An Accurate System for Face Detection and Recognition

Abdelmgeid A. Ali[1], Tarek Abd El-Hafeez[1,2*] and Yosra Khalaf Mohany[1]

[1]*Department of Computer Science, Faculty of Science, Minia University, EL-Minia, Egypt.*
[2]*Computer Science Unit, Deraya University, EL-Minia, Egypt.*

Authors' contributions

This work was carried out in collaboration among all authors. Author AAA designed the study, performed the statistical analysis and wrote the protocol. Authors TAEH and YKM managed the analyses of the study, managed the literature searches and wrote the first draft of the manuscript. All authors read and approved the final manuscript.

Original Research Article

Abstract

During the last few years, Local Binary Patterns (LBP) has aroused increasing interest in image processing and computer vision. LBP was originally proposed for texture analysis, and has proved a simple yet powerful approach to describe local structures. It has been extensively exploited in many applications, for instance, face image analysis, image and video retrieval, environment modeling, visual inspection, motion analysis, biomedical and aerial image analysis, remote sensing. Face recognition is an interesting and challenging problem, and impacts important applications in many areas such as identification for law enforcement, authentication for banking and security system access, and personal identification among others. In this paper we are concerned with face recognition in a video stream using Local Binary Pattern histogram with processed data. First we will detect faces by using a combination of Haar cascade files that uses skin detection, eye detection and nose detection as input of LBP to increase the accuracy of the proposed recognition system. Also, our system can be used to build a dataset of faces and names to be used in a recognition step. The experimental results have shown that the proposed system can achieve accuracy of recognition up to 96.5% which was better than the relevant methods.

*Corresponding author: E-mail: tarek@mu.edu.eg, tarek@deraya.edu.eg, tarek_1_2@yahoo.com;

1 Introduction

The 21st century is a modern and scientific era, which have great achieves .Nowadays computer technology has been an essential part of life. Computers are being used in pyramids of applications, which range from simple to complex problem solving methods. at this time face recognition technology has appeared as useful tool to recognize features of faces through their inherent traits. And it has been one of the most researched areas in the field of pattern recognition and computer vision. However, because of its wide use in multitude of applications like in biometrics, information security, law enforcement access control, surveillance system and smart cards. But it has many challenges for researcher that needs to be addressed. Face an object depends on facial expressions, which constitute meaningful features. For instance, pose invariance, illuminations and aging which are potential areas that require further investigation over previous work. The result of previous researches reveals that facial expressions are changing with respect to aging; therefore, they could not be permanently modeled in face recognition. There is many ways which concern in detection techniques, approaches, viz. Eigen face, Artificial Neural Networks (ANN), Support Vector Machines (SVM), Principal Component Analysis (PCA), Independent Component Analysis (ICA), Gabor Wavelets, Elastic Bunch Graph Matching, 3D morphable Model and Hidden Markov Models. We will try to show some of them in this paper [1].

Authentication is an important issue in system control in computer based communication. Human face recognition is a significant branch of biometric verification and has been used widely in many applications, such as video monitor system, human-computer interaction, and door control system and network security [1]. In this paper we are concerned with objects motion so we try to modify face recognition technique to recognize faces in a video stream using Local Binary Pattern histogram with processed data. First, we will detect faces by using a combination of Haar cascade files that uses skin detection, eye detection and nose detection as input of LBP to increase the accuracy of the proposed recognition system.

Biometric recognition system can be built through various techniques. However, most commonly used are fingertips and iris methods. These require individual's participation or involvement to access the system. Moreover, latest systems provide participant access without its intervention. Among such methods face recognition is one of the most viable technique in which an individual can easily be captured and monitored through the system. Face recognition databases vary with controllable photos to uncontrollable videos, where LFT is used for controllable photos and YTM is used for videos. Face recognition system comprises of three main modules: 1) pre-processing, 2) feature selection, and 3) Classification [2].

Motion detection is the process to detect a change in the position of an object comparative to its surroundings also or a variation in the environment which surround to an object. By mechanical or electronic methods motion detection can be achieved. Motion detection is done by natural organisms, we named it motion perception [2].

One of algorithm which used to detect the motion in image processing is motion segmentation algorithm for the detection of objects which are moving in image sequences needed from a moving platform which contains two steps. First, estimate the image plane transformation induced by the moving platform using a sub pixel accuracy image registration algorithm [3]. As known that the registration algorithm is fully automatic and performs well under significant camera rotation and translation. The input images are then transformed into a common coordinate system (for example, the coordinate system of the first image). Then, segments the changed regions from the camera motion-compensated frame difference. Finally, the moving object is detected from the closure of changing segments, and the object motion parameters are estimated [4].

In this paper, we are concerned with face recognition in a video stream. The structure of the paper is as follows: Section 2 gives a brief overview of tow face recognition methods which are Eigenfaces and LBPH.

In Section 3, the Proposed Methodology is introduced. Section 4 presents the experimental results. We presented the conclusion and future work in section 5 and 6.

2 Face Recognition

Face Detection has the target of finding the faces (location and size) in an image and may extract them to be used by the face recognition algorithm. But Face Recognition with the facial images already extracted, cropped, resized and usually converted to grayscale, the face recognition algorithm is responsible for finding characteristics which best describe the image [5].

There is clear difference between Face Detection VS Face Recognition. Face Detection: it has the target of finding the faces (location and size) in an image and may extract them to be used by the face recognition algorithm. But Face Recognition: with the facial images already extracted, cropped, resized and usually converted to grayscale, the face recognition algorithm is responsible for finding characteristics which best describe the image [5].

A facial recognition system is a technology able to identify or verify a person from a digital image or a video frame from a video source. There are multiple methods in which facial recognition systems work, but in general, all of them work by comparing selected facial features from given image with faces within a dataset or database. It is also described as a Biometric Artificial Intelligence based application which can uniquely identify a person by analyzing patterns based on the person's facial textures and shape [6].

Face recognition is a process which humans routinely perform and effortlessly in their daily lives. Wide availability of powerful and low-cost desktop and embedded computing systems has created a huge interest in automatic processing of digital images in a variety of applications, like biometric authentication, human-computer interaction, surveillance, and multimedia management. Research and development in automatic face recognition naturally follows [7].

In face recognition system, the input is normally a digital image or video stream, and the output is an identification or verification of the person that appear in the image or video stream [7]. Several approaches define a face recognition system process as three steps as shown in Fig. 1.
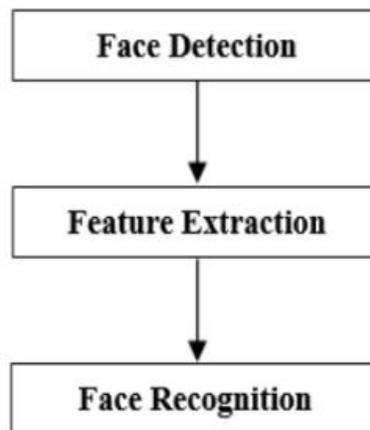

Fig. 1. Face recognition process [7]

Two major methods are used for facial features extraction, which can be classified into appearance-based and Model-based methods.

- **Appearance-Based**

Appearance-Based methods try to recognize faces using global representations which based on the whole image rather than local facial features. Many methods for object recognition and computer graphics are directly based on images without intermediate 3 dimensional models, most of them depends on image representation which induce a vector space structure and requires dense correspondence in principle [8].

Global facial information is represented basically by a small number of features which are derived from the pixel information of face images, this small number of features distinctly capture the variance between different individual faces and are used also to identify unique individuals. In appearance-based method, the whole face region is considered as an input for face detection system to perform face recognition [9].

- **Model-Based Methods**

Model-based face recognition methods help to construct a model of the human face that captures facial variations. Prior knowledge of the human face is highly used to design the model (11). For example, model-based matching derives the distance and relative position features from the placement of internal facial elements. It can be made invariant to size, orientation, and lighting. The other advantages of these schemes are the compactness of the representation of face images and rapid matching. Three different extraction methods are distinguished (generic methods based on edges, lines, and curves; feature template-based methods, and structural matching methods that consider geometrical feature constraints).

While initially a form of computer application, in recent times it has seen wider uses on mobile platforms and in other forms of technology, such as robotics. It is typically used as access control in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems. Although the accuracy of facial recognition system as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless and non-invasive process. Recently, it has also become popular as a commercial identification and marketing tool. Other applications include advanced human-computer interaction, video surveillance, automatic indexing of images, and video database, among others. And over the last ten years or so, face recognition has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding [10].

Face Recognition systems use computer algorithms to pick out specific, distinctive details about a person's face. These details, such as distance between the eyes or shape of the chin.

In this paper we will use our method that detects faces based on a group of identifiers which were carefully examined and were explained in more details in [11].

There are many algorithms for face recognition, we concentrated on two algorithms –OpenCV and LBPH- and we will modify the second one to get more accurate results [9].

Analyzing different results drawn from the researchers the accuracy ratio has been much improved in recent years as compared previous results. It is expected to have an effective and efficient output in upcoming years.

A comparative study of the analysis results obtained by different researchers by applying face recognition techniques on the basis of Eigen Faces is given in Table 1 [12].

OpenCV(Open Source Computer Vision Library), which is an image and video processing library with bindings in C++, C, Python, and Java. OpenCV is used for all sorts of image and video analysis, like facial recognition and detection, license plate reading, photo editing, advanced robotic vision, optical character recognition, and a whole lot more [10].

Table 1. Comparative study of face recognition techniques based on PCA

S#	Year	Database	Technique	Accuracy	Reference
1	2012	ORL Faces	PCA	70.0 %	Slvkovic et al.
2	2012	Face94	PCA	100.0 %	Abdullah et al.
3	2013	FRAV Face DB	Eigen Face	96.0 %	Saha, Rajib et al.
4	2014	-	PCA Eigen Faces	70.0%	Rahman Armanadurni Abd, et al
5	2014	Yale Database	PCA	92 % to 93 %	MuzammilAbdulrahman et al.
6	2014	AT & T	PCA		Johannes Reschke et al.
7	2016	Computer Vision Research Projects dataset	PCA	93.6%	Md.Al-Amin Bhuiyan
8	2017	EmguCV Library	PCA +RMF	93.0%	Jacky Efendi et al.
9	2017	Yale Database	PCA	98.18%	Riddhi A & S.M. Shah

OpenCV has three built-in face recognizers and thanks to its clean coding, we can use any of them just by changing a single line of code. Here are the names of those face recognizers and their OpenCV calls:

- EigenFaces—cv2.face.createEigenFaceRecognizer()
- FisherFaces—cv2.face.createFisherFaceRecognizer()
- Local Binary Patterns Histograms (LBPH) cv2.face.createLBPHFaceRecognizer() [11]

Face recognition using Eigenfaces

The word eigenface coined by German ―Eigen wert‖ The ―Eigen‖ literally mean characteristic and ―wert‖ mean value. Eigenface is well establish algorithms that was used to recognize a feature in a face image. It is based on Principle Component Analysis (PCA). In this method the basic concept is to recognize the face by taking its unique information about the face in question. Then encode it to compare with the decode result of previously taken image. In eigenface method, decoding is performed with the calculation of eigenvector and then it is represented as a matrix. However, Eigenface based face recognition systems is only suitable for images having the frontal faces but some researches identify a face with different poses have also been made.

Eigenfaces is the name given to a group of eigenvectors when they are used in the computer vision problem of human face recognition. Sirovich and Kirby developed the approach of using eigenfaces for recognition in (1987) and this system used by Matthew Turk and Alex Pentland in face classification. The eigenvectors are derived from the covariance matrix of the probability distribution over the high-dimensional vector space of face images. The eigenfaces themselves form a basis set of all images used to construct the covariance matrix. This produces dimension reduction by allowing the smaller set of basis images to represent the original training images. We could achieve a good Classification by comparing how faces are represented by the basis set [12].

The problem with the image representation we are given is its high dimensionality. For Example Two-dimensional $p \times q$ gray scale images span a $m = pq$ -dimensional vector space, so an image with 100×100 pixels lies in a $10,000$-dimensional image space already. The question is: Are all dimensions equally useful for us? We can only make a decision if there's any variance in data, so what we are looking for are the components that account for most of the information. The Principal Component Analysis (PCA) was independently proposed by Karl Pearson (1901) and Harold Hotelling (1933) to turn a set of possibly correlated variables into a smaller set of uncorrelated variables. The idea is, that a high-dimensional dataset is often described by correlated variables and therefore only a few meaningful dimensions account for most of the information. The PCA method finds the directions with the greatest variance in the data, called principal components [13].

2.1 Algorithm description

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a random vector with observations $x_i \in \mathbb{R}^d$.

- Compute the mean μ

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

- Compute the Covariance Matrix S

$$S = \frac{1}{n} \sum_{i-1}^{n} (x_i - \mu)(x_i - \mu)^{T}$$

- Compute the eigenvalues λ_i and eigenvectors v_i of S

$$S v_i = \lambda_i v_i, i = 1, 2, \ldots, n$$

- Order the eigenvectors descending by their eigenvalue. The k principal components are the eigenvectors corresponding to the k largest eigenvalues.

The k principal components of the observed vector x are then given by:

$$y = W^T(x - \mu)$$

where $W = (v_1, v_2, \ldots, v_k)$

The reconstruction from the PCA basis is given by:

$$x = Wy + \mu$$

where $W = (v_1, v_2, \ldots, v_k)$.

The Eigenfaces method then performs face recognition by:

- Projecting all training samples into the PCA subspace.
- Projecting the query image into the PCA subspace.
- Finding the nearest neighbor between the projected training images and the projected query image.

But still there's one problem left to solve. Imagine we are given 400 images sized 100×100 pixel. The Principal Component Analysis solves the covariance matrix $S = XX^T$, where $size(X) - 10000 \times 400$ in our example. You would end up with a 10000×10000 matrix, roughly 0.8GB. Solving this problem isn't feasible, so we'll need to apply a trick. From your linear algebra lessons you know that a $M \times N$ matrix with $M > N$ can only have $N - 1$ non-zero eigenvalues. So it's possible to take the eigenvalue decomposition $S = X^T X$ of size $N \times N$ instead:

$$X^T X v_i = \lambda_i v i$$

and get the original eigenvectors of $S = XX^T$ with a left multiplication of the data matrix:

$$XX^T(X v_i) = \lambda_i (X v_i)$$

The resulting eigenvectors are orthogonal; to get orthonormal eigenvectors they need to be normalized to unit length [14].

Fig. 2. EigenFace in OpenCV algorithm [14]

2.2 Face recognition by LBPH algorithm

Local binary patterns (LBP) are a kind of visual descriptor which used for classification in computer vision. The first described for (LBP) in 1994 [15]. It has since been found to be a powerful feature for texture classification; it has been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets [16,17].

Although the concept of LBPs were introduced as early as 1994, Local Binary Patterns, or LBPs for short, are a texture descriptor made popular by the work of Ojala et al. in their 2002 paper, Multi-resolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns [18]. LBPH gives less false positive results even after capturing an image from a height of 4 feet to 7 feet for database of training sets [19].

LBPs calculate a local representation of texture (LR). This LR is constructed by comparing each pixel with its surrounding neighborhood of pixels. Unlike Haralick texture features which compute a global representation of texture based on the Gray Level Co-occurrence Matrix [20].

To construct the LBP texture descriptor we should first to convert the image to gray scale. Then, we select a neighborhood of size r surrounding the center pixel for each pixel in the gray scale image. Then calculate a LBP value is for this center pixel and stored in the output 2D array with the same width and height as the input image [21].

The idea isn't to look at the whole image as a high-dimensional vector, but describe only local features of an object. The features were extracted by this way have a low-dimensionality implicitly. But we observed that the image representation we are given doesn't only suffer from illumination variations [22].

Take in consideration things like scale, translation or rotation in images -at least our local description has to be a little strong against these things. The Local Binary Patterns methodology has its roots in 2D texture analysis. The main idea of LBP is to summarize the local structure in an image, it do this by comparing each pixel with its neighborhood. Take a pixel as center and threshold its neighbors against [23]. If the intensity of the center pixel is greater-equal its neighbor, write it down with 1 and the value 0 if not.so we'll end up with a binary number for each pixel, just like 11001111. So with 8 surrounding pixels you'll end up with 2^8 possible combinations, called Local Binary Patterns or sometimes referred to as LBP codes. The first LBP operator described in literature actually used a fixed 3 x 3 neighborhood just like this [24].
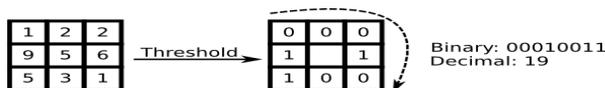
A more formal description of the LBP operator can be given as:

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

, with (x_c, y_c) as central pixel with intensity i_c; and i_n being the intensity of the the neighbor pixel. s is the sign function defined as:

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases}$$

This description enables you to capture very fine grained details in images. In fact the authors were able to compete with state of the art results for texture classification. Soon after the operator was published it was noted, that a fixed neighborhood fails to encode details differing in scale. So the operator was extended to use a variable neighborhood. The idea is to align an arbitrary number of neighbors on a circle with a variable radius, which enables to capture the following neighborhoods [24]:
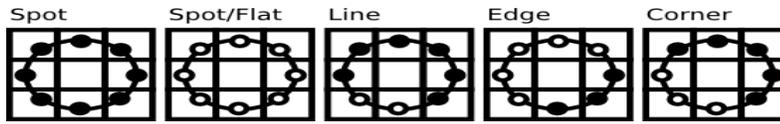
For a given Point (x_c, y_c) the position of the neighbor $(x_p, y_p), p \in P$ can be calculated by:

$$x_p = x_c + R\cos(\frac{2\pi p}{P})$$

$$y_p = y_c - R\sin(\frac{2\pi p}{P})$$

Where R is the radius of the circle and P is the number of sample points [25].

The operator is an extension to the original LBP codes, so it's sometimes called Extended LBP (also referred to as Circular LBP). If a point coordinate on the circle doesn't correspond to image coordinates, the point get's interpolated. Computer science has a bunch of clever interpolation schemes; the OpenCV implementation does a bilinear interpolation:

$$f(x, y) \approx \begin{bmatrix} 1-x & x \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1-y \\ y \end{bmatrix}.$$

By definition the LBP operator is robust against monotonic gray scale transformations and can be easily verify this by looking at the LBP image of an artificially modified image [13].

3 Proposed Methodology

LBPH uses 4 parameters:

- Radius: the circular local binary pattern was built by the radius and that represents the radius around the central pixel. It is often set to 1.
- Neighbors: they are considered as the number of sample points which used to build the circular local binary pattern. We should keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8
- Grid X: they are considered as the number of cells in the horizontal direction. Note that the more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is often has a value 8.

- Grid Y: The number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. Also it is usually set to 8 [26].

Training the Algorithm:

Applying the LBP operation: The creation of an intermediate image which describes the original image in a better way is the first computational step of the LBPH, it was done by highlighting the facial characteristics. The algorithm uses a concept of a sliding window, which based on the parameters radius and neighbors. The following image shows this procedure:
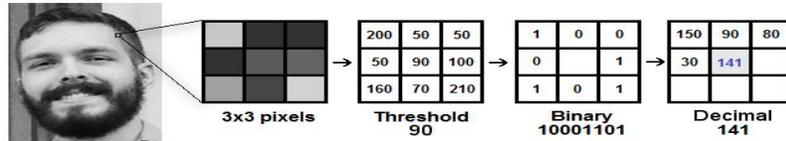

Fig. 3. LBPH example [27]

Based on the previous image, let's slice it into several small steps to can understand it easily:

- Assume we have a facial image in gray scale.
- We can get one part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix which containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- We will use this value to define the new values from the 8 neighbors.
- We set a new binary value for each neighbor of the central value (threshold). We set 1 for values equal or higher than the threshold and set 0 for values lower than the threshold.
- So now, the matrix will only contain binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10101001).
- Then, this binary value will be converted to a decimal value and set it to the central value of the matrix, which actually is a pixel from the original image.
- Finally, we have a new image which represents better the characteristics of the original image.

We must note: The LBP procedure was expanded to use a different number of radius and neighbors, it is called Circular LBP [28].

Extracting the Histograms: Now, by using the image that generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids, as can be seen in the following image:

Fig. 4. Extracting Histogram [27]

Based on the image above, we can extract the histogram of each region as follows:

As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

The LBPH algorithm is pretty much it [27].

Performing the face recognition: Here, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we repeat the steps again for this new image and create a histogram which represents the image.

So to find the image which matches the input image we just need to compare two histograms and return the image with the closest histogram.

We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: Euclidean distance, chi-square, absolute value, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement [28].

Note: here we shouldn't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer. Then we can use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can suppose that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

Our modification is that, when we apply LBPH on data processed which is given by our face detection method in [20], this makes the recognition process is much easier and it isolates the background of the images and focus on the face extracted only which helps the recognizer to work better than before. So we could name the new method "MLBPH" Modified Local Binary Pattern Histogram.

The Proposed MLBPH Algorithm Steps:

- Using webcam/video stream to get image frames.
- Detect Faces of the image frames then draw a square around it.
- Ask user to input names for each face and create a dataset of faces and its names.
- Apply LBP Algorithm.
- Divide the examined window into cells (e.g. 16x16 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 8 neighbors.
- If the center pixel's value is greater than the neighbor's value, write "0" Else, write "1".
- Compute the histogram, over the cell, of the frequency of each "number" occurring
- (i.e., each combination of which pixels are smaller and which are greater than the center).
- This histogram can be seen as a 256-dimensional feature vector.
- Optionally normalize the histogram.
- Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window [29].

Fig. 3 shows a flowchart of Creating Dataset Steps, Fig. 4 shows a Flow Chart for face Detection System which is represented in our paper [11], and Fig. 5 shows a Face recognition System Flow Chart.
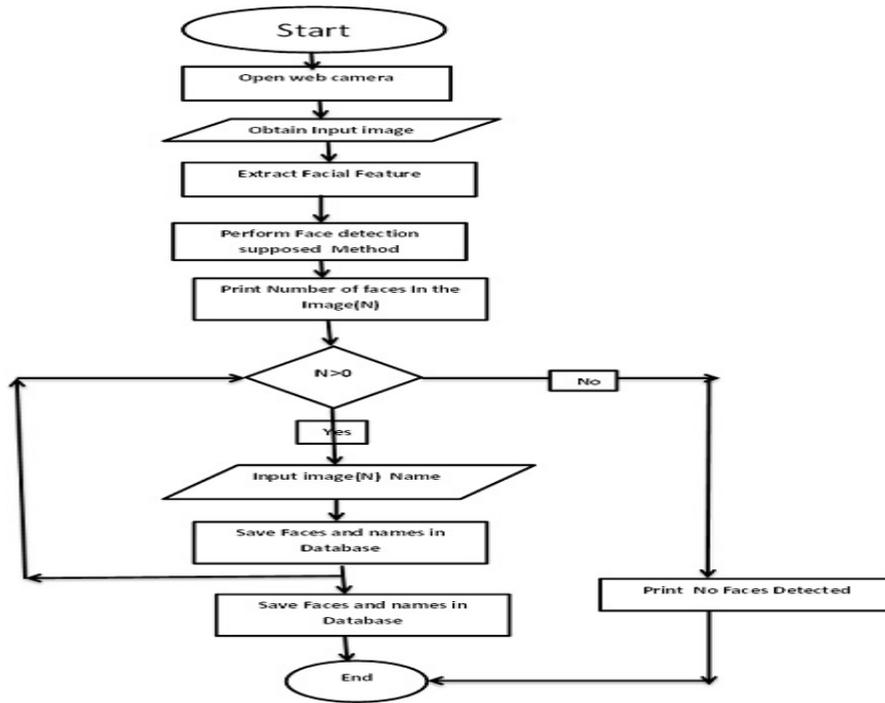
**Fig. 5. Creating dataset steps flowchart**
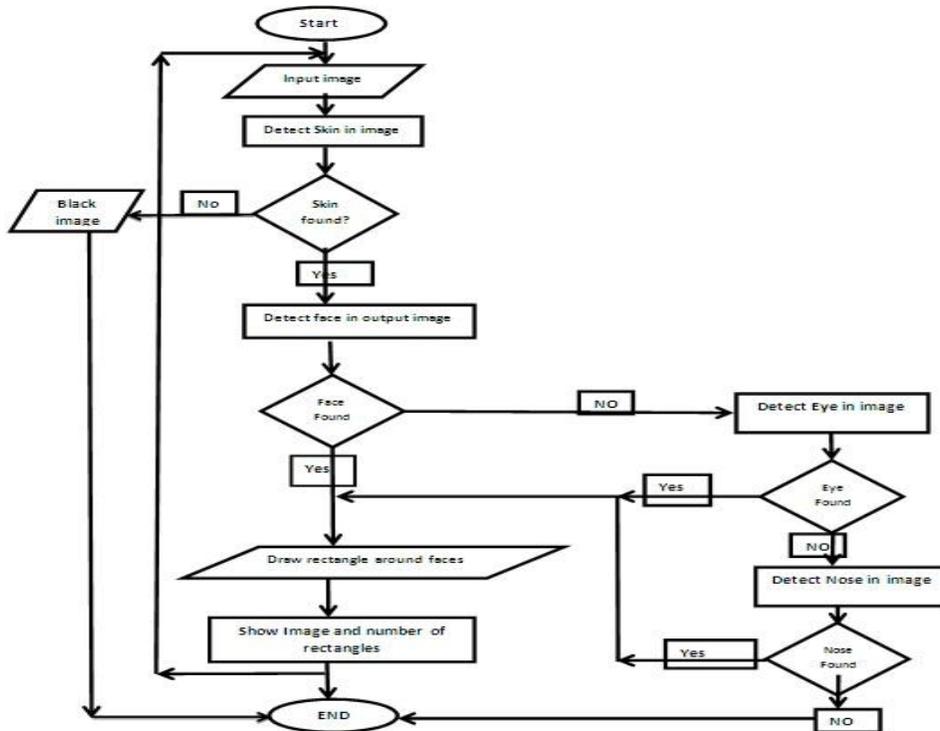
**Fig. 6. Flow Chart for face Detection System [11]**
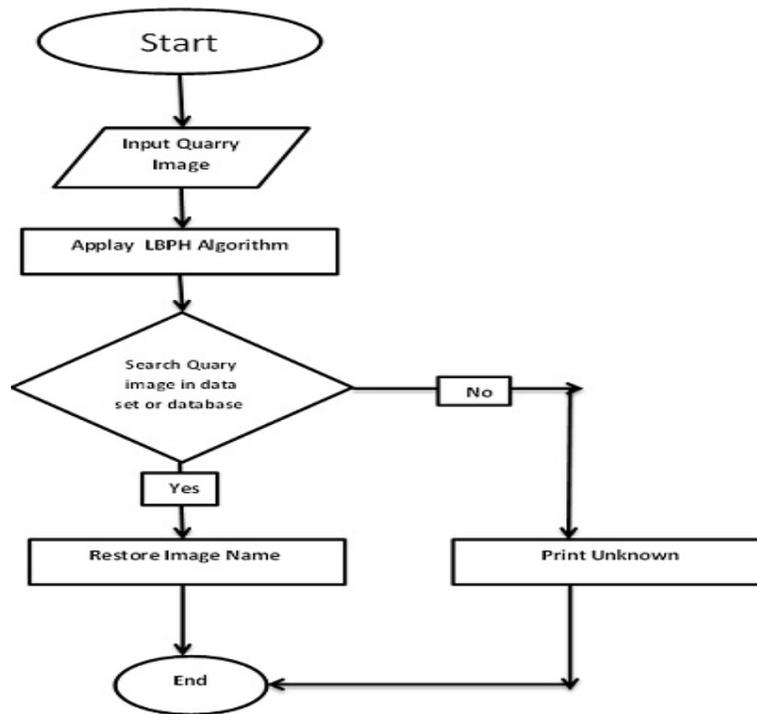
Fig. 7. Face recognition System Flow Chart

At the beginning we should open the camera to get an input image, the first step is to get an input image and apply the face detection process to improve the performance, then we should extract facial feature of the face, give it a name and save it to create dataset to make it as a reference Later. Second part of the system is to input an image to know if it belongs to our data set or not then apply local binary pattern face recognition algorithm LBPH) third we get the output name if this image was found in the dataset if not the system will get the image default name like 'Unknown'. The idea of LBPH is not to find the local features of an image. LBPH algorithm tries to find the local structure of an image, and it does that by comparing each pixel with its neighboring pixels. Simply the modification was represented in using smart method of recognition that helps effectively in the process of recognition [30].

4 Difference between Supposed Methods and Other

There are a number of studies already conducted in face recognition and detection like Fisherfaces [31], EigenFace, LBPH and our supposed model which called MLBPH. We know that Eigenfaces and Fisherfaces are both affected by light and in real life; we cannot guarantee perfect light conditions. LBPH face recognizer is an improvement to overcome this drawback.

5 Experimental Results

Using 25 sample images, captured from video and webcams, have 57 faces appearance in different illuminations and backgrounds. Table 2 shows sample results of the implemented face recognition techniques. The first column shows the image number, the next column shows the original images, and the next 3 columns shows the Face Recognition with EigenFace, Face Recognition with LBPH only, and Face Recognition with LBPH with the proposed face detection method, respectively.

Table 2. Results of the implemented face recognition techniques

#	Original image	Face Recognition with EigenFace	Face Recognition with LBPH only	Face Recognition with LBPH + proposed face detection method
1				
2				
3				
4				
5				
6				
7				
8				

#	Original image	Face Recognition with EigenFace	Face Recognition with LBPH only	Face Recognition with LBPH + proposed face detection method
9				
10				
11				
12				
13				
14				
15				
16				

#	Original image	Face Recognition with EigenFace	Face Recognition with LBPH only	Face Recognition with LBPH + proposed face detection method
17				
18				
19				
20				
21				
22				
23				
24				
25				
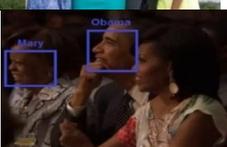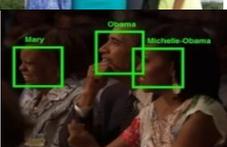
Table 3 shows the results of the true positive and true negative of each face recognition technique.

Table 3. The results of the true positive and true negative of each face recognition technique

#	No.of.Faces	True Positive			True Negative		
		EigenFace	LBPH	LBPH+Proposed	EigenFace	LBPH	LBPH+Proposed
1	1	1	1	1	0	0	0
2	1	1	1	1	0	0	0
3	1	1	1	1	0	0	0
4	2	1	1	2	1	1	0
5	4	2	3	4	2	1	0
6	2	2	2	2	0	0	0
7	2	1	1	1	1	1	1
8	2	2	2	2	0	0	0
9	3	1	2	3	2	1	0
10	2	1	2	2	1	0	0
11	2	2	2	2	0	0	0
12	4	3	4	4	1	0	0
13	6	4	5	5	2	1	1
14	2	1	1	2	1	1	0
15	1	1	1	1	0	0	0
16	2	2	2	2	0	0	0
17	1	1	1	1	0	0	0
18	1	0	1	1	1	0	0
19	3	2	2	3	1	1	0
20	2	1	2	2	1	0	0
21	4	2	3	4	2	1	0
22	3	2	2	3	1	1	0
23	3	3	3	3	0	0	0
24	1	1	1	1	0	0	0
25	2	2	2	2	0	0	0
Total	57	40	48	55	17	9	2

Table 4 summarizes the Experimental results for the recognition success rate and error rate.

Table 4. The experimental results for summary

	EigenFace	LBPH only	Proposed
Recognition Rate	70%	84.2%	96.5%
Average Error Rate	29.8 %	**%15**	3.5%

As shown in Tables 3 and 4 the modified LBPH method improved the success rate to 96.5% with error rate 3.5%, as it detected faces more than other ways so it could recognize faces better while using EigenFace only has a high error rate 29% and only 70% recognition rate. Also using LBPH has approximate 84% of recognition rate and 15% error rate. We can say that, the success rate of the proposed recognition system was better than the EigenFaces and LBPH only techniques.

6 Conclusion

Authentication is an important issue in system control in computer based communication. Human face recognition is a significant branch of biometric verification and has been used widely in many applications, such as video monitor system, human-computer interaction, and door control system and network securit. In this paper we are concerned with faces on a video stream so we built a modified face recognition technique based on Local Binary Pattern histogram with processed data (faces detected before by using a combination of Haar cascade files that uses skin detection, eye detection and nose detection) as input of LBP to increase the accuracy of the proposed recognition system. The modified LBPH method improved the success rate to 96.5% with error rate 3.5%, while using EigenFace only has a high error rate 29% and only 70% recognition rate. Also using LBPH has approximate 84% of recognition rate and 15% error rate. The success rate of the proposed recognition system was better than the EigenFaces and LBPH only techniques.

7 Discussion and Future Work

There is many possible data could be processed to recognize faces like face expression like happiness sadness this enable us to analysis the character of the person and will help to get more information about the person which was recognized. It will be helpful in many fields Like Psychiatry. In the future we will try to obtain more accurate results by using sophisticated technique as face recognition by eye iris.

Competing Interests

Authors have declared that no competing interests exist.

References

[1]   Kar N, Debbarma MK, Saha A, Pal DR. Study of implementing automated attendance system using face recognition technique. International Journal of Computer and Communication Engineering. 2012;100.

[2]   wikipedia.org. [Online]; 2015.
        Available:https://en.wikipedia.org/wiki/Motion_detection

[3]   Zheng Q, Chellappa R. Motion detection in image sequences acquired from a moving platform. IEEE International Conference on Acoustics, Speech, and Signal Processing. 1993;5:201-204.

[4]   Mitiche A, Bouthemy P. Computation and analysis of image motion: A synopsis of current problems and methods. International Journal of Computer Vision. 1996;19(1):29-55.

[5]   Zhao W, Chellappa R, Phillips PJ, Rosenfeld A. Face recognition: A literature survey. ACM Computing Surveys (CSUR). 2003;4(35):399-458.

[6]   Hjelmås E, Low BK. Face detection: A survey. Computer Vision and Image Understanding. 2001;83(3):236-274.

[7]   Bradski G, Kaehler A. Learning OpenCV: Computer vision with the OpenCV library. s.l.: O'Reilly Media; 2008.

[8]   Jafri R, Arabnia HR. A survey of face recognition technique., s.l.: Jips. 2009;2:5.

[9]   Bhat VS, Pujari JD. Human face recognition using combined approaches PCA and ICA., s.l.: Digital Image Processing. 2015;7(10):301-307.

[10]   Patel R, Yagnik SB. A literature survey on face recognition techniques. International Journal of Computer Trends and Technology (IJCTT). 2013;4(5):189-194.

[11]   Ali AA, El-Hafeez TA, Mohany YKA. Robust and efficient system to detect human faces based on facial features. s.l.: Asian Journal of Research in Computer Science. 2018;2(4):1-12.

[12]   Madan Lal, Kamlesh Kumar. Study of face recognition techniques: A survey. s.l.: International Journal of Advanced Computer Science and Applications. 2018;9.

[13]   OpenCV 2.4.13.7 documentation. opencv.org. [Online]; 2019.
        Available:https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html

[14]   Shrivastava K, Manda S, Chavan PS, Patil TB, Sawant-Patil ST. Conceptual model for proficient automated attendance system based on face recognition and gender classification using Haar-Cascade, LBPH algorithm along with LDA model. International Journal of Applied Engineering Research. 2018;10:8075-8080.

[15]   Turk MA, Pentland AP. Face recognition using eigenfaces. In Proceedings. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 1991;586-591.

[16]   Kaehler A, Bradski G. Learning OpenCV 3: Computer vision in C++ with the OpenCV library. O'Reilly Media, Inc.; 2016.

[17]   Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2006;12:2037-2041.

[18]   Özdil A, Özbilen MM. A survey on comparison of face recognition algorithms. IEEE 8th International Conference on Application of Information and Communication Technologies (AICT); 2014.

[19]   Shan C. Learning local binary patterns for gender classification on real-world face images. Pattern Recognition Letters. 2012;33(4):431-437.

[20]   Chintalapati S, Raghunadh MV. Automated attendance management system based on face recognition algorithms. In 2013 IEEE International Conference on Computational Intelligence and Computing Research. 2013;1-5.

[21]   Zhao X, Wei C. A real-time face recognition system based on the improved LBPH algorithm. 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP). 2017;72-76.

[22]   Ranganatha S, Gowramma YP. Image training and LBPH based algorithm for face tracking in different background video sequence. International Journal of Computer Sciences and Engineering. 2018;9(6):349-354.

[23]   Face Recognition LBPH. Towards Data Science. [Online]; 2017.
        Available:https://towardsdatascience.com

[24]   Bakhshi Y, Kaur S, Verma P. An improvement in face recognition for invariant faces. s.l.: International Journal of Current Engineering and Technology. 2016;6(2):423-426.

[25]   Barnouti NH. Improve face recognition rate using different image pre-processing techniques. s.l.: American Journal of Engineering Research (AJER). 2016;5(4).

[26]   Shah DH, Shah DJ, Shah DTV. The exploration of face recognition techniques. s.l.: International Journal of Application or Innovation in Engg. And Management (IJAIEM). 2014;2(3).

[27]   Li Y, Wang G, Nie L, Wang Q, Tan W. Distance metric optimization driven convolutional neural network for age invariant face recognition. s.l.: Pattern Recognition. 2018;75.

[28]   Kurmi US, Agrawal D, Baghel RK. Study of different face recognition algorithms and challenges. s.l.: IJER. 2018;3:112-115.

[29]   Kadam KD. Ace recognition using principal component analysis with DCT. s.l.: International Journal of Engineering Research and General Science. 2017;2091-2730.

[30]   Kadir K, Kamaruddin MK, Nasir H, Safie SI, Bakti ZAK. 4th International Conference on Engineering Technology and Technopreneuship. 2014, August. A comparative study between LBP and Haar-like features for Face Detection using OpenCV. s.n. 2015;335-339.

[31]   Mu Y, Yan S, Liu Y, Huang T, Zhou B. Discriminative local binary patterns for human detection in personal album. s.l.: Conference on Computer Vision and Pattern Recognition. 2018;1-8.