Face recognition-based real-time system for surveillance

Article *in* Intelligent Decision Technologies · September 2016

DOI: 10.3233/IDT-160279

Intelligent Decision Technologies 11 (2017) 79–92
DOI 10.3233/IDT-160279
IOS Press

Face recognition-based real-time system for surveillance

Fahad Parvez Mahdi[a,*], Md. Mahmudul Habib[b], Md. Atiqur Rahman Ahad[b], Susan Mckeever[c], A.S.M. Moslehuddin[b] and Pandian Vasant[a]

[a]*Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, 32610 Bandar Seri Iskandar, Perak, Malaysia*
[b]*Department of Electrical and Electronic Engineering, University of Dhaka, Dhaka, Bangladesh*
[c]*School of Computing, Dublin Institute of Technology, Dublin, Ireland*

Abstract. The ability to automatically recognize human faces based on dynamic facial images is important in security, surveillance and the health/independent living domains. Specific applications include access control to secure environments, identification of individuals at a particular place and intruder detection. This research proposes a real-time system for surveillance using cameras. The process is broken into two steps: (1) face detection and (2) face recognition to identify particular persons. For the first step, the system tracks and selects the faces of the detected persons. An efficient recognition algorithm is then used to recognize detected faces with a known database. The proposed approach exploits the Viola-Jones method for face detection, the Kanade-Lucas-Tomasi algorithm as a feature tracker and Principal Component Analysis (PCA) for face recognition. This system can be implemented at different restricted areas, such as at the office or house of a suspicious person or at the entrance of a sensitive installation. The system works almost perfectly under reasonable lighting conditions and image depths.

Keywords: Computer vision, face detection, face tracking, face recognition, real time, video surveillance, Viola and Jones, principal component analysis

1. Introduction

Computer vision is a field of computer science that includes methods for acquiring, processing and analyzing images and other high-dimensional data to produce numerical or symbolic information. Computer vision plays a crucial role in smart surveillance systems. It helps us detect and track people. It then recognizes their faces and actions [1]. These components are the main ingredients of a modern surveillance system. Video surveillance systems have become a key component in securing different key areas such as airports, police stations, government buildings, military installations, and banks [2]. This is why there is a growing demand for applications to support monitoring indoor and outdoor environments [3–6]. Research in video surveillance systems is a multidisciplinary field that includes image analysis and processing, pattern recognition, signal processing, embedded computing, and communication [7]. Video-based face recognition systems usually operate via three modules: one for detecting the face, the second for tracking it and the third for recognizing it [8]. In our proposed and developed system, we too focus on these combined modules to create an efficient system for surveillance. The system can then be used in any restricted area or suspicious installation of interest to monitor people by their faces.

To detect single as well as multiple faces in a real-time video, we use the Viola-Jones method [9]. The main components of this face detection framework (Viola-Jones) are integral imaging, adaboosting and cascading. A new representation of the image, called the integral image, motivated by the work of Papa-

[*] Corresponding author: Fahad Parvez Mahdi, Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, 32610 Bandar Seri Iskandar, Perak, Malaysia. E-mail: fahadapecedu@gmail.com.

georgiou et al. [10], allows very fast feature evaluation and is the first contribution of this method. The second contribution of this method is a simple and efficient classifier that is built by selecting a small number of important features from a huge library of potential features using AdaBoost. The final major contribution of this method is its successful combination of more-complex classifiers in a cascade structure, which dramatically increases the speed of the detector by focusing on promising regions of the image.

Face tracking is basically a front-end for facial recognition. It (face tracking) works by tracking facial feature points like landmarks surrounding facial components such as eyebrows, eyes, nose, and mouth [11, 12]. It has been used for security as well as for monitoring so as to avoid fraud by substituting live face with an image [13]. We use the Kanade-Lucas-Tomasi feature tracker to track features. This tracker is based on the early work of Lucas and Kanade [14], was developed fully by Tomasi and Kanade [15], and was explained in detail in the paper by Shi and Tomasi [16]. Here, good features are usually located by examining the minimum eigenvalue of each 2 by 2 gradient matrix. The features are then tracked using a Newton-Raphson method [17] of minimizing the difference between the two windows. Multi-resolution tracking allows for relatively large displacements between images.

Automatic face recognition has seen a gradual increase in use for the identification of individuals to enhance security and situational awareness [18]. Under manually controlled conditions, efficient, precise and timely responses are required to identify faces captured on surveillance cameras. However, the recognition of captured faces suffers from error due to different adverse factors, such changes in illumination conditions, pose, resolution, expression, occlusion and blur [19]. Different methods have been developed to counter these problems. The most widely used method is Eigenfaces [20], which is based on feature extraction. Methods based on Principal Component Analysis (PCA), Fisherfaces [21], Linear Discriminant Analysis (LDA) and Independent Component Analysis (ICA) [22,23] have also been frequently used. We have used PCA in our developed system for face recognition. This technique was pioneered by Kirby and Sirovich in 1988 [24].

2. Related works

Face detection is one of the most heavily studied topics in computer vision literature. The Viola-Jones [9] face detection algorithm is one of the most popular and simple algorithms used in this field. Its feature-based technique is relatively robust to position variations, interested reader may read [25] for details. In principle, feature-based schemes can be made invariant to size, orientation and/or lighting [26]. Other benefits of these schemes include the compactness of representation of the face images and high-speed matching [27]. The problem of these approaches is the difficulty of automatic feature detection and that the implementer of any of these techniques has to make arbitrary decisions about which features are important [28]. The features of the Viola-Jones algorithm were further extended, leading to many other face detection methods. A similar approach was proposed in [29], where instead of computing Haar-like features (see Section 3.1 for more information about Haar-like features), individual pixels are compared. The Viola-Jones method [9], which considers only 4 types of horizontal/vertical features, was later extended by Lienhart and Maydt [30] by introducing rotated features. In the subsequent years, many variations of the Haar-like features were proposed. Interested readers may read [31] for more information.

Face recognition is another well-known topic of computer vision and pattern recognition that has been extensively studied over the past few decades. One of the most well-known face recognition methods for face classification is the early work of Turk and Pentland [20], which is based on the notion of eigenfaces. It was previously shown [24] that any face image can be represented as a linear combination of pictures (called eigenfaces) and their coefficients. This method is based on Principal Component Analysis (PCA), with the eigenfaces serving as the principal components of the initial training set of face images. The face space generated by the PCA is one where the variance of all the samples is minimized. By projecting the faces in a space where the variance of the samples within one class (same individual) is minimized and, at the same time, the variance of the samples between classes (different individuals) is maximized; one can achieve better classification results. This is the idea behind the Fisher-faces method [32]. The basis vectors for such a sub-space can be generated by Linear Discriminant Analysis (LDA). A more detailed comparison of the eigenfaces (PCA) versus the Fisherfaces (LDA) methods can be found in [21,33].

Real-time recognition of faces is needed for surveillance to ensure security. It involves real-time recognition of faces from a sequence of images captured by

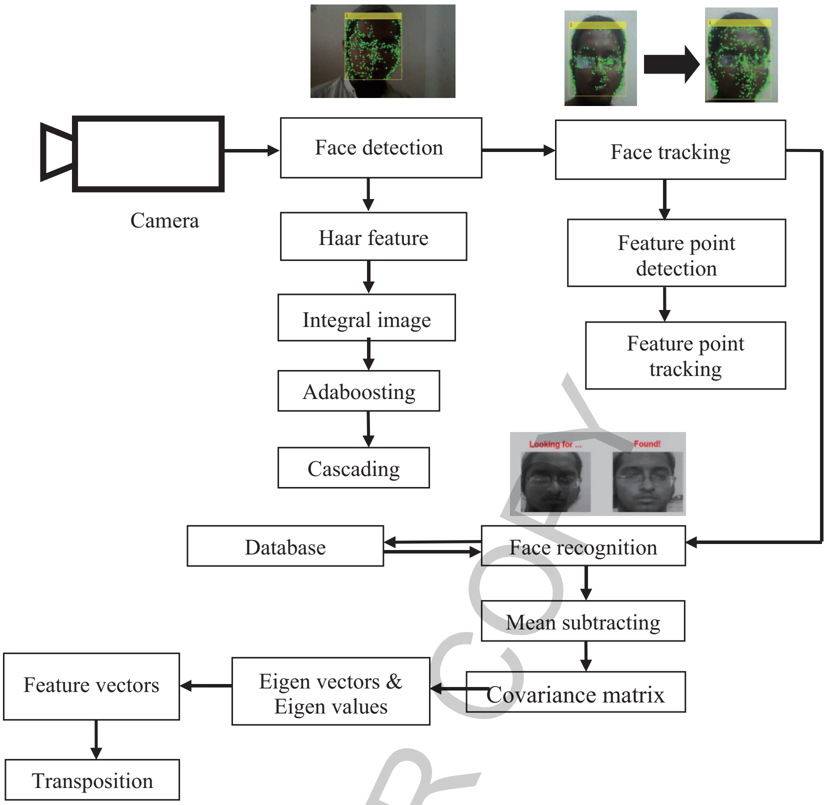

Fig. 1. Flowchart of our proposed and developed system.

a video camera [34]. Much research has been directed to this area. In most cases, recognition has been performed through choosing a few good frames and then applying a suitable recognition technique for intensity images to those frames in order to identify the individual [35]. A two-layer Radial Basis Function network [36,37] and Difference of Gaussian (DoG) filtering as well as Gabor wavelets were proposed and used by Howell and Buxton [38] for learning/training and to analyze the feature representation, respectively, while other schemes, such as combined motion and model-based face tracking [39], have been utilized for face detection and tracking. A skin color modeling [40] system was proposed by Campos et al. [41] to detect the face. GWN [42] was then utilized to detect prominent facial points such as the eyes, nose, and mouth and to track those features. Here, the eigenfeatures [43] of each individual frame are then extracted and all the eigenfeatures are combined by a feature selection algorithm [44]. Later, the best eigenfeatures are selected to form the feature space. A couple of classifiers [45] have been applied to identify individual persons in the frame. Finally, a super classifier [46] performs the final classification for the entire video sequence. The recognition rate has been found to be high, at nearly 98 percent.

Some approaches [47,48] have utilized a video-to-video paradigm that collects information from a sequence of frames from a video segment and then combines and associates it with an individual face. This approach involves a temporal analysis of the video sequence and a condensing of the tracking and recognition problems, but these operations are a matter of ongoing research, as the reported experiments were performed without any real variations in orientation and facial expressions [49]. It is also important to mention here that several other schemes have incorporated information from other modalities to recognize faces from images acquired from video clips. System such as [50] makes use of stereo information, with high recognition accuracy (of 90%) reported, while [51] exploits both audio and video as well as 3D information about the head to achieve the highest accuracy rate (i.e., 100%). More recently, many other methods and techniques have been developed and introduced for face recognition from video cameras. It is outside the scope of this paper to explain these methods in detail. The interested reader may refer to Table 1 below to

Table 1
Recently used techniques for face recognition from video camera

Author(s)	Year	Title	Technique(s)
Miguel De-la-Torre, Eric Granger, Robert Sabourin & Dmitry O. Gorodnichy	2015	Adaptive skew-sensitive ensembles for face recognition in video surveillance [52]	Adaptive skew-sensitive ensembles
Bogdan Raducanu, Alireza Bosaghzadeh and Fadi Dornaika	2015	Multi-observation face recognition in videos based on label propagation [53]	Adaptive graph construction
N.M. Khan, Xiaoming Nan, A. Quddus, E. Rosales and Ling Guan	2015	On video based face recognition through adaptive sparse dictionary [54]	Adaptive sparse dictionary
N. Hassanpour and L. Chen	2015	A hierarchical training and identification method using Gaussian process models for face recognition in videos [55]	Ensemble of Abstract Sequence Representatives
Z. Huang, R. Wang, S. Shan and X. Chen,	2015	Face recognition on large-scale videos in the wild with hybrid Euclidean-and-Riemannian metric learning [56]	Hybrid Euclidean-and-Riemannian Metric Learning
M.U. Ragashe, M.M. Goswami and M.M. Raghuwansh	2015	A face recognition technique for partial occluded faces in streaming video [57]	Modified AdaBoost-based face detector
X.S. George Zhao, Hongzong Si and Peilong Xu	2014	A cooperative dual-camera system for face recognition and video monitoring [58]	Class Specific Hyper Graph
L. An, B. Bhanu and S. Yang	2012	Face recognition in multi-camera surveillance videos [59]	Unified Face Image
O. Arandjelović	2012	Color invariants under a non-linear photometric camera model and their application to face recognition from videos [60]	Discriminative power of color-based invariants
L. An, M. Kafai and B. Bhanu	2012	Face recognition in multi-camera surveillance videos using dynamic Bayesian network [61]	Dynamic Bayesian network
S. Damavandinejadmonfared	2012	Kernel entropy component analysis using local mean-based k-nearest centroid neighbor (LMKNCN) as a classifier for face recognition in video surveillance camera systems [62]	Local mean-based k-nearest centroid neighbor

see the recent methods and techniques applied for face recognition from video cameras.

3. Proposed system

In our proposed system, we focus on the identification of faces from video sequences. The basic mechanism proposed in this system is to detect, track and identify people entering the surveillance area. The basic features of our proposed system are the following (Fig. 1):

- Detect single as well as multiple faces from real-time video.
- Crop detected face image and save it to the system memory to create a database.
- Recognize cropped face with a known database (already-stored images or recently taken images by the surveillance camera).
- Detect and track multiple people if they are trying to enter the surveillance area simultaneously.

The system is able to detect faces from real-time video using the Viola-Jones method [9]. To detect faces accurately from the video, efficient features tracker has been used. The detected faces are then used for recognition by an automatic face recognition system. To achieve this goal, reliable modules for face detection, tracking and face recognition are required. Although considerable progress has been made in this area, these systems still suffer from slow training time and limited accuracy in challenging conditions such different illumination conditions, face poses and image depth.

3.1. Face detection

The accurate detection of human faces from arbitrary images is the first important step of facial recognition. Thus, the goal of face detection is to determine whether there are any faces in a given arbitrary image and, if present, to return the image location and extent of each face. The Viola-Jones [9] method is one of the most popular and exploited methods in face detection history. The major components of this algorithm are Haar features, the integral image, adaboost and cascading (Fig. 2).

Here, the features are rectangular in shape. Each feature results in a single value that is calculated by subtracting the sum of pixels under white rectangles from the sum of pixels under black rectangles (Fig. 3). The Viola-Jones algorithm uses a 24 × 24 window as the base window size to start evaluating these features in any given image. If we consider all possible param-

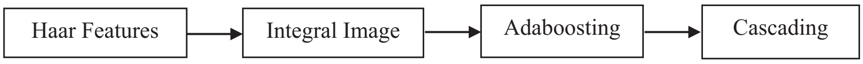

Fig. 2. Flowchart of Viola-Jones algorithm.

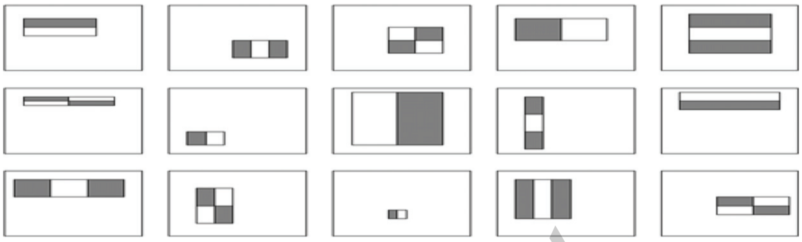

Fig. 3. Different sizes of Haar features for Viola-Jones face detection.

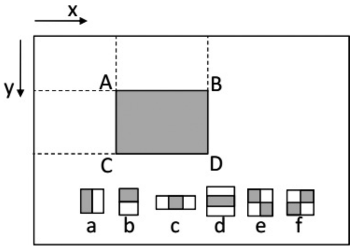

Fig. 4. Illustration of the integral image and Haar-like rectangle features (a)–(f).

eters of the Haar features, such as position scale and type, we end up calculating approximately 180,000+ features in the window.

The integral image, also known as a summed area table, is an algorithm for quickly and efficiently computing the sum of values in a rectangular subset of a grid. Viola and Jones applied the integral image for rapid computation of Haar-like features, as detailed below. The integral image is constructed as follows:

$$ii(x,y) = \sum_{x' \leqslant x, y' \leqslant y} i(x',y') \tag{1}$$

where $ii(x,y)$ is the integral image at pixel location (x,y), and $i(x'y')$ is the original image. The integral image can be constructed in one pass over the original image using the following recurrent formulas:

$$s(x,y) = s(x,y-1) + i(x,y)\ldots \tag{2}$$

$$ii(x,y) = ii(x-1,y) + s(x,y)\ldots \tag{3}$$

where $s(x,y)$ is the accumulated pixel values of row x, $s(x,y-1) = 0$, $ii(x-1,y) = 0$.

Using the integral image to compute the sum of any rectangular area is extremely efficient, as shown in Fig. 4. The sum of pixels in rectangle region *ABCD* can be calculated as:

$$\sum_{(x,y)\in ABCD} i(x,y) = ii(D) + ii(A) - ii(B) - ii(C)\ldots \tag{4}$$

This only requires four array references. The integral image can be used to compute simple Haar-like rectangular features, as shown in Figs 4(a)–(f). The features are defined as the (weighted) intensity difference between two to four rectangles. For instance, in feature (a), the feature value is the difference in average pixel value in the gray and white rectangles. Since the rectangles share corners, the computation of two rectangle features (a and b) requires six array references, the computation of three rectangle features (c and d) requires eight array references, and the computation of four rectangle features (e and f) requires nine array references [29,63]. As discussed earlier, there can be approximately 180,000+ feature [29] values within a detector at a 24 × 24 base resolution that need to be calculated. However, it is to be understood that only a few sets of features will be useful among all the features that could be used to identify a face [29]. Adaboost [64] is a machine learning algorithm that helps find only the best among all 180,000+ features. After these features are found, a weighted combination of all the features is used in evaluating and deciding whether any given window has a face. Each of the selected features is considered sufficient for inclusion if they can at least perform better than random guessing (detecting more than half the cases) [29].

These firstly acquired features are also called weak classifiers as most of their detection rates are slightly better than random guess. Adaboost [64] constructs a strong classifier as a linear combination of these weak classifiers.

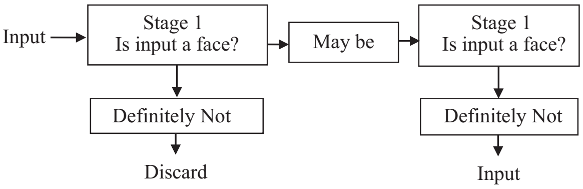

Fig. 5. Different cascading stages.

$$\underbrace{F(x)}_{\text{Strong classifier}} = \underbrace{\alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \dots}_{\text{Weak classifier}} \tag{5}$$

The basic principle of the Viola-Jones face detection algorithm is to scan the detector many times through the same image, each time with a new size. Even if an image contains one or more faces, it obvious that an excessively large number of evaluated sub-windows would still be negatives (non-faces). Thus, the algorithm should concentrate on discarding non-faces quickly and spend more time on probable face regions. Hence, a single strong classifier formed out of a linear combination of all of the best features is impractical for evaluating each window because of computation cost. Therefore, a cascade classifier is used that is composed of stages, each containing a strong classifier (Fig. 5). Therefore, all the features are grouped into several stages, where each stage has a certain number of features. The job of each stage is used to determine whether a given sub-window is definitely not a face or may be a face. A given sub window is immediately discarded as not a face if it fails in any of the stages [29,63].

3.2. Face tracking

The automatic detection and tracking of (typically corner-like) feature points throughout an image sequence is a necessary prerequisite for many algorithms in computer vision. The gathered information about the feature points and their motion can subsequently be used for pose estimation, camera self-calibration and tracking various kinds of objects such as people and vehicles. One of the most popular methods for feature point tracking is the Kanade-Lucas-Tomasi (KLT) algorithm [14], which was introduced by Lucas and Kanade [14], later extended in the works of Tomasi and Kanade [15], and Shi and Tomasi [16].

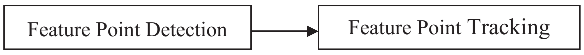

Fig. 6. Flowchart of the KLT algorithm.

The KLT algorithm operates into two main parts (Fig. 6). During the detection process, salient feature points are found and added to the already existing ones. Afterward, in the tracking process for each feature point, its corresponding motion vector is calculated. In the following, we describe each part in more detail.

Note that in the following, Greek letters denote scalars, lowercase letters denote column vectors and uppercase letters denote matrices. We denote I as the current image and J as the subsequent image in the sequence. We write $\nabla I = \delta I/\delta(x, y)$ as the spatial image gradient of I, which is typically performed using the Sobel or Sharr operator for robustness. Also, we define $W(p)$ as a small rectangular region centered at a given point p. Typically, $W(p)$ will be a 5 × 5 or 7 × 7 pixel neighborhood. Because the tracking is done with sub-pixel precision, p will have non-integer coordinates. Its neighbors are then calculated using bilinear interpolation [65].

The task here is to detect new feature points in a given image I and add them to the already existing feature points. To track feature points reliably, their pixel neighborhood should be richly structured. As a measure of 'structuredness' of the neighborhood of a pixel p, one can define the structure matrix G:

$$G = \sum_{x \in W(p)} \nabla I\,(x)\,.\nabla I(x)^T \tag{6}$$

Its eigenvalues λ_1, λ_2 (which are guaranteed to be $\geqslant 0$, as the matrix is positive-semi definite) deliver useful information about the neighborhood region W. If W is completely homogenous, then $\lambda_1 = \lambda_2 = 0$. In contrast, $\lambda_1 > 0$, $\lambda_2 = 0$ indicates that W contains an edge, and $\lambda_1 > 0$, $\lambda_2 > 0$ indicates a corner. The smaller eigenvalue $\lambda = \min(\lambda_1, \lambda_2)$ can now be used as a measure of the cornerness of W, where larger values mean stronger corners. The feature detection is now composed of the following steps (Fig. 7): First, calculate structure matrix G and cornerness λ for each pixel in the image I. Second, calculate the maximum cornerness $\lambda_{\max}$ occurring in the image. Third, keep all pixels that have a cornerness λ larger than a certain percentage (5%∼10%) of $\lambda_{\max}$. Performa non-maxima suppression within the 3 × 3 pixel neighborhood of the remaining points to keep only the local maxima. From the remaining points, add as many new points to the already existing points as needed, starting with the points with the highest cornerness values. To avoid the concentration of points in some area of the

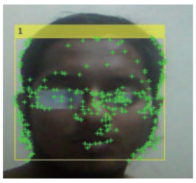

Fig. 7. Feature point detection.

image, newly added points must have a specific minimum distance (e.g., 5 or 10 pixels) to the already existing points as well as to other newly added points (Minimum Distance-Enforcement) [14,66].

In the tracking step, we want to calculate for each feature point p in image I its corresponding motion vector v so that its tracked position in image J is $p+v$. As 'goodness' criterion of v, we take the SSD error function:

$$\varepsilon\left(v\right) = \sum_{x\in W(p)} \left(J\left(x+v\right) - I(x)\right)^2 \tag{7}$$

The SSD error function measures the image intensity deviation between a neighborhood of the feature point position in I and its potential position in J and should be zero in the ideal case. Setting the first derivative of $\varepsilon(v)$ to zero and approximating $J(x+v)$ by its first-order Taylor expansion around $v = 0$ results in a better estimate, $v = 1$. By repeating this multiple times, we obtain an iterative update scheme for v, as summarized in Table 2.

Due to the Taylor expansion around zero, the given scheme is only valid for small motion vectors, v. To allow tracking of large motions of feature points (Fig. 8), which is quite common, we generate an image pyramid and apply the scheme for all points in each pyramid level. We do this from a coarse pyramid level to a fine one, using the result of the previous pyramid level as the initial guess for the next one [14,66].

Table 2 KLT pseudo-code of the calculation of the motion vector v for a given feature point p. $W(p)$ is a window centered at p. Typically, the window size is set to 5 × 5 pixels, *maxIter* to 10 pixels and esp. to 0.03 pixels [67].

3.3. Face recognition

PCA has been extensively exploited for face recognition algorithms. It not only reduces the dimensionality of the image but also retains some of the variations in the image data. The system functions by projecting the face image onto a feature space that spans the significant variations among known face images. The significant features are known as "eigenfaces" [20], because they are the eigenvectors (Principal Component) of the set of faces that do not necessarily correspond to features such as eyes, ears, and noses. The projection operation characterizes an individual face by a weighted sum of the eigenface features, so to recognize a particular face, it is necessary only to compare these weights to those individuals. The Eigen Object Recognizer applies PCA to each image, the results of which will be an array of eigenvalues that a Neural Network can be trained to recognize. The method to which PCA is applied can vary at different stages, so what will be demonstrated is a clear method for PCA application that can be followed.

```
1. Set initial motion vector $v_1 = (0,0)^T$
2. Spatial image gradient $\nabla I = \partial I/\partial(xy)$
3. Calculate structure matrix $G = \sum_{x\in W(p)} \nabla I\left(x\right).\nabla I(x)^T$
4. For $k = 1$ to maxlter
   (a) Image difference $\eta_x = I\left(x\right) - J(x+v^k)$
   (b) Calculate mismatch vector $b = \sum_{x\epsilon W(p)} \eta\left(x\right).\nabla I(x)$
   (c) Calculate updated motion $v_{k+1} = v_k + G^{-1}b$
   (d) If $\| v_{k+1} - v_k < eps$, then stop (converged)
5. Report final motion vector $v$
```

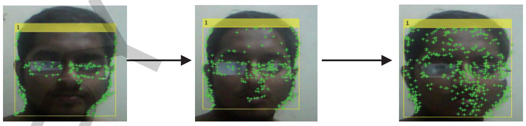

Fig. 8. Feature point tracking.

These data (array of eigenvalues) are fairly simple, making the calculation of our covariance matrix a little simpler now that it is not the subtraction of the overall mean from each of our values, as we need at least two dimensions of data for covariance. Rather, it is the subtraction of the mean of each row from each element in that row. Alternatively the mean of each column from each element in the column could be used; however, this would change the way we calculate the covariance matrix.

The basic Covariance equation for two-dimensional data is:

$$cov\left(x,y\right) = \frac{\sum_{i=1}^{n} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\left(n-1\right)} \tag{8}$$

which is similar to the formula for variance; however, the change of x is in respect to the change in y rather

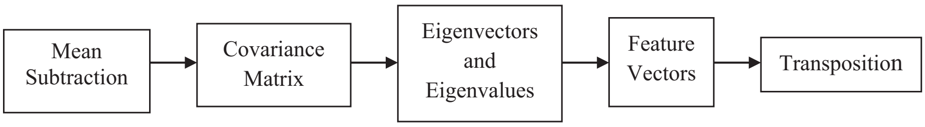

Fig. 9. Flowchart of PCA algorithm.

than solely the change of x in respect to x. In this equation, x represents the pixel value, x is the mean of all x values, and n is the total number of values. The covariance matrix formed from the image data represents how much the dimensions vary from the mean with respect to each other.

Eigenvalues are a product of multiplying matrices; however, they are a special case. Eigenvalues are found by multiples of the covariance matrix by a vector in 2-dimensional spaces (i.e., an eigenvector). This makes the covariance matrix the equivalent of a transformation matrix. Usually, the results of Eigenvalues and eigenvectors are not as clean as in the example above. In most cases, the results provided are scaled to a length of 1.

Once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalues from highest to lowest. This yields the components in order of significance. Here, the data can be compressed and the weaker vectors removed, resulting in a lossy compression method, with the data lost deemed insignificant. The final stage in PCA is to take the transpose of the feature vector matrix and multiply it on the left of the transposed adjusted data set (the adjusted data set is from Stage 1, where the mean was subtracted from the data).

The Eigen Object Recognizer class performs all of this and then feeds the transposed data as a training set into a Neural Network. When it passes along an image for recognition, it performs PCA and compares the generated eigenvalues and eigenvectors to those from the training set. The Neural Network produces a match if one has been found or a negative match if no match is found [68].

Most face recognition algorithms are extremely sensitive to lighting conditions, so if it is trained to recognize a person when they are in a dark room, it will probably not recognize them in a bright room, etc., and there are many other issues. For example, the face should be in a consistent position within the images (such as the eyes being in the same pixel coordinates), and there should be a consistent size, rotation angle, hair and makeup, emotion (smiling, angry, etc.), and position of lights (to the left or above, etc.).

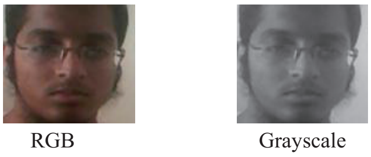

Fig. 10. RGB image and Gray scale image.

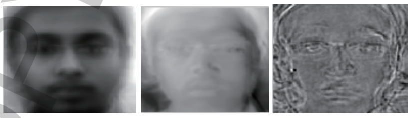

Fig. 11. Average face, eigenface No. 1 and eigenface No. 100.

This is why it is so important to use good image preprocessing filters before applying face recognition. We present an example of this preprocessing stage in the following.

Here, we have selected 15 students from a class and taken 10 photos of each person for a total of 150 preprocessed facial images of the same size (e.g., 92 × 112 pixels). We then use PCA to convert all 150 training images into a set of "eigenfaces" that represent the main differences among the training images. First, it will find the 'average face image' of these 150 images by obtaining the mean value of each pixel. Then, the eigenfaces are calculated in comparison to this average face, where the first eigenface has the most dominant face differences, and the second eigenface has the second most dominant face differences, and so on, until we have approximately 50 eigenfaces that represent most of the differences in all the training set images.

From our experiment, we have shown the average face and the first and last eigenfaces generated from a collection of 150 images (Fig. 11). Note that the average face will show the smooth face structure of a generic person. The first few eigenfaces will show some dominant features of faces, and the last few eigenfaces (e.g., eigenface 119) are mainly image noise. We can see the first 8 eigenfaces in the image below (Fig. 12).

To explain eigenfaces (PCA) in simple terms, the main differences among all the training images are determined, followed by how best to represent each training image using a combination of those differences.

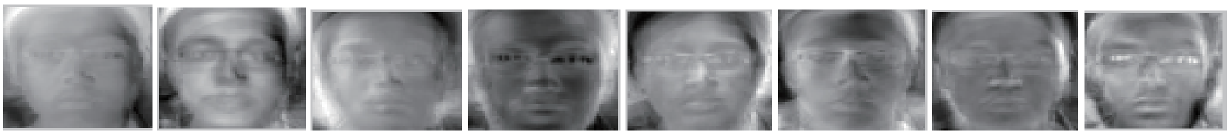

Fig. 12. The first 8 eigenfaces.

For example, one of the training images might be composed of the following:

(Average Face) $+$ (13.5$\%$ of eigenface0) $-$ (34.3$\%$ of eigenface1) $+$ (4.7$\%$ of eigenface2) $+ \ldots +$ (0.0$\%$ of eigenface199).

Once it has figured this out, it can think of that training image as the 200 ratios:

$$\{13.5, -34.3, 4.7 \ldots 0.0\}$$

It is indeed possible to generate the training image back from the 200 ratios by multiplying the ratios with the eigenface images and adding the average face. However, since many of the last eigenfaces will be image noise or will not contribute much to the image, this list of ratios can be reduced to just the most dominant ones, such as the first 50 numbers, without affecting the image quality much. To recognize a person in a new image, it can apply the same PCA calculations to find the 200 ratios for representing the input image using the same 140 eigenfaces. Once again, it can just keep the first 50 ratios and ignore the rest, as they are less important. It can then search through its list of ratios for each of its 20 known people in its database to determine which have the top 50 ratios that are most similar to the 50 ratios for the input image. This is basically a method of checking which training image is most similar to the input image out of all 150 training images that were supplied [68].

The methodology for image capture for our experimental work was as follows: We captured facial video footage for 15 different people, with each person randomly moving their faces during the video capture in order to capture different positions, poses and distances. The system then took many photos (cropped faces) but randomly selected just 10 photos (faces) for each individual person. These 150 images were used as our training images. For the test images, we again captured images of the same 15 people in front of the video camera, with variation on distance, poses and positions. The system randomly captured 10 photos of each of the 15 individuals, with these 150 test images used to validate our system.

4. Results

The proposed and developed system for surveillance has tested under different lighting conditions, facial positions and distances. We have used surveillance video camera to locate faces from video and then used face recognition method to test the credibility of the system. Although the main goal of this developed system is to detect individuals by their faces from a video camera, we also discuss the performances of face detection methods under different conditions. Finally, the effectiveness of the face recognition method used in this system has been tested using 150 facial images of 15 persons. Below, we will describe our findings in terms of face detection and face recognition based on our experiments.

4.1. Results of face detection

As stated earlier, we used Viola-Jones method for face detection [9]. One of its main advantages we found that its feature-based technique is relatively robust to position variations. It also offered compactness of representation of the face images and high-speed matching. However, in our experiment, we found that extreme variations in lighting conditions make it highly error-prone, as these approaches have difficulty selecting the features; consequently, they make arbitrary decisions about which features are important. Below, we describe the method's behavior under different lighting conditions.

4.1.1. Face detection under different lighting conditions

We conducted our experiment under various lighting conditions to observe the effectiveness of our detection method. The detection method will not work efficiently under extreme lighting conditions, especially when the background exhibits powerful lighting (Fig. 13(a)).

Our detection procedure worked well under normal lighting conditions (300–400 lux), as sufficient features can be gathered with ease (Fig. 13(c)). We can see from Fig. 13(c) that the feature tracker successfully extracted sufficient features for facial recognition. This

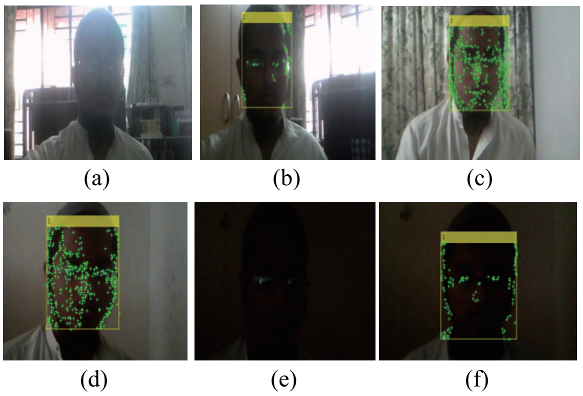

Fig. 13. (a) In the presence of extreme light conditions, the Viola-Jones method fails to locate features or (b) only can locate very few features, (c) and (d) the exploited algorithm can locate the highest number of features undergood lighting and proper distance (e). In the absence of sufficient light, the Viola-Jones method fails to locate features or (f) locates very few features.

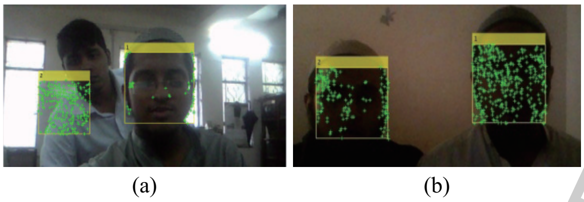

Fig. 14. (a) In the presence of extreme light conditions, the Viola-Jones method fails to locate features or gives false features when more than one face is present. (b) Under good lighting conditions, it can locate the maximum number of features, even for multiple faces.

is also true for the below-average lighting condition (Fig. 13(d)). Here, we show such two examples, one underaverage lighting conditions (100–150 lux) and another underbelow-average conditions (25–50 lux).

Like the extreme illumination condition, a deteriorating lighting condition can also affect face detection. When we began to deteriorate the lighting condition from below-average, we have found out that our face detection method started to gather less features (Fig. 13(f)), and at a certain point, it failed to collect any features (Fig. 13(e)).

We observed nearly the same results for multiple face detection as observed for single face detection. Under extreme or dark lighting conditions, this detection procedure yields almost nothing, whereas under sufficient lighting conditions, it gives a good result. In the following figure, we show these two conditions.

4.2. Results of face recognition

We have exploited PCA for the recognition of faces in our system, and we have described PCA in detail earlier (see Section 3.3). Face recognition is much more complex (in terms of higher error rate and requires more mathematical operations) than face detection. The success of face recognition depends primarily on image depth, lighting conditions, facial poses and expressions. Variations of faces in terms of poses, image depths and lighting conditions in known database play a crucial role in face recognition. We conducted our experiment under different lighting conditions and facial poses.

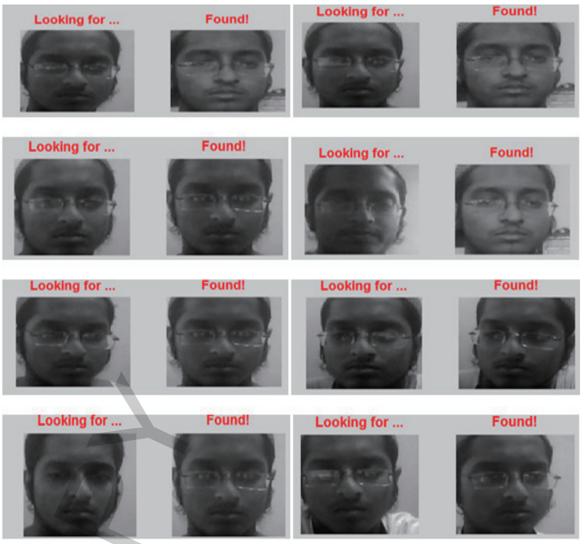

Fig. 15. Face recognition under the same lighting conditions.

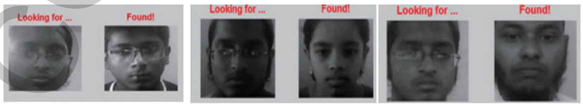

Fig. 16. Face recognition under different lighting conditions.

4.2.1. Face recognition under different lighting conditions

In our experiment, the recognition rate is much higher for the same or nearly same lighting conditions as in the database. We found that the accuracy rate is nearly 100% (Fig. 15). However, under different lighting conditions, whether extreme, dark or varying, the system has a low rate of 30% or less (Fig. 16).

4.2.2. Face recognition with variation in facial positions

Another major concern for face recognition is different facial positions. At different facial positions, it is very difficult to recognize face without a trained database containing multiple positioned examples.

Table 2
Accuracy of face recognition under different conditions

Conditions	Accuracy (in percentage %)
Same facial position under same lighting condition	100%
Same facial position but different lighting condition	30%
Different facial positions under same lighting condition	80%
Different facial positions under different lighting condition	$<$ 30%
Same lighting condition but different image depths	62.5%

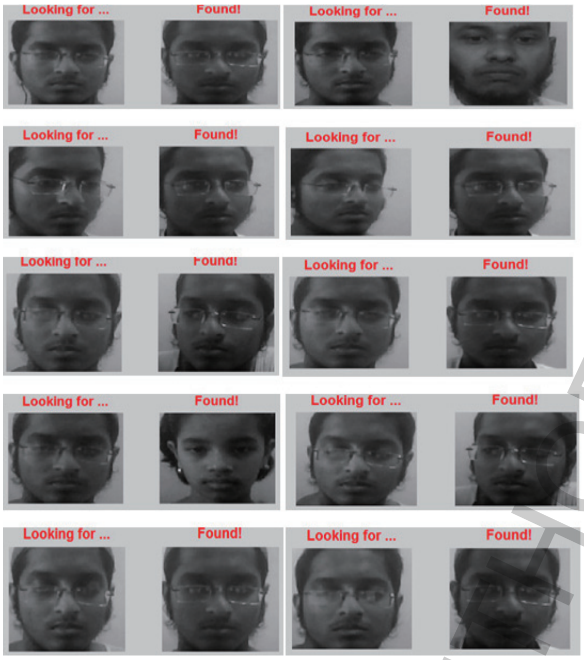

Fig. 17. Face recognition with some variation in facial positions.

Therefore, we created a database consisting of some variation in facial positions and then tested a sample of different facial positions with the trained database. We found that the face recognition accuracy for different facial positions in a trained database is 80% (Fig. 17) under the same lighting conditions. However, with different input images under different lighting conditions, the accuracy is much lower, even lower than the 30% we found earlier for the same facial position but different lighting conditions.

4.2.3. *Face recognition from different distances*

Distance of the facial position or image depth is another major factor we tested using images with different image depths with a trained database. We found that the accuracy rate under such varying image depths is approximately 62.5% (Table 3). This clearly shows how vulnerable the method is if we consider image depth in the system. Here, we show some of our examples.

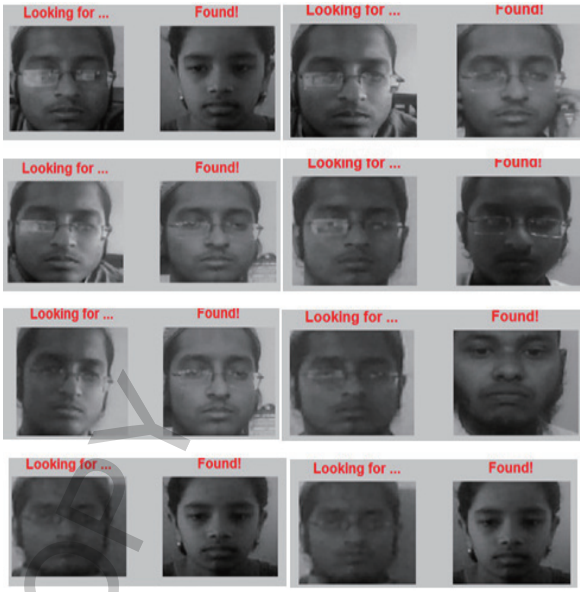

Fig. 18. Face recognition from different distances.

Different lighting conditions remain the primary challenge of face detection, so in our system, we considered whether the place under surveillance has good lighting conditions. In addition to lighting, there are two other key factors that have to be considered to improve the system and make it useful for face recognition: different facial positions and image depths. Below, we show some of the outcomes of our experiment under different conditions (Table 3).

5. Conclusion

In this paper, we have proposed and developed a system that focuses on face detection and face recognition from a video camera. Implementation simplicity, inexpensive computation, real-time nature and smart acquisition of facial images are some of the features of this system. Face detection is always a challenge, especially face recognition under different lighting conditions, positions and image depths. We found good results when we gathered faces under reasonably varied illumination conditions, positions and depths similar to those of our database. However, the system is highly error-prone under varied illumination conditions, positions and image depths. As the areas under surveillance usually have sufficient illumination condition and cameras can be placed anywhere, our proposed surveillance system can be implemented at offices, educational institutions and different sensitive installations. To address the problem of different facial positions or image depths, we may consider more than one camera at different positions and angles. Intelligent techniques can then be used to help combining all the videos from the camera to make the system more efficient, robust and less error-prone to position variations and image depths. Although, we have not been able to demonstrate a multi-camera surveillance system with intelligent agents here, we think the system will work well at avoiding the constraints posed by the different facial positions and image depths. We also need to keep in mind that a well-trained database plays a crucial role in face recognition. Therefore, to have a high percentage of face recognition rates, we need to train our database with as much variations as possible. This work can be a basic step toward a more advanced system with hybrid methods for face detection and recognition. In addition, many intelligent algorithms can be included with this system to make it a more robust and error free computer-based automatic system for surveillance.

Acknowledgments

This research has been supported by the Center for Natural Science & Engineering Research (CNSER). The authors are grateful to the students (of the Dept. of Electrical and Electronic Engineering, University of Dhaka) for volunteering in experiments.

References

[1] Md. A.R. Ahad, Computer vision and action recognition: A guide for image processing and computer vision community for action understanding, Atlantis/Springer, 2011.

[2] R. Rangaswami, Z. Dimitrijevic, K. Kakligian, E. Chang and Y.-F. Wang, The SfinX video surveillance system, *IEEE Conference on Multimedia and Expo*, (2003).

[3] E. Heilmann, Video surveillance and security policy in France: From regulation to widespread acceptance, *Information Polity* **16** (2011), 369–377.

[4] T.D. Räty, Survey on contemporary remote surveillance systems for public safety, *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews* **40** (2010), 493–515.

[5] D.D. Paola, A. Milella, G. Cicirelli and A. Distante, An autonomous mobile robotic system for surveillance of indoor environments, *International Journal of Advanced Robotic Systems* **7** (2010), 19–26.

[6] J. Zhang, G. Song, G. Qiao, T. Meng and H. Sun, An indoor security system with a jumping robot as the surveillance terminal, *IEEE Transactions on Consumer Electronics* (2011), 57.

[7] F.F. Chamasemani and L.S. Affendey, Systematic review and classification of video surveillance systems, *International Journal of Information Technology and Computer Science* **7** (2013), 87–102, ISSN 2074–9007, ESSN: 2074–9015.

[8] L. Torres, L. Lorente and J. Vilà, Face recognition using self-eigenfaces, in: *International Symposium on Image/Video Communications Over Fixed and Mobile Networks*, Rabat, Morocco, (2000), 44–47.

[9] P. Viola and M.J. Jones, Robust real time face detection, *International Journal of Computer Vision* **57**(2) (2001), 137–154.

[10] C. Papageorgiou, M. Oren and T. Poggio, A general framework for object detection, in: *International Conference on Computer Vision*, (1998).

[11] T. Yan, Y. Wang, Z. Zhu and Q. Ji, Robust facial feature tracking under varying face pose and facial expression, *Pattern Recognition* **40**(11) (2007), 3195–3208.

[12] Y. Nishina, M.A.R. Ahad, J.K. Tan, H.S. Kim and S. Ishikawa, A robust face tracking method by employing color-based particle filter, *Biomedical Fuzzy and Human Sciences: The Official Journal of the Biomedical Fuzzy Systems Association* **16** (2011), 127–134.

[13] N. Markus, Overview of algorithms for face detection and tracking, retrieved from https://www.fer.unizg.hr/_download/repository/KDI_Nenad_Markus.pdf, (accessed at 22-Jan-16).

[14] B.D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, *IJCAI* (1981), 674–679.

[15] C. Tomasi and T. Kanade, Detection and tracking of point features, Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[16] J. Shi and C. Tomasi, Good features to track, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Jun 1994), 593–600.

[17] A. Ben-Israel, A Newton-Raphson method for the solution of systems of equations, *Journal of Mathematical Analysis and Applications* **15.2** (1966), 243–252.

[18] F. Matta and J.L. Dugelay, Person recognition using facial video information: A state of the art, *Journal of Visual Languages and Computing* **20** (2009), 180–187.

[19] R. Chellappa, M. Du, P. Turaga and S.K. Zhou, Face tracking and recognition in video, Springer-Verlag London Limited, Princeton, 2011, pp. 323–351.

[20] M.A. Turk and A.P. Pentland, Face recognition using eigenfaces, *Proc IEEE* (1991), 586–591.

[21] P. Belhumeur, J. Hespanha and D. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7) (1997), 711–720.

[22] M.S. Bartlett, J.R. Movellan and T.J. Sejnowski, Face recognition by independent component analysis, *IEEE Trans Neural Network* **13**(6) (2002), 1450–1464.

[23] M. Sharkas and M.A. Elenien, Eigenfaces vs. Fisherfaces vs. ica for face recognition: A comparative study, in: *9th International Conference on Signal Processing (ICSP)*, Beijing, China, (2008), 914–919.

[24] L. Sirovich and M. Kirby, A low-dimensional procedure for the characterization of human faces, *J Optical Soc Am A* **4**(3) (1987), 519–524.

[25] T. Jebara, 3D pose estimation and normalization for face recognition, Center for Intelligent Machines, McGill Univer-

sity, undergraduate Thesis May 1996.

[26] I.J. Cox, J. Ghosn and P.N. Yianilos, Feature based face recognition using mixture distance, in: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (1996), 209–216.

[27] R. Brunelli and T. Poggio, Face recognition through geometrical features, in: *Proceedings of the Second European Conference on Computer Vision*, G. Sandini, ed., Vol. 588, Lecture Notes In Computer Science, London, UK: Springer-Verlag, 1992, pp. 782–800.

[28] R. Cendrillon and B.C. Lowell, Real-time face recognition using eigenfaces, in: *Proceedings of the SPIE International Conference on Visual Communications and Image Processing* **4067** (2000), 269–276.

[29] C. Zhang and Z. Zhang, Boosting-based face detection and adaptation, Sams Python, Chapter 1, 2010, pp. 1–8.

[30] R. Lienhart and J. Maydt, An extended set of haar-like features for rapid object detection, in: *International Conference on Image Processing (ICIP)*, Rochester, 2002.

[31] C. Zhang and Z. Zhang, A survey of recent advances in face detection, Microsoft Research, Redmond, 2010.

[32] D. Swets and J. Weng, Using discriminant eigenfeatures for image retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(8) (1996), 831–836.

[33] A. Martinez and A. Kak, PCA versus LDA, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(2) (2001), 228–233.

[34] R. Jafri and H.R. Arabnia, A survey of face recognition techniques, *Journal of Information Processing Systems* **5**(2) (2009), 41–68.

[35] R. Chellappa, C.L. Wilson and S. Sirohey, Human and machine recognition of faces: A survey, *Proceedings of the IEEE* **83** (1995), 705–740.

[36] J. Moody and C. Darken, Learning with localized receptive fields, in: *Proceedings of the 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton and T. Sejnowski, eds, Morgan Kaufmann, 1988, pp. 133–143.

[37] J. Moody and C. Darken, Fast learning in networks of locally-tuned processing units, *Neural Computation* **1** (1989), 281–294.

[38] A. Howell and H. Buxton, Towards unconstrained face recognition from image sequences, in: *Proceedings of the Second IEEE International Conference on Automatic Face and Gesture Recognition*, (1996), 224–229.

[39] S. McKenna and S. Gong, Combined motion and model-based face tracking, in: *Proceedings of British Machine Vision Conference*, Edinburgh, UK, (1996), 755–765.

[40] R.S. Feris, T.E. Campos and R.M. Cesar, Jr, Detection and tracking of facial features in video sequences, in: *Mexican International Conference on Artificial Intelligence (MICAI 2000)*, Lecture Notes in Artificial Intelligence, Acapulco, Mexico: Springer-Verlag **1793** (2000), 129–137.

[41] T.E. de Campos, R.S. Feris and R.M. Cesar, Jr, A framework for face recognition from video sequences using GWN and eigenfeature selection, in: *Workshop on Artificial Intelligence and Computer Vision*, Atibaia, Brazil, (2000).

[42] V. Kruger and G. Sommer, Affine real-time face tracking using a wavelet network, in: *ICCV'99 Workshop: Recognition, Analysis, and Tracking of Faces and Gestures in R Real-Time Systems*, Corfu, Greece, (1999), 141–148.

[43] A. Pentland, B. Moghaddam and T. Starner, View-based and modular eigenspaces for face recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition*, (1994), 84–90.

[44] T.E. de Campos, I. Bloch and R.M. Cesar, Jr, Feature selection based on fuzzy distances between clusters: First results on simulated data, in: *Proceedings of International Conference on Advances on Pattern Recognition-ICAPR'2000*, Lecture Notes In Computer Science. Rio de Janeiro, Brasil: Springer-Verlag **2013** (2001), 186–195.

[45] R. Duda and P. Hart, Pattern classification and scene analysis, New York, USA: Wiley, 1973.

[46] A.K. Jain, R.P.W. Duin and J. Mao, Statistical pattern recognition: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000), 4–37.

[47] V. Krüger and S. Zhou, Exemplar-based face recognition from video, in: *Computer Vision – ECCV 2002: 7th European Conference on Computer Vision Copenhagen*, Denmark, May 28–31, 2002 Proceedings, Part IV, A. Heyden, G. Sparr, M. Nielsen and P. Johansen, eds, ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 732–746.

[48] S. Zhou, V. Krueger and R. Chellappa, Face recognition from video: A condensation approach, in: *Proc of Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, Washington D.C., USA, (2002), 221–228.

[49] G.L. Marcialis and F. Roli, Fusion of face recognition algorithms for video-based surveillance systems, in: *Multisensor Surveillance Systems: The Fusion Perspective*, G.L. Foresti, C.S. Regazzoni and P.K. Varshney, eds, Kluwer, 2003, pp. 235–250.

[50] J. Steffens, E. Elagin and H. Neven, Person Spotter fast and robust system for human detection, tracking and recognition, in: *Proceedings, International Conference on Audio- and Video-Based Person Authentication* (1999), 96–101.

[51] T. Choudhry, B. Clarkson, T. Jebara and A. Pentland, Multimodal person recognition using unconstrained audio and video, in: *Proceedings, International Conference on Audio and Video-based Person Authentication* (1999), 176–181.

[52] M. De-la-Torre, E. Granger, R. Sabourin and D.O. Gorodnichy, Adaptive skew-sensitive ensembles for face recognition in video surveillance, *Pattern Recognition* **48**(11) (2015), 3385–3406.

[53] B. Raducanu, A. Bosaghzadeh and F. Dornaika, Multi-observation face recognition in videos based on label propagation, in: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on* (7–12 June 2015), 10–17.

[54] N.M. Khan, X. Nan, A. Quddus, E. Rosales and L. Guan, On video based face recognition through adaptive sparse dictionary, in: *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on* **1** (4–8 May 2015), 1–6.

[55] N. Hassanpour and L. Chen, A hierarchical training and identification method using Gaussian process models for face recognition in videos, in: *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on* **1** (4–8 May 2015), 1–8.

[56] Z. Huang, R. Wang, S. Shan and X. Chen, Face recognition on large-scale video in the wild with hybrid euclidean-and-riemannian metric learning, *Pattern Recognition (PR)* **48**(10) (2015), 3113–3124.

[57] M.U. Ragashe, M.M. Goswami and M.M. Raghuwansh, A face recognition technique for partial occluded faces in streaming video, *International Journal of Applied Engineering Research* **10**(20) (2015), 17860–17865.

[58] Z. Chen, H.B. Wu and S.P. Xia, A cooperative dual-camera system for face recognition and video monitoring, *Advanced Materials Research* **998–999** (Jul 2014), 784–788.

[59] L. An, B. Bhanu and S. Yang, Face recognition in multi-camera surveillance videos, in: *Pattern Recognition (ICPR), 2012 21st International Conference on* (11–15 Nov 2012), 2885–2888.

[60] O. Arandjelović, Colour invariants under a non-linear photometric camera model and their application to face recognition from video, *Pattern Recognition* **45**(7) (July 2012), 2499–2509.

[61] L. An, M. Kafai and B. Bhanu, Face recognition in multi-camera surveillance videos using dynamic Bayesian network, in: *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on* (30 Oct 2012–2 Nov 2012), 1–6.

[62] S. Damavandinejadmonfared, Kernel entropy component analysis using local mean-based k-nearest centroid neighbour (LMKNCN) as a classifier for face recognition in video surveillance camera systems, in: *Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference on* (30 Aug 2012–1 Sept 2012), 253–256.

[63] H. Fassold, P. Schallaue, J. Rosner and W. Bailer, Real time KLT feature point tracking for high definition video, GraVisMa, 2009.

[64] Y. Freund and R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* **55**(1) (Aug 1997), 119–139.

[65] R. Feris, Y. Tian and A. Hampapur, Capturing people in surveillance video, *Procedure IEEE Int Workshop Visual Surveillance* (2007), 1–8.

[66] E. Hjelmas and B.K. Low, Face detection: A survey, *Computer Vision and Image Understanding* **81** (2001), 236–274.

[67] Y. Abramson, B. Steux and H. Ghorayeb, YEF real-time object detection, in: *International Workshop on Automatic Learning and Real-Time (ALaRT)*, Siegen, (2005).

[68] G. Dashore and V.C. Raj, Efficient method for face recognition using principal component analysis (PCA), *International Journal of Advanced Technology & Engineering Research*, (2012).