Resbee Publishers

Multimedia Research (MR)

Received 9 June, Revised 22 August, Accepted 20 September

Hybrid Optimization based DBN for Face Recognition using Low-Resolution Images

Renjith Thomas
GISAT
Kottayam, Kerala, India
renjiththomas80@gmail.com

MJS. Rangachar
Department of School of Electrical Sciences
Hindustan University
Chennai, Tamil Nadu, India

Abstract: The recognition of faces has gained immense interest in image processing. The conventional face recognition techniques provide improved performance using the frontal images with high resolution. However, the major problem in face recognition is the Low-Resolution face images. To address this challenge, this paper proposes the face recognition system by integrating the Gabor Filter + Wavelet + Texture (GWTM) operator and the Deep Belief Network (DBN) to increase the classification performance, while deploying the low-resolution images. Initially, the input image is subjected to the preprocessing, and the low-resolution image is generated. Then, these low-resolution images employed kernel regression model for generating an image with high-resolution. Then, both the low-resolution and the high-resolution images are applied to the GWTM operator for extracting significant features. The result of the GWTM is provided to the fractional Bat algorithm for producing the intermediary images. Finally, the intermediary images are given to the DBN classifier for optimal face detection. The proposed method is analyzed with the existing methods using three evaluation measures, like the false acceptance rate (FAR), accuracy, and false rejection rate (FRR). Thus, the proposed method outperformed other methods with higher accuracy of 0.98, minimum FAR and FRR of 0.05.

Keywords: Face recognition, image processing, classification, preprocessing, DBN

1. Introduction

The progression in face recognition has fascinated much interest in several methods based on face recognition. There exist two main practices for processing the face recognition systems, namely face matching process and face representation process. The goal of face representation is to refine the discriminative features so as to split the face images into many persons, and the face matching process is to develop an effectual classifier for recognizing different persons [1]. Face recognition poses many advantages as compared to other biometric modalities, like iris and fingerprint. The main benefit of face recognition is that it poses the ability to capture the faces residing in distant places [3]. The two key stages for each face recognition models are Feature extraction and classification. Thus, for improving the recognition rate, it is important to determine a high-quality feature extractor and a valuable classifier [4]. Face recognition is done without any overt action on the users and can be captured from a distance using a camera. Thus, it is valuable for the purpose that is to provide security in face recognition systems. Moreover, the data acquisition is laden with issues for several biometrics methods, which depends on hands and finger gestures as it can be rendered, if the epidermis tissue is injured [5]. Furthermore, face recognition is devised for several applications, such as national securities and defence. [6].

Several algorithms based on face recognition are devised for distinguishing the faces by examining and differentiating the pattern of the facial image. However, face recognition is considered a major challenge like variations in illuminations, expressions, and poses [3]. Different face representation methods are devised for the recognition of faces and are categorized into two sets, namely local feature representation [9], and Representative holistic feature representation [8]. The features based on representative holistics are Linear Discriminant Analysis (LDA) [8], and Principal Component Analysis (PCA) [10]. The Representative local features contain Gabor descriptor [12], Compact Binary Face Descriptor (CBFD) [14], Local Binary Patterns (LBP) [9], and Discriminant Face Descriptor (DFD) [13]. In general, the local features attain improved performance than holistic features due to robustness and stableness in feature description [1]. Different classifiers are devised in the previous works, which involves neural networks, Fisher's discriminant analysis, and minimum distance classification using eigenspace. The global methods are used to classify the frontal views of the faces, but they are not robust with respect to the changes as the global features are extremely sensitive with face rotation [7]. The analysis with respect to the different techniques confirmed that the variations in ages, poses, and illumination are some of the issues faced by several face recognition systems [5]. Numerous face recognition algorithms are devised like Laplacianface, Eigenface, and Fisherface. Moreover, the dimensionality reduction techniques are utilized to obtain the significant characteristic of data, by processing the data in low dimensionality space for face recognition [9].

This paper proposes a face recognition system by integrating the GWTM operator, with Fractional Bat algorithm based DBN classifier. Initially, the input image is employed in the preprocessing stage to produce a low-resolution image. Then, this low-resolution image is adapted in the kernel regression model for generating the high-resolution image. Then, both the low resolution and the high-resolution images are applied with respect to the GWTM operator for extracting the facial features. The GWTM result is provided to the Fractional Bat based DBN for generating the intermediate images. At last, the intermediate images are subjected to DBN classifier for optimal face recognition.

The organization of the paper is represented as: Section 2 illustrates the Literature survey, along with the challenges of face recognition. Section 3 deliberates the illustration of the proposed method based on face recognition. Section 5 provides the results of the proposed method with existing methods and finally, section 5 provides the summary of the research.

2. Literature Review

This section illustrates eight existing techniques based on face recognition techniques. Lu et al. [15] developed a face recognition algorithm on the basis of regularized robust coding and discriminative dictionary learning to recognize the faces based on facial images. Here, the Gabor amplitude images of the faces are acquired using Gabor filters. Then the extraction of the Fisher criterion and a uniform local binary histogram was done to solve the sparse represented problem and to obtain a new face. At last, the test image was used to classify the existing class with sparse representation Coding. Thomas, R. and Rangachar, M.J.S. [16], designed a classifier named multi-kernel-based spherical Support Vector Machine (SVM) classifier with fractional Bat algorithm. At first, the low-resolution images are transformed into high-resolution images using the steps of the kernel regression method. Then The GWTM process was employed for extracting the features using the Gabor filter, wavelet transform, and Local Binary Pattern (LBP). Then, the super-resolution images are used for fusing the feature level by employing the fractional Bat algorithm, which was designed by integrating the fractional theory with Bat algorithm. Finally, the multi-kernel-based spherical SVM classifier was utilized for recognizing the faces. Masi, I.et al.[11] developed a pose-aware method for performing face recognition using facial images with extreme variations in poses. The method demonstrated the single frontal Convolutional Neural Network (CNN) model using half-profile and full profile models for performing the face recognition, particularly in full-profile images. Here, the method focussed on rendering the faces using the estimated poses captured in an adaptive manner. Duan, Y.et al.[1] developed a Context-Aware Local Binary Feature Learning (CA-LBFL) technique for recognizing the faces from the face images. The CA-LBFL utilized the appropriate information obtained from the neighbouring bits by restraining the number of shifts from different binary bits, so as to cover the feasible information required for the face representation. Initially, the method extracted Pixel Difference Vectors (PDV) from local paths and learned discriminative mapping for projecting the PDV into the context-aware binary vector. Then, clustering was performed on the learned binary codes for constructing the codebook and to extract the histogram feature from each face image.

2.1 Challenges

The challenges faced by existing methods are enlisted as follows:

- Face recognition is an obstacle in several systems, due to its computationally challenging tasks, which needs fine discrimination between the similar images of different identities and generalization between different images with the same entities [2].
- The face recognition is a major challenge because of its huge variations in the appearance of face images and different factors that leads to the degradation in recognizing the faces. The factors involved in the degradation are an expression of faces, different subjects, lighting conditions, and different viewpoints [16].

- The majority of the conventional local feature descriptors are based on hand-craft, which needs strong domain knowledge, whereas the learning-based methods like DFD and CBFD use feature representations based on unprocessed pixels to discover the feature code and are more vulnerable to noise [1].
- The existing wavelet-based approaches for recognizing the faces are used in low-frequency subbands for extracting the facial features, but these frequency sub-bands contains significant information related to the edge and contour to reflect the face details and may lead to poor performance in recognizing the faces [3].
- Many classifiers suffer from the handling of huge training data samples for the classification and recognition without any information loss [17].

3. Proposed Fractional Bat Based Multi-Kernel DBN Classifier for Face Recognition

This section illustrates the proposed fractional bat algorithm for fusing the features and DBN classifier for the recognition of low-resolution face images. The block diagram of the proposed face recognition system is portrayed in Fig. 1. The proposed system undergoes five steps for recognizing the face images. The face recognition is processed in the following manner: i) Pre-processing, ii) High-resolution image, iii) GWTM, iv) Fractional Bat algorithm and v) Recognition using multi-kernel DBN classifier. At first, the pre-processing is performed on the input face images. Then, the obtained pre-processed image is of the low-resolution image. Thus, the low-resolution image is transformed into a high-resolution image based on the kernel regression. The Gabor filter and wavelet transform are employed on the high-resolution image for extracting the significant features. Then for local texture analysis, the LBP and the histogram is employed. Then, the obtained features are utilized in the GWTM process. Thus, the high resolution and low-resolution feature images are combined using the feature level fusion based on the proposed Fractional Bat algorithm and DBN. Finally, face recognition is carried out using multi-kernel based DBN classifier for improved performance.

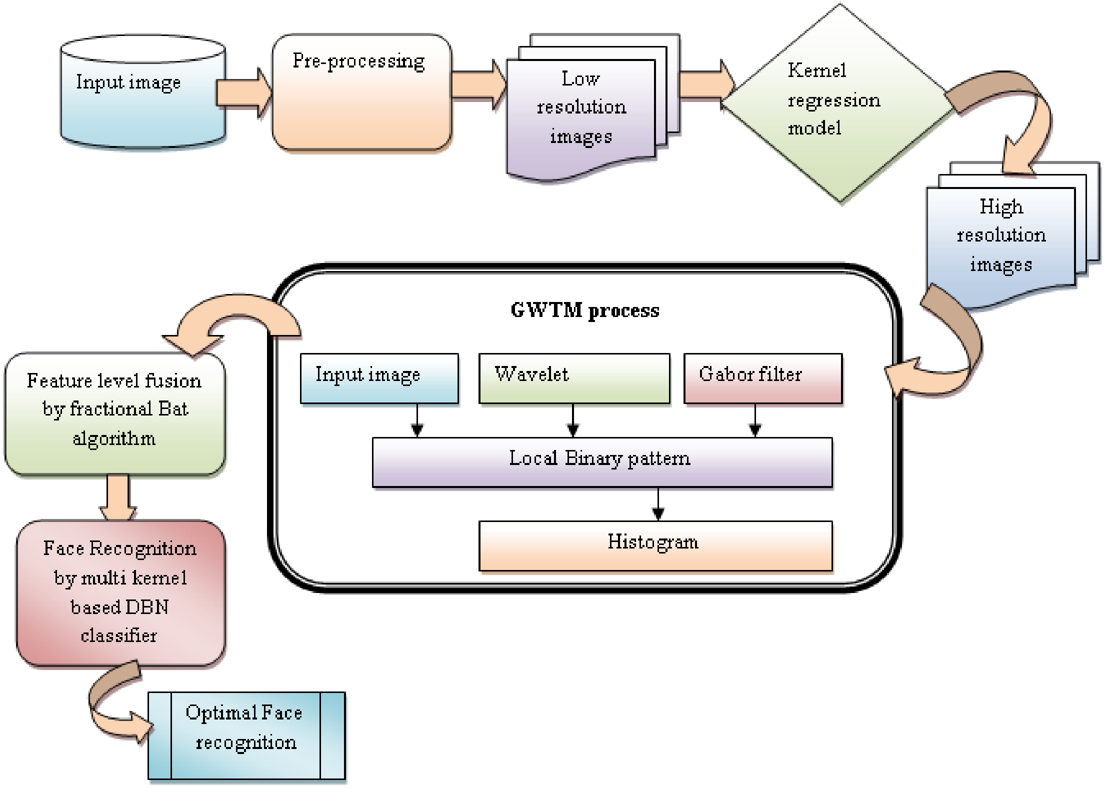


***Fig. 1** Architecture of proposed fractional Bat based DBN classifier*

3.1 Pre-Processing

At first, the input images undergo pre-processing for eliminating the noise and artifacts present in the image. Assume the face image database D with l face images. Then, the input images are pre-processed for extracting the Region of Interest (ROI). After pre-processing, the low-resolution image $D^L(e,f)$ is generated.

3.2 High-Resolution Image by Kernel Regression

Here, the low-resolution image is converted into the high-resolution image using the kernel regression in order to increase the performance rate. The attribute window $W(i,j)$is created using a low resolution image based on the kernel regression and is expressed as,

$$D^{H}(e,f)=\frac{1}{b_1 * b_2}\sum_{i=1}^{b_1}\sum_{j=1}^{b_2}W(i,j)*K(i,j) \tag{1}$$

where, $W(i,j)$denote the attribute window that generates pixel for high-resolution image, $D^{H}(e,f)$, b_1 denote row size and b_2 represents columns size. $W(i,j)\in(e,f)$; $k(i,j)\in(ef)$

Then, the kernel function is formulated as,

$$K_1 = \exp\left(\left(\frac{1}{2\times m^2}\right)*P\right) \tag{2}$$

where, b_2 refers smoothing factor.

3.3 GWTM Process

The GWTM process is adapted for extracting the features using the facial image for face recognition. The GWTM process contains Gabor filter, wavelet transform and texture operator. The low and high-resolution image and the input image are applied to the GWTM process. At first, the wavelet transform is used for the high and low-resolution image. The GWTM process consists of four steps: i) Wavelet transform ii) Gabor filter iii) Local Binary Pattern (LBP) and iv) Histogram.

The wavelet transform is employed to distinguish the low resolution and high-resolution image into many frequency ranges known as wavelets. The wavelets are used to extract the features at various positions and scales and are expressed as,

$$X^{H}(e,f)=\frac{1}{\sqrt{GH}}\sum_{e=0}^{G\ 1}\sum_{f=0}^{H\ 1}D^{H}(e,f)*\ \ {}_{p,q}(e,f) \tag{3}$$

where $X^{H}(e,f)$ is the wavelet transform scale, $D^{H}(e,f)$ refers high-resolution image and ${}_{p,q}(e,f)$ denote a scaling function.

The Gabor filter is used for extracting the features which are formulated as,

$$\delta^{H}_{p,q}(s)=\frac{\left\|t_{p,q}\right\|^2}{\sigma^2}e^{\left(\frac{\left\|t_{p,q}\right\|^2\ \ \left\|s^2\right\|}{2\sigma^2}\right)}\left[e^{n,t_{p,q},s}\quad e^{\ \frac{\sigma^2}{2}}\right] \tag{4}$$

where, $t_{p,q}$ represents the magnitude computation factor given by,

$$t_{p,q}=\frac{t_{\max}}{r} \tag{5}$$

Where, $t_{p,q}$ denote the frequency ratio, p is the orientation and q scale of the Gabor filters, σ represents standard deviation and s spatial position.

Then, the Gabor filter output image is obtained by convoluting the face image response $X^{H}(e,f)$ with $\delta^{H}_{p,q}(s)$ is given as,

$$C^{H}_{p,q}(e,f)=X^{H}(e,f)-\delta^{H}_{p,q}(s) \tag{6}$$

Then, the complex representation of Eq. (6) with magnitude Z and the phase φ is given by,

$$C^{H}_{p,q}(e,f)=Z_{p,q}(e,f).e^{p\theta(e,f)} \tag{7}$$

The LBP operator is utilized for extracting the local features of the image. Thus, the LBP pattern is applied in Gabor filtered and wavelet transforms output along with input image for the texture description. Firstly, the LBP pattern of wavelet transform is represented as,

$$X^{H}_{LBP}(e,f)=LBP\left[X^{H}(e,f)\right] \tag{8}$$

where, $X^{H}(e,f)$ is a wavelet transform the output of the high-resolution image and $X^{H}_{LBP}(e,f)$ is a local binary pattern of the wavelet transform output. Then, the input image and Gabor filtered image is applied to the LBP pattern, which can be specified as follows:

$$D^{H}_{LBP}(e,f) = LBP\left[F^{H}(e,f)\right] \tag{9}$$

$$C^{H}_{LBP}(e,f) = LBP\left[C^{H}(e,f)\right] \tag{10}$$

where, $D^{H}(e,f)$ and $C^{H}(e,f)$ denotes the high-resolution image of the input image and Gabor filter image.

Once the LBP pattern is completed, the histogram is applied which is used to estimate the pixel values. Then, the histogram is applied to the LBP pattern of Gabor filter and wavelet transform output image along with the input image. Thus, the obtained histogram is given by x^{H}, y^{H}, z^{H}. These are the high-resolution feature images represented by a histogram. The histogram is utilized and represented as x^{L}, y^{L}, z^{L}. Finally, the output images obtained after GWTM process for high-resolution images are grouped into one set $Z = \left\{z^{H}, y^{H}, x^{H}\right\}$. Likewise, for the low resolution, the GWTM images are grouped into another set $V = \left\{z^{L}, y^{L}, x^{L}\right\}$. Then, the features of both resolution images are fused to form the one function Y with the help of optimal value β which is denoted as follows,

$$Y = \beta Z + (1-\beta)V \tag{11}$$

3.4 Proposed Fractional Bat Algorithm Based DBN

This section explains the proposed Fractional Bat based DBN classifier introduced for face recognition. The Fractional Bat is proposed by integrating fractional theory and Bat algorithm in the DBN model for the optimal selection of weights in the network. The DBN classifier classifies the interesting points in the images and derives the classes. For the effective classification of the images, DBN classifier is used and the weights-biases are determined using the proposed Fractional Bat algorithm in an optimal manner.

3.4.1 Architecture of Deep Belief Network

The DBN [21] is a part of Deep Neural Network (DNN) and consists of different layers of Restricted Boltzmann Machines (RBMs) and Multilayer Perceptrons (MLPs). RBMs contain hidden and visible units, which are linked based on weighted connections. The MLPs consist of hidden, output, and input, layers. The network with multiple layers has the ability to solve any complicated tasks and thereby, make the classification of health records more effective for evaluating the health risks. The training of the DBN is done using the proposed Fractional Bat algorithm.

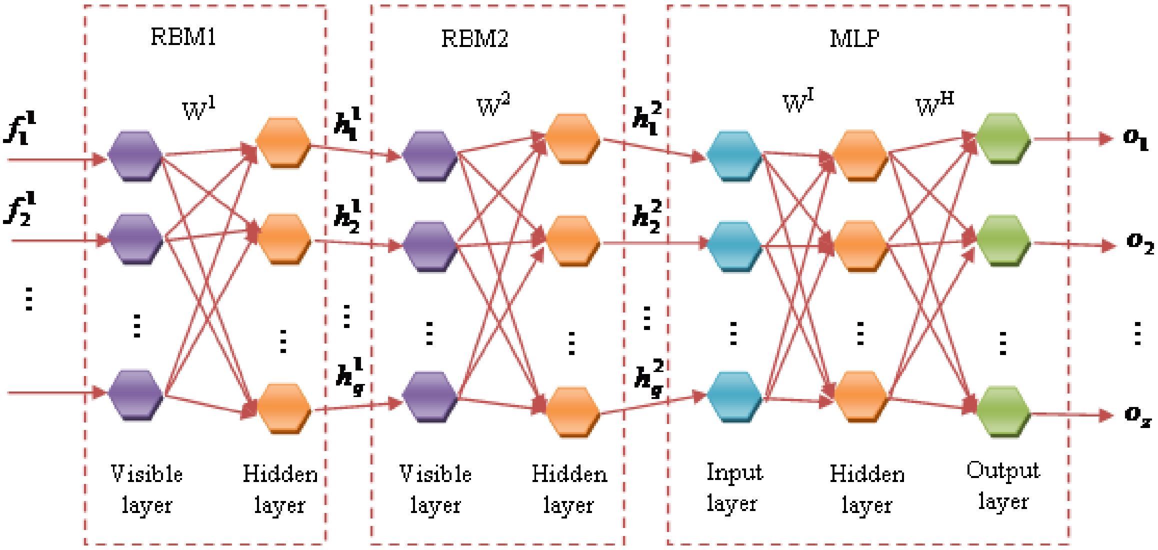


Fig. 2. *Architecture of DBN classifier*

The visible layer, which has the feature vector as its input, and the hidden layer of the first RBM can be represented as follows,

$$a^{1} = \left\{a^{1}_{1}, a^{1}_{2}, \ldots, a^{1}_{k}, \ldots, a^{1}_{10}\right\}; 1 \le k \le 10 \tag{12}$$

$$h^{1} = \left\{h^{1}_{1}, h^{1}_{2}, \ldots, h^{1}_{m}, \ldots, h^{1}_{d}\right\}; 1 \le m \le d \tag{13}$$

where, a_k^1 is the k^{th} visible neuron in the first RBM, h_m^1 refers to the m^{th} hidden neuron and c is the total hidden neurons. Each neuron in the visible layer and the hidden layer has a bias. Let u and v denote the biases in the visible layer and the hidden layer and is given by,

$$S^1 = \{S_1^1, S_2^1, \ldots, S_k^1, \ldots, S_{10}^1\} \tag{14}$$

$$T^1 = \{T_1^1, T_2^1, \ldots, T_h^1, \ldots, T_c^1\} \tag{15}$$

where, S_k^1 represents bias corresponding to k^{th} visible neuron and T_m^1 is the bias corresponding to m^{th} hidden neuron. The weights of first RBM is given by,

$$w^1 = \{w_{km}^1\}; 1 \le k \le 10; 1 \le l \le c \tag{16}$$

where, w_{km}^1 specifies the weight between k^{th} visible neuron and m^{th} hidden neuron and the size of the weight vector is $10 \times c$. Therefore, the output of the hidden layer in the first RBM is given as,

$$h_m^1 = \sigma\left[v_m^1 + \sum_k a_k^1 w_{km}^1\right] \tag{17}$$

where, σ refers activation function. Thus the output of first RBM is given as,

$$h^1 = \{h_m^1\}; 1 \le m \le c \tag{18}$$

The number of hidden neurons in the first RBM and is expressed as,

$$a^2 = \{a_1^2, a_2^2, \ldots, a_c^2\} = \{h_m^1\}; 1 \le m \le c \tag{19}$$

where, $\{h_m^1\}$ represents output vector of the first RBM. The hidden layer of second RBM is expressed as,

$$h^2 = \{h_1^2, h_2^2, \ldots, h_m^2, \ldots, h_c^2\}; 1 \le m \le c \tag{20}$$

The biases of visible layer and the hidden layer of second RBM is given as u^2 and v^2, respectively. For the second RBM, the weight vector is represented as,

$$w^2 = \{w_{mm}^2\}; 1 \le m \le c \tag{21}$$

where, w_{mm}^2 represents the weight between m^{th} visible neuron and m^{th} hidden neuron in the second RBM such that the dimension of the weight vector is $c \times c$. The output of the m^{th} hidden neuron is measured similar to the first case as,

$$h_m^2 = \sigma\left[v_m^2 + \sum_k a_k^2 w_{mm}^2\right] \forall a_k^2 = h_m^1 \tag{22}$$

where, v_l^2 is the bias associated with the m^{th} hidden neuron. Thus, the hidden layer output obtained is given by,

$$h^2 = \{h_l^2\}; 1 \le l \le q \tag{23}$$

The above Eq. forms the input to the MLP, and the input layer of MLP is given as,

$$N = \{N_1, N_2, \ldots, N_m, \ldots, N_c\} = \{h_m^2\}; 1 \le m \le c \tag{24}$$

where, c are the total neurons present in the input layer, which is given to the hidden layer output of the second RBM $\{h_m^2\}$. The hidden layer of the MLP is given as,

$$S = \{S_1, S_2, \ldots, S_A, \ldots, S_B\}; 1 \le A \le B \tag{25}$$

where, B represents total hidden neurons. Assume S_A as the bias of A^{th} hidden neuron, where $A = 1, 2, \ldots, B$. The third output of the MLP is given as,

$$o = \{o_1, o_2, \ldots, o_I, \ldots, o_J\}; 1 \le I \le J \tag{26}$$

where, J represents the number of neurons in the output layer. MLP has two weight vectors, one between the input layer and the hidden layer, and the other between the hidden layer and the output layer. Let w^E be the weight vector between the input and the hidden layers, as given below,

$$w^E = \{w_{mA}^E\}; 1 \le l \le c; 1 \le A \le B \tag{27}$$

where, w_{mA}^E is the weight between m^{th} input neuron and A^{th} hidden neuron so that the size of w^E is $c \times B$. Based on the weights in the neurons together with the bias, the hidden layer output is calculated as,

$$S_A = \left[\sum_{m=1}^{c} w_{mA}^E - S_m\right] = h_m^2 \tag{28}$$

where, X_K is the bias of hidden neuron and $S_m = h_m^2$, since the input to the MLP is the output of the second RBM. The weights between the hidden layer and the output layer are denoted as w^H and are given by,

$$w^H = \left\{w_{Kr}^H\right\}; 1 \leq K \leq M; 1 \leq r \leq z \tag{29}$$

Thus, the output vector can be computed based on the weight w^H and the hidden layer output, as formulated below,

$$o_r = \sum_{A=1}^{B} w_{Ar}^H - S_A \tag{30}$$

where, w_{Ar}^H is the weight between the A^{th} hidden neuron and r^{th} output neuron and S_A is the output of the hidden layer.

3.4.2 Fractional Bat algorithm for optimal weight update in DBN

The optimal tuning of the weights in the DBN facilitates the better classification of the input images and ensures effective classification accuracy. The proposed algorithm, Fractional Bat, is the integration of the Fractional calculus [20] in the Bat algorithm [19] so that the developed algorithm offers better benefits to classification. Thus, Bat algorithm is said to offer the convergence to the local optimum solution and is capable of dealing with the real solutions associated with the unknown search space. The fractional theory uses real or complex numbers for integral operators. Thus, the proposed fractional bat algorithm is used to determine the optimal weight value β for the fusion of low resolution and high-resolution feature images. The integration of Fractional theory increases the efficiency that formulates the result based on the entire training set. However, the algorithm takes a longer time to converge, which is nullified using bat algorithm that renders greater convergence rate. The algorithmic steps of the proposed Fractional Bat are illustrated in the following steps:

i) Weight Initialization: The total number of the weights is initialized as,

$$w_u ; (1 \leq u \leq v) \tag{31}$$

where, v is total weights and w_u is the u^{th} weight.

ii) Evaluate the fitness: The fitness of the solutions is evaluated for individual weight, to determine the optimal weight. The fitness is evaluated on the basis of minimum error, and the solution providing minimum error is evaluated as the optimal weight. The error is determined as,

$$MSE = \frac{1}{g}\left[\sum_{h=1}^{g} R_{t\arg et} \quad R_{est}\right]^2 \tag{32}$$

where, R_{est} represents the estimated output and $R_{t\arg et}$ represents the target output of the classifier.

iii) Computation of optimal weights for tuning DBN: Thus, the weight update between the layers of the MLP is performed using the position update of the Fractional Bat algorithm is represented as,

$$w_i^{l+1} = \alpha w_i^l + \frac{1}{2}\alpha w_i^{l\;\,1} + \frac{1}{6}\alpha(1 \quad \alpha) w_i^{l\;\,2} + \frac{1}{24}\alpha(1 \quad \alpha)(2 \quad \alpha) w_i^{l\;\,3} + \alpha w^l \tag{33}$$

where, α represents the discrete derivative order, w_i^l is i^{th} weight at iteration l, $w_i^{l\;\,1}$ denote i^{th} weight at iteration $l \quad 1$, $w_i^{l\;\,2}$ denote i^{th} weight at iteration $l \quad 2$, $w_i^{l\;\,3}$ denote i^{th} weight at iteration $l \quad 3$.

iv) Derive the optimal solution: The optimal weight is decided based on the minimum value of the fitness and is used for tuning the DBN.

v) Terminate: The update is continued for the maximum number of iterations to derive the optimal weights.

4. Discussion of Results

The analysis of the methods based on the evaluation metrics is done to prove the effectiveness of the proposed method.

4.1 Set-Up of the Experiment

The proposed algorithm for face recognition is executed in MATLAB using Windows 10 OS, 2GB RAM, and Intel i3 processor.

4.2 Dataset Description

The CVL Face database [18] is donated by Peter Peer in the year 1999. The dataset comprises different face image, which can be used for performing the analysis and for the experimentation purpose. The dataset is composed of the face images consisting of 114 persons with 7 images of each person. The image resolution is 640*480 pixels with jpeg format. The image contained in the dataset is captured with no flash and projection screens in the background. Here, the experimentation is carried out using four sets of input images for the evaluation

4.3 Evaluation Metrics

The analysis of the performances is done using three evaluation metrics, namely FAR, Accuracy, and FRR.

4.4 Comparative Methods

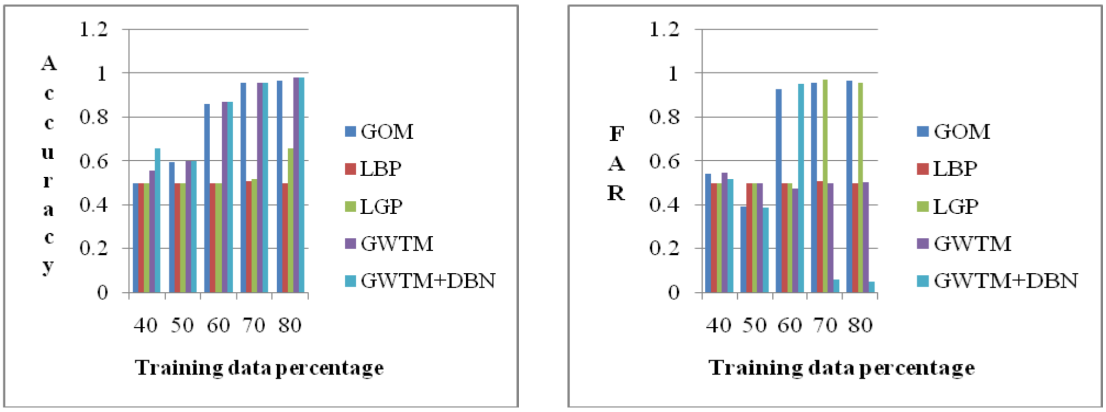
The comparative methods taken into consideration for evaluating the performance are Gabor Ordinal Measure (GOM), Local Binary Pattern (LBP), Local Gradient Pattern (LGP), Gabor Filter, Wavelet Transforms And Texture Operators (GWTM) and proposed GWTM+DBN.

4.5 Analysis Based on Comparative Methods

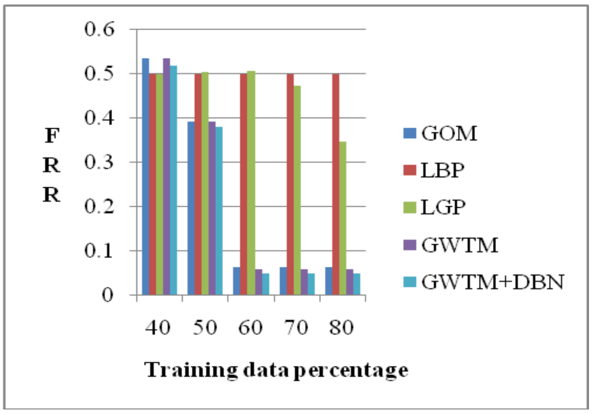
The comparative analysis of the existing methods with respect to the proposed method is done in terms of FAR, FRR and Accuracy.

a) Comparative analysis based on training data percentage

Fig. 3 depicts the comparative analysis of existing and proposed methods in terms of accuracy, FAR, and FRR by varying the training data from 40% to 80% data. The analysis in terms of accuracy parameter is depicted in Fig. 3a. When the training data percentage is 80, then the corresponding accuracy values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.97, 0.5, 0.66, 0.98, and 0.98 respectively. The analysis based on the FAR parameter is depicted in Fig. 3b. For 80% data, the corresponding FAR values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.97, 0.5, 0.956, 0.501, and 0.05 respectively. The analysis based on FRR parameter is depicted in Fig. 3c. When the training data percentage is 80, then the corresponding accuracy values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.065, 0.5, 0.3479, 0.06, 0.05 respectively. Based on the produced results, it is noted that proposed GWTM+DBN outstands other methods with maximal accuracy and minimal FAR, and FRR.



(a) (b)

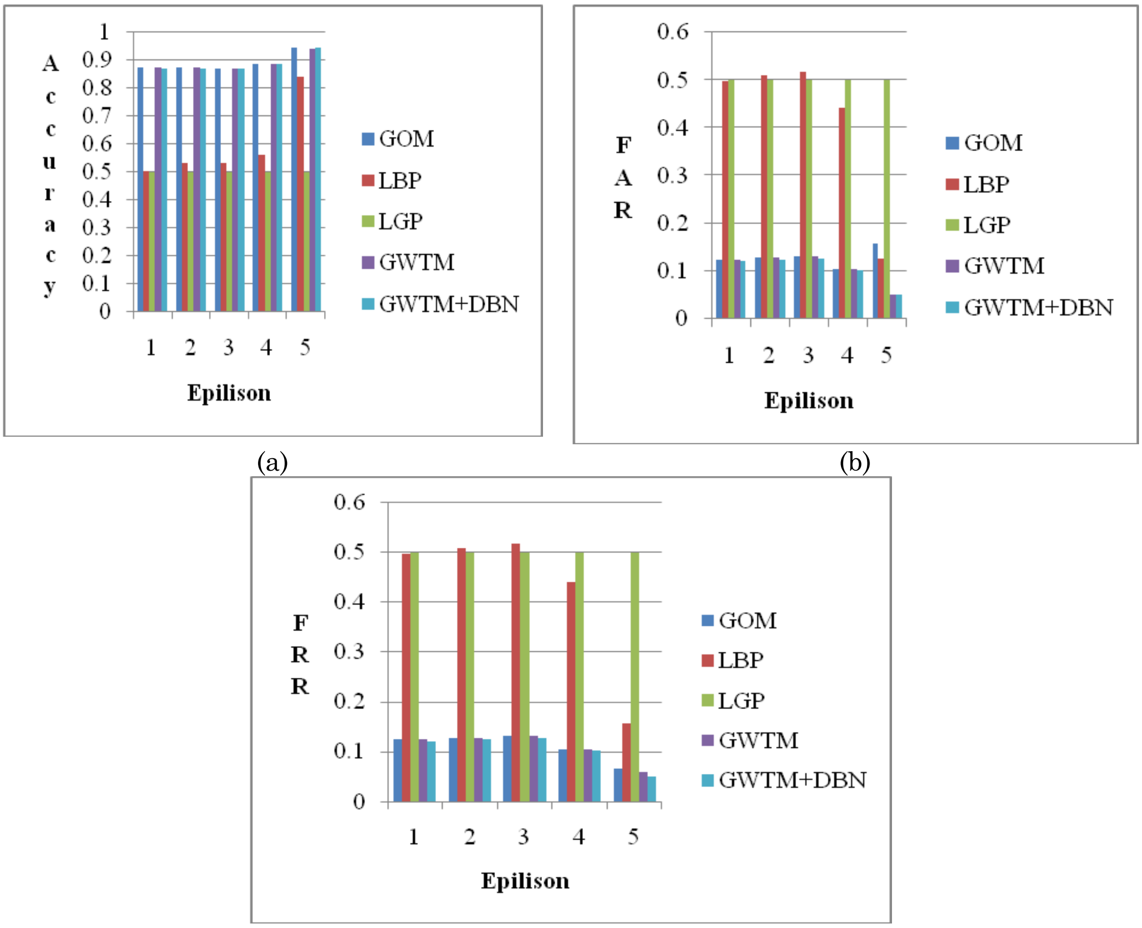


(c)

Fig. 3.*Comparative analysis based on training data percentage a) Accuracy b) FAR c) FRR*

b) Comparative analysis based on varying epsilon

Fig. 4 depicts the comparative analysis of existing and proposed methods based on FAR, accuracy and FRR by varying the epilison from 40 to 80 data. The analysis in terms of accuracy parameter is depicted in Fig. 4a. When the epilison is 5, then the corresponding accuracy values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.9436, 0.8423, 0.5, 0.9435, and 0.945 respectively. The analysis based on the FAR parameter is depicted in Fig. 4b. When the epilison is 5, the corresponding FAR values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.1577, 0.1255, 0.5, 0.05, and 0.051 respectively. The analysis based on FRR parameter is depicted in Fig. 4c. When the epilison is 5, then the corresponding accuracy values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.065, 0.1577, 0.5, 0.06, and 0.051 respectively. With respect to the results, it is noted that the proposed GWTM+DBN outstands other methods with maximal accuracy and minimal FAR, and FRR.



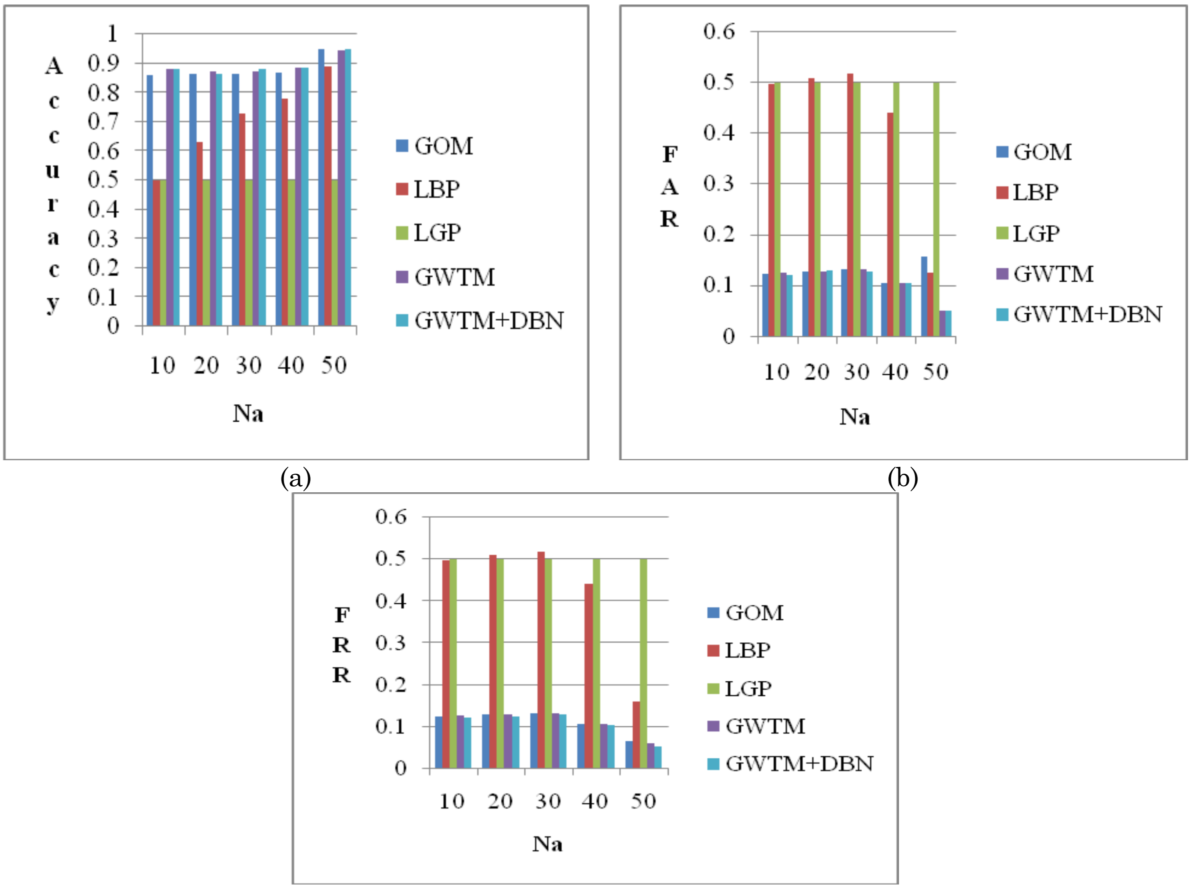
(a) (b)

(c)

Fig. 4.*Comparative analysis based on epilison a) Accuracy b) FAR c) FRR*

c) Comparative analysis based on number of draw attempts (Na)

Fig. 5 depicts the comparative analysis of existing and proposed methods in terms of accuracy, FAR, and FRR by varying the Na values. The analysis in terms of accuracy parameter is depicted in Fig. 5a. When the Na is 50, then the corresponding accuracy values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.95, 0.888, 0.502, 0.9435, and 0.95 respectively. The analysis based on the FAR parameter is depicted in Fig. 5b. When the epilison is 5, the corresponding FAR values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.1577, 0.1255, 0.5, 0.05, and 0.051 respectively. The analysis based on FRR parameter is depicted in Fig. 5c. When the epilison is 5, then the corresponding accuracy values measured by existing GOM, LBP, LGP, GWTM, and GWTM+DBN are 0.064, 0.1577, 0.5, 0.06, and 0.052 respectively. Based on the results, it is noted that the proposed GWTM+DBN outstands other methods with maximal accuracy and minimal FAR, and FRR.



(a) (b)

(c)

Fig. 5.*Comparative analysis based on number of attempts (Na) a) Accuracy b) FAR c) FRR*

5. Conclusion

In this research work, a low-resolution face recognition model is proposed using fractional bat technique and multi-kernel based DBN classifier. In this method, the input images are pre-processed to attain low-resolution images. To accomplish pre-processing, ROI extraction is employed. Based on the kernel regression model, the low-resolution image is converted to the high-resolution image for processing the subsequent steps. Then, the GWTM process is utilized for extracting the features. The wavelet transforms and Gabor filter is further processed using high and low-resolution images. The obtained images are fed to LBP for texture description and histogram representation. Lastly, the proposed technique is adapted for fusing the features using fractional bat algorithm, which is the integration of bat algorithm and fractional theory. At last, face recognition is performed using facial features with multi-kernel based DBN classifier. The proposed method outperformed other methods with higher accuracy of 0.98, minimum FAR and FRR of 0.05.

References

[1] Duan, Y., Lu, J., Feng, J. and Zhou, J., "Context-aware local binary feature learning for face recognition," IEEE transactions on pattern analysis and machine intelligence, vol.40, no.5, pp.1139-1153, 2018.

[2] Naphtali Abudarham, Lior Shkiller, GalitYovel, "Critical features for face recognition," Cognition vol.182, pp.73-83, January 2019.

[3] Huang, Z.H., Li, W.J., Wang, J. and Zhang, T., "Face recognition based on pixel-level and feature-level fusion of the top-level's wavelet sub-bands," Information Fusion, vol.22, pp.95-104, 2015.

[4] J.W. Zhao, Z.H. Zhou, F.L. Cao, "Human face recognition based on ensemble of polyharmonic extreme learning machine," Neural Comput. Appl. vol.24, no.6, pp.1317–1326, 2014.

[5] Jafri, R. and Arabnia, H.R., "A survey of face recognition techniques," Jips, vol.5, no.2, pp.41-68, 2009.

[6] S. W. Park and M. Savvides, "A multifactor extension of linear discriminant analysis for face recognition under varying pose and illumination", Advanced signal processing, 2010.

[7] Heisele, B., Ho, P. and Poggio, T., "Face recognition with support vector machines: Global versus component-based approach," In Proceedings Eighth IEEE International Conference on Computer Vision, vol. 2, pp. 688-694, 2001.

[8] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," TPAMI, vol.19, no.7, pp.711–720, 1997.

[9] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," TPAMI, vol.28, no.12, pp.2037–2041, 2006.

[10] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol.3, no.1, pp.71–86, 1991.

[11] Masi, I., Chang, F.J., Choi, J., Harel, S., Kim, J., Kim, K., Leksut, J., Rawls, S., Wu, Y., Hassner, T. and AbdAlmageed, W., "Learning pose-aware models for pose-invariant face recognition in the wild," IEEE transactions on pattern analysis and machine intelligence, vol.41, no.2, pp.379-393, 2019.

[12] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," TIP, vol.11, no.4, pp.467–476, 2002.

[13] [13] Z. Lei, M. Pietikainen, and S. Z. Li, "Learning discriminant face descriptor," TPAMI, vol.36, no.2, pp.289–302, 2013.

[14] J. Lu, V. E. Liong, Z. Xiuzhuang, and J. Zhou, "Learning compact binary face descriptor for face recognition," TPAMI, pp.2041–2056, 2015.

[15] Z. Lu and L. Zhang, "Face recognition algorithm based on discriminative dictionary learning and sparse representation," Neurocomputing vol.174, pp.749–755, 2016.

[16] Thomas, R. and Rangachar, M.J.S., "Fractional bat and multi-kernel-based spherical SVM for low resolution face recognition," International Journal of Pattern Recognition and Artificial Intelligence, vol.31, no.8, pp.1756014, 2017.

[17] [17] Robert Strack, Vojislav Kecman, Beata Strack and QiLi, "Sphere Support Vector Machines for large classification tasks", Neurocomputing., pp. 59-67, 2013.

[18] "CVL Face Database," http://www.lrv.fri.uni-lj.si/facedb.html Accessed on May 2019.

[19] Xin-She Yang A New Metaheuristic Bat-Inspired Algorithm Department of Engineering, University of Cambridge 2010.

[20] E. J. Solteiro Pires, J. A. Tenreiro Machado, "Particle swarm optimization with fractional-order velocity", springer, pp. 296-301, Jan 2011.

[21] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," Neural Comput., vol. 18, pp. 1527–1554, 2006.