International Journal of Soft Computing 2 (2): 216-224, 2007

Novel Techniques for Face Recognition Identification and Labeling

[1]Mikhled Alfaouri, [2]Hilal M. Al-Bayatti and [3]Nada N. Al-Ramahi
[1]Department of Communication and Electrical Engineering,
Philadelphia University, Amman, Jordan
[2]Department of Computer Science,
Amman Arab University for Graduate Studies, Amman, Jordan
[3]Researcher at Philadelphia, Amman, Jordan

Abstract: In this study the method is presented for face image recognition identification and labeling. The method is based on using the combination of the Discrete Multi-Wavelet Transform (DMWT) and the Inverse Discrete Wavelet Transform (IDWT) followed by a Neural Network (NN). In this method the resulting coefficients were computed by the proposed multi-wavelets transform for single-level decomposition. It can be readily observed that the lowpass block (upper left corner) actually contains one lowpass subband and three bandpass subbands. The LL subbands resemble a smaller version of the original image. In this study the Inverse Discrete Wavelet Transform (IDWT) of LL coefficients will be obtained. The resultant of feature extraction is obtained by the Inverse Discrete Wavelet Transform, (IDWT) and applied to the first column to the Neural Network (NN) for recognition identification of the face image. This method gave an excellent result: 99% for a database of 50 different face images were excellent which indicates that the suggested algorithm in an excellent tool to process the database of standard pose of face image. A detection rate of 90% was achieved for facial movement such as a smile or other moves of face identification. The algorithm is implemented using MATLAB programming languages version 7.

Key words: Discrete multi-wavelet transform, inverse discrete wavelet transform, neural network, novel transformation, human face recognition, labeling, subband

INTRODUCTION

This study provides the algorithm and step by step method of commercial face image recognition identifi-cation to increase the capability of security and surveillance systems using the Novel transformation and Neural Network (NN).

The proposed transform is considered as feature extractor of the decomposed reference images into different frequency subbands and amid-range frequency subband for caricature image for the representation of the given face image. Recent researches on the methods of Discrete Multi-Wavelet Transform (DMWT) and the Inverse Discrete Wavelet Transform (IDWT) have been focused on the development of the basic theory for face image recognition. The novelty of this research is based on constructing new multi-filter for de-noising, comp-ressing and recovering the original image. The result obtained by the suggested method is promised and will shed some light to researchers to adapt this technique. Evaluations have generally shown interesting promise for Discrete Multi-Wavelet Transform (DMWT) and the Inverse Discrete Wavelet Transform (IDWT), they have been limited and are more theoretical still needs to find a simple and easy algorithm to use for computing Discrete Multi-Wavelet Transform (DMWT) and the Inverse Discrete Wavelet Transform (IDWT) coefficients. The Neural Network (NN) is used here as a classifier of two phases one is called the training and the other is called the testing.

THE COLLECTED DATABASE

In order to take a sufficient database, a total of 50 face images are collected. The following procedure is used for the data collection (Al-Ramahi, 2006):

Step (1): Input the color face image of any size.
Step (2): Give the label for each face image as a sequence number.
Step (3): Convert the face image into gray-scale form.
Step (4): Resize the face image of size of power of 2 if is not to get the nearest size because it must be the power of 2 of each image as suggested by Al-Ramahi (2006).
Step (5): Save the resized face image when it is done.

Corresponding Author: Mikhled Alfaouri, Department of Communication and Electrical Engineering, Philadelphia University, Amman, Jordan

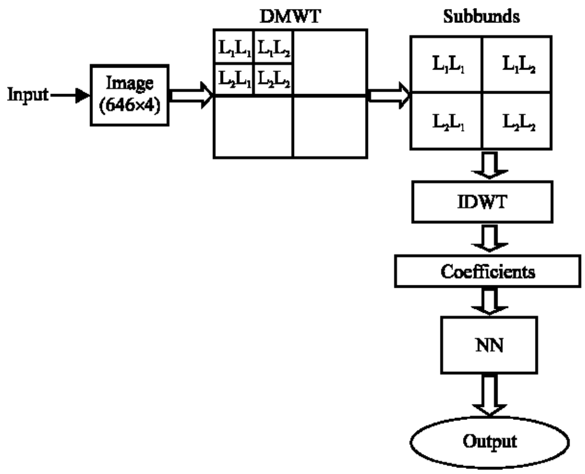

Fig. 1: Main block diagram of the combined transform

THE MAIN BLOCK DIAGRAM OF THE PROPOSED IDENTIFICATION ALGORITHM USING NEURAL NETWORK (NN)

The main block diagram of this method consists of first: The preprocessing method of the Discrete Multi-Wavelet Transform (DMWT) and the Inverse Discrete Wavelet Transform (IDWT) algorithm, the aim of the preprocessing is to associate the given scalar input signal of length N to a sequence of length-2 vector in order to start multi-wavelet transformation process, in other word, because the given scalar signal consists of one stream of the DMWT algorithm the results of (L_1L_1, L_1L_2, L_2L_1, L_2L_2) of sub bands as the input data Wavelet Transform is obtained from them. The result obtained is placed serially in a vector form in order to be the input of the Neural Network (NN) (Fig. 1).

WAVELET TRANSFORM

A particular set of wavelets is specified by a particular set of numbers, called wavelet filter coefficients (Alfaouri, 1997). In this study, will largely restrict our self to wavelet filters in a class discovered by Daubechies (Laurenz, 1997). This class includes members ranging from highly localized to highly smooth. The simplest (and most localized) member, often called Daubechies four (DAUB4 or D4), has only four coefficients, $c_0,\dots c_3$. For moment specialize this case for ease of notation (Alfaouri, 1997).

$$\begin{aligned} c_0 &= (1+\sqrt{3})/4\sqrt{2} & c_1 &= (3+\sqrt{3})/4\sqrt{2} \\ c_2 &= (3-\sqrt{3})/4\sqrt{2} & c_3 &= (1-\sqrt{3})/4\sqrt{2} \end{aligned} \quad (1)$$

Consider the following transformation matrix acting on a column vector of data to its right:

$$\begin{bmatrix} C_0 & C_1 & C_2 & C_3 & 0 & 0 & \cdots & & & & \\ C_3 & -C_2 & C_1 & -C_0 & 0 & 0 & \cdots & & & & \\ 0 & 0 & C_0 & C_1 & C_2 & C_3 & \cdots & & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & & & \\ & & & & & & & C_0 & C_1 & C_2 & C_3 \\ & & & & & & & C_3 & -C_2 & C_1 & -C_0 \\ C_2 & C_3 & & & & & & & & C_0 & C_1 \\ C_1 & -C_0 & & & & & & & & C_3 & -C_2 \end{bmatrix} \quad (2)$$

For such a characterization to be useful, it must be possible to reconstruct the original data vector of length N from its N/2 smooth or s-components and its N/2 detail or d-components. That is affected requiring the matrix (2) to be orthogonal, so that its inverse is just its transpose of the matrix.

$$\begin{bmatrix} C_0 & C_3 & & \cdots & & & C_2 & C_1 \\ C_1 & -C_2 & & \cdots & & & C_3 & -C_0 \\ C_2 & C_1 & C_0 & C_3 \cdots & & & & \\ \cdots & \cdots & \cdots & \cdots\cdots & & & & \\ & & & C_2 & C_1 & C_0 & C_3 & \\ & & & C_3 & -C_0 & C_1 & -C_2 & \\ & & & & C_2 & C_1 & C_0 & C_3 \\ & & & & C_3 & -C_0 & C_1 & -C_2 \end{bmatrix} \quad (3)$$

One sees immediately that matrix (3) is the inverse to matrix (2). In fact, DAUB4 is only the most compact of a sequence of wavelet sets.

INVERSE DISCRETE WAVELET TRANSFORM COMPUTATION ALGORITHM

To reconstruct the original 2-D signal (N×N matrix) from the discrete wavelet transformed 2-D signal (N×N matrix), the Inverse Discrete Wavelet Transform (IDWT) should be used. For wavelet decomposed signal, it is possible to reconstruct the original image matrix using inverse transformation matrix of (3) (Laurenz, 1997).

A general procedure for computing inverse DWT wavelet: To compute a single-level 2-D Inverse Discrete Wavelet Transform of D4 wavelet, the next step is followed as in (Laurenz, 1997).

Constructing an inverse transformation matrix: Using D4 inverse transformation matrix (3) format, an N×N inverse transformation matrix should be constructed using c0,….,c3 coefficients values given in Eq. 1. The inverse transformation matrix is the transpose of the transformation matrix since the transform is based on orthogonal principle.

Column reconstruction: Column reconstruction steps are applied to the DWT matrix as follows:

- Transpose the wavelet transformed matrix as in the first step.
- Apply shuffling to the wavelet transformed matrix transpose by rearranging the rows from 1 to N/2 to be the odd rows of the resultant shuffled matrix starting from the first row and rearranging the rows from N/2 to N to be the even rows of the resultant shuffled matrix starting from the second row.
- Apply matrix multiplication by multiplying the inverse transformation matrix with the resultant shuffled matrix which results in computing the column reconstructed matrix.

Row reconstruction: Row reconstruction steps are the same for column reconstructed matrix.

MULTI-WAVELET TRANSFORM

Since the wavelet and multi-wavelets transformations are directly applicable to one-dimensional signals only. But face images are considered to be two- dimensional signals, so there must be away to process them with a 1-D transform (Aleix and Avinash, 2001).

The two main categories of methods for doing this: *separable* and *non separable* algorithms.

Methods simply work on each dimension in series. The typical approach is to process each of the rows in order and then process each column of the result. Non-separable methods work in both image dimensions at the same time. While non-separable methods can offer benefits over separable methods, such as savings in computation, they are generally more difficult to implement. Computing Discrete Multi-wavelet Transform, scalar wavelet transform can be written as follows:

$$\begin{matrix} H_0 & H_1 \\ G_0 & G_1 \\ 0 & 0 \\ 0 & 0 \end{matrix} \quad \begin{bmatrix} H_2 & H_3 & 0 & 0 \\ G_2 & G_3 & 0 & 0 \\ H_0 & H_1 & H_2 & H_3 \\ G_0 & G_1 & G_2 & G_3 \end{bmatrix}$$

Where H_i and G_i are low and high pass filter impulse responses. They are 2-by-2 matrices which can be written as follows:

$$\begin{bmatrix} H_0 & H_0 & H_1 & H_1 & \dots & \dots & \dots & \dots & \dots \\ H_0 & H_0 & H_1 & H_1 & \dots & \dots & \dots & \dots & \dots \\ G_0 & G_0 & G_1 & G_1 & \dots & \dots & \dots & \dots & \dots \\ G_0 & G_0 & G_1 & G_1 & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & H_0 & H_0 & H_1 & H_1 & \dots \\ 0 & 0 & 0 & 0 & H_0 & H_0 & H_1 & H_1 & \dots \\ 0 & 0 & 0 & 0 & G_0 & G_0 & G_1 & G_1 & \dots \\ 0 & 0 & 0 & 0 & G_0 & G_0 & G_1 & G_1 & \dots \end{bmatrix} \quad (5)$$

By examining the transform matrices of the scalar wavelet and multi-wavelets, one can see that in the multi-wavelets transform domain there are first and second low-pass coefficients followed by first and second high pass filter coefficients rather than one low-pass coefficient followed by one highpass coefficient, Therefore, if we separate these four coefficients, there are four subbands in the transform domain.

Since multi-wavelet decompositions produce two low-pass subbands and two highpass subbands in each dimension, the organization and statistics of multi-wavelet subband differ from the scalar wavelet case. A closer examination of the differences suggests a method for improving the performance of multi-wavelets in image applications. During a single level of decomposition using a scalar wavelet transform, the 2-D image data is replaced with four blocks corresponding to the subbands representing either lowpass or highpass in both dimensions. These subbands are illustrated in (Fig. 2a).

The subband labels in (Fig. 2a) indicate how the subband data was generated. For example, the data in subband LH was obtained from highpass filtering of the rows and then lowpass filtering of the columns. The multi-wavelets used here have two channels, so there will be two sets of scaling coefficients and two sets of wavelet coefficients. Since there is multiple iteration over the lowpass data are desired, the scaling coefficients for the two channels are stored together. Likewise, the wavelet coefficients for the two channels are also stored together. The multi-wavelet decomposition subbands are shown in (Fig. 2b).

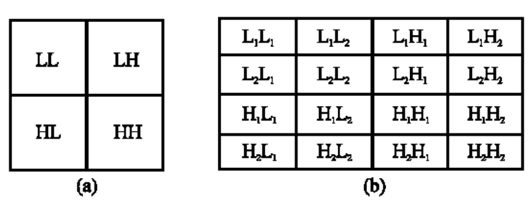

Fig. 2: Image subbands after a single-level decomposition, for (a) Scalar wavelets and (b) multi-wavelets

For multi-wavelets the L and H have subscripts denoting the channel to which the data corresponds. For example, the subband labeled L_1H_2 corresponds to data from the second channel highpass filter in the horizontal direction and the first channel lowpass filter in the vertical direction. This shows how a single level of decomposition is done.

In practice, there is more than one decomposition is performed on the image data. Successive iterations are performed on the low pass coefficients from the pervious stage to further reduce the number of low pass coefficients. Since the lowpass coefficients contain most of the original signal energy, this iteration process yields better energy compaction. After a certain number of iterations, the benefits gained in energy compaction becomes rather negligible compared to the extra computational effort.

Usually five level of decomposition are used, a single level of decomposition with a symmetric-anti-symmetric multi-wavelet is roughly equivalent to two levels of wavelet decomposition. Thus a 3-level multi-wavelet decomposition effectively corresponds to 6-level scalar wavelet decomposition

Scalar wavelet transform give a single quarter-sized subband from the original larger subband, as shown in (Fig. 2a).

The multi-level decomposition is performed in the same way. The multi-wavelet decomposition iterates on the lowpass coefficients from the pervious decomposition, as shown in (Fig. 3). In the case of the scalar wavelets, the lowpass quarter image is a single subband. But when the multi-wavelet transform is used, the quarter image of lowpass coefficients is actually a 2×2 block of subbands (the L_iL_j subbands in (Fig. 2b).

Due to the nature of the preprocessing and symmetric extension method, data in these different subbands becomes intermixed during iteration of the multi-wavelet transform. The intermixing of the multiwavelet lowpass subbands leads to suboptimal results.

Consider the multi-wavelets transform coefficients resulting from a single-level decomposition. It can be readily observed that the 2×2 "lowpass" block (upper left corner) actually contains one lowpass subband and three bandpass subbands. The L_1L_1 subband resembles a smaller version of the original image, which is a typical characteristic of a true lowpass subband. In contrast, the L_1L_2, L_2L_1 and L_2L_2 subbands seem to process characteristics more like those of high subbands. Also only L_1L_1 subband contains coefficients with a large DC value and a relatively uniform distribution. The L_1 ,L_1,H_1 and H_2 subbands ,measured along the vertical direction. Note that L_2 subband looks more like the highpass bands H_1 and H_2 than the L_1 subband.

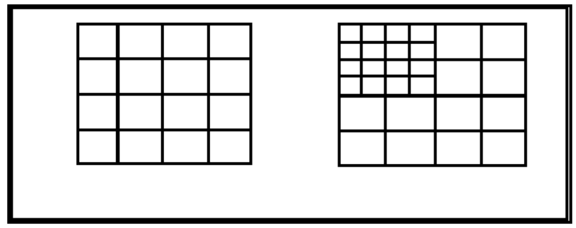

Fig. 3: Conventional iteration of multiwavelet decomposition

A GENERAL PROCEDURE FOR COMPUTING DMWT USING AN OVER-SAMPLED SCHEME OF PREPROCESSING (REPEATED ROW PREPROCESSING)

A general procedure for computing can be made for computing a single-level 2-D DMWT using GHM four multi-filter and using an over-sampled scheme of preprocessing (Michael and Amy, 2001).

By using an over-sampled scheme of preprocessing (repeated row preprocessing), The DMWT matrix is doubled in dimension compared with that of the input which should be a square matrix N x N where N must be power of 2 (Sidney *et al*., 1998). Transformation matrix dimensions equal image dimensions after preprocessing.

To compute single-level 2-D Discrete multi-wavelets Transform, the next steps should be followed:

Checking image dimensions: Image matrix should be a square matrix, N x N matrix, where N must be power of 2. So that the first step of the transform procedure is checking input image dimensions .If the image is not a square matrix some operation must be done to the image like resizing the image or adding rows or column of zeros to get a square matrix.

Constructing a transformation matrix: Using the transformation matrix format, an N×N transformation matrix should be constructed using GHM low-and highpass filter matrices (Contronei *et al*., 2000). After substituting GHM matrix filter coefficients values as a 2N×2N transformation matrix results with the same dimensions as the input image matrix dimensions after preprocessing.

Where H_k for GHM system are four scaling matrices as follows:

$$H_0 = \begin{bmatrix} 3+5\sqrt{2} & 4/5 \\ -1/20 & -3/10\sqrt{2} \end{bmatrix} \quad H_1 = \begin{bmatrix} 3+5\sqrt{2} & 0 \\ 9/20 & 1/1\sqrt{2} \end{bmatrix} \qquad (6)$$
$$H_2 = \begin{bmatrix} 0 & 0 \\ 9/20 & -3/10\sqrt{2} \end{bmatrix} \quad H_2 = \begin{bmatrix} 0 & 0 \\ -1/20 & 0 \end{bmatrix}$$

Where G_k for GHM system are four scaling matrices as follows:

$$G_0 = \begin{bmatrix} -1/20 & -3/10\sqrt{2} \\ 1/10\sqrt{2} & 3/10 \end{bmatrix} \quad G_1 = \begin{bmatrix} 9/20 & -1\sqrt{2} \\ -9/10\sqrt{2} & 0 \end{bmatrix}$$
$$G_2 = \begin{bmatrix} 9/20 & -3/10\sqrt{2} \\ 9/10\sqrt{2} & -3/10 \end{bmatrix} \quad G_3 = \begin{bmatrix} -1/20 & 0 \\ -1/10\sqrt{2} & 0 \end{bmatrix} \tag{7}$$

Preprocessing rows: Row preprocessing doubles the number of the input matrix rows. So if the 2-D input is N x N matrix elements, after row preprocessing the result are 2N x N matrix. The odd rows 1, 3… 2N-1 of this resultant matrix are the same original matrix rows values 1,2,3…,N, respectively. While the even rows numbers 2, 4…2N are the original rows values multiplied by á. For GHM system functions $\alpha = 1\sqrt{2}$.

Transformation of image rows: They can be done as follows:

- Apply matrix multiplication to the 2N×2N constructed transformation matrix by the 2N×2N preprocessed input image matrix.
- Permute the resulting 2N×2N matrix rows by arranging the row pairs 1,2 and 5,6, ….,2N-3,2N-2 after each other at the upper half of the resulting matrix rows, then the row pairs 3,4 and 7,8,….,2N-1,2N below them at the next lower half.

Preprocess columns: To repeat the same procedure used in preprocessing rows:

- Transpose the row transformed 2N×N matrix resulting from step 4.
- Repeat step 3 to the N×2N matrix (transpose of the row transformed 2N×N matrix) which results in 2N×2N column preprocessed matrix.

Transformation of image columns: To get the final transformation of image columns is applied next to the 2Nx2N column preprocessed matrix as follows:

- Apply matrix multiplication to the 2N×2N constructed transformation matrix by the 2N×2N column preprocessed matrix.
- Permute the resulting 2N×2N matrix rows by arranging the row pairs 1,2 and 5,6…,2N-3,2N-2 after each other at the upper half of the resulting matrix rows, then the row pairs 3,4 and 7,8,…,2N-1,2Nbelow them at the next lower half.

The final transformed matrix: To get the final transformed matrix the following should be applied:

- Transpose the resulting matrix from column transformation step.
- Apply coefficient permutation to the resulting transpose matrix. Coefficient permutation is applied to each of the basic four subbands of the resulting transpose matrix so that each subband permute rows then permute columns.

Finally, a 2N×2N DMWT matrix results from the N×N original image matrix using repeated row preprocessing.

THE TRAINING PHASE AND TEMPLATES GENERATION

The Novel transformation is proposed to be used for feature extraction of the images. Figure 4 Shows the flowchart of this algorithm. The algorithm consists of the following steps (Al-Ramahi, 2006).

Step (1): Input the face image into Discrete Multi-wavelet Transform (DMWT) *proposed* repeated row algorithm.

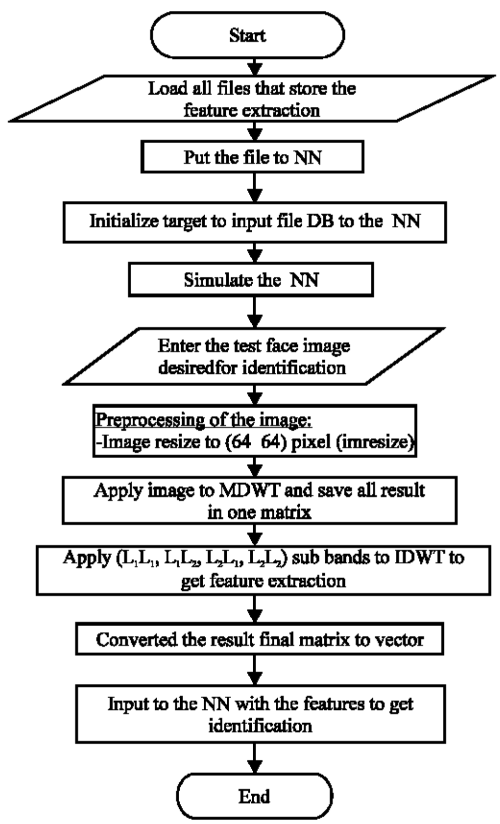

Fig. 4: The learning and testing algorithm

Step (2): Multi-wavelet filter bank require a vector-valued input signal. This is another issue to be addressed when multi-wavelets are used in the transform process, a scalar-valued input signal must some how be converted into a suitable vector-valued signal. There are a number of ways to produce such a signal from 2-D image data.

Step (3): Input face image (64×64) matrix and apply the following steps:

- Let X the input 2-D signal.
- Matrix input 2-D signal, X and construct a transformation matrix using GHM low- and high-pass filters
  As GHM filters, H's and G's, are 2×2 matrices, the transformation matrix, W, dimension will be 16×16 (2N×2N) after substituting filters coefficients values.
- Apply row preprocessing to the input 2-D matrix, X using repeated row preprocessing and $\alpha = 1\sqrt{2}$.

$$\xrightarrow[\quad X \quad]{\text{Preprocess rows}} \tag{8}$$

- Apply row transformation
  - Let, $[Z]=[W1]\times[x]$ (9)
  - Permute[Z],

$$\xrightarrow{\text{Permute}} \tag{10}$$

- Apply column transformation
  - Transpose [P] matrix.
    $[P] = [P]^t$ (11)
  - Preprocess $[P]^t$ to get [p].

$$\xrightarrow{\text{Preprocess Column}} \tag{12} p$$

  - Let $[b] = [W1]\times [p]$
  - Permute [b] to get [B] matrix which is (128×128) matrix.

$$\xrightarrow{\text{Permute}} b \tag{14}$$

- The final DMWT matrix$[Y]$ result from the following steps,
  - Transpose [B] matrix to get [y] matrix .
  - $[y] = [B]^t$

- Apply coefficients permutation to each of the four basic subbands of matrix[y] each subband to get [Y] matrix.

So, [Y] is the final single-level DMWT matrix this algorithm is given in (Fig. 5).

Step (4): Input the (L1L1,L1L2,L2L1,L2L2) subbands each subband size (64×64) to compute 2D in the Inverse Discrete Wavelet Transform (IDWT).

- For an (128×128) 2D signal [Y], constructs a Daubechies four (D4) inverse transformation matrix which is the transpose of transformation matrix, W.
- Apply column reconstruction as follows:
  Transpose the LL subbands ,that is,

$$[P] = [Y]^t \tag{16}$$

Apply to the [P] to get [Z].

$$[P] \text{ shuffling } [Z] \tag{17}$$

Obtain the Inverse Discrete Wavelet Transform (IDWT) using Eq. 3.

$$\text{Let, } [y] = [W]^t \times [Z] \tag{18}$$

- Apply row reconstruction as follows:

Transpose [y], that is.

$$[b] = [y]^t \tag{19}$$

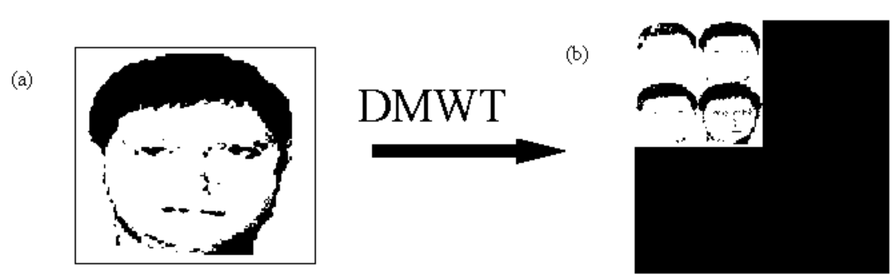

Fig. 5: Face image (a) original (b) after single-level of DMWT using a repeated row preprocessing

Fig. 6: (c) After single-level of DMWT using repeated row preprocessing take just LL subbands (d) after single-level of IDWT using caricature image

Apply shuffling to the [b] to get matrix[a].

$$[P] \text{ Shuffling } [Z] \quad (20)$$

The reconstructed 2-D signal is,

$$[X] = [W]^t \times [a] \quad (21)$$

So, [X] is the final single-level IDWT matrix this algorithm is given in (Fig. 6) then go back face image like the orignal.

Step (5): The result matrix from IDWT is the feature extraction.
Step (6): Convert the final result matrix to a vector and to the Data Base file.
Step (7): Put the Data Base file (face image) and feed it to the Neural Network (NN).
Step (8): Store the weights of the Neural Network (NN) for the identification phase.

The Neural Network consists of one input node and to each of 50 output nodes corresponding to the 50 different images under classification. Thus each output node represents only one given image.

Neural network training: The summarized structure of Neural Network (NN) is given in Fig. 7.

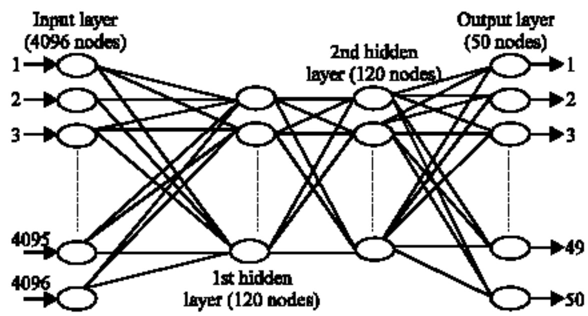

Fig. 7: The neural network classifier architecture for image Identification

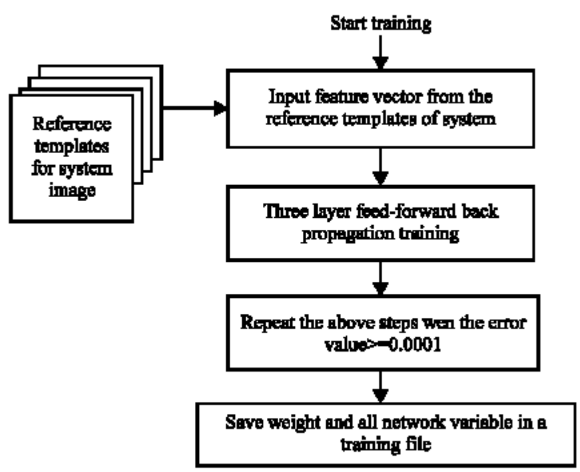

Fig. 8: Block diagram for the (NN) training procedure

NEURAL NETWORK TRAINING

The identification algorithm: The output of the Novel transformation is applied to the Neural Network to identify the images. It uses Back Propagation neural network (BP) with training Algorithm of Batch Gradient and Momentum (traingdx). The block diagram of the training procedure is given in (Fig. 8) (Al-Ramahi, 2006).

The flowchart of this algorithm is given in (Fig. 9), the algorithm consists of the following steps (Al-Ramahi, 2006):

Step (1): Load all the files obtained from the training phase to the Neural Network (NN).
Step (2): Compute the Discrete Multi-Wavelet Transform (DMWT) and Inverse Discrete Wavelet Transform (IDWT) as the given training, phases for the testing image.
Step (3): Train the Neural Network (NN) for the testing image classification.

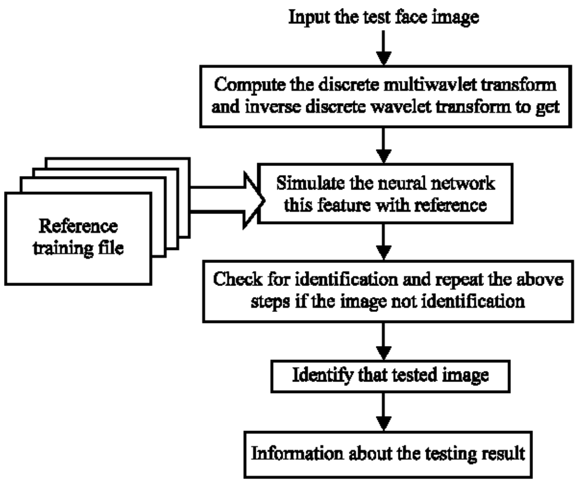

Fig. 9: Algorithm of identification image

RESULTS AND DISCUSSION

This study contains of the results of face image identification depending on the training set that generated using Neural Network (NN) (Al-Ramahi, 2006).

To identify any images, first it is converted to a gray image of size (64×64) then obtain the image in MDWT approximates for (L1L1, L1L2, L2L1, L2L2), the damnation for each sub band (64 (64× pixel for faster convergence, then inter the 4 sub band to Inverse Discrete Wavelet Transform IDWT and the other results from all vectors are saved in one matrix. This result matrix is a feature extraction convert to vector to take the first column from the final result to be used for the identification. The Back Propagation neural network is simulated with this vector, by choosing the larger output to get the corresponding target of that output. The maximum output of the *image,* Back Propagation (BP) is related to the corresponding image which means if the corresponding label for the output is (1) the corresponding image of label (1) has to be found. The above paragraph summarized in a flowchart given in (Fig. 10).

The main step of the algorithm includes the following:

Step (1): Enter the testing image.
Step (2): Apply the (DMWT and IDWT) to the given image to get the feature extraction.
Step (3): The feature extraction is applied to Neural Network (NN) for recognition.
Step (4): If the Neural Network (NN) identifies the label of this image will be printed.

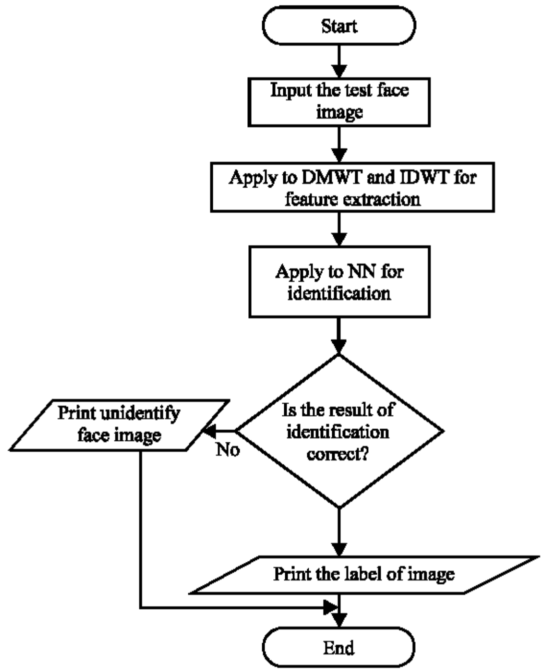

Fig. 10: Algorithm of test image

CONCLUSION

It has been shown throughout this study that facial transforms have a great impact on the capabilities of all facial recognition techniques. Method of face images recognition identification was implemented and tested based upon the introduced combination. A method which depends on combining the Discrete Multi-Wavelet Transform (DMWT) Repeated Row Algorithm (RRA) and the Inverse Discrete Wavelet Transform (IDWT) in conjunction with Neural Network (NN).

After testing the suggested algorithm with the Neural Network (NN) both give good result. From the previous discussion the following points can be concluded:

- Combined techniques give rise to successful results. This was achieved through the suggested combination of Discrete Multi-Wavelet Transform (DMWT) and using only the LL subbands obtained by the Inverse Discrete Wavelet Transform (IDWT).
- Using the Neural Network (NN) can help to achieve good results: 99% was achieved for standard pose face identification and result 96% was achieved for smile or move of face identification.
- The use of Neural Network (NN) to identify matches has been shown to be a good method of face images recognition and labeling especially as they are capable of dealing with facial occlusion and transforms.

REFERENCES

Al-Ramahi and Nada, N., 2006. Wavelet Based Automatic Image Identification And Labeling. Paper Accepted for Publication at The Five International Conference on: Computational Aspects and their Application in Electrical Engineering, Philadelphia University-Amman Jordan and IEE-Jordan.

Alfaouri Mikhled, 1997. Time-Frequency Analysis of Non-Stationary Signal, AL-Balqa. J. Res. Studies, Amman-Jordan Vol. 5, No.1.

Aleix, M. Martinez and Avinash C. Kak, 2001. PCA versus LDA, IEEE Trans. Pattern Anal. Matching Intellig., 23: 228-233.

Contronei, M., D. Lazzaro, L.B. Montefusco and L. Puccio, 2000. Image Compression Through Embedded Multiwavelet Transform Coding, IEEE Trans. Image Processing, 9: 184-189.

Laurenz Wiskott, Jean-Marc Fellous, Norbert Kruger and Christoph von der Malssburg, 1997. Face Recognition by Elastic bunch Graph Matching, IEEE Trans. Pattern Anal. Matching Intellig., 19: 775-779.

Michael B. Martin and Amy E. Bell, 2001. New Image compression Techniques Using Multiwavelets and Multiwavelet Packets, IEEE Trans. Image Processing, Vol. 10, No.4.

Sidney Burrus, Ramesh A. Gopinath and Haitao Guo, 1998. Introduction to wavelets and wavelet Transforms, Prentice-Hell, Inc., Lena Image.