply the greedy selection process for the onlookers are considered as features

Step 10: Determine the abandoned feature solution for the scout, if exists, and replace it with a new randomly produced solution χ_i^j by (9)

Step 11: Memorize the best solution achieved so far

Step 12: cycle = cycle + 1

Step 13: until cycle = MCN

For those selected features from ABC then perform classification methods to classify feature samples results into fake and real images. FFFNN is used for classification and it makes decision either fake or real image.

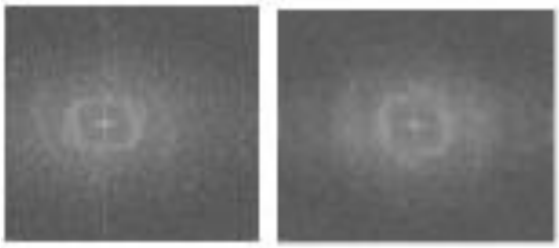
Table II and Table III show the results of the features of the real and fake fingerprint image samples. The slight variations in the features of the real and fake fingerprint images are clearly seen in the above tables.

TABLE II: SAMPLE FEATURES FOR REAL FINGERPRINT IMAGES

0.1565	0.1562	0.1534	0.1461	0.1339	0.1188	0.1037	0.0912	0.0819	0.0751
0.1563	0.1555	0.1521	0.1443	0.1318	0.1168	0.1023	0.0906	0.0822	0.0761
0.1541	0.1527	0.1484	0.1396	0.1266	0.1117	0.0980	0.0877	0.0808	0.0759
0.1482	0.1463	0.1411	0.1314	0.1178	0.1027	0.0895	0.0805	0.0754	0.0722
0.1382	0.1361	0.1306	0.1204	0.1061	0.0904	0.0772	0.0690	0.0656	0.0644
0.1253	0.1236	0.1184	0.1084	0.0938	0.0773	0.0632	0.0552	0.0531	0.0539
0.1118	0.1108	0.1069	0.0979	0.0838	0.0669	0.0516	0.0426	0.0411	0.0434
0.0993	0.0994	0.0972	0.0900	0.0773	0.0612	0.0457	0.0356	0.0334	0.0357
0.0884	0.0896	0.0891	0.0839	0.0733	0.0590	0.0447	0.0348	0.0312	0.0318
0.0790	0.0810	0.0819	0.0784	0.0696	0.0575	0.0453	0.0364	0.0320	0.0304

TABLE III
SAMPLE FEATURES FOR FAKE FINGERPRINT IMAGES

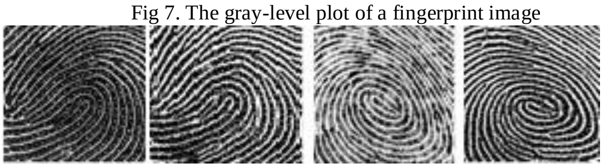
0.2357	0.2380	0.2399	0.2381	0.2315	0.2210	0.2089	0.1972	0.1867	0.1775
0.2335	0.2355	0.2368	0.2345	0.2276	0.2172	0.2058	0.1951	0.1860	0.1781
0.2263	0.2278	0.2283	0.2253	0.2180	0.2078	0.1975	0.1887	0.1818	0.1760
0.2109	0.2120	0.2120	0.2084	0.2008	0.1911	0.1821	0.1755	0.1711	0.1674
0.1862	0.1875	0.1875	0.1839	0.1764	0.1674	0.1598	0.1553	0.1531	0.1510
0.1544	0.1565	0.1576	0.1550	0.1484	0.1403	0.1342	0.1313	0.1304	0.1288
0.1198	0.1237	0.1274	0.1272	0.1226	0.1160	0.1111	0.1089	0.1078	0.1053
0.0882	0.0953	0.1032	0.1067	0.1049	0.1002	0.0959	0.0932	0.0907	0.0862
0.0652	0.0764	0.0890	0.0962	00969	0.0936	0.0894	0.0857	0.0816	0.0757
0.0552	0.0691	0.0842	0.0930	0.0946	0.0915	0.0870	0.0830	0.0791	0.0742



(a) Real fingerprint images (b) Fake fingerprint images

Fig 6. The power spectra of (a) Real and (b) fake fingerprint images

Fig 7. The gray-level plot of a fingerprint image



(a) (b)

Fig 8. Two sets, (a) and (b), of live and fake fingerprint images captured with a 1000-dot/ in high resolution optical sensor Left: real image; right: corresponding fake image.

For each one of the feature samples, the results are shown specifically. Power spectra of real and fake images exhibit similar ring patterns, but their overall geometric structures are alike, as shown in Fig 6. Fig 7 shows the gray-level plot of a fingerprint image. The ridge thickness is calculated using the gray-level values of each block in a direction normal to the ridge orientation. However, in this observation, it is determined that pores could be easily detected in fake fingerprints but the pores of real fingerprint images are invisible, as shown in Fig 8.

Table IV shows the average values of the ridge thickness of the real and fake fingerprints for the four datasets considered. It is clear that the ridges of the fake fingerprint images were thicker than those of the real images because of the best optimal features selected by ABC.

E. Fuzzy Feed Forward Neural network for classification

In this work, the multilayer FFFNN method is used to classify fake and real fingerprint images from selected features. Multilayer feed forward neural network can represent a very broad set of nonlinear functions to classify fake and real fingerprint images for selected multiple static features from the ABC optimization algorithm. FFFNN

TABLE IV
COMPARISON OF RIDGE THICKNESS FOR ABC

Ridge thickness (pixels)				
Fingerprint Images	Low-cost Optical sensor	Low-cost Capacitive Optical sensor	Optical sensor	Synthetic generator
Real	6.35	8.12	7.98	3.95
Fake	9.98	13.46	12.54	4.36

initiates through input layer multiple static feature results obtained from ABC for fake and real fingerprint image samples. The input multiple static features from ABC for fake and real fingerprint images are connected to the hidden layer. In ANN system, the networks are known as feed forward, since input layer from one multiple static features neurons feed forward into next layer of neurons. Typically, all the input samples with feature selected results of all nodes are entirely connected to hidden nodes and output node comprises of the real, fake fingerprint classification results. So, it is easy to solve the difficulty of classification results for those selected features. To perform activation function, weight values are to be assigned between connected nodes in FFFNN of input multiple static selected features. Assigning weight values randomly do not give an exact result for classification. In order to overcome this problem, in this work, a special weight has been used for both hidden layer and output layer process. The weight value w_0 feeds into every selected multiple static feature node at the hidden layer and a special weight (called z_0) feeds into every node at the output layer to classify fake and real fingerprint class names. These types of special weights are known as bias. Initially, all the weight values of nodes are set to zero or almost zero. The back propagation training of the selected features from ABC will adjust these weights so that the output fake and real fingerprint generated by the network is matched with the correct fake and real fingerprint classification results. Every input from selected multiple static features are connected to hidden layer and in the output classification layer, classification is performed through its weight value from input node to classify results (fake or real fingerprint images). Each layer of FFFNN working principles varies according to conditions and their own characteristics. Input units: The input unit takes in the selected features from ABC. The results from input units (feature selection results in ABC) unit is labeled x_j, for $j \in [1, d]$, where d input units. There is also a special type of input labels named as x_0, which always has the value of 1. It is used to provide bias values to the hidden nodes.

Hidden units: The connections coming out of input selected static features result have weight values connected with them. A weight going to hidden unit z_h from important selected features unit x_j would be labeled w_{hj}.The special form of input node with static features, x_0, is connected to hidden nodes in the network along with weight value w_{h0}. In the training of important multiple selected features from ABC, these base weight values are not considered and the remaining weight of nodes is updated through the back propagation algorithm. It is known that the weight value of the specialized input node is always one. Each hidden node calculates the weighted sum of its fake and real fingerprint samples for selected features from ABC and applies a thresholding function to determine the fake and real fingerprint output of the hidden node z_h as defined in equation (10) as:

$$\sum_{j=1}^{d} w_{hj} x_j \tag{13}$$

The activation function of input selected features nodes with threshold value is in the form of equation (13):

$$\text{sigmoid}(a) = \frac{1}{1 + e^{-a}} \quad (14)$$

At the hidden node, equation (10) is applied to the weighted sum of the selected features input to the hidden node, to get the output (fake and real fingerprint) images classification results from hidden node z_h is:

$$z_h = \text{sigmoid}\left(\sum_{j=0}^{d} w_{hj} x_j\right) = \frac{1}{1 + e^{-\sum_{j=0}^{d} w_{hj} x_j}} \quad (15)$$

For $h \in [1, H]$, where H is the total number of hidden nodes. Now, the output of classification result for feature selected from ABC is represented as,

$$o_i = \sum_{h=0}^{H} v_{ih} z_h \quad (16)$$

To differentiate each input node, weight values are assigned to output results y_i of unit i. Consequently, the sigmoid function is applied to get an output classification result unit, y:

$$y = \text{sigmoid}(o) = \text{sigmoid}\left(\sum_{h=0}^{H} v_h z_h\right) = \frac{1}{1 + e^{-\sum_{h=0}^{H} v_h z_h}} \quad (17)$$

Training the neural network to produce the correct output for given selected features from ABC is an iterative process, in which the output of this classification (fake and real fingerprint images) results is compared with required correct output, and weight values of hidden nodes and output nodes are updated continuously to reduce error in equation (21,22). FFFNN results in the correct classification of fake and real fingerprint images through BP learning process and weight updating among nodes in the network. Calculating the weight updates for a given single instance (x^t, r^t) where x^t represent the input training sample with selected feature from ABC and r^t is the target output either fake or real image classification result, and y^t is the correct classification result from network. Here, superscript $'t'$ represents the current running features in the training phase. Classification for 2 classes fake and real fingerprint weight updates for this case are:

$$\Delta v_h = \eta(r^t - y^t) z_h^t \quad (18)$$

$$\Delta w_{hj} = \eta(r^t - y^t) v_h z_h^t (1 - z_h^t) x_j^t \quad (19)$$

Classification for K > 2 classes, weight updates for this case is:

$$\Delta v_{ih} = \eta(r_i^t - y_i^t) z_h^t \quad (20)$$

$$\Delta w_{hj} = \eta\left(\sum_{i=1}^{K} (r_i^t - y_i^t) v_{ih}\right) z_h^t (1 - z_h^t) x_j^t \quad (21)$$

The result of the classification process is evaluated by computing the error results of the output node (fake and real fingerprint classification results) for one complete training epoch which is given by the following equation

$$E(W, V|\chi) = \frac{1}{2} \sum_{(x^t, r^t) \epsilon \chi} (r_i^t - y_i^t)^2 \quad (22)$$

One easy way to speed up the learning process of the fake and real fingerprint image classification is to use momentum. The general procedure of learning process use momentum function from equation (21) & (22) to update the weight values of fake and real fingerprint input features to reduce error value of classification. So, it is necessary to save the weight of earlier one training process for each time step. Then, on the next process of weight updating, this earlier update information is used. It is observed that the weight updates of earlier one were as follows:

$$v_{ih} = v_{ih} + \Delta v_{ih} \quad (23)$$

$$w_{hj} = w_{hj} + \Delta w_{hj} \quad (24)$$

Now the updated equation after constant momentum parameters is given as follows,

$$v_{ih}^t = v_{ih}^t + \Delta v_{ih}^t + \alpha \Delta v_{ih}^{t-1} \quad (25)$$

$$w_{hj}^t = w_{hj}^t + \Delta w_{hj}^t + \alpha \Delta w_{hj}^{t-1} \quad (26)$$

$$\mu_A(w) = \begin{cases} 0 & \text{if } w \le a \\ \dfrac{w-a}{b-a} & \text{if } a \le w \le b \\ \dfrac{c-w}{c-b} & \text{if } b \le w \le c \\ o & \text{if } w \ge c \end{cases} \quad (27)$$

Where $a \in (0 - 0.3)$, $b \in (0.3 - 0.7)$ and $c \in (0.7 - 1)$

In equation (26), still the weight value updating based on the constant moment parameters does not provide higher classification results which reduce the performance for detection rate of fake fingerprint images. In order to solve the weight updating problem, a new calculation of the weight values is introduced with the fuzzy membership function to calculate the weight values to each selected multiple static feature samples from ABC. Membership functions can take any form, but there are some common examples that appear in pattern.

In this work, the superscript t refers to the current image samples with the feature selection results for fake and real fingerprint images and $t - 1$ refers to the previous training example by the Feed Forward neural network. So, constant momentum parameter α is used to adjust new weight values to improve classification accuracy. Here, α is a constant called momentum, with $0 \le \alpha < 1$. Once, the training process for multiple static feature selection results is over, the accuracy of classification result is evaluated based on the testing samples. The results are compared based on the parameters like sensitivity, specificity, precision and classification accuracy.

IV. Experimental Results

In this section, the classification results of proposed FFFNN and existing SVM classification methods are compared. The real fingerprint images are collected from Fingerprint Verification Competition (FVC2000) [22] and fake fingerprints samples are generated from SFinGE. Each and every database samples are different from each other since each optical sensor works in a different manner. DB1 and DB2 images are gathered by using two small-size image samples with low cost sensors namely optical and capacitive, correspondingly. DB3 is collected through a large quality of optical sensors. Finally, databases DB4 is created synthetically by SFinGE in [15].

TABLE V
THE FOUR FVC2000 DATABASES

Database names	Sensor type	Image Size
DB1	Low-cost Optical sensor	388 × 374
DB2	Low-cost Capacitive Optical sensor	560 × 296
DB3	Optical sensor	300 × 300
DB4	Synthetic generator	288 × 384

Four different databases were collected by using the following sensors/technologies: DB1: Secure Desktop Scanner by KeyTronic for low-cost optical sensor , DB2: TouchChip by ST Microelectronics for low-cost optical capacitive sensor , DB3: DF-90 by Identicator Technology for optical sensor , DB4: synthetic generation by SFinGE

TABLE V summarizes the global features of the four databases from each of them

The performance of classification methods are measured based on classification methods as described below. Fig 9. shows the Sample images taken from DB1, DB2, DB3 and DB4.In order to show the different image size of each database, the four images are reported at the same scale factor. Fig 10 shows the Sample images from DB1; each row shows different impressions of the same finger. Fig 11 shows the Images from DB1; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality). Fig 12 shows the Sample images from DB2; each row shows different impressions of the same finger. Fig 13 shows the Images from DB2; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality). Fig 14 shows the Sample images from DB3; each row shows different impressions of the same finger. Fig 15 shows the Images from DB3; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality). Fig 16 shows the Sample images from DB4; each row shows different impressions of the same finger. Fig 17 shows the Images from DB4; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality).

Fig 9. Sample images taken from DB1, DB2, DB3 and DB4.

Fig 10. Sample images taken from DB1 with different impression

Fig 11. Images from DB1; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality).

Fig 12. Sample images taken from DB2 with different impression

Fig.13. Images from DB2; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality).

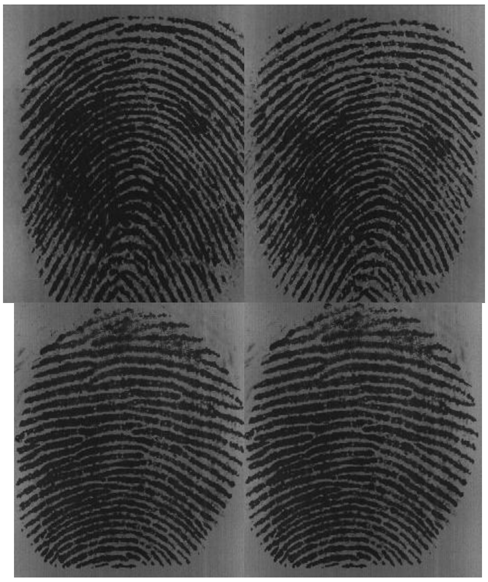


Fig.14. Sample images taken from DB3with different impression

Fig.15. Images from DB3; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality).

Fig 16. Sample images taken from DB4 with different impression

Fig 17. Images from DB4; all the samples are from different fingers and are ordered by quality (top-left: high quality, bottom-right: low quality).

A. Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio (PSNR) is an important metric to measure image quality after and before preprocessing methods is applied.

$$\mathrm{PSNR} = 10 \ \log_{10}\left(\frac{\mathrm{R}^2}{\mathrm{MSE}}\right) \tag{28}$$

MSE represents the increasing squared error between the filtered image and original images before filtering and after normalization.

$$\mathrm{MSE} = \frac{\sum_{\mathrm{M,N}}[\mathrm{I}_1(\mathrm{m,n})-\mathrm{I}_2(\mathrm{m,n})]^2}{\mathrm{M}*\mathrm{N}} \tag{29}$$

Where M and N represent the total number of rows and columns in the image samples respectively. In the previous equation, R is the maximum changeability in the input image data type. From Table VI, it is observed that PSNR and MSE values after normalization and median filtering with different Gaussian noise levels are added. In this work, the quality of the fingerprint and fake image sample using PSNR ratio parameter is evaluated using preprocessing step.

TABLE VI
PSNR AND MSE VALUE COMPARISON

Preprocessing schemes	Noise $\sigma = 10$		Noise $\sigma = 20$	
	PSNR	MSE	PSNR	MSE
Min max normalization	53.4	1.5	35.5	1.8
Median filtering	60.12	0.8	39.2	1.6

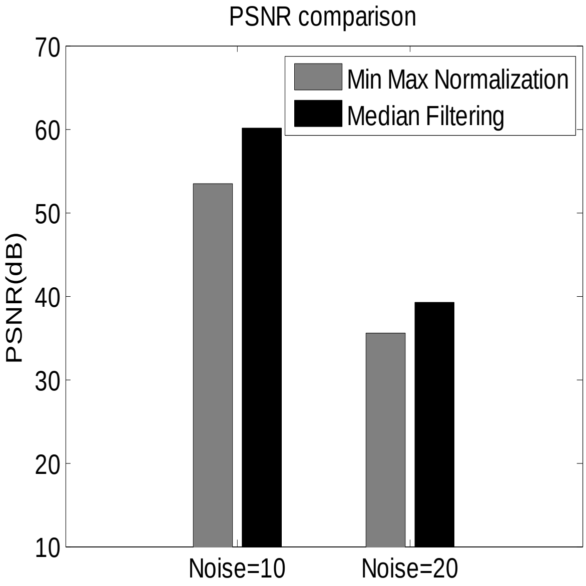


Fig 18. PSNR for preprocessing methods

Fig 18 shows that PSNR results obtained for Gaussian noise $\sigma = 10$ is higher when compared with Gaussian noise $\sigma = 20$ for both the normalization and median filtering methods.

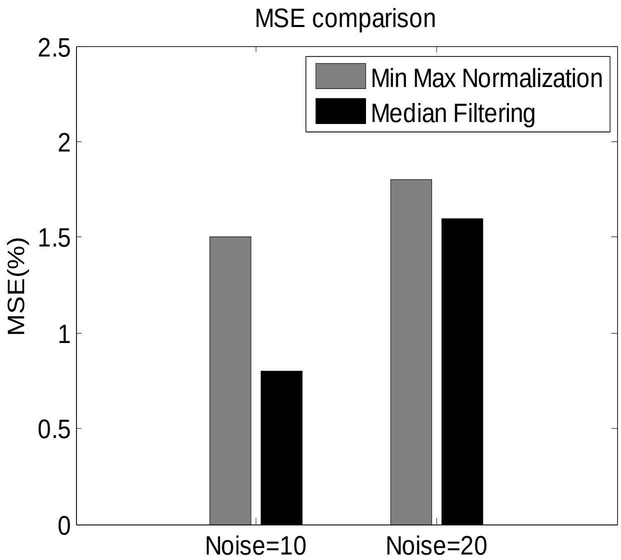


Fig 19. MSE for preprocessing methods

Fig 19 shows that MSE results for median filtering approach is observed to be lesser when compared with the normalization method for different levels of Gaussian noise $\sigma = 10$ & $\sigma = 20$.

TABLE VII
PERFORMANCE COMPARISON OF EER FOR DIFFERENT DATABASES

Database names	SVM	FFNN	FFFNN
Selected Optimal features in DB1	1.25	1.05	0.25
Selected Optimal features DB2	1.2	1.01	0.3
Selected Optimal features DB3	0.9	0.85	0.2
Selected Optimal	0	0	0

Table VII shows the EER result comparison for the selected set of features in all the four databases. EER results of the proposed FFFNN is observed to be lesser when compared with the SVM and FFNN approaches for all the databases taken for consideration. For instance with DB1, it is observed that, the proposed FFFNN approach attains only 0.25 EER whereas SVM and FFNN approach attains 1.25 and 1.05 EER respectively.

Table VIII shows the EER results comparison of the feature sets like power spectrum, first order histogram, Directional contrast and ridge thickness in four different databases. It is observed that the proposed FFFNN approach results in lesser Equal Error Rate (EER) for all the four static features taken into consideration. Among the four static features, power spectrum feature is observed to provide lesser EER comparatively for the proposed FFFNN approach.

There are certain important aspects to be taken into consideration while using fake fingerprint images. The main factor is that, the fake fingers should be able to interact with fingerprint recognition system. If the fake finger is of very low quality, it could be taken as a non matched finger and gets simply rejected. So, ensuring the image quality is very vital and this work utilizes Natural Image Quality Evaluator (**NIQE**) for quality assessment. Quality of the distorted image is expressed based on the multiple static feature selection models from the distorted image:

$$D(v_1, v_2, \Sigma_1, \Sigma_2) = \sqrt{\left((v_1 - v_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (v_1 - v_2)\right)} \quad (31)$$

v_1, v_2 be the mean value of input and distorted image, Σ_1, Σ_2 be the covariance matrix of input and distorted image.

Each fingerprint image was assigned to one of five quality levels namely excellent, very good, good, fair, and bad according to the quality measure. Fig 20, Fig 21, Fig 22 and Fig 23 shows the NIQE quality-checking results for the four different databases taken into consideration. It is observed that most of the fake fingerprint images are of good quality and is applicable to be used in the evaluation.

TABLE VIII
PERFORMANCE COMPARISON OF EACH FEATURE SET

Static features	DB1 ,EER(%)			DB2 ,EER(%)			DB3 ,EER(%)			DB4 ,EER(%)		
	SVM	FFNN	FFFNN	SVM	FFNN	FFFNN	SVM	FFNN	FFFNN	SVM	FFNN	FFFNN
Power spectrum	1.78	1.25	0.58	1.78	1.04	0.56	3.48	0.58	0.25	0.5	0.25	0.12
First order histogram	9.81	5.8	2.3	8.59	4.5	2.18	12.01	9.12	5.65	11.45	5.68	2.35
Directional contrast and ridge thickness	11.5	5.6	1.89	9.45	2.35	1.45	12.14	9.14	4.56	9.54	5.86	2.24
Ridge signal	20.9	15.2	5.85	12.58	4.15	1.32	2.1	1.4	0.56	15.26	11.45	5.12

Fig 20.The results of the NIQE quality check on optical_1 sensor database

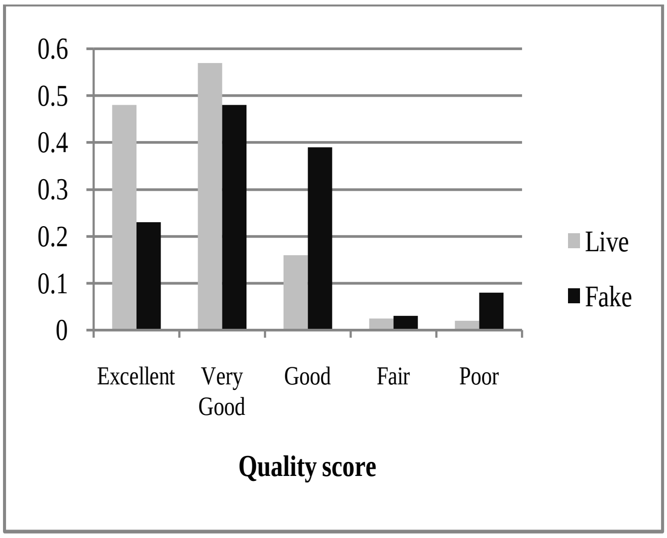


Fig 21.The results of the NIQE quality check on optical_2 sensor database

Fig 22. The results of the NIQE quality check on optical_3 sensor database

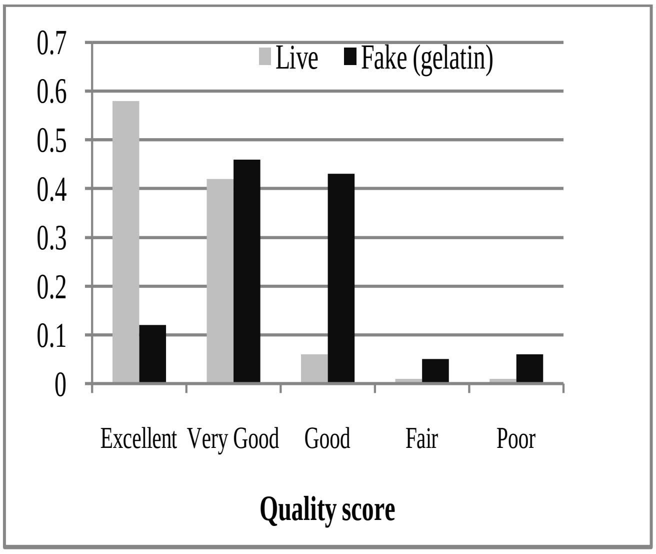


Fig 23.The results of the NIQE quality check on optical_3 Capacitive database

B. Sensitivity

Sensitivity evaluates the percentage of actual positive results of fake and real fingerprint subjects class. It is observed that the classified percentage of real and fake fingerprint results for the proposed approach is higher. The sensitivity is defined as below:

$$\text{Sensitivity} = \frac{T_p}{T_p + F_n} \tag{30}$$

T_p defines the fake fingerprint as fake and similarly, it classifies real fingerprint as real.

F_p defines the real fingerprint incorrectly as the fake fingerprints

T_n is defines the fake fingerprint correctly identified as fake fingerprint

F_n is defines the fake fingerprint incorrectly identified as real fingerprints

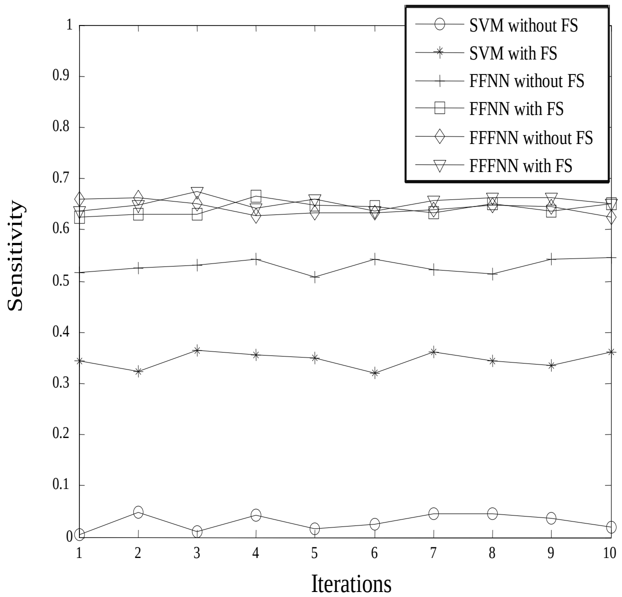


Fig 24. Sensitivity for classification

C. Specificity

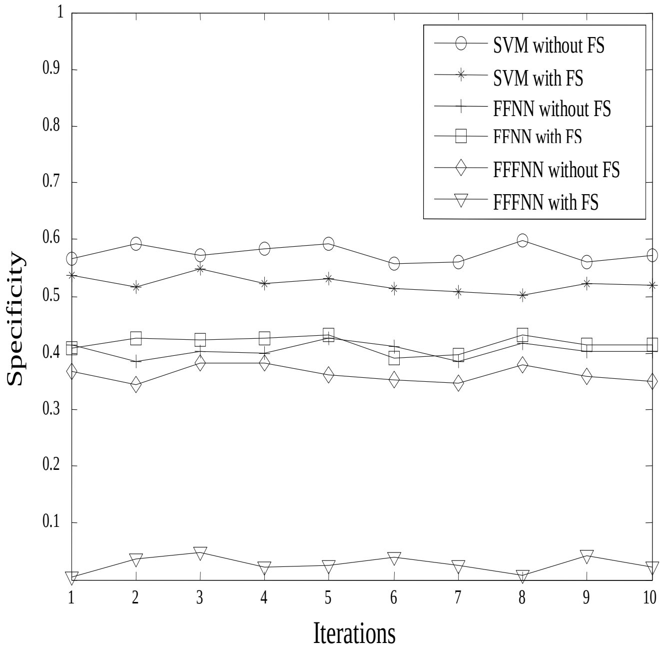


Fig 25. Specificity for classification

Fig 24 shows the sensitivity results for Fuzzy Feed Forward Neural Network (FFFNN), feed forward Neural network (FFNN) and Support Vector Machine (SVM) classification methods. The performance is evaluated based on the influence of the feature selection method. The sensitivity results obtained with and without feature selection approach is clearly shown in the figure. It is observed that the proposed FFFNN have higher sensitivity results than FFNN, SVM methods with ABC based feature selection.

Specificity evaluates the percentage of actual negatives which are related to negative subjects class that is fake image is classified as real fingerprint images and real images are classified as fake images.

$$\text{Specificity} = \frac{T_n}{T_n + F_p} \qquad (32)$$

Fig 25 shows specificity results of proposed classification methods with and without ABC feature selection. The proposed FFFNN classification approach with ABC feature selection is observed to have lesser specificity results when compared with FFNN and SVM classification methods.

D. Precision

Precision is defined as the proportion of the true positives against both true positives and false positives results for fake and real fingerprint images .It is defined as follows:

$$\text{Precision} = \frac{T_p}{T_p + F_p} \qquad (33)$$

Fig 26 shows the precision results of FFFNN, FFNN and SVM methods with and without ABC feature selection. It is observed that the proposed FFFNN with ABC feature selection have higher precision accuracy than classification methods FFNN, SVM without feature selection.

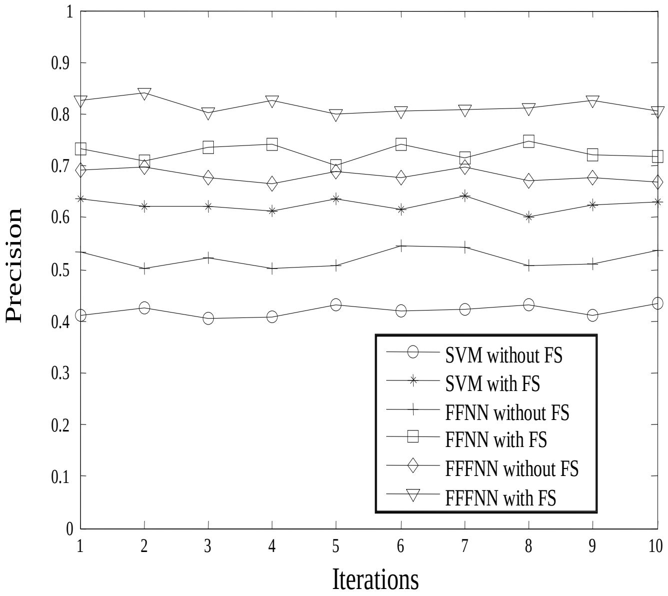


Fig 26. Precision for classification

E. Classification Accuracy

Accuracy is defined as the overall correctness of the model and is calculated as the sum of actual classification parameters $(T_p + T_n)$ separated by the total number of classification parameters $(T_p + T_n + F_p + F_n)$

$$\text{Accuracy} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \qquad (34)$$

Fig 27 evaluates the classification results of FFFNN, FFNN and SVM classification methods. The classification result is evaluated with and without feature selection and it is observed that the proposed FFNN with ABC feature selection approach has higher accuracy results than the classification method without feature selection. This significant performance of the proposed FFFNN approach is mainly due to continuous updating of weight values according to gradient momentum function in FFFNN, it reduces error values in FFFNN. Moreover, the results also show the importance of the ABC feature selection algorithm in classification.

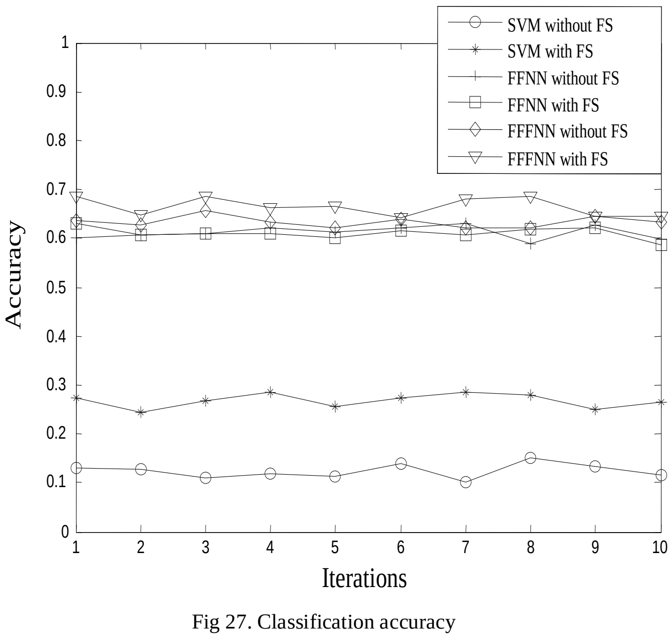


Fig 27. Classification accuracy

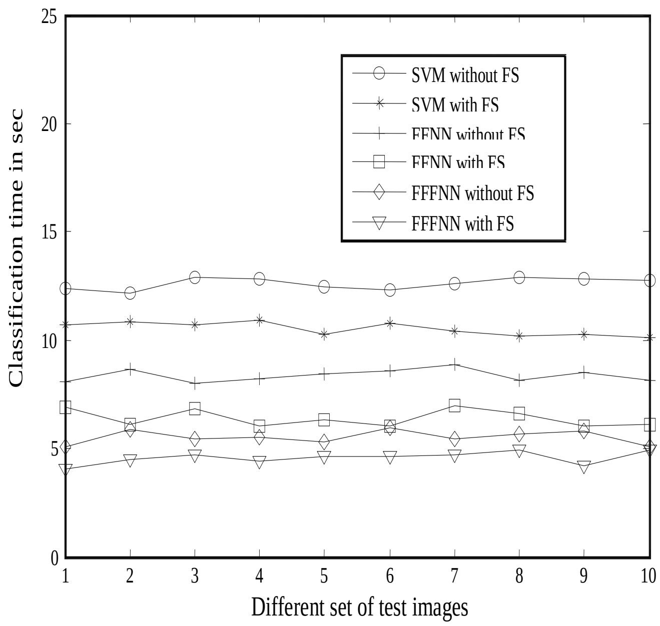


Fig 28. Time Comparison for classification methods

Fig 28 evaluates the time comparison of FFFNN, FFNN and SVM classification methods. The time comparison results for classification methods is evaluated separately with and without feature selection and it is observed that the proposed FFFNN with ABC feature selection approach has taken less time to complete classification process as best features are selected for the classification. On the other hand, the existing classification approaches without feature selection is observed to take higher processing time.

Table IX shows the confusion matrix results of the methods based on FAR and FRR values. It is observed from the table that the True positive (TP) predicated outcome value of proposed FFFNN classifier is 27 which is higher than the other SVM and FFNN classifiers taken for consideration. It shows that the proposed FFFNN classifier correctly matches fake and real fingerprint images. Moreover, false negative results of the proposed FFFNN classifier are also less when compared to with existing SVM and FFNN classifiers. It is observed that, when FAR increases, FRR rate automatically decreases and viz versa. False acceptance rate (FAR) = type I error = 1 − specificity = FP /(FP + TN) , False rejection rate (FRR) = type II error = 1 − sensitivity = FN / (TP + FN)

TABLE IX: CONFUSION MATRIX SAMPLE RESULTS

Predicated outcome for SVM with (original = 40, fake =20 images)		
Actual value	True positive (TP) = 23	False positive (FP) = 17
	False negative (FN) = 8	True negative (TN) = 12
Predicated outcome for FFNN with (original = 40, fake =20 images)		
Actual value	True positive (TP) = 26	False positive (FP) = 12
	False negative (FN) = 14	True negative (TN) = 8
Predicated outcome for FFFNN with (original = 40, fake =20 images)		
Actual value	True positive (TP) = 27	False positive (FP) = 16
	False negative (FN) = 13	True negative (TN) = 14

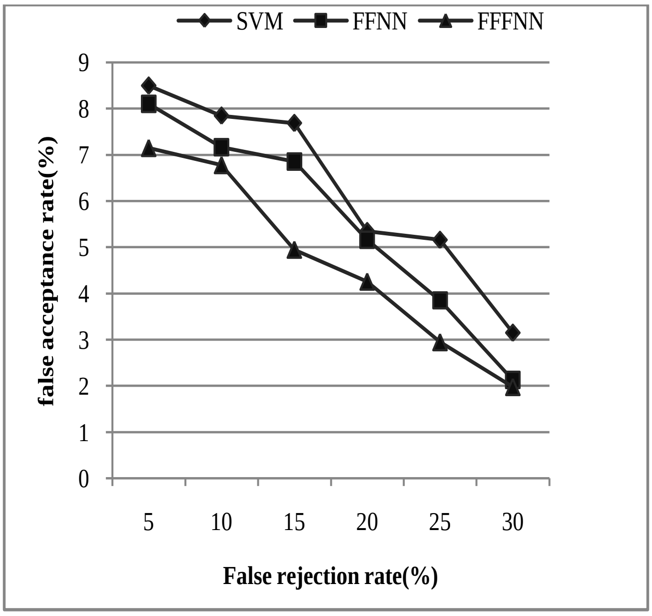


Fig 29. Performance evaluation of FAR vs FRR for classification methods

Fig 29 show the performance of the proposed FFFNN method evaluated using the FAR and FRR. FAR is the probability of accepting a fake fingerprint as a real one, where as FRR is the probability of rejecting a real fingerprint as a fake one. Fig 20 clearly shows that the performance of the proposed FFFNN approach is significant when compared with the existing FFNN and SVM classification methods.

V. CONCLUSION

This paper describes a new method of classifying fingerprint images for best multiple static features selection result with optimization methods. Initially, the fingerprint images are preprocessed to enhance the quality of image and clarity. This work uses normalization for contrast enhancement and then median filtering is also performed for noise removal. Best features of images are extracted after preprocessing completion using 2D Gabor filtering method. 2D Gabor filtering method extracts multiple static features among real and fake fingerprints instead of considering dynamic images. In order to select best static features, ABC optimization algorithm is applied. To improve the classification performance, Fuzzification process is integrated with FFNN. Experimental results are evaluated for each fingerprint images through the parameters like sensitivity, specificity, precision and classification accuracy for samples from FVC2000 and synthetically generated database images are collected. Experimental results confirm that proposed with ABC feature selection provided best classification accuracy than classification methods without feature selection. The results are compared with SVM classifier with and without feature selection approaches.

REFERENCES

[1] E. Guillen, L. Alfonso, K. Martinez and M. Mejia, "Vulnerabilities and Performance Analysis over Fingerprint Biometric Authentication Network," *Proceedings of the World Congress on Engineering and Computer Science* 2012, WCECS 2012, San Francisco, USA, 24-26 October, 2012, vol.2, pp. 908-913.

[2] M. Gomez-Barrero, J. Galbally and J. Fierrez, “Efficient software attack to multimodal biometric systems and its application to face and iris fusion,” *Pattern Recognition Letters*, vol.36, pp.243–253, April 10, 2013.
[3] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, “Impact of Artificial Gummy Fingers on Fingerprint Systems,” *In: Proceedings of SPIE, Optical Security and Counterfeit Deterrence Techniques IV*, 24-25 January, 2002, pp. 275-289.
[4] K. Thaiyalnayaki, K.S.S. Abdul and P.P. Varsha, “Finger Print Recognition using Discrete Wavelet Transform,” *International Journal of Computer Applications*, vol.1,no.24, pp 96-100,2010.
[5] L. Wang, and N. Bhattacharjee, “Adaptive approach to fingerprint image enhancement,” *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia, ACM New York, NY, USA*, November, 2010, pp. 42- 49.
[6] G.L. Marcialis, F. Roli, and A. Tidu, “Analysis of Fingerprint Pores for Vitality Detection,” *In: Proc. of 20th IEEE/IAPR International Conference on Pattern Recognition (ICPR 2010),* Instanbul (Turkey), 23-26 August, 2010, pp. 1289–1292.
[7] P. Coli, “Vitality detection in personal authentication systems using fingerprints,” PhD Thesis, 2007, pp 1-113.
[8] S.A.C. Schuckers and A. Abhyankar, “A wavelet based approach to detecting liveness in fingerprint scanners,” *In: Biometric Authentication Workshop (ECCV)*, August 5, 2004, pp 278-286.
[9] B. Tan and S. Schuckers, “Comparison of ridge and intensity based perspiration liveness detection methods in fingerprint scanners,” *In:* Proc. *SPIE, Biometric Technology for Human Identification III,* April 17,2006, pp 62020A-62020A.
[10] B. Tan and S. Schuckers, “New approach for liveness detection in fingerprint scanners based on valley noise analysis,” *Journal of Electronic Imaging*, vol.17, no.1, pp 011009-011009, March-3, 2008.
[11] G. Qinghai and Z. Xiaowen, “A Study of Distortion Effects on Fingerprint Matching,” *Computer Science and Engineering*, vol.2, no.3, pp.37-42, 2012.
[12] C. Akinyokun Oluwole, K. Alese Boniface, and B. Iwasokun Gabriel, “Fingerprint Matching Using Spatial Characteristics,” *Proceedings of the World Congress on Engineering*. WCE 2014, 2-4 July, 2014, London, U.K., vol.1, pp 466-474.
[13] A. Abhyankar and S. Schuckers, “Fingerprint liveness detection using local ridge frequencies and multiresolution texture analysis techniques,” *in IEEE International Conference on Image Processing*, October 8-11, 2006, pp 321–324.
[14] A. Franco and D. Maltoni, “Fingerprint Synthesis and Spoof Detection,” In Ratha N.K., Govindaraju V., (Eds.), *Advances in Biometrics: Sensors, Algorithms and Systems, Springer, London*, 2007, pp 385–406.
[15] M. Sepasian, W. Balachandran, and C. Mares, “Image enhancement for fingerprint minutiae-based algorithms using CLAHE, standard deviation analysis and sliding neighborhood,” *Proceedings of the World congress on Engineering and Computer Science*, WCECS 2008, 22-24 October, 2008, San Francisco, USA, pp 1199-1203.
[16] S. Yoon, J. Feng and A.K. Jain, “Altered Fingerprints: Analysis and Detection,” *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.34, no.3, pp 451−464, March 2012.
[17] T. Elham, and G. Patrick, “Fingerprint Image Quality,” *Encyclopedia of Biometrics*, 2009, pp 482-490.
[18] A.A. Aburas, and S.A. Rehiel, “Fingerprint patterns recognition system using huffman coding,” *Proceedings of the World Congress on Engineering*, WCE 2008, 2-4 July, 2008, London, U.K., vol.3, pp 1794-1796.
[19] D. Maltoni, D. Maio, A. K. Jain, & S. Prabhakar, “Synthetic fingerprint generation,” *Handbook of fingerprint recognition*, 2009, pp 271-302.
[20] L. Bao and Z. Jian-chao, “Comparison and Analysis of the Selection Mechanism in the Artificial Bee Colony Algorithm,” *Proceedings of IEEE Ninth International Conference on Hybrid Intelligent Systems*, August 12-14, 2009, vol.1, pp 411-416.
[21] N. Suguna and K.G. Thanushkodi, “An Independent Rough Set Approach Hybrid with Artificial Bee Colony Algorithm for Dimensionality Reduction,” *American Journal of Applied Sciences,* vol.8, no.3, pp 261 – 266, 2011.
[22] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman and A. K. Jain, “FVC2000: Fingerprint verification competition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.24,no.2, pp 402-412 ,August-07, 2002.