A COMBINATION OF LOCATION AVERAGING FEATURE REDUCTION TECHNIQUE WITH RECOGNITION ALGORITHMS FOR FACE RECOGNITION SYSTEM

Parivazhagan.A[1*], Dr.BrinthaTherese.A[2]

[1&2]School of Electronics Engineering, VIT University, Chennai Campus, Chennai 600127, INDIA
E-mail:parivazhagan.a2013@vit.ac.in

Abstract

Nowadays in biometric recognition, face recognition is the pioneering process, today all the important high alert areas are monitored by cameras, but still now there are few drawbacks in facial recognition due to several reasons. Pose variation is one of the major drawbacks, in this work methods are explained and practised to solve the pose variation problem. The recognition of person's face using new recognition algorithms is the major research area going all around the world. In this proposed work an efficient feature extraction/reduction technique called Location averaging technique is combined with four object recognition techniques for facial recognition. Location averaging technique combines separately with Gaussian kernel, Eigen vector, Max-Min comparison and Haar function techniques, and produces effective results. Results are analysed using runtime, mismatching and accuracy for 170 images of two standard face databases. Accuracy percentage of about 99% is obtained through these novel recognition methods.

Keywords: Location averaging technique, Gaussian kernel, Eigen vector, Max-Min comparison, Haar function, Feature reduction

I. INTRODUCTION

In security systems identifying a person or an object, with good accuracy is the basic criteria; during security activities biometrics plays a vital role in identifying human beings. Every day new techniques are arriving massively for human recognition, even though a perfect accuracy is questionable, due to several errors. Face recognition [5, 6,15,16] provides a perfect accuracy for human identification, compared to all other modem of recognition ; at the same time it also have some difficulties, to overcome that here a novel feature reduction technique called Location averaging technique is practiced. In this proposed work pose variation problem is handled with Location averaging technique [10], and this technique is combined with four recognition techniques to recognize the face images. A novel recognition technique called as Max-Min comparison is also practiced. Gaussian kernel [1-3], Eigen vector [7, 8], Max-Min comparison and Haar function [11, 12] methods are combined with Location averaging technique for facial recognition.

Rest of the paper is organized as, *1. Reduction technique*: 2. Location averaging technique. *Recognition methods*: 3.Location averaged Gaussian kernel 4.Location averaged Eigen vector 5.Location averaged Max-Min comparison 6.Location averaged Haar function 7.Results and Discussions 8.Comparisons 9.Conclusions, and References the overall process is shown Fig.1.
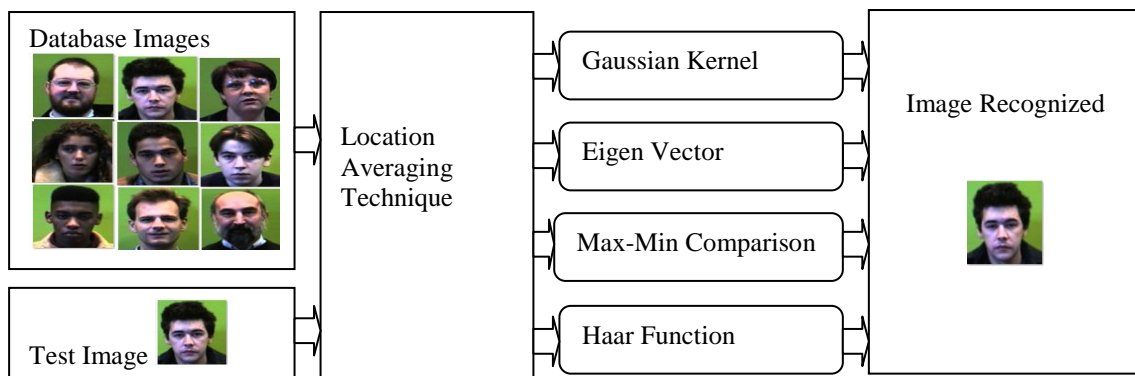

Fig. 1.Block diagram of Location averaged recognition system

A. Location averaging technique

In feature extraction and feature reduction fields Location averaging technique [10] is the novel idea and it is briefly explained in this work. It is the process of averaging image's row elements using column size. Each cell value (Intensity value) of an image is multiplied with its location (position) value and it is divided by its column size (X_{kt}). It is called as Location averaged value of the Cell (X). This process is same for all the Cell values in an image.

$$X_{kt} = (C_{kt}*L_{kt})/C_s$$

After all the cells are transformed in to Location averaged value of the Cells, then a row wise averaging process is done, i.e., addition of all the Location averaged value of the Cells in a row, and it is divide by the column size (Y_k). It is called as Location averaged value of the Row (Y), First row values are reduced in to a Location averaged value of the First Row (Y_1), similarly for all rows. So finally an image matrix is reduced in to a column vector.

$$Y_k = (X_{11}+X_{12}+X_{13}+...+X_{1N})/C_s \quad ; \quad \text{Here k=1;}$$

1	2	3	4	5

21	20	29	15	16
13	12	15	17	18
12	14	17	19	11
14	15	16	17	19
23	24	26	27	22

Fig. 2 (a) Model image's matrix

$$\text{Location averaged value for 'N' Rows} = \begin{bmatrix} Y_1 \\ ... \\ ... \\ ... \\ Y_N \end{bmatrix}$$

C – Cell value (Intensity value)
L – Cell Location value
C_s – Column Size
k – Row; t – Column
N – Last element
X – Location averaged value of the Cell
Y – Location averaged value of the Row
Y_1 –Location averaged value of the First Row
Y_N – Location averaged value of the 'Nth' Row

The Location averaging reduces the whole size of an image in to its column size (column vector). It is one of the best feature extraction/ feature reduction method, for object recognition. The basic working model of Location averaging is shown in Fig. 2. (a) represents a model image's matrix and (b) represents the Location averaged value for the model image's matrix.

Example:

The total number of elements in an image is 36,000 (image size 180x200) and it is reduced to its column size of 200 elements, a single vector, due to Location averaging technique.
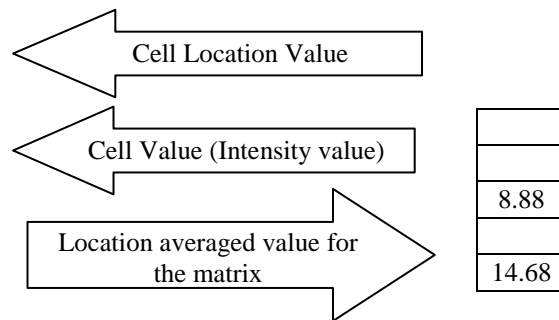
Cell Location Value

Cell Value (Intensity value)

Location averaged value for the matrix

8.88
14.68

Fig. 2 (b) Location averaged value (vector) for the model image's matrix

Fig. 2.Location averaging technique

Notes:

Location averaged value is calculated using the cell's location and intensity value, here both these values are bounded together and divided by the column size. So there is a unique set of values for all images, it provides easy differentiation between the images and leads to greater accuracy. It follows the basic averaging principle; the main aim is size reduction. All the columns in an image are reduced in to a single column i.e., an image can be reduced into a column vector. It is a way of representing/reducing an image, and its outputs are data.

B. *Location averaged Database images and Test image:*

This Location averaging technique is applied to the set of database images and the test image (Fig. 1.) to get Location averaged database images (Ld) and Location averaged test image (Lt).

C. *Location averaged Gaussian kernel*

The Gaussian process [1-3] is combined with the location averaging process (Fig. 3) for face recognition [4]. The location averaged values of the test image and database images are given as the input to the Gaussian process.

In Gaussian method 'difference of value' is found between the Location averaged test image and Location averaged database images, and its square value is divided by twice the square value of the total number of images. Finally the exponential function is derived for above values. It is represented in below equation.

$$G = \exp(-(x-a)^2 / 2b^2)$$

x – Location averaged database images (Ld)

a – Location averaged test image (Lt)

b – Total Number of images in the database

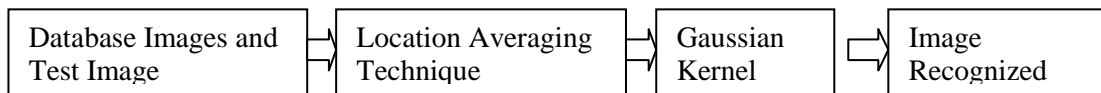Database Images and Test Image → Location Averaging Technique → Gaussian Kernel → Image Recognized

Fig. 3.Location averaged Gaussian kernel method

After deriving the exponential value results, it is to find the maximum value column; it denotes the highest matching column value of test image with database image. The position of the maximum value column indicates the matched image in the database. Output of Location averaged Gaussian kernel is shown is shown in Fig. 4.
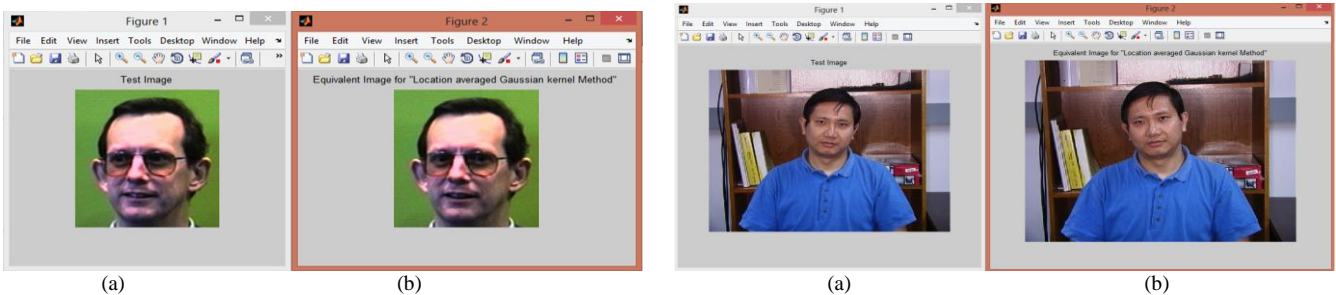

(a)   (b)   (a)   (b)

Fig. 4.Location averaged Gaussian kernel method output for face images; (a) Test image; (b) Database image (Matched image)

D. *Location averaged Eigen vector*

Eigen method [7, 8] is widely practised useful method for recognition [9]. In this paper the results from the Location averaging technique is used by the Eigen method (Fig. 5). The Location averaging values for database images (Ld) are multiplied by its transpose to get the square matrix (S). Eigen values and Eigen vectors are calculated for this square matrix. The Eigen vectors are used to obtain the Eigen faces, and then the projected matrix is produced by multiplying transpose of Eigen faces with Ld.

Database Images and Test Image → Location Averaging Technique → Eigen Vector → Image Recognized

Fig.5.Location averaged Eigen vector method

$$S = (Ld)^T(Ld)$$

Ld – Location averaged database images; Lt – Location averaged test image

Projected Matrix = Eigen faces ^{T} x Ld

Location averaging value for a test image (Lt) is also calculated and the projected test image is formed by multiplying transpose of Eigen faces with Lt.
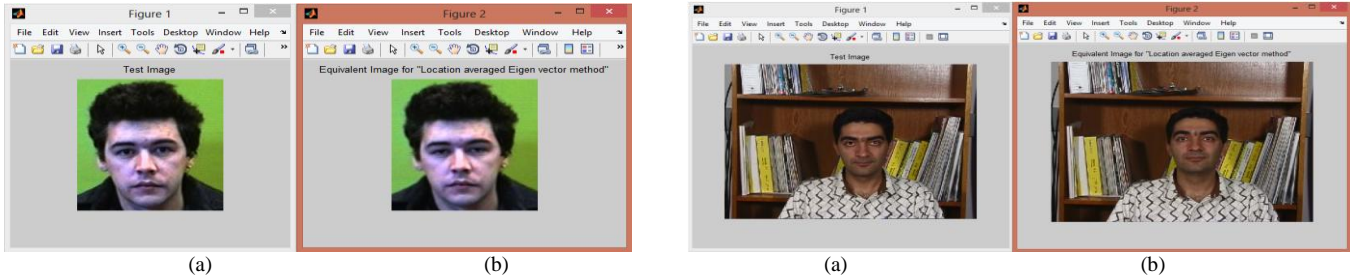
Projected Test Image = Eigen faces ^{T} x Lt.

Fig. 6.Location averaged Eigen vector method output for face images; (a) Test image; (b) Database image (Matched image)

The normalization process is done between the projected matrix and the projected test image, the minimum difference represents the matched image. The minimum difference shows the maximum resemblance of matching an image in the database with the test image, and the recognized image is obtained. Output of Location averaged Eigen vector is shown in Fig. 6.

E. Location averaged Max-Min comparison

In this system the Location averaging technique is combined with the new Max-Min comparison process. The Max-Min comparison operation is done between the Location averaged test image (Lt) and the Location averaged database images (Ld).

It is to find the maximum intensity and minimum intensity values between the Location averaged database images and Location averaged test image, for these values a comparison process is done to find the recognized image, shown in Fig. 7, Fig. 8. In this process,
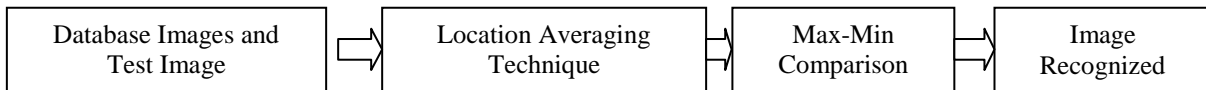
all the 2D images in database are converted in to Location averaged 1D vectors, with these 1D vectors a 2D matrix is (Fig. 10) formed, in this matrix (Location averaged database's Matrix) 1st column shows the first image, 2nd column shows the second image and so on. A 2D test image is also converted in to Location averaged test image 1D vector (Fig: 9).

This Location averaged test image is compared with all Location averaged database images, i.e., Comparison between the Location averaged test image's 1D vector and Location averaged database's Matrix, it is a Row by Row comparison. While comparing the rows between the Location averaged test image's 1D vector and the Location averaged database's Matrix, the maximum intensity values, are detected. The detected maximum intensity values are collected in a matrix called Location averaged Maximum Matrix (LMaxM) (Fig. 11).

Fig 7.Location averaged Max-Min Comparison method

Fig 8.Location averaged Max-Min Comparison block diagram

15				

12	13	19	16	11
13	14	12	14	17
15	16	18	13	12
11	19	15	17	16
17	20	12	11	18

Fig 9.Location averaged test Image 1D Vector

Fig 10.Location averaged Database Images 1D Vector
(Location averaged Database's Matrix)

(Model - Not actual images matrix)

19	19	19	19	19
13	14	12	14	17
15	16	18	13	12
13	19	15	17	16
17	20	15	15	18

12	13	19	16	11
11	11	11	11	11
09	09	09	09	09
11	13	13	13	13
15	15	12	11	15

Fig 11.Location averaged Maximum Matrix (LMaxM)

Fig 12.Location averaged Minimum Matrix (LMinM)

In Location averaged Maximum matrix the first column represents the maximum intensity value collected between the test image and the first image of the database images, and so on up the last column. Next step is to detect the minimum intensity values, from the Location averaged database images and the Location averaged test image. The procedure is same as like as maximum intensity value detection. The above processes are proceeded with an important change, (i.e.) instead of extracting maximum intensity values, here it is to extract the minimum intensity values. The resultant matrix from this step is called as Location averaged Minimum matrix (LMinM) (Fig. 12).

7	6	0	3	8
2	3	1	3	6
6	7	9	4	3
2	6	2	4	3
2	5	3	4	3

Euclidean distance

19	27	15	18	23

Row wise addition

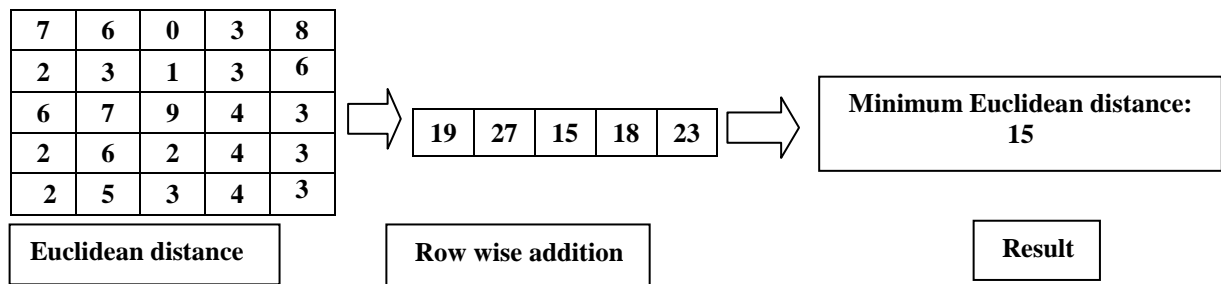**Minimum Euclidean distance:
15**

Result

Fig 13.Calculation of Minimum Euclidean Distance value, between the Maximum and Minimum Matrices

Finally with the Euclidean distance phenomenon, (Fig. 13) the shortest Euclidean distance is found between the Location averaged Maximum Matrix (LMaxM) and Location averaged Minimum Matrix (LMinM). This shortest distance represents the minimum variation among the images and it shows the correct database image that is matched with the given test image (Fig. 14). Euclidean distance for Max Min Comparison = $\sqrt{[(LMaxM-LMinM)^2]}$.
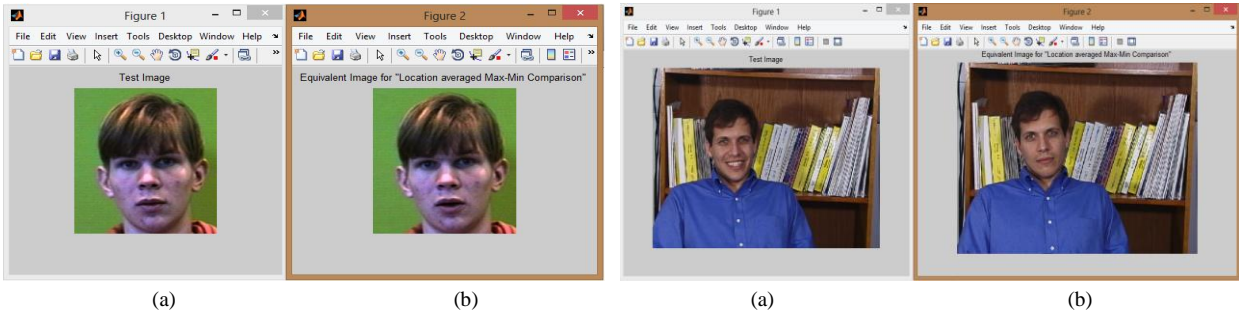
<div align="center">

(a)   (b)

</div>

Fig 14.Location averaged Max-Min comparison method output for face images; (a) Test Image (b) Matched Image (Database Image)

F.   Location averaged Haar function

Decomposition of a discrete signal in to two signals takes place in haar transform [11, 12], the decomposed two signals are called as sub signals, and its length is half of its original signal. One sub signal is called as running average and another is running difference.

The location averaged database images results are combined with haar wavelet (Fig.15) [13, 14] for recognition. The overall database image's location averaged values are considered as a discrete signal and it is decomposed in to two sub signals called averaging and difference.
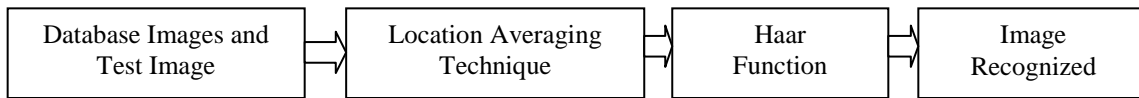

Fig. 15.Location averaged Harr function method

The Location averaging values for database images (Ld) are multiplied by its transpose to get the square matrix (x). If an image size is 120x120 (for example), after the haar transform decomposition two sub matrix (sub signals) are formed, the average matrix size is 120x60 and the difference matrix size is 120x60.

$$x = (Ld)^{T}(Ld)$$

x - Square matrix (Location averaged database images)

$$\text{Average matrix} = \phi (x) = \phi (2x) + \phi (2x-1)$$

$$\text{Difference matrix} = \Psi(x) = \Psi (2x) - \Psi (2x-1)$$

Next the average and difference matrix are combined together to get the Haar values. The Haar projected matrix is produced by multiplying transpose of Haar values with Ld.

$$\text{Haar Projected Matrix = Harr values}^{T} \text{ x Ld}$$

Location averaging value for a test image (Lt) is also calculated and then the haar projected test image is formed by multiplying transpose of Haar values with Lt.

$$\text{Haar Projected Test Image = Haar values}^{T} \text{ x Lt.}$$

The normalization process is done between the haar projected matrix and the haar projected test image, the minimum difference represents the matched image. Output of Location averaged Haar function is shown in Fig. 16.
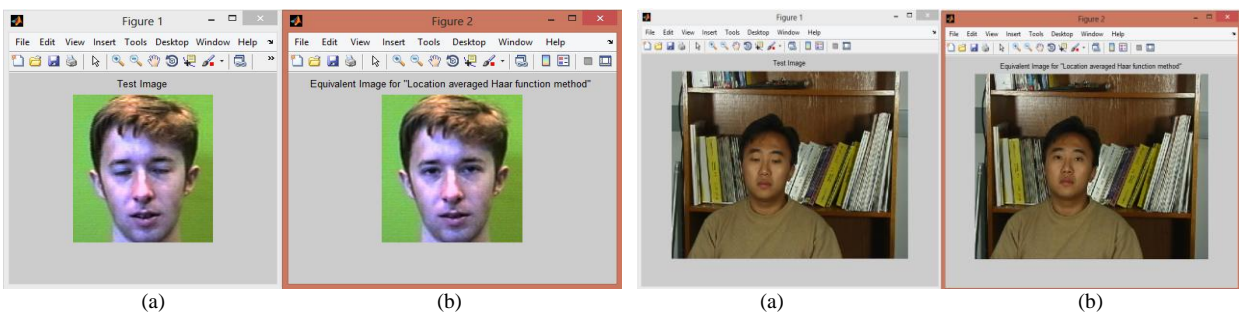

<div align="center">

(a)   (b)

</div>

Fig. 16.Location averaged Haar function method output for face image; (a) Test image; (b) Database image (Matched image)

G. Results and Discussions

The above recognition methods are concentrated on each and every intensity value of an image, so small variations in between the group of images can be easily identified, and are very much useful to identify the pose varied face images. The above four recognition methods called, Location averaged Gaussian kernel, Location averaged Eigen vector, Location averaged Max-Min comparison and Location averaged Harr function are tested on 120 subjects of Libor Spacek's Facial Images Database (Image size: 180x200) and 50 subjects of Georgia tech face database (Image size: 640x480) and its results are tabulated as Table 1 and Table 2.

Table 1, shows that the Location averaged Gaussian kernel method and Location averaged Max-Min comparison method yields minimum mismatching of 0.8% and 1% in minimum runtime and its accuracies are about 99%. Location averaged Eigen vector and Location averaged Haar function produced accuracy of about 93.3% and 85.0%. It is observed that Location averaged Harr function method yields higher mismatching of 15%. The run time variations for all the three methods are between 1.35 to 1.60 s.

Table 2, shows that the Location averaged Gaussian kernel method and Location averaged Max-Min comparison method yields minimum mismatching of 2% and its accuracies are about 98%. Higher mismatching of 8% is from Location averaged Harr function method. The run time variations for all the three methods are between 2.0 to 2.67 s. The Runtime of the recognition performance results in Table 2 is higher than the Table 1 because the image (image size) used in Table 2 is bigger than in Table 1. So there is a small hike in the runtime.

.

Table 1.Face recognition results for Libor Spacek's facial images database

S.N	Proposed Recognition Techniques (Tested on 120 subjects)	Runtime (Sec)	Mismatching (%)	Recognition Performance (%)
1	Location averaged Harr function method	1.50-1.59 s	15.0%	85.0%
2	Location averaged Eigen vector method	1.53-1.59 s	6.7%	93.3%
3	Location averaged Max-Min comparison method	1.35-1.39 s	1.0%	99.0%
4	Location averaged Gaussian kernel method	1.49-1.60 s	0.8%	99.0%

Table 2.Face recognition results for Georgia tech face database

S.N	Proposed Recognition Techniques (Tested on 50 subjects)	Runtime (Sec)	Mismatching (%)	Recognition Performance (%)
1	Location averaged Harr function method	2.58-2.63 s	8%	92.0%
2	Location averaged Eigen vector method	2.0-2.67 s	6%	94.0%
3	Location averaged Max-Min comparison method	2.50-2.57 s	2%	98.0%
4	Location averaged Gaussian kernel method	2.53-2.60 s	2%	98.0%

The Recognition performance results from Table 1 and Table 2 are shown as, Recognition performance comparison plots in Fig. 17.and Fig. 18. The Runtime results from Table 1 and Table 2 are shown as Runtime comparison plots in Fig. 19.and Fig. 20. The Mis-Matching results from Table 1 and Table 2 are shown as Mis-Matching Comparison plots in Fig. 21 and Fig. 22
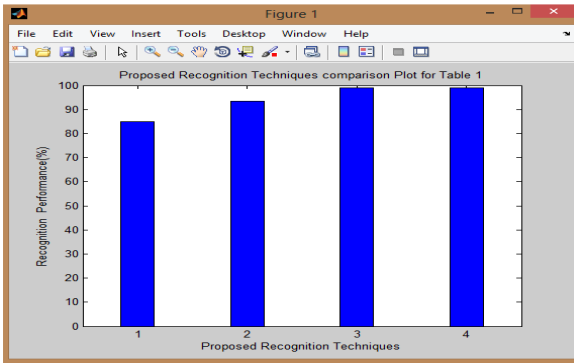
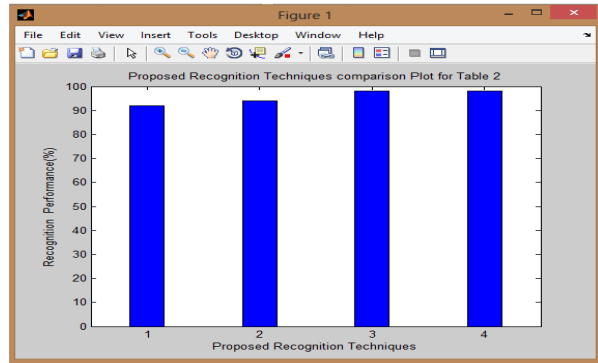Fig. 17.Recognition performance comparison plot for Table 1

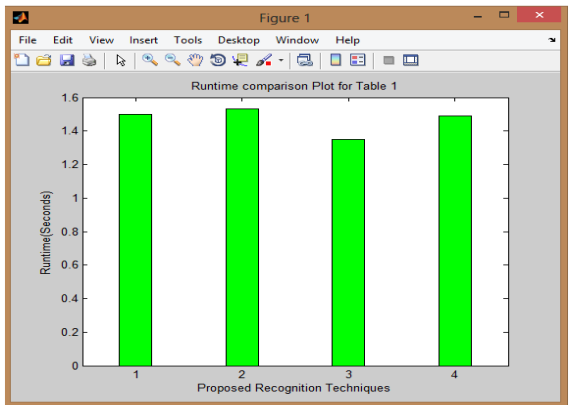Fig. 18.Recognition performance comparison plot for Table 2
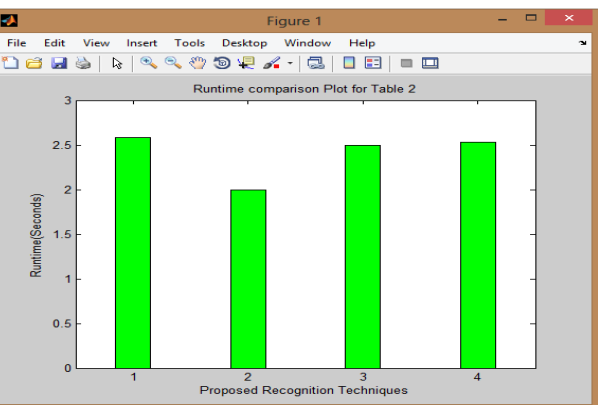

Fig. 19.Runtime comparison plot for Table 1
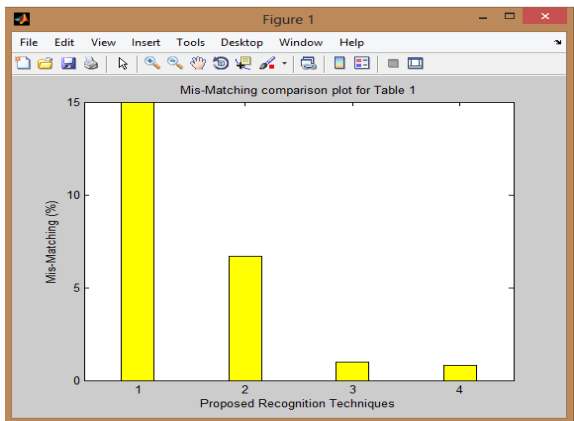

Fig. 20.Runtime comparison plot for Table 2

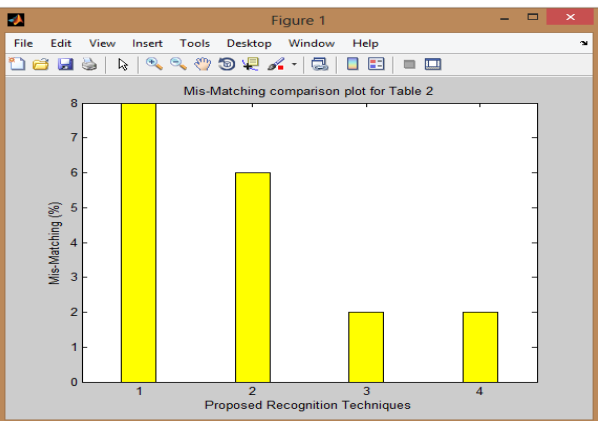Fig. 21.Mis-Matching comparison plot for Table 1

Fig. 22.Mis-Matching comparison plot for Table 2

II.   COMPARISONS

In our previous work, the Location averaging technique [10] was explained and compared with few other methods, in this work it was combined with new Max-Min comparison, Gaussian kernel, Eigen vector, Haar function techniques. The comparison results are shown in Table 3. The location averaging technique alone yields runtime variation of about 1.32-1.42 s for 100 subjects with mismatching of 1%, the Location averaged Gaussian kernel, Location averaged Max-Min comparison, yields runtime variation of about 1.49-1.60 s and 1.35-1.39 s for 120 subjects with mismatching of 0.8% and 1.0%. It shows that, in this proposed work the subject's quantity was increased and the Performance of the system is still 99%, i.e., its Recognition performance was high even though the numbers of subjects are increased. Therefore the Gaussian kernel, Max-Min comparison combined with Location averaging technique

yields a perfect system for biometric object recognition. The proposed methods Location averaged Max-Min comparison, Location averaged Gaussian kernel are compared with the few existing techniques called PCA, Direct LDA, LDA, and ICA are tabulated as Table 4.

Table 3. Comparison table between our previous work and proposed work

S.N	Reference	Feature	Subject	Runtime (Sec)	Mismatching (%)	Recognition Performance (%)
1	Parivazhagan.A (10)	Location averaging	100	1.32-1.42 s	1%	99%
2	Proposed Method	Location averaged Max-Min comparison	120	1.35-1.39 s	1%	99%
3	Proposed Method	Location averaged Gaussian kernel	120	1.49-1.60 s	0.8%	99%

Table 4. Comparison table of various recognition techniques

S.N	Reference	Feature	Image Size	Recognition Performance (%)
1	Martinez and Kak (17)	PCA	85x60	70.0%
2	Yu and Yang (18)	Direct LDA	112x92	90.8%
3	Martinez and Kak (17)	LDA	85x60	88.0%
4	Bartlett Marian et al (19)	ICA	60x50	89.0%
5	Parivazhagan.A (10)	Location averaging	180X200	99.0%
6	Proposed Method	Location averaged Max-Min comparison	180x200	99.0%
7	Proposed Method	Location averaged Gaussian kernel	180x200	99.0%

III.   CONCLUSIONS

In this proposed work four novel recognition methods are presented to identify a person. From the results it is concluded that the Location averaging technique can act as a good feature extraction/reduction process. It can act as an effective pair with Gaussian kernel, Eigen vector, Max-Min comparison and Haar function for facial recognition system. All the four recognition methods produce results in shortest runtime with good recognition performance, for pose varied face images.   Location averaged Gaussian kernel and Location averaged Max-Min comparison methods tops the recognition performance list with recognition value of about 99%.

REFERENCES

[1]. David J.C.Mackay, Introduction to Gaussian Processes,1998.

[2]. Maria Isabel Ribeiro, Gaussian Probability Density Functions: Properties and Error Characterization, M. Isabel Ribeiro, 2004.

[3]. Gaussian function, Wikipedia, https://en.wikipedia.org/wiki/Gaussian_function

[4]. Ognjen Rudovic, Maja Pantic, Ioannis (Yiannis) Patras, Coupled Gaussian Processes for Pose-Invariant Facial Expression Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 6, June 2013, 1357-1369.

[5]. Zahid Mahmood, Tauseef Ali, Samee U. Khan, Effects of pose and image resolution on automatic face recognition, IET Biometrics, 2016, Vol. 5, Issue 2, pp. 111–119.

[6]. Haifeng Hu, Illumination invariant face recognition based on dual-tree complex wavelet transform, IET Computer Vision, 2015, Vol. 9, Issue 2, pp. 163–173.

[7]. Matthew Turk, Alex Pentland, Eigenfaces for Recognition, Journal of Cognitive Neuroscience, vol. 3, no. 1, 1991, 71-86.

[8]. Lindsay I Smith, A tutorial on Principal Components Analysis, February 26, 2002

[9]. N. Rajagopalan, Rama Chellappa, Nathan T. Koterba, Background Learning for Robust Face Recognition With PCA in the Presence of Clutter, IEEE Transactions on Image Processing, Vol. 14, No. 6, June 2005, 832-843.

[10]. Parivazhagan.A and BrithaTherese.A, A Novel, Location averaging, Linear equation and Exponential function techniques for face recognition in Human identification system, International Journal of Applied Engineering Research (IJAER), Special Issue, Volume 10, Number 87, 2015, pp.21-26.

[11]. Brani Vidakovic, Peter Mueller, Wavelets for Kids a Tutorial Introduction, Institute of Statistics and Decision Sciences, DukeUniversity,1991.

[12]. Albert Boggess, Francis J Narcowich, A First Course in Wavelets with Fourier Analysis, Pearson, 2001.

[13]. Piotr Porwik, Agnieszka Lisowska, The Haar–Wavelet Transform in Digital Image Processing: Its Status and Achievements, Machine Graphics & Vision Vol . 13, No 1/2, 2004, 79-98.

[14]. Yujian Li, Houjun Li, and Zhi Cai, Fast Orthogonal Haar Transform Pattern Matching via Image Square Sum, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 36, No. 9, September 2014, 1748-1760.

[15]. Mohammad Reza Faraji, Xiaojun Qi, Face recognition under varying illumination based on adaptive homomorphic eight local directional patterns, IET Computer Vision, 2015, Vol. 9, Issue 3, pp. 390–399.

[16]. S L Happy, Aurobinda Routray, Automatic Facial Expression Recognition Using Features of Salient Facial Patches, IEEE Transactions on affective computing, vol. 6, No. 1, January-March 2015.

[17]. Martinez, A.M., Kak, A.C, PCA versus LDA. IEEE Trans. Patter Anal. Machine Intell. 23 (2), 228 – 233, 2001.

[18]. Yu, H., Yang, J, A direct LDA algorithm for high-dimensional data with application to face recognition. Pattern Recognition 34 (October), 2067–2070, 2001.

[19]. Bartlett Marian, Stewart, Movellan Javier, R., Sejnowski Terrence, J, Face recognition by independent component analysis. IEEE Trans. Neural Networks 13 (6), 1450–1464, 2002

.