



Feature redundancy approach to efficient face recognition in still images



Moses E. Ekpenyong¹  · Philip M. Wilson¹ · Aniekan S. Brown²

© Springer Nature Switzerland AG 2019

Abstract

This paper proposes the use of redundant features for efficient recognition of faces in still images using a novel system framework that offers a detailed systematic workflow for solving the facial recognition problem. It accepts still frontal face images, processes and represents salient facial features (face, eyes, nose, and mouth) using facial detection and extraction techniques. The extracted features are then modeled by an ensemble of self-organizing maps. The ensemble outputs are later reassembled into a single dataset consisting of a normalized image Euclidean distance matrix to enhance the search space for optimal convergence and classification. The feasibility of the framework is tested using an experiment facial database captured during the study, and three benchmark facial expression databases, namely the Extended Cohn–Kanade (CK+) database, the Japanese Female Facial Expressions database, and the MMI Facial Expression database. The results suggest that feature redundancy is indeed useful for efficient facial recognition, as the support vector machine classification recorded high accuracies across the various databases, with the normalized image Euclidean distance dataset producing the highest performance, when compared with the localized principal component analysis and unnormalized image Euclidean distance datasets. Furthermore, overall classification accuracy of above 99% was achieved for the experiment (nonexpressive still face) database, compared with the benchmark facial expression databases, which yielded slightly lower results. A future direction of this work is further improvement of the framework to robustly handle severe facial variations.

Keywords Biometric · Ensemble SOM · Image Euclidean distance · Face recognition · Redundant feature · Support vector machine

1 Introduction

The term “biometric” describes the application of modern statistical methods for measuring biological traits. Hence, it should include distinctive physiological and behavioral characteristics for automatic recognition of a person. To ensure the selection quality of biometric characteristics for recognition purposes, certain properties are necessary. These properties include:

- *Robustness*—biometric traits should be sufficiently invariant over a period of time;

- *Distinctiveness*—biometric traits should uniquely discriminate any two individuals, i.e., should possess wide interclass variability;
- *Availability*—biometric traits should be universal or exist in everyone;
- *Accessibility*—biometric traits should be easy to acquire.

The science of biometrics has used several approaches suitable for automatic identification of a person (using personal biological traits and characteristics). The various connections between “biometrics” and “forensics” primarily hinge on the identification of these biological

✉ Moses E. Ekpenyong, mosesekpenyong@uniuyo.edu.ng; mosesekpenyong@gmail.com | ¹Department of Computer Science, University of Uyo, Uyo, Nigeria. ²Department of Sociology and Anthropology, University of Uyo, Uyo, Nigeria.



characteristics. The central question now rests on the utilization of biometric data to reliably identify persons from a scene, or on the contrary, exclude them with reliability within reasonable doubt. Typical personal characteristics often exploited as biometrics are the face, eyes, nose, mouth, chin, finger, palm, gait, iris, ear electroencephalogram (EEG), hand writing, and voice. The face is preferred over other biometrics because of the wide use of digital cameras via smartphones and close-circuit television (CCTV), as well as frequent transmission through the internet and mobile devices. In comparison with other biometrics, the face offers the following advantages, which explain its preference over other personal characteristics:

- *Nonintrusion*: The face can be captured from a distance without physical contact with the person(s) and can also be collected in a user-friendly manner, which is acceptable by the public, as compared with other biometrics such as fingerprints or the iris;
- *Easy biological natural recognition*: Humans, and even animals, recognize objects mostly by facial features. This facilitates authentication and authorization purposes, whereas such recognition systems based on fingerprints and iris would require additional professional skills;
- *Little or no cooperation*: Compared with other biometrics that require individual cooperation, the use of surveillance applications can identify an individual without active participation with the investigator.

Face recognition as a biometric method offers high accuracy and low intrusiveness [74], and has attracted tremendous interest from not only computer scientists but also neuroscientists and psychologists [70]. The fields of biometrics, access control, law enforcement, security, and surveillance systems are the various application areas of face recognition. Biometrics is generally concerned with discrimination of persons based on their physical or behavioral attributes. So far, biometric techniques have predominantly flourished in various sectors such as immigration and customs, and commerce, and are being successfully utilized for the purpose of assured identification. When comparing a facial image from a crime scene with a suspect's photograph (for instance), forensic experts pay close attention to morphologic–anthropologic features, in line with prescribed protocol. Biometric face recognition does not use morphologic–anthropologic features; instead, it extracts and compares abstract features, resulting in a quantitative similarity score. When above a predetermined threshold, the compared faces are accepted as being of the same person. Given a still image or video of a scene, the problem of face recognition is to identify one or more persons in the scene by using a stored database

of faces [16]. This is mainly a classification problem, and involves training of the face recognition system using images from known individuals and classifying new test images into one of the classes, which constitutes the main aspect of face recognition systems.

The face recognition problem can generally be formulated as follows: *Given a facial image, identify or verify one or more persons, using a (stored or enrolled) facial database.* Although existing information such as age, gender, race, expression, and speech may serve as means to narrow the search space or enhance the recognition process, the solution to this problem is often accomplished using the following steps: segmentation or detection, feature extraction, and recognition or verification. In identification, the input to the system is an unknown face, while in verification, the system must confirm or reject the claimed identity of an input face.

Two most prominent challenges in face recognition are the illumination and pose problems [31]. The illumination problem occurs when the same face appears differently due to a change in lightning, or the changes induced by illumination become more pronounced, resulting in an incorrect comparison or misclassification of the image identity. Difficulties due to illumination and pose variation are well discussed in literature [2, 9, 61]. To address the illumination problem, several approaches utilizing domain knowledge have been proposed [96]. These approaches exploit the assumption that all images belong to the same face class, and include heuristic image comparison as well as class- and model-based approaches [96]. To address the pose problem [8], researchers have proposed various methods for the rotation problem. The notable approaches include multi-image based, single-image based, and hybrid approaches.

Applications of face recognition technology have progressed from still controlled-format images, to uncontrolled video images, thereby presenting numerous technical challenges, and demanding a range of image processing techniques. Furthermore, variations in image quality, background clutter, availability of well-defined matching criterion, and the nature, type, and amount of input data present significant difficulties in categorizing deployed matching algorithms. Face recognition is therefore regarded as a pattern recognition problem, performed specifically on faces. It is used to classify or discriminate a face as either “known” or “unknown,” after due comparison with a collection of known faces. The accuracy of a face recognition system therefore lies in its ability to optimally learn unknown faces and discern a particular face using available face information. The discernment process is particularly intractable in that all faces appear similar and contain the same set of features such as eyes, nose, chin, and mouth, positioned roughly at

the same places. Hence, some major weaknesses associated with face recognition systems include [67]:

- *Relatively low accuracy* compared with the proven performance of a single feature, e.g., fingerprint and iris recognition;
- *Robustness against fraud*—the relative ease with which many systems can be compromised or defeated;
- *Existence of inadequate constraints* to deal with image variability that could further complicate recognition (rotation in depth and broad lighting changes as well as a personal change in appearance).

Currently, no recognition system can claim to handle all these issues effectively, and we also are not claiming a solution. In particular, not much research has gone into making face recognition robust to the effect of ageing, although in general, constraints on the application scenario and capture situation are often exploited to limit the amount of variance in face image samples, which are required to be scaled algorithmically. Today, our ability to quickly recognize familiar faces and subtle facial gestures continues to be a source of great interest to anthropologists, psychologists, neuroscientists, and doctors, because discovering whether face recognition is a specialized human ability may provide new insights into how our cognitive mind functions. This paper is significant because it exploits *redundant facial features* for efficient facial recognition. Specifically, the introduction of localized principal components and training, as opposed to global training, is proposed to help minimize the effect of poor illumination and eliminate the unusual outlier effects inherent in most recognition algorithms. This process assists in reversing the *curse of dimensionality*. To reduce the effect of noise, a Gaussian blur (or Gaussian smoothing) is applied. The Gaussian smoothing operator is a two-dimensional (2D) convolution operator used to “blur” images for removal of noise and unwanted detail.

The remainder of this paper is structured as follows: Sect. 2 extensively discusses developments in face recognition approaches; Sect. 3 presents the proposed system framework with a practical methodological workflow to accomplish the various phases of the framework; Sect. 4 implements the proposed framework and reports on the results obtained from an empirical facial database and three benchmark facial databases; Sect. 5 concludes the paper and summarizes its contributions.

2 Developments in face recognition approaches

Automated facial recognition is a quickly developing technology. Its early stages of development date back to the 1960s and 1970s [12, 24, 64]. The first semiautomated

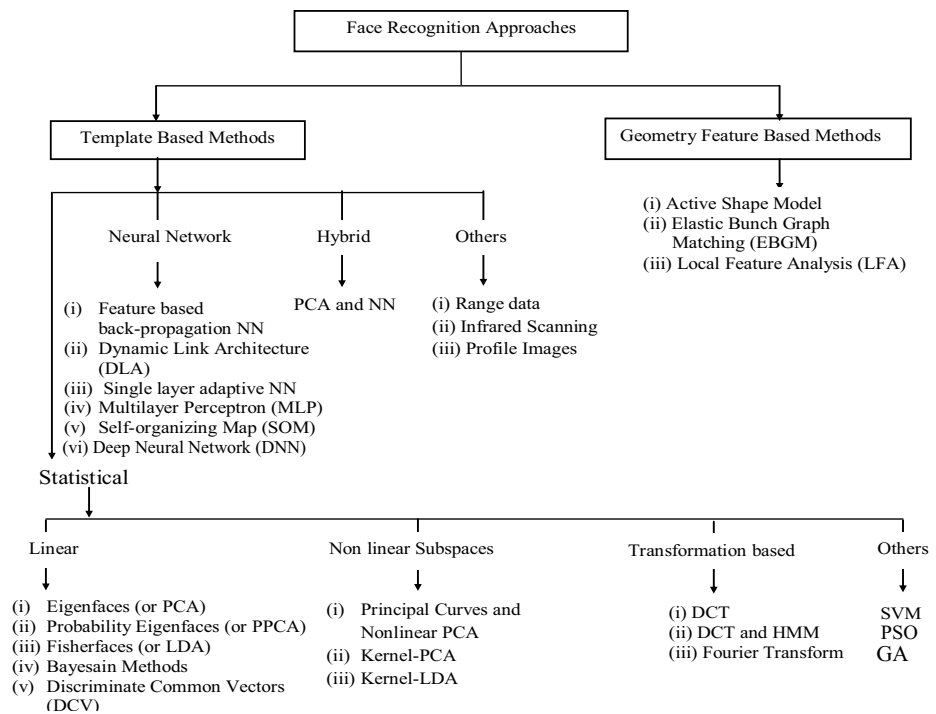
system for facial recognition required an administrator to locate salient features such as the eyes, ears, nose, and mouth on photographs, before figuring out the distances and ratios to a common reference point, then comparing these with a reference dataset. Face recognition therefore measures and matches unique features for the purposes of identification or authentication. Often leveraging a digital capture tool, facial recognition software can detect faces in images, quantify their features, then match them against templates stored in a database. However, as scientific research into evolving robust face recognition algorithms deepened, three main categories emerged [97]:

- *Holistic matching*, where the entire face region is used as raw input to the recognition system. Here, one of the most widely explored representations of the face region is the eigenpicture [37], usually based on principal component analysis (PCA);
- *Feature-based (structural) matching* first extracts local features such as the eyes, nose, and mouth, including their locations, as well as local statistics (geometric and/or appearance) before feeding them into a structural classifier;
- *Hybrid methods* use both local features and the whole face region to recognize a face.

Different approaches to face recognition abound. Most of the current face recognition algorithms can be categorized into two classes, namely image template-based and geometry feature-based methods. Developments in approaches to face recognition are presented in Fig. 1 [4, 7, 45].

Template-based methods compute the correlation between a face and one or more model templates to establish or estimate the face identity. Brunelli and Poggio [14] suggest that the optimal strategy for face recognition corresponds to template matching [13, 42], and compared a geometric feature-based technique with a template (matching)-based system. Their results showed very high accuracy, considering a database of 97 faces. Support vector machine (SVM), principal component analysis (PCA), linear discriminant analysis (LDA), kernel methods, and neural networks (NN) are statistical tools that have been developed to construct a suitable set of face templates [9, 26, 47, 59, 66, 78, 88]. Apart from statistical and neural network approaches, other approaches are hybrids that combine statistical pattern recognition techniques and neural network systems. Examples of hybrid approaches include the combination of PCA and radial basis functions (RBFs) or a neural network [77]. Transformation-based methods include the discrete cosine transform (DCT) [68], DCT

Fig. 1 Developments in face recognition approaches



and hidden Markov model (HMM), DCT and neural networks [75], Weber local binary image cosine transform (WLBI-CT) [33], and Fourier transform (FT). Khan et al. [33] proposed the Weber local binary image cosine transform (WLBI-CT) technique, to extract and integrate the frequency components of images obtained using the Weber local descriptor and local binary descriptor. The frequency components assist accurate classification of various facial expressions of the Japanese Female Facial Expressions (JAFFE), MMI Facial Expression, and Extended Cohn–Kanade (CK+) datasets. Results obtained demonstrate the superiority of this technique over contemporary techniques, as regards recognition rate and computation time. Among other methods, range, infrared-scanned, and profile images have also been used for face recognition. Evolutionary methods and their combinations have also been explored [33–35]. In Khan et al. [33, 34], a computationally intelligent method based on particle swarm optimization (PSO) utilizing features extracted from the texture and wavelet domain was developed. An informative wavelet subband was selected using the discrete Fourier transform (DFT) to compensate for the translational variance problem of the discrete wavelet transform (DWT). The results indicated robustness to illumination and variation in expressions, as the average accuracy obtained over the CK+, MMI, and JAFFE datasets yielded impressive performance of 98.6%, 95.5%, and 98.8%, respectively. Khana et al. [35] proposed an approach where the features collected contain local and global face information. The global features

were obtained using DCT, while the local facial features were obtained using the local binary pattern. The local and global features were then combined, resulting in an increase in features. PSO and genetic algorithm (GA) (PSO–GA) was then applied to eliminate the redundant features, resulting in optimized feature sets. Their system was evaluated using the ORL and Labelled Faces in the Wild (LFW) face databases, and the results obtained revealed a promising classification accuracy rate of 98% with a reduced number of features, making the system robust to variations such as facial expression changes, illumination effects, and occlusions.

This paper adopts a framework that exploits the hybrid form of template-based statistical methods [43]. The framework combines a modified form of PCA (LPCA), to extract localized features from the global image features, an ensemble of SOMs (E-SOMs), to cluster the local features, resulting in an image Euclidean distance matrix sufficient to establish spatial relationships between the pixels, and SVM, to discriminate the group of patterns assembled by the E-SOMs. Unlike the last three works, the proposed framework robustly handles variant illumination by first identifying salient facial features. Then, through the process of image binarization, background effects including illumination are completely dissolved, leaving only the desired features. This approach improves the processing speed (or computational time), avoiding the need for a compensation technique. A succinct discussion of notable template-based methods, viz. statistical methods, neural network-based approaches, and other

methods for face recognition, is presented in the following subsections.

2.1 Template-based methods

2.1.1 Statistical approach

Principal component analysis (PCA) The main idea behind PCA is to find the best set of projection directions in the sample space that maximizes the total scatter across all images [78]. This is accomplished by computing a set of eigenfaces from the eigenvectors of a total scatter matrix, S_t , defined as

$$S_t = \sum_{i=1}^N (x_i - m)(x_i - m)^T, \tag{1}$$

where N is the total number of samples, m is the mean face of the sample set X , and M is the total number of pixels in the images. For dimensionality reduction, $K : K < M$ eigenvectors, $U = [u_1, u_2, \dots, u_K]$, corresponding to the first K largest eigenvalues of S_t , are selected as eigenfaces. The reduced-dimension training samples, $Y = [y_1, y_2, \dots, y_N]$, can be obtained by the transformation $Y = U^T X$. Now, when a probe image x_t is presented for identification or verification, it is projected onto U to obtain a reduced vector $y_t = U^T x_t$. A response vector of length $C \in R(x_t) = [r_1, r_2, \dots, r_C]$ is then computed by measuring the distances from the probe to the nearest training samples from each class. The distance function between two vectors can then be expressed as

$$d(y_i, y_j) = y_i - y_j^2. \tag{2}$$

Hence, the desired class label for the probe image can be obtained by the minimum membership rule, thus

$$\mathcal{L}(x_t) = \arg \min_c r_c. \tag{3}$$

Linear discriminant analysis (LDA) The objective of LDA is to determine the subspace that best discriminates different face classes by maximizing between-class scatter, while minimizing the within-class scatter. The eigenvectors chosen by LDA provide the best separation among the class distributions; while PCA selects the eigenvectors which provide the best representation of the overall sample distribution, the eigenvectors for LDA can be obtained by computing the eigenvectors of $S_w^{-1} S_b$. Here, S_b and S_w are the between-class and within-class scatter matrices of the training samples, defined as

$$S_w = \sum_{i=1}^C \sum_{x_k \in C_i} (x_k - m_i)(x_k - m_i)^T, \tag{4}$$

$$S_b = \sum_{i=1}^C n_i (m_i - m)(m_i - m)^T, \tag{5}$$

where m_i is the mean face for the i th class and n_i is the number of training samples in the i th class. The LDA subspace is spanned by a set of vectors W , which maximizes the criterion J , defined as

$$J = \frac{\text{tr}(S_b)}{\text{tr}(S_w)}. \tag{6}$$

w can be constructed from the eigenvectors of $S_w^{-1} S_b$. In most image processing applications, the number of training samples is usually smaller than the dimension of the sample space, which leads to the so-called small-sample-size (SSS) problem due to the singularity of the within-class scatter matrix. To overcome the SSS problem, the following approaches are usually attempted: a two-stage PCA + LDA (PLDA) approach [72], the Fisherface method [9], or discriminant component analysis [95]. In all cases, the higher-dimension face data are projected onto a lower-dimension space using PCA, then LDA is applied to the PCA subspace.

Discriminative common vector (DCV) DCV solves the SSS problem of LDA by optimizing a variant of Fisher’s criterion. It searches for the optimal projection vectors in the null space of within-class scatter S_w [see Eq. (4)] and satisfies the criterion

$$J(W_{\text{opt}}) = \arg \max_{|W^T S_w W|=0} |W^T S_b W| = \arg \max_{|W^T S_w W|=0} |W^T S_t W|. \tag{7}$$

Hence, to find the optimal projection vectors in the null space of S_w , the face samples are projected onto the null space of S_w , in order to generate common vectors for each class, and then obtain the projection vectors by performing PCA on common vectors. A new set of vectors, referred to as discriminative common vectors, are obtained by projecting face samples onto the projection vectors. At the end, each class is represented by a single discriminative common vector. Among the two algorithms to extract the discriminant common vectors for representing each person in the training set of the face database, one algorithm uses the within-class scatter matrix of the samples in the training set while the other uses the subspace methods and the Gram–Schmidt orthogonalization procedure to obtain the discriminative common vectors.

These discriminative common vectors are used for the classification of new faces.

2.1.2 Neural network approach

Artificial neural networks (ANNs) are powerful tools for pattern recognition problems [25, 89]. The use of neural networks (NN) on faces has addressed several problems such as gender classification, face recognition, and classification of facial expressions. A few NN-based face recognition techniques and their implementation are discussed as follows:

Single-layer adaptive NN (SLANN) A SLANN (one for each person) for face recognition, expression analysis, and face verification was reported by Stonham [71]. The developed system was called a Wilke–Aleksander–Stonham recognition device (WISARD) and required, typically, 200–400 presentations to train each classifier—where the training patterns include translation and variation in facial expressions. A classifier was constructed corresponding to one subject in the database, and classification was achieved by determining the classifier with the highest response for a given input image. Much of the present literature on face recognition using neural networks presents results for only a small number of classes.

Multilayer perceptron (MLP) In Demers and Cottrell [21], for instance, the first 50 principal components of images were extracted and reduced to 5 dimensions using an auto associative neural network. The resulting representation was classified using a standard multilayer perceptron (MLP).

Self-organizing map (SOM) An SOM represents a set of high-dimensional data items as an ordered quantized two-dimensional image, where each data item is mapped to a unit node on the map, and the distances between the items in the map reflect the similarities between them [40]. In Lawrence et al. [47], a hybrid neural network solution which combines local image sampling using a self-organizing map (SOM) and a convolutional neural network was presented. The SOM provided a quantization of the image samples into a topological space (also nearby in the output space), thereby providing dimensionality reduction and invariance to minor changes in the image sample. The convolutional neural network provided partial invariance to translation, rotation, scaling, and deformation. Their recognizer provided a measure of confidence in its output, as the classification error approached zero when rejecting as few as 10% of the examples on a database of 400 images which contained a high degree of variability in expression, pose, and facial details. In Lefebvre and Garcia [49], a method that quantifies the visual similarity between

an image and a class model is presented. They proposed the labeling of a self-organizing map (SOM) to measure image similarity, by feeding local signatures associated with the regions of interest into the neural network. After learning, each neural unit is tuned to a specific local signature prototype, and facial recognition is performed using a probabilistic decision rule. The scheme produced very promising results for identification of faces when dealing with variation in illumination, facial pose, and expression. An evaluation of the automatic training samples selection method based on a self-organizing map (SOM) in face recognition systems is presented in [30]. They used PCA and SVM, compared the results with random (uncontrolled and controlled) training samples selection, and evaluated the recognition accuracy of each method.

Deep neural network DNNs are fast becoming the dominant technique in machine learning [6]. Their performance has been most successful for a wide variety of tasks, including image classification, speech synthesis, speech recognition, and face recognition. For a review and application of state-of-the-art deep learning approaches to face recognition, see [6, 60, 93]. Recently, Facebook employed DeepFace—a deep learning facial recognition system created by a research group at Facebook. DeepFace identifies human faces in digital images by employing a nine-layer neural net with over 120 million connection weights, trained on four million images uploaded by Facebook users [73]. The system is said to be 97% accurate, compared with 85% for the FBI's Next Generation Identification system. Such databases have raised serious ethical concerns, and could potentially be used (or abused) by governments for facial recognition purposes. The largest face recognition systems in the world with over 75 million photographs are actively used for visa processing and operate in the US Department of State. Recently, tools for collaborative software development have remarkably improved, with the biometrics community enjoying an active field of research that has generated algorithms and modalities suitable for real-world applications. Klontz et al. [39], for instance, motivate the need for more community-driven software in the field of biometrics, and have proposed OpenBR—an open-source face recognition software whose architecture considers front-view still images, to support open-source software development. More details of their research work can be found at: <http://www.openbiometrics.org>.

2.1.3 Hybrid approaches

Hybrid approaches use both statistical pattern recognition techniques and neural networks. An example is

the use of PCA and RBF [22]. They suggest the use of RBF on data extracted by discriminant eigenfeatures—where a hybrid learning algorithm was used to decrease the search space dimension in the gradient method, which is crucial for optimization of high-dimensional problems. First, they attempted to extract face features using PCA and LDA methods, and presented a hybrid learning algorithm to train RBF neural networks, such that the dimension of the search space was significantly decreased in the gradient method. In Thomaz et al. [76], PCA and RBF neural network were combined for face recognition, with PCA projecting the inputs of a face image over the principal components into a RBF network acting as a classifier. More hybrid approaches have emerged recently [4].

2.1.4 Other approaches

Other well-known template-based approaches include the use of range data, infrared scanning, and profile images. When using the *range data method* in the task of face recognition, image data are obtained by scanning the individual using a laser scanner. This system also provides depth information. Hence, the system processes three-dimensional data to classify the face images.

In the *infrared scanning method*, the face image is scanned using an infrared light source. Yoshitomi et al. [90] employed thermal sensors to detect the temperature distribution of a face, where the front-view face in the input image is normalized in terms of location and size, followed by measurement of the temperature distribution to obtain the locally averaged temperature and the shape factors of the face. The measured temperature distribution and locally averaged temperature were used separately as input data to feed a neural network, and supervised classification was used to identify the face. The disadvantage of visible-ray image analysis is that its performance is strongly influenced by lighting conditions, including variation of shadow, reflection, and darkness, which can be overcome by using the infrared method.

The *profile images method* was adopted by Lipocak and Loncaric [53] instead of frontal images. This method was used to represent the original and morphological properties (the outline that bounds the face and hair) derived from profile images. They took a graylevel profile image and thresholded it to produce a binary image representing the face region. Furthermore, they normalized the area and orientation of this shape using dilation and erosion, and simulated hair growth and haircut to produce two new profile silhouettes. From these three profile shapes, they obtained the feature vectors, and used the Euclidean distance measure to measure the similarity of the feature vectors derived from different profiles.

2.2 Geometry feature-based methods

Geometry feature-based methods exploit facial feature measures such as the distance between the eyes, or the ratio of the distance between the eyes and nose, but are significantly different from feature-based techniques in that they construct a topological graph using the facial features of each subject. Some such methods include the elastic bunch graph matching method, active shape model, and local feature analysis. Examples of this approach include the graph matching-based method (GMM) and feature-based PCA (F-PCA). A typical case of GMM is found in Lades et al. [44], where a dynamic link architecture for distortion-invariant object recognition which employs elastic graph matching to find the closed stored graph was presented. The objects are represented as sparse graphs with vertices labeled by geometrical distances. Individual faces are then represented by a rectangular graph, with each node labeled with a set of complex Gabor wavelet coefficients, called a jet. Only the magnitudes of the coefficients are used for matching and recognition. When recognizing a face in a new image, each graph in the model gallery is matched to the image separately, and the best match indicates the recognized person. With a database of 87 subjects and test images composed of different expressions and faces turned by 15°, the matching process was computationally expensive, taking roughly 25 s to compare an image with 87 stored objects when using a parallel machine with 23 transputers. A case of F-PCA is found in Cagnoni and Poggio [15]. They suggest a feature-based approach instead of a holistic approach to face recognition, and apply the eigenface method to sub-images (eye, nose, and mouth). Then, a rotation correction is applied to the faces to improve the results.

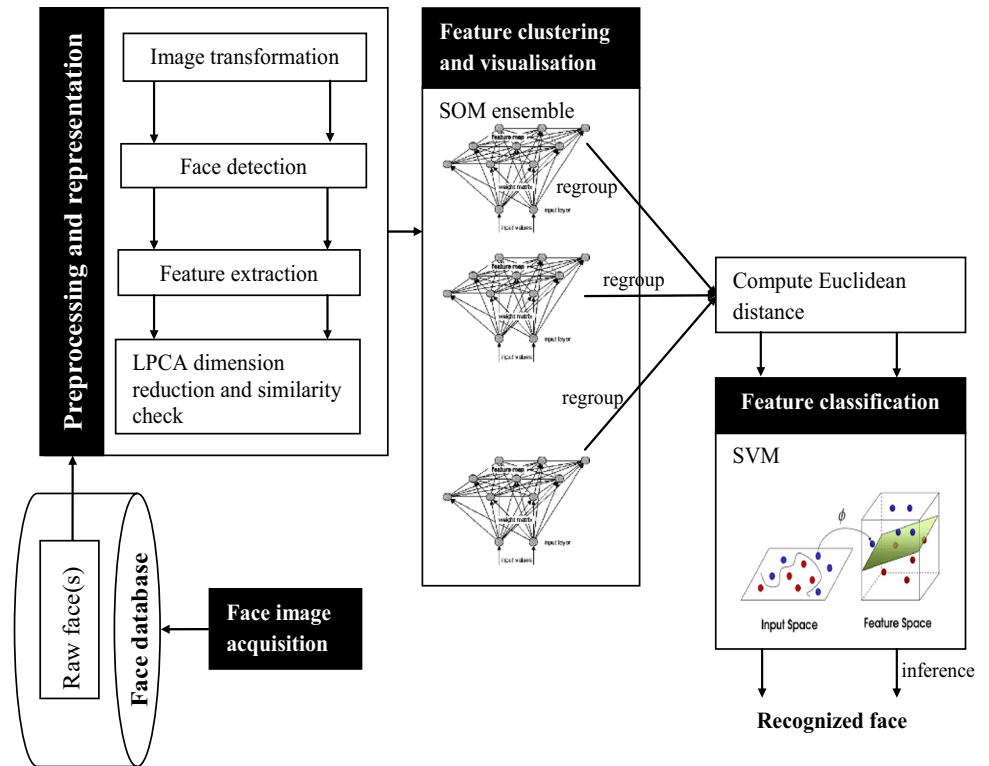
3 Proposed system framework

The proposed system framework as shown in Fig. 2 consists of four phases, namely *face image acquisition*, *pre-processing and representation*, *feature clustering/visualization*, and *feature classification*. These phases are discussed in the following subsections.

3.1 Face image acquisition

Before any image is processed, it must be captured by a camera and then converted into a form that is convenient for processing. This process is called image acquisition, and comprises [56]: (1) the energy reflected from the object of interest, (2) focusing of the energy by an optical system, and (3) measurement of the amount of energy by the sensor of the capturing device. On acquisition of

Fig. 2 Proposed face recognition system framework



the facial images, a database of raw faces is formed. These faces must undergo preprocessing and representation to enable efficient modeling and classification of facial features. During the preprocessing stage, the image is first converted into its numerical equivalent to enable easy manipulation of the facial features. In our framework, the steps to accomplish this process include: (1) image transformation, (2) face detection, (3) feature extraction, and (4) dimension reduction and similarity check.

3.1.1 Image transformation

Transforming color images to grayscale is one of the image preprocessing steps, and requires extensive knowledge about the color image. A pixel color in an image is a combination of three colors: red, green, and blue (RGB). The RGB color values are represented in three dimensions: XYZ, illustrated by the attributes of lightness, chroma, and hue. The quality of a color image depends on the color, represented by the number of bits that the digital device can support. A basic color image is represented as 8 bits, whereas a high-color image is represented as 16 bits, a true-color image as 24 bits, and a deep-color image as 32 bits. Hence, the number of bits determines the maximum number of colors supported by the digital device, and represents the pixel value in the color or grayscale image. A grayscale image is represented by luminance using 8-bit values, and the luminance of a pixel value of a

grayscale image ranges from 0 to 255. Consequently, converting a color image into a grayscale image implies converting the RGB values (24 bit) into grayscale values (8 bit). Most image processing techniques and software applications can convert color to grayscale images. However, image processing techniques or applications cannot handle the disparity in chromaticity and luminance. Although several linear and nonlinear techniques for transforming images have been discussed in literature, our goal is to eliminate color information that is unlikely to aid in the recognition process, as our attention is on specific facial features. Figure 3 shows a typical image conversion from RGB to grayscale.

On transforming the image from color to grayscale, cropping of the image in grayscale is necessary to exclude unnecessary background information that would affect the recognition process. This step is handled by a face detection algorithm [77, 84].

3.1.2 Face detection and feature extraction

Face detection is used to determine the presence and location of a face in an image. It requires the absolute distinction of a face from other patterns present in a scene, and involves the use of appropriate face modeling and segmentation techniques. The goal of facial feature detection is to locate the presence of facial features such as the eyes, eyebrows, nose, nostrils, mouth, lips, and ears, with the assumption that

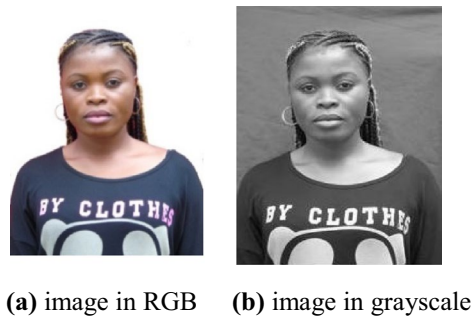


Fig. 3 Image conversion from RGB to grayscale

only one face exists in the image. The intended approach should therefore consider sources of variation of the facial appearance, such as the view geometry (pose), illumination (color and shadowing), imaging process (focus, effect, and resolution), and other factors such as occlusion [80]. Numerous methods have been proposed to resolve the variations that exist when detecting faces, and the prominent ones include (1) template-matching methods [19, 46]—used for locating and detecting a face—by computing the correlation between the input image and the face pattern; (2) feature-invariant approaches [50, 57]—used for detecting feature such as the eyes, mouth, ears, and nose; (3) appearance-based methods [37, 78]—used for face detection with eigenfaces; (4) neural network approaches [32]—used for robust modeling of facial features; and (4) information theory approaches [17, 51]. Despite the advancements in face detection, implementing these methods still presents great challenges due to the variations that exist in face images. Using the Viola–Jones algorithm, desired and prominent features of the face, such as the eyes, nose, and mouth, can be detected and extracted. Facial (feature) extraction is an important step in face recognition, but before this process, a face image must be registered or enrolled in a standard-size frame. To ensure that these features are localized within the face boundary, we also extracted the global facial feature (a superset of all the salient features in the face). Feature selection that aims to identify a subset of features among a possible target set of features that are relevant for predicting a response is an important precursor step to machine learning (ML). Feature selection (FS) is a process of choosing a subset of features that improves the performance of the ML method. Formally, for N data samples with M features in each, the FS problem is to find from the M -dimensional observation space \mathbb{R}^M , a subspace of m features \mathbb{R}^m that best predict the target. This is achieved by scaling down the number of features, to eliminate irrelevant, redundant, or noisy information from the feature set. FS primarily assists in alleviating the *curse of dimensionality*, and this in turn speeds up the learning task. In practice, it helps to optimize the data collection methodologies by identifying or sieving relevant

data, and serves as a key preprocessing step to improve predictions. The facial features selected for this experiment consist of four sets $\{F: \text{face}, E: \text{eyes}, N: \text{nose}, M: \text{mouth}\}$, where F is global (or redundant dataset) or a superset of all other facial features, and E , N , and M are subsets of F .

Feature redundancy involves a duplication of features or information that contributes to avoiding failure or errors. Applying this principle to our problem, we use the (global) face feature to optimize the search space of other (localized) facial features. Consequently, we hypothesize that: *feature redundancy is necessary and sufficient for efficient face recognition*. The feature redundancy problem can be formulated as follows [91, 92]: Given a dataset S including a set of relevant features S' and the set of relevant features F_i , $S_i^+ = \{F_j | F_j \in S', SU_{j,c} > SU_{i,c}\}$, $S_i^- = \{F_j | F_j \in S', j \neq i, SU_{j,c} \leq SU_{i,c}, F_j \in S_i^+\}$, and $S_i^{(RO)} = \{F_j | SU_{i,j} \geq SU_{j,c}, F_j \in S_i^-\}$. To achieve minimum redundancy and maximum relevance [5, 62], the facial features are ranked according to their correlations to the output target vector.

Definition 3.1 (*Predominant correlation*) The correlation between a relevant feature F_i and the class C is predominantly $S_i^{(RS)} = \emptyset$.

Definition 3.2 (*Redundancy*) A relevant feature F_i is redundant with respect to the existence of any feature F_j (also called the redundant subject) if $F_j \in S_i^{(RO)}$. Consequently, feature F_k (also called the redundant object) $\in S_i^{(RO)}$ is redundant with respect to the existence of F_i .

The Viola–Jones algorithm is considered as one of the first robust real-time face detectors being actively used in practice. Figure 4 presents images showing the detection of features considered in the experiment.

Table 1 presents the average three-dimensional (3D) datasets obtained from the various features after applying the Viola–Jones algorithm.

The Viola–Jones algorithm employed in this paper integrates three major algorithms: the *integral image*, *Adaboost*, and *cascading*. The integral image algorithm is used to efficiently compute the sum of intensity values in a simple rectangular subset of an image, and is defined as

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y'), \quad (8)$$

where $i(x', y')$ represents the intensity of a grayscale image at pixel (x, y) . The Viola–Jones algorithm exploits Haar-like features [87], and is defined as the difference between the sum of the intensities in the dark- and light-shaded regions of simple rectangular patterns. Now, let I and P denote an image and its corresponding pattern, both of the same

Fig. 4 Detection of facial features

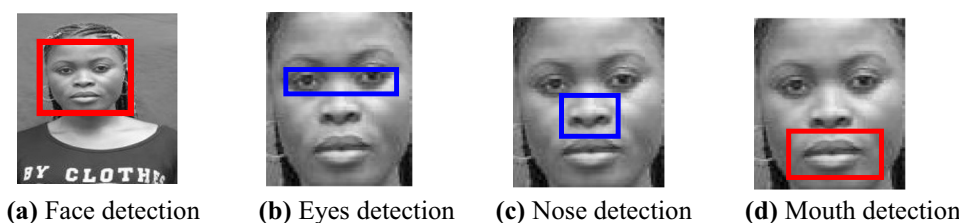


Table 1 Average datasets obtained from facial features

	Face	Eye	Nose	Mouth
Average 3D dataset	448 × 542 × 3	259 × 285 × 3	259 × 285 × 3	259 × 285 × 3

size, say $(N \times N)$. The features associated with pattern P are defined by

$$\sum_{1 \leq i \leq N} \sum_{1 \leq j \leq N} I(i,j) 1_{p(i,j) \text{ is white}} - \sum_{1 \leq i \leq N} \sum_{1 \leq j \leq N} I(i,j) 1_{p(i,j) \text{ is black}} \tag{9}$$

Hence, consider a rectangle with sides A, B, C , and D ; then the sum of the intensity of the pixels is calculated as

$$\sum_{(x,y) \in ABCD} i(x,y) = ii(D) + ii(A) - ii(B) - ii(C). \tag{10}$$

The computational advantage of the Viola–Jones algorithm is that it enables multifeature scale detection at no extra cost, as the same number of operations is required regardless of the size. Adaboost [83] is then used to select the features and train the classifier. The *weak learner* is designed to select the feature which best separates the weighted positive and negative training exemplars. A weak classifier $wcl(x, f, p, \theta)$ is defined as

$$wcl(x, f, p, \theta) = \begin{cases} 1, & pf(x) < p\theta \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where f is a feature from a set that spans various sizes of the Haar-like features, p is a polarity indicating the direction of the inequality, θ is the threshold, and x is a training subwindow of size 24×24 pixels. Figure 5 describes the Adaboost algorithm used in Viola–Jones.

The process of normalization is achieved by the Adaboost algorithm (Fig. 5). Here, there are two intensity features associated with the image, viz. a white background and a black foreground. The coloration intensity now falls between 0 and 255. A binarization of the normalized data (transforming the detected face features into numbers that fall between 0 and 1) is then performed before dimension reduction. Table 2 presents 15 binarized sample data for the first four (enrolled) faces.

Given: $(x_1, y_1), \dots, (x_N, y_N)$; $x_i \in X, y_i \in \{-1, 1\}$
 Initialise: $D_1(i) = \frac{1}{2n}, \frac{1}{2l}$; for $y_i = -1, 1$, respectively
 for $t = 1, \dots, T$:

1. Normalise the weights $D_t(i) = \frac{D_t(i)}{\sum_{i=1}^N D_t(i)}$
2. Select the best weak classifier w.r.t. weighted error:

$$\epsilon_t = \min_{f, p, \theta} \sum_{i: h(x_i, f, p, \theta) \neq y_i} D_t(i)$$
3. Define $h_t(x) = h(x_i, f, p, \theta)$; $f_t, p_t, \theta_t = \underset{f, p, \theta}{\operatorname{argmin}} \sum_{i: h(x_i, f, p, \theta) \neq y_i} D_t(i)$
4. Chooses $\alpha_t = \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
5. Update weights using $D_{t+1}(i) = D_t(i) \left(\frac{\epsilon_t}{1-\epsilon_t}\right)^{1-e_i}$
 where $e_i = 0$, if x_i is correctly classified, and $e_i = 1$, if misclassified.
 The final hypothesis $H(x) = \operatorname{sign}(\sum_{t=1}^T \alpha_t h_t(x))$.

Fig. 5 Adaboost algorithm used in Viola–Jones algorithm. (Source: [84])

3.1.3 PCA dimension reduction and similarity check

Principal component analysis (PCA)—a simple, unsupervised, nonparametric method—has been used successfully to extract relevant information from face datasets. PCA is advantageous for finding the maximum variance in the original space. The linear transformation maps the original d -dimensional space onto an s -dimensional feature subspace. Most discriminating features are then identified as feature vectors. Another advantage of using PCA is its ability to compress data patterns through dimension reduction, without loss of information, and it may also be exploited in image compression. PCA was employed in this paper for the purpose of dimension reduction. Assuming that there are some observations on p variables, X_1, X_2, \dots, X_p , the main goal of PCA is to find a subspace with dimension $q < p$ onto which the original data can be projected with minimal error. This process is known as dimension reduction. Mathematically, we can formulate the PCA problem thus: Given a p -dimensional random variable, $X = X_1, X_2, \dots, X_p$, and considering the transformation

$$Y = m + P(X - m) \tag{12}$$

Table 2 Sample binarized feature data

S. no.	Face 1				Face 2				Face 3				Face 4			
	Face	Eyes	Nose	Mouth	Face	Eyes	Nose	Mouth	Face	Eyes	Nose	Mouth	Face	Eyes	Nose	Mouth
1	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2	0.99	0.97	1.00	0.99	0.96	0.98	0.99	1.00	1.00	0.99	0.42	0.99	1.00	1.00	0.49	0.64
3	0.98	0.79	1.00	0.96	1.00	1.00	0.99	0.98	1.00	0.99	0.44	0.97	1.00	1.00	0.48	0.62
4	0.97	0.85	1.00	0.95	1.00	0.99	0.99	0.98	0.99	1.00	0.45	0.48	1.00	1.00	0.48	0.60
5	0.96	0.96	0.99	1.00	1.00	1.00	0.99	0.95	1.00	1.00	0.42	0.49	0.99	1.00	0.48	0.60
6	0.95	0.97	0.99	1.00	1.00	1.00	0.99	0.97	1.00	0.99	0.44	0.50	0.97	0.99	0.48	0.60
7	0.95	0.98	0.98	0.96	1.00	1.00	1.00	1.00	0.99	0.97	0.46	0.50	0.44	0.97	0.48	0.59
8	0.93	0.98	1.00	1.00	1.00	0.99	1.00	0.97	0.97	0.44	0.47	0.50	0.44	0.47	0.48	0.58
9	0.89	0.98	1.00	0.97	1.00	0.98	1.00	0.97	0.47	0.44	0.47	0.51	0.45	0.46	0.48	0.58
10	0.86	0.98	1.00	0.78	1.00	0.98	1.00	1.00	0.49	0.44	0.46	0.50	0.45	0.46	0.48	0.58
11	0.83	0.98	1.00	0.35	1.00	1.00	1.00	0.28	0.49	0.44	0.46	0.49	0.46	0.46	0.48	0.58
12	0.82	0.98	0.98	0.26	1.00	0.98	1.00	0.06	0.49	0.45	0.46	0.48	0.47	0.46	0.48	0.58
13	0.81	0.97	0.97	0.31	1.00	0.84	0.97	0.72	0.50	0.46	0.46	0.48	0.48	0.47	0.48	0.60
14	0.81	0.98	1.00	0.43	1.00	0.11	1.00	0.34	0.51	0.48	0.44	0.48	0.49	0.48	0.48	0.59
15	0.81	0.98	0.98	0.44	0.98	0.11	0.85	0.00	0.51	0.48	0.42	0.48	0.48	0.49	0.47	0.58

where $m \in \mathbb{R}^p$ and P is an orthogonal projection onto a subspace of dimension $q < p$, we seek m and P such that $\mathbb{E}(\|X - Y\|^2)$ is minimal. If $p=2$ and q is unity, then the problem becomes a search of the line r such that the square distance (measured as the orthogonal distance) between the distribution of X and the line is minimal. Let $\mathbb{E}(X) = \mu$, and the covariance matrix, $\text{Cov}(X) = S$. To obtain m and P , we recall some basic properties with trivial solutions:

1. If x is a p -dimensional random variable with $\mathbb{E}(X) = \mu$, then $\mathbb{E}(\|X - \mu\|^2) \leq \mathbb{E}(\|X - m\|^2) \forall m \in \mathbb{R}^p$.
2. If X is a p -dimensional random variable with $\mathbb{E}(X) = \mu$, and $\text{Cov}(X) = S$, and given A is a matrix $k \times p$, then $\mathbb{E}(AX) = A\mu$, and $\text{Cov}(AX) = ASA^T$.
3. A projection is an operator such that $p^2 = P$, and the projected subspace is $V = \text{Im}(P)$. If P is a projection, $I - P$ is thus also a projection. P represents an orthogonal projection if $P^T = P$. As such, the orthogonal projection onto the subspace V can be formulated thus: Suppose that A is a $(p \times q)$ matrix with orthogonal columns of base V (i.e., $A^T A = I_q$, the q -dimensional identity matrix); then $P = AA^T$.

Manipulating Y in Eq. (12), we obtain

$$\begin{aligned} \mathbb{E}(\|X - Y\|^2) &= \mathbb{E}(\|X - m - P(X - m)\|^2) \\ &= \mathbb{E}(\|(I - P)X - (I - P)m\|^2) \\ &\geq \mathbb{E}(\|(I - P)X - (1 - P)\mu\|^2). \end{aligned} \tag{13}$$

Considering the first property, for any projection P , the optimal choice that minimizes $\mathbb{E}(\|X - Y\|^2)$ is $m = \mu$. Now, consider that

$$\begin{aligned} \mathbb{E}(\|(I - P)X - (1 - P)\mu\|^2) &= \sum_{i=1}^p \mathbb{E}(((I - P)(X - \mu))_i^2) \\ &= \sum_{i=1}^p (\text{Cov}((I - P)X))_{ii} = \text{tr}(\text{Cov}((I - P)X)) \\ &= \text{tr}((I - P)S(I - P)^T). \end{aligned} \tag{14}$$

Based on the property that, for all A and B , AB and BA are well defined, and $\text{tr}(AB) = \text{tr}(BA)$, and that P is an orthogonal projection, then Eq. (14) becomes

$$\begin{aligned} \mathbb{E}(\|(I - P)X - (1 - P)\mu\|^2) &= \text{tr}((I - P)S(I - P)^T) \\ &= \text{tr}((I - P)^2 S) = \text{tr}((I - P)S) \\ &= \text{tr}(S) - \text{tr}(PS). \end{aligned} \tag{15}$$

Since $\text{tr}(S)$ is a known number, minimizing $\mathbb{E}(\|X - Y\|^2)$ is the same as maximizing $\text{tr}(PS)$. For simplicity, we restrict to a case where $q=1$. Hence, $P = vv^T$ is a vector of norm 1 ($v^T v = 1$), and S is symmetric and positive semidefinite, such that $S = C^T \Lambda C$, where C is orthogonal ($C^T C = I$) and Λ is a diagonal matrix with elements $\lambda_1, \lambda_2, \dots, \lambda_p$, ordered as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$. Then,

$$\text{tr}(PS) = \text{tr}(vv^T C^T \Lambda C) = \sum_{i=1}^p \lambda_i (C_v)_i^2. \tag{16}$$

Since v has norm 1, and C is orthogonal, then C_v has norm 1, i.e., $\sum_{i=1}^p (C_v)_i^2 = 1$. It then becomes clear that, if $\lambda_1 \geq \lambda_2$, the choice of C_v that maximizes Eq. (16) is

$$C_v = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}, \text{ such that, } C^T = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}.$$

Besides the maximum value of $\text{tr}(PS) = \lambda_1$, we observe that v is the eigenvector of S relative to λ_1 , and indeed

$$S_v = C^T \wedge C C^T \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} = C^T \wedge \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} = \lambda_1 C^T \wedge \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} = \lambda_1 v.$$

It has been proved that:

Theorem 1 Suppose Y is given by Eq. (1) with $\text{dim}(\text{Im}(P)) = 1$. Then,

$$\mathbb{E}(\|X - Y\|^2) \geq \mathbb{E}(\|X - \bar{Y}\|^2) = \text{tr}(S) - \lambda_1 \text{ with} \quad (17)$$

$$\bar{Y} = \mu + v v^T (X - \mu),$$

where v is a normalized eigenvector relative to λ_1 . Through this proof, it has also been proved that, if $Z = v^T X$ (a one-dimensional random variable), the vector v with $\|v\| = 1$, such that $V(Z)$ is maximal, is the eigenvector of S relative to λ_1 . More generally:

Theorem 2 Let Y be given by Eq. (1) with $\text{dim}(\text{Im}(P)) = q$. Then the projection \bar{P} that minimizes $\mathbb{E}(\|X - Y\|^2)$ under rule (1) is the orthogonal projection with $\text{Im}(\bar{P})$, the subspace generated by the eigenvalues of S relative to $\lambda_1, \dots, \lambda_q$. Then,

$$\mathbb{E}(\|X - Y\|^2) = \text{tr}(S) - (\lambda_1, \dots, \lambda_q) = \lambda_{q+1}, \dots, \lambda_p. \quad (18)$$

We can then consider projections onto subspaces of increasing dimension each time a new eigenvector is added and is orthogonal to the previous one. In this paper, a localized PCA (LPCA) learning approach is adopted. The algorithm concentrates on the salient details extracted by the Viola–Jones algorithm (focusing more on specific feature points and their neighboring region). Precisely, the LPCA replaces the binarized features with eigenfeatures for enhanced detail. LPCA learning is similar to guided training, and assumes a priori knowledge of the facial salient points $\{f\}$, which only feeds the network with the surrounding image patch for learning. In reality, the learning represents a specific feature in the localized subset of the

input facial distribution. So, if we are interested in facial features such as the eyes, nose, and mouth, then the learning statistics is constrained by the statistical distribution in these areas—to yield principal components for the feature points of interest. Furthermore, the localized approach converges more rapidly than the global approach [36], because the learning is a one-pass process and the number of trials is the same as the training set.

3.1.3.1 Localized PCA learning algorithm The procedure for implementing the LPCA learning is as follows:

Input Given $\{x_i\}_{i=1}^N$, of N training faces and $\{f'\} = \{f'_{ij} = (x'_{ij}, y'_{ij})\}_{i=1, j=1}^{N, Q}$ of feature points, there exist a total of $N \times Q$ feature points. If there are k PCA masks, then the face dataset has $N \times Q \times k$ values.

Step 1 Obtain the average matrix by:

- Selecting the patch x'_i from x_i and from $\{f'_{ij}\}$, where x'_i is a $p \times p$ array, $p < n$
- Find $\bar{x}' = \frac{1}{N} \sum_{i=1}^N x'_i$, across all N image samples

Step 2 Compute the covariance matrix C using the N local image patches, thus

$$C = \frac{1}{N} \sum_{i=1}^N (x'_i - \bar{x}')^T (x'_i - \bar{x}')$$

Step 3 Compute the eigenvectors and the associated eigenvalues from the covariance matrix C to obtain p^2 eigencomponents thus $C \bar{e}_i = \lambda_i \bar{e}_i$, where λ_i represents the eigenvalues and \bar{e}_i the associated eigenvector

The LPCA results in a 4×4 feature matrix, with each column representing a feature dataset for (face, eyes, nose, mouth). The feature dataset was then transformed into a unit vector, and compiled for all the faces used in the experiment. A scatter plot showing the distribution of the eigenfeatures for the enrolled faces is given in Fig. 6.

A total of 108 faces were used in the experiment. Participants willingly accepted to be captured for the experiment and gave their consent. Out of these faces, 100 were captured using a Nikon D40 6.1 MP Digital SLR camera, while the remaining 8 images (excluded for verification purposes) were captured using an Infinix Hot 4 8.0 MP. Figure 7 presents the enrolled or registered faces with their respective face numbers, as well as faces excluded for verification purposes.

Fig. 6 Distribution of eigenfeatures across faces

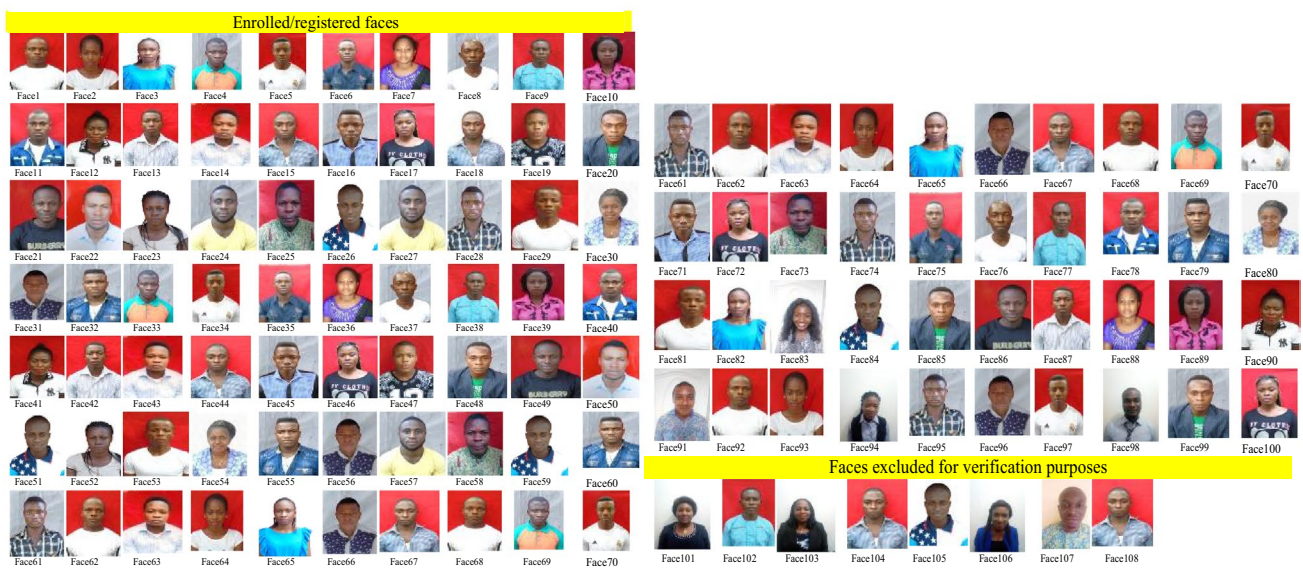
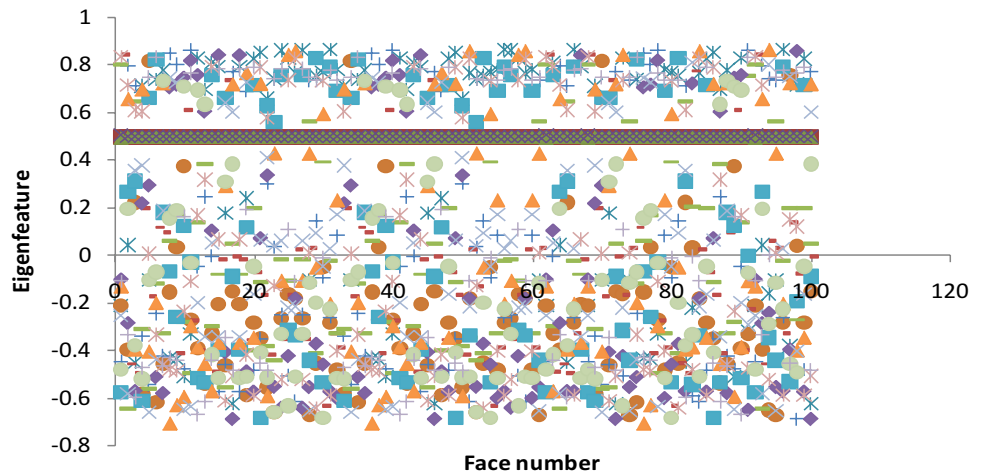


Fig. 7 Facial images used in the experiment

3.2 SOM clustering

Clustering has been widely used in digital image processing and face recognition [20, 28], with appreciable face recognition accuracies recorded. SOM provides a topology-preserving mapping from a high-dimensional space to map units. The property of topology-preserving means that the mapping preserves the relative distances between nodes. SOM is a neural network model and algorithm that implements a nonlinear projection from a high-dimensional space of input vectors to a low-dimensional array of neurons. The experiment is carried out on a high-dimensional and sparse space using a frontal (still) image database of faces. Each neuron has the same dimensionality as the input samples. Suppose that the dimensionality of the neurons is d , then each

neuron can be explained as a combination of linear LPCA eigenfeatures. A visualization of each neuron exploits the inner product of its weights vector with the sum of the eigenfeatures and mean face \bar{x} , thus [38]

$$\tilde{x} = \bar{x} + \sum_{j=1}^d x_{ij} p_j; \quad i = 1, 2, \dots, N, \tag{19}$$

where n is the dimension of all faces. SOM is applied to cluster and visualize the face data. It is a set of nodes, connected to one another via a rectangular or hexagonal topology. The rectangular SOM topology is adopted in this paper and is presented in Fig. 8. Here, a circle represents a node, and the connections between the inputs and nodes have weights, so a set of weights corresponds to each

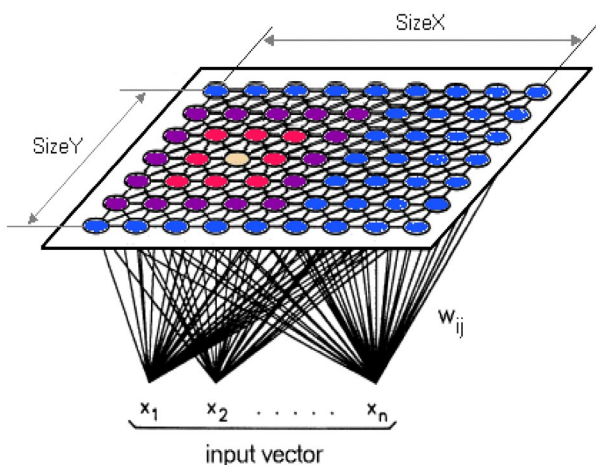


Fig. 8 A typical SOM topology

node. The set of weights forms a vector $M_{ij:i=1,\dots,k_p;j=1,\dots,k_m}$, usually called a neuron or codebook vector. In a rectangular SOM, the dimension of the codebook vector is equivalent to the number of the inputs $M_{ij} = \{m_{ij}^1, m_{ij}^2, \dots, m_{ij}^n\}$ [41]. During training, the input vectors are passed to the network, and a set of winning neurons computed. Each winning neuron corresponds to one or more input vector (i.e., the inputs are distributed among the map elements). After training the SOM network, its quality must be evaluated. Two errors (quantization and topographic) are usually calculated. The quantization error E_{QE} shows how well neurons of the trained network adapt to the input vectors. The quantization error, as shown in Eq. (20), represents the average distance between the data vectors X_p and the winning neurons $M_{c(p)}$ [41],

$$E_{QE} = \frac{1}{m} \sum_{p=1}^m \left\| x_p - M_{c(p)} \right\|. \tag{20}$$

The topographic error E_{TE} shows how well the trained network preserves the topology of the data, calculated as [41]

$$E_{TE} = \frac{1}{m} \sum_{p=1}^m u(x_p). \tag{21}$$

If the winning neuron vector x_p is closer to the neuron, the distance from x_p to it is smallest, regardless of the winning neuron, then $u(x_p) = 0$; otherwise, $u(x_p) = 1$.

3.3 Image Euclidean distance

A central problem in image recognition and computer vision is the determination of the distance between

images [86]. Considerable efforts have been made to define image distances that are intuitively reasonable. Notable metrics include: (1) tangent distance [69]—developed specifically for handwritten digit recognition, a metric that is locally invariant to seven transformations: line thickening, line thinning, X translation, Y translation, rotation scaling, and two hyperbolic transforms; (2) generalized Hausdorff distance [27]—useful for comparing shapes (binary images in particular); (3) fuzzy image metric [52]—primarily useful for image quality assessment, and specifically useful for compressed images. Among these metrics, the Euclidean distance has been the most commonly used, because of its simplicity [86]. The Euclidean distance of images is the distance between their corresponding points in the image space.

Let x, y be two images M and N , $x = (x^1, x^2, \dots, x^{MN}), y = (y^1, y^2, \dots, y^{MN})$, where x^{kN+1}, y^{kN+1} are the graylevels at location (k, l) . The Euclidean distance $d_E^2(x, y)$ is given by

$$d_E^2(x, y) = \sum_{k=1}^{IJ} (x^k - y^k)^2. \tag{22}$$

The distance measure is known to suffer from high sensitivity, even to slight perturbation or deformation, as a reasonable image metric should present smaller distances. Given the metric coefficients $g_{ij} : i, j = 1, 2, \dots, MN$, defined by

$$g_{ij} = \langle e_i, e_j \rangle \geq \sqrt{\langle e_i, e_i \rangle} \sqrt{\langle e_j, e_j \rangle} * \cos \theta_{ij}. \tag{23}$$

where $\langle \cdot \rangle$ represents the scalar product and θ_{ij} is the angle between e_i and e_j , the Euclidean distance between two images x, y becomes

$$d_E^2(x, y) = \sum_{i,j=1}^{MN} g_{ij} (x^i - y^i)(x^j - y^j) = (x - y)^T G (x - y), \tag{24}$$

where the symmetric matrix $G = (g_{ij})_{MN \times MN}$ is the metric matrix. For images with fixed size $M \times N$, every MN th-order symmetric and positive-definite matrix G induces a Euclidean distance.

The image Euclidean distance should however satisfy the following conditions:

1. The metric coefficient g_{ij} depends on the distance between pixels P_i and P_j . Let f represent the dependency, then $g_{ij} = f(|P_i - P_j|); i, j = 1, 2, \dots, MN$.
2. f is continuous, and g_{ij} decreases monotonically as $|P_i - P_j|$ increases.
3. f is universal.



Fig. 9 Sample faces from the CK+ Facial Expression benchmark database



Fig. 10 Sample faces from the JAFFE benchmark database

The above conditions imply that the image Euclidean distance is characterized by the following properties:

1. Slight perturbation yields small image distance. Hence, the stronger the deformation, the larger the distance.
2. The distance between two images remains invariant if the same translation, rotation, and reflection are applied to the images.
3. The metric (Euclidean distance) applies to images of diverse size and resolution.

After regrouping the parallel clusters obtained from the ensemble SOMs (E-SOMs), an image Euclidean distance matrix is formed. The Euclidean distance metric is chosen because it accounts for the spatial relationship between pixels and, hence, is robust against perturbation or deformation, and can easily be embedded into most image classifiers such as SVM, LDA, and PCA. The embedding proposed in this paper is rather efficient, as a standard transform of the Euclidean distances is required to obtain a domain-smoothing transform. Hence, this study demonstrates a consistent performance improvement over previous methods which hitherto ignored domain smoothing.

4 Results and discussion

The face recognition experiment was performed using an HP Intel® Core™ i5-9250, CPU 1.60 GHz laptop. To prove the feasibility of the proposed framework, it was tested across different face recognition databases. The first database is the experiment database of (nonexpressive) still faces collected for the purpose of this study. This database contains consented exposures of still faces with several duplicates of same faces (Fig. 7). The second database is the CK+ database [54]—a public benchmark dataset for action unit and

emotion recognition. The CK+ database comprises a total of 593 sequences across 123 subjects. Sequences range from neutral to peak expression. The third database is also a benchmark database known as the JAFFE database [55]. This database contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. The fourth database is another benchmark database called the MMI Facial Expression database [79]—an ongoing project that aims to deliver large volumes of visual data of facial expressions to the facial expression analysis community. The MMI Facial Expression database consists of over 2900 videos and high-resolution still images of 75 subjects, and is freely available to the scientific community. To maintain a test balance, 100 still faces were selected from the benchmark databases for comparison. Figures 9, 10, and 11 show sample faces from the benchmark databases.

4.1 Face similarity measure

A similarity check was performed to discover similar faces within the experiment and benchmark databases. We achieved this using the Pearson product-moment correlation coefficient ($r : -1 \leq r \leq 1$)—a measure of the linear association between two variables. The Pearson correlation coefficient of two (facial) eigenfeatures i and j can be obtained using Eq. (25).

$$\rho_{ij} = \frac{N \sum f_i f_j - (\sum f_i)(\sum f_j)}{\sqrt{[N \sum f_i^2 - (\sum f_i^2)][N \sum f_j^2 - (\sum f_j^2)]}} \tag{25}$$

We found that similar face features are contained in

$$\rho_{ij} = \begin{cases} 1; & \rho_i = \rho_j \\ [0, 1); & \rho_i \neq \rho_j \end{cases} \tag{26}$$

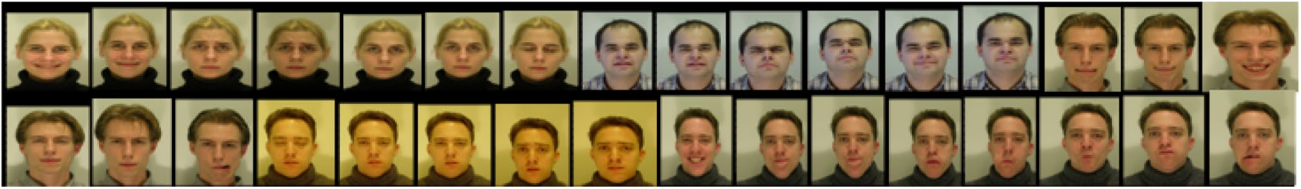


Fig. 11 Sample faces from the MMI Facial Expression benchmark database

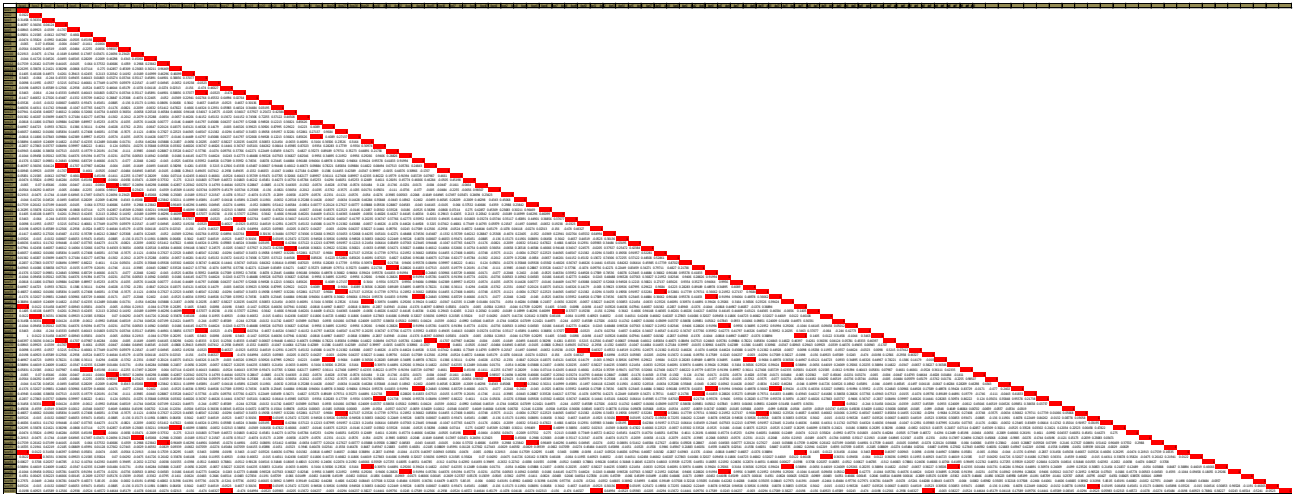


Fig. 12 Face similarity matches for the experiment database



Fig. 13 Face similarity matches for the CK+ database

A comparison across all the image data points resulted in a 100×100 (correlation) matrix. Correlation matrices showing similarity matches of facial pairs for the experiment and the CK+, JAFFE, and MMI benchmark databases are presented in Figs. 12, 13, 14, and 15, respectively.

A collection of similar face pairs is documented in Table 3. Table 3 confirms that, even though the benchmark databases (CK+, JAFFE, and MMI) contained

evidence from some subjects, few cases of similar pairs were observed (mainly neutral or with not too severe facial distortions) when compared with the experiment database. This proves the efficiency of our framework to identify similar facial expressions, as severe changes in facial features such as the eyes and mouth may contribute to image dissimilarities.



Fig. 14 Face similarity matches for the JAFFE database



Fig. 15 Face similarity matches for the MMI database

4.2 Feature clustering and visualization

Clustering data involves the grouping of data by similarities. The inefficiency of modeling unnormalized data can be seen in Fig. 16. The figure shows a scatter plot of a section of the unnormalized (raw) facial features generated by the Viola–Jones algorithm from the experiment database. We observe from the plot that no defined pattern clusters exist for the different facial features (face, eyes, nose, and mouth). Hence, we seek dimension reduction and clustering techniques to reveal the nature of patterns exhibited by the respective facial features. The LPCA method as described in Sect. 3.1.3.1 was used to locally extract eigenfeatures specific to the domain where the facial features are. To maintain *purity* of the extracted (redundant) features, an ensemble of SOMs was employed to separately cluster these local features. The initial condition for the E-SOMs was determined as follows: (1) transform the eigenfeatures obtained from the LPCA (a 4×4 matrix) representing the facial features into a (16×1) unit vector, and cluster them into four groups or segments, with each segment containing four eigenfeatures

describing a facial feature; (2) train the resulting eigenfeature dataset (16×100) using the required number of face models. Each model contained a Kohonen 10×20 SOM, and was trained using the batch unsupervised weight/bias algorithm (*trainbu*)—where weights and biases are only updated after all the inputs are fed into the network. The batch version of the SOM algorithm is computationally more efficient [40, 82].

During each training step, all the input data vectors are simultaneously used to update the weight vectors; (3) using the SOM results, transform the map into $m = 4$ clusters of minimum Euclidean distance; (4) update the corresponding weight vector using each cluster set or cluster. The updated weight vectors are computed thus:

$$m_i(t+1) = \frac{\sum_{j=1}^m n_j h_{ij}(t) \bar{x}_j}{\sum_{j=1}^m n_j h_{ij}(t)}, \quad (27)$$

where \bar{x}_j is the mean of the n data vectors in group j , and $h_{ij}(t)$ denotes the value of the neighborhood function at

Table 3 Similar face pairs for the different databases

Experiment database	Benchmark databases			
	CK+	JAFFE	MMI	
(62,1); (68,1); (92,1); (64,2); (93,2); (65,3); (82,3); (33,4); (69,4); (34,5); (70,5); (35,6); (75,6); (36,7); (88,7); (37,8); (76,8); (38,9); (77,9); (39,10); (89,10); (40,11); (78,11); (41,12); (90,12); (42,13); (87,13); (43,14); (63,14); (18,15); (44,15); (67,15); (45,16); (71,16); (46,17); (72,17); (100,17); (44,18); (67,18); (47,19); (48,20); (85,20); (99,20); (49,21); (86,21); (50,22); (52,23); (27,24); (57,24); (58,25); (73,25); (51,26); (59,26); (84,26); (57,27); (61,28); (74,29); (95,29); (54,30); (80,30); (56,31); (66,31); (96,31); (55,32); (60,32); (79,32); (69,33); (70,34); (75,35); (88,36); (76,37); (77,38); (89,39); (78,40); (90,41); (87,42); (63,43); (67,44); (71,45); (72,46); (85,48); (99,48); (86,49); (59,51); (84,51); (81,53); (80,54); (60,55); (79,55); (66,56); (96,56); (73,58); (84,59); (79,60); (74,61); (95,61); (68,62); (92,62); (93,64); (82,65); (96,66); (92,68); (100,62); (95,74); (99,85); (54,30); (80,30); (56,31); (66,31); (96,31); (55,32); (60,32); (79,32); (69,33); (70,34); (75,35); (88,36); (76,37); (77,38); (89,39); (78,40); (90,41); (87,42); (63,43); (67,44); (71,45); (72,46); (85,48); (99,48); (86,49); (59,51); (84,51); (81,53); (80,54); (60,55); (79,55); (66,56); (96,66); (92,68); (100,62); (95,74); (99,85)	(10,9); (44,43); (46,45); (83,82); (93,92)	(60,59); (61,59); (61,60); (62,60); (62,61); (65,64); (72,71); (74,73); (75,73); (75,74); (80,78); (81,78); (81,80); (83,82); (86,85); (87,85); (87,86); (96,95)	(19,18); (25,24); (28,17); (29,27); (31,30); (33,32); (49,42); (50,39); (55,54); (57,56); (59,58); (62,60); (66,65); (67,65); (67,66); (71,70); (74,73); (76,75); (78,77); (81,80); (86,85); (91,90); (92,90); (93,90); (92,91); (93,91); (93,92)	

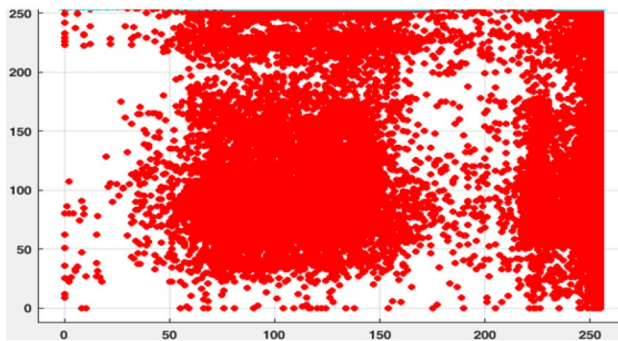


Fig. 16 A scatter plot showing a section of the unnormalized facial dataset from the experiment database

cluster unit j when the neighborhood function is centered on cluster unit i . A constant learning rate is adopted in the study, and the performance metric used to evaluate the SOM is the mean squared error (MSE). Proofs indicating the grid equilibrium, map stability, and convergence are provided in the following subsections.

4.2.1 Grid equilibrium

If we denote the mean field in \mathbb{R}^d of the SOM algorithm update by h , with a neighboring function ϕ , the stability of an equilibrium point depends on the eigenvalues of the gradient, h [18]. Hence, the l th component of h_i is obtained as

$$\begin{aligned} \frac{\partial h_i^l}{\partial x_j} &= \sum_{k \neq j} \phi(i, k) \mu(C_k(x)) \delta_{ij} e_l \\ &+ \sum_{k \neq j \in I} \phi(i, k) - \phi(i, j) \int_{\tilde{C}_k(x) \cap \tilde{C}_j(x)} (x_i^l - w^l) \\ &\times \left(\frac{1}{2} n_x^{k_j} + \frac{1}{\|x_k - x_j\|} \left(\frac{x_k - x_j}{2} - w \right) \right) f(w) \lambda_x^{k_j}(dw), \end{aligned} \tag{28}$$

where I represents a facial feature dataset, $\lambda_x^{k_j}(dw)$ is the Lebesgue measure on the Voronoi boundary [11] between x_k and x_j (precisely the median hyperplane), $n_x^{k_j}$ is the normal unit vector $\frac{x_j - x_k}{\|x_j - x_k\|}$ to the hyperplane, and $(e_i)_{1 \leq i \leq d}$ is the canonical basis of \mathbb{R}^d . The proof of uniqueness of the equilibrium point is found in [10]. The outputs of our SOMs were the Euclidean distances between the input vectors and the network’s weight vectors. At the end of the training, a regrouping process was performed. Our algorithm clustered the data such that a minimal total error criterion during regrouping was achieved. Suppose that $d_{n,k}^j(r)$ is the Euclidean distance between the n th vector of the k th

segment $(v_{n,k})$; then the closest centroid in the r th model per iteration i is

$$d_{n,k}^{(i)}(r) = d(v_{n,k}, C^{(i)}(r)) = (v_{n,k}, C^{(i)}(r))^\top (v_{n,k}, C^{(i)}(r)). \tag{29}$$

At the i th iteration, the total distance between the k th segment and the r th model is

$$d_k^{(i)}(r) = \sum_{n=1}^m d_{n,k}^{(i)}(r). \tag{30}$$

Hence, $d_k^{(m)}$ for the 100 faces trained in the experiment is $d_k^{(i)}(r) = \sum_{n=1}^{100} d_{n,k}^{(i)}(r)$.

The k th segment, S_k , is assigned to model j (SOM $_j$) to achieve a minimum total error of

$$j = \arg \min_{r = 0, \dots, R} \{ D_k^{(i)}(r) \} \Rightarrow S_k \in \text{SOM}. \tag{31}$$

Hence, the iterative sequence is defined by the following steps:

1. Retrain the models with new clusters, obtained from the previous iteration
2. Regroup the data using Eq. (30)
3. Check for termination criterion; if this criterion applies, exit; else, go to step 1.

The above algorithm has been proved to converge [85].

4.2.2 Map convergence and stability

First, we assume that $\phi(i, i) = 1$, and that for $i \neq j$, $\phi(i, j) < 1$. Consider a constant learning rate ϵ , and assume that the probability distribution has a positive p.d.f. f on a compact nonempty interior set. We define the distance in variation between two probability measures on \mathbb{R}^d , P and Q , by $\|P - Q\| = \sup_{A \subset \mathbb{R}^d} |P(A) - Q(A)|$, and arrive at the following results [63]:

Theorem 3 Assume:

1. There exists a set $I_0 \neq \emptyset$, such that at least there is one instance of k_0 for which $\phi(k_0, j) = 0, \forall j \in I_0$.
2. If $i \neq k_0$ then there exists $j \in I_0$ such that $\phi(i, j) > 0$.

So, the SOM algorithm $x^n(t)$ weakly converges with a uniform geometric rate: Let P_x^T represent the distribution of $x^n(t)$ when $x^0(t) = x$; there exists a unique stationary probability distribution such that

$$\exists \mathbb{R}_\epsilon \in \mathbb{R}^+, \exists r_\epsilon, 0 < r_\epsilon < 1, \left\| P_x^T - \pi^\epsilon \right\| \leq \mathbb{R}_\epsilon r_\epsilon^T. \tag{32}$$

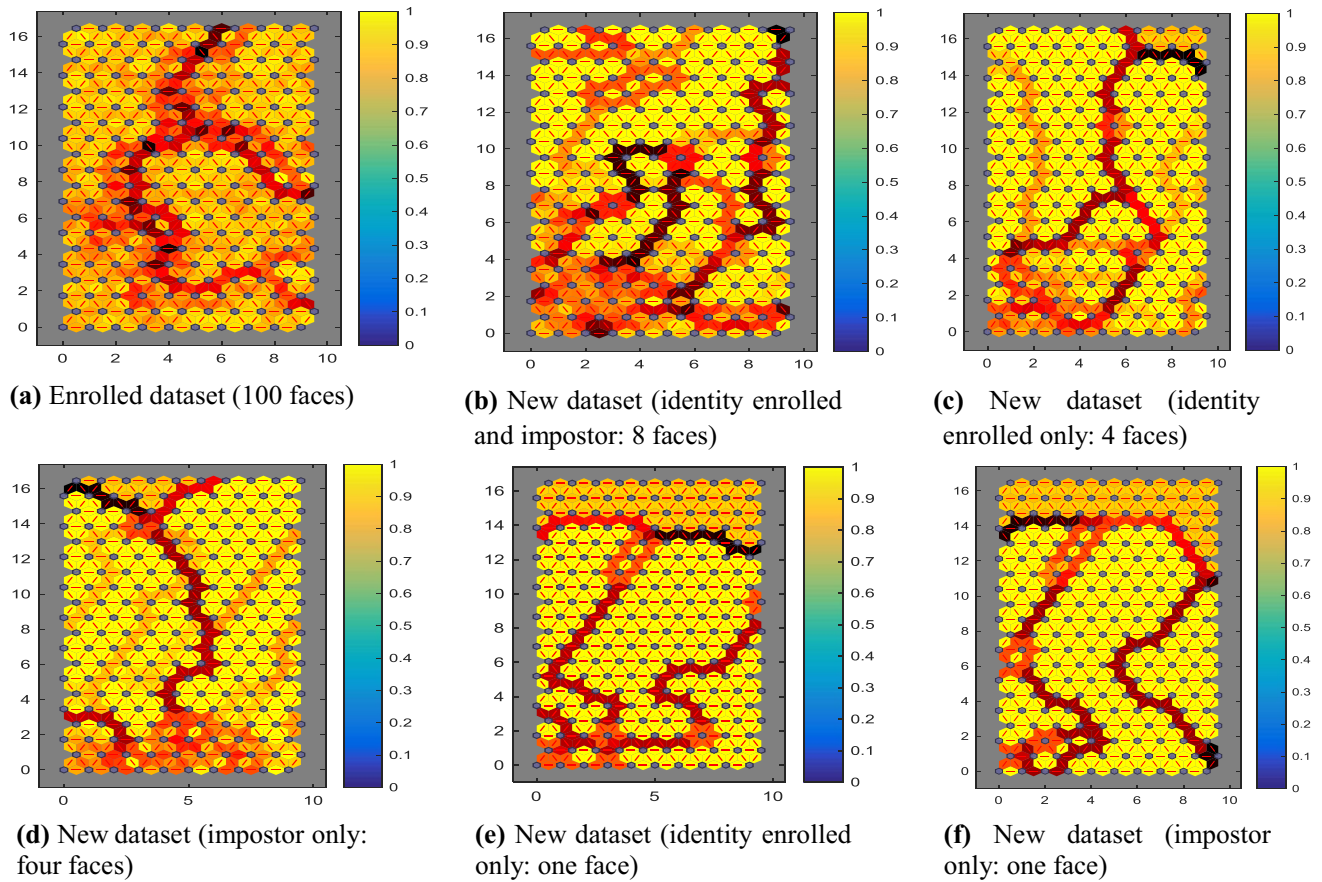


Fig. 17 U-matrices for enrolled and new face datasets

Remarks The assumptions on the topology of the unit set are satisfied in all useful cases, except that the neighborhood function must have a finite range.

Theorem 4 Assuming that the convergence to some stable state of the SOM is true, we expect that the expected values of $m_i(t + 1)$ and $m_i(t) : t \rightarrow \infty$ must be equal [41]. Mathematically,

$$\forall i, E_t \{ h_{c_i}(x(t) - m_i(t)) \} = 0, \tag{33}$$

where h_{c_i} is nonzero, $c = c(x(t))$ is the index of the winning node of input $x(t)$, and E_t represents an expected value operator over t . In the assumed asymptotic state, $t \rightarrow \infty$, $m_i(t)$ are independent of t and are denoted by m_i^* . If the expected values $E_t(\cdot)$ are generated as $\left(\frac{1}{t}\right) \sum_t (\cdot) : t \rightarrow \infty$, then

$$m_i^* = \frac{\sum_t h_{c_i}(t)x(t)}{\sum_t h_{c_i}(t)}. \tag{34}$$

Equation (34) is still implicit, since c depends on $x(t)$ and m_i , and must be iteratively solved.

4.3 SOM results

Using the experiment database, results of clustering (visualization) of the extracted LPCA (clustered SOM) dataset are presented in Figs. 17, 18, 19, 20, and 21. The SOM was found to converge at a maximum training cycle of 200, and was sufficient to guarantee a stable map. The U-matrix gives a quantitative summary of the topological relationships between neighboring neurons. The result represents the Euclidean distances between neighboring neurons. The chain of high values in the U-matrix represented by darker colors is an indication of prototypes that are far away from the groups, probably representing several outliers with very distinct face information. The U-matrix shows the relations between the neighboring neurons, and can be represented mathematically as

$$U\text{-matrix} = \begin{bmatrix} u_{11} & u_{11|12} & u_{12} & u_{12|13} & \cdots & u_{1k_p} \\ & \vdots & & & \ddots & \vdots \\ & & u_{k_n,1} & u_{k_n,1|12} & u_{k_n,3} & u_{12|13} & \cdots & u_{k_n,k_p} \end{bmatrix}, \tag{35}$$

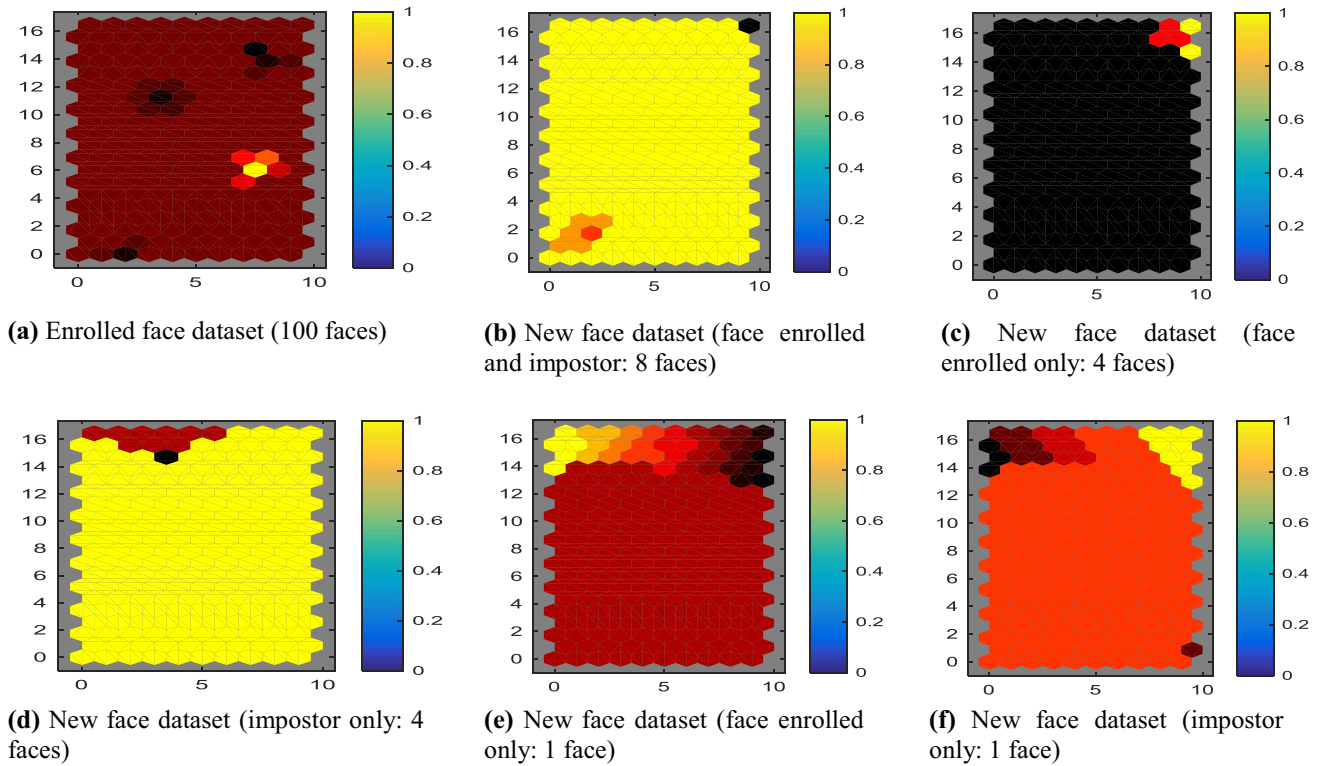


Fig. 18 Face component planes for enrolled and new (identity enrolled and impostor) dataset

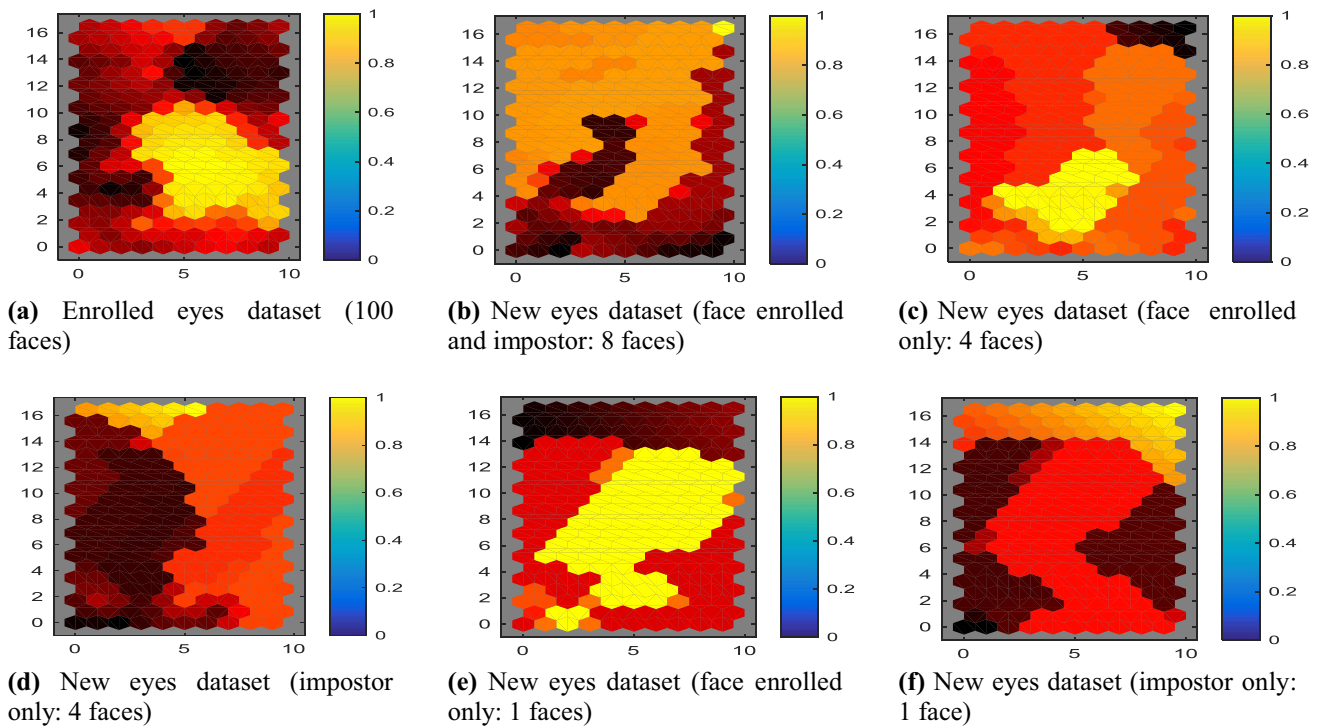


Fig. 19 Eyes component planes for enrolled and new (for identity enrolled and impostor) dataset

where the values of the elements $u_{ij|i(j+1)}u_{ij|(i+1)j}$ are the distances between neighboring neurons M_{ij} and $M_{i(j+1)}$ (M_{ij} and $M_{(i+1)j}$), respectively. Hence, the averages of smaller neighbors are computed with smaller numbers of elements. The SOM is colored by values of the U-matrix elements—a dark color between the neurons corresponds to large distance, while light color between the neurons signifies that the codebook vectors are close to each other in the input space. Light areas can be thought of as clusters, and dark areas as cluster separators. A color scale is displayed near the SOM, where the number denotes the values of the U-matrix elements and that of the distances between neighboring neurons. The U-matrices of the enrolled and new face datasets are presented in Fig. 17. From the U-matrices, it is easy to notice the formation of clear clusters with sharp boundaries, indicating distinct separation between the facial features. Figure 17a attempts to segment the enrolled facial dataset (consisting of 100 faces) into four clusters, representing the facial features selected for the experiment (face, eyes, nose, and mouth). Although the cluster segments appear darkened, indicating moderately large distances between the neurons, perfect convergence of our SOMs is here confirmed. Closer distances between neurons (lighter cluster segments) with closely similar patterns are observed for the SOM maps of the new dataset with (only) enrolled and (only) impostor faces, for four faces and one face, respectively (Fig. 17c–f). However, the distinction of the new dataset with enrolled and impostor faces (Fig. 17b) can be attributed to the quality of the device used to capture the impostor faces, which causes larger variances between the neurons, hence forcing the cluster segments to form subclusters (e.g., the last few rows of the map). More conspicuous are the top three rows of Fig. 17e, f, which form large distances between the neurons, compared with other segments within the maps.

In Fig. 18a–f, the facial component—a superset of the other facial features within the facial domain (i.e., eyes, nose, and mouth)—appears to be more dominant without a very useful pattern, as few cluster-like patches are noticed in the maps, but the maps of Fig. 18b, d present very light and clear segments compared with the maps of Fig. 18a, c, e, f, in which the segments appear darker. This implies that the extracted facial features of the enrolled and impostor faces differ greatly, and confirms the difference in quality of the capture devices. Again, the facial illumination (grayscale effect) and number of faces may have introduced larger distance between the neurons and marred the visibility of patterns, as useful patterns started emerging in the first three rows of the maps—for the new face dataset with only one enrolled face and only one impostor face (Fig. 18e, f), respectively.

Localizing facial features contributes to narrowing the iteration space and yields faster map convergence, with more informative patterns for each facial component plane. Figures 19, 20, and 21 present the component (feature) planes for the eyes, nose, and mouth datasets, respectively. In these figures, clear and sharp cluster shapes are obtained from the different datasets, but with distinct pattern dissimilarities, indicating inherent variability in the characteristics (or feature patterns). These feature component plane results confirm the distinctiveness of the biometric (face) characteristics and indicate wide interclass variability between the facial features.

The above E-SOMs reveal the existence of similar pattern formation for one face only (Fig. 21e, f), thus confirming the appearance of similarity of the facial features (in terms of shape) and the position of the same set of features (eyes, nose, and mouth) at roughly the same places.

The image Euclidean distance matrix obtained from the E-SOMs is symmetric and has dimension of 400×400 with zero diagonal. The full and partial symmetries of the image Euclidean distance matrix are presented as three-dimensional (3D) contour plots in Fig. 22.

As can be seen in Fig. 22, the image distances are well distributed with clearly defined clusters. The search consists of eight levels, two of which appear as thin films (or shadows of an existing film). To reduce the search space, Gaussian blurring (or smoothing) was applied to the unnormalized dataset. Gaussian blurring is a spatial filtering process for smoothing an image using a Gaussian function, and is widely used as a preprocessing stage in computer vision algorithms to enhance image structures at different scales. Applying Gaussian blur to an image is similar to convolving the image with a Gaussian function—a 2D Weierstrass transform [94]. The Gaussian smoothing operator is a 2D convolution operator used to “blur” images and remove unnecessary details and noise. The kernel exploited is an isotropic (circularly symmetric) Gaussian filter of the form [58]

$$\frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(x_i-\bar{x})^2}. \quad (36)$$

We observe in Fig. 23 that the blurring process, which results in a normalized image Euclidean matrix, indeed further compressed the search space into three layers while maintaining the fidelity of the data clusters.

4.4 SVM pattern classification

SVM utilizes the statistical learning theory—a framework for machine learning drawn from the fields of statistics and functional analysis—and can be applied to both classification and regression problems. In Farooq et al. [23], feeding hybrid features to an SOM for facial (expression) recognition

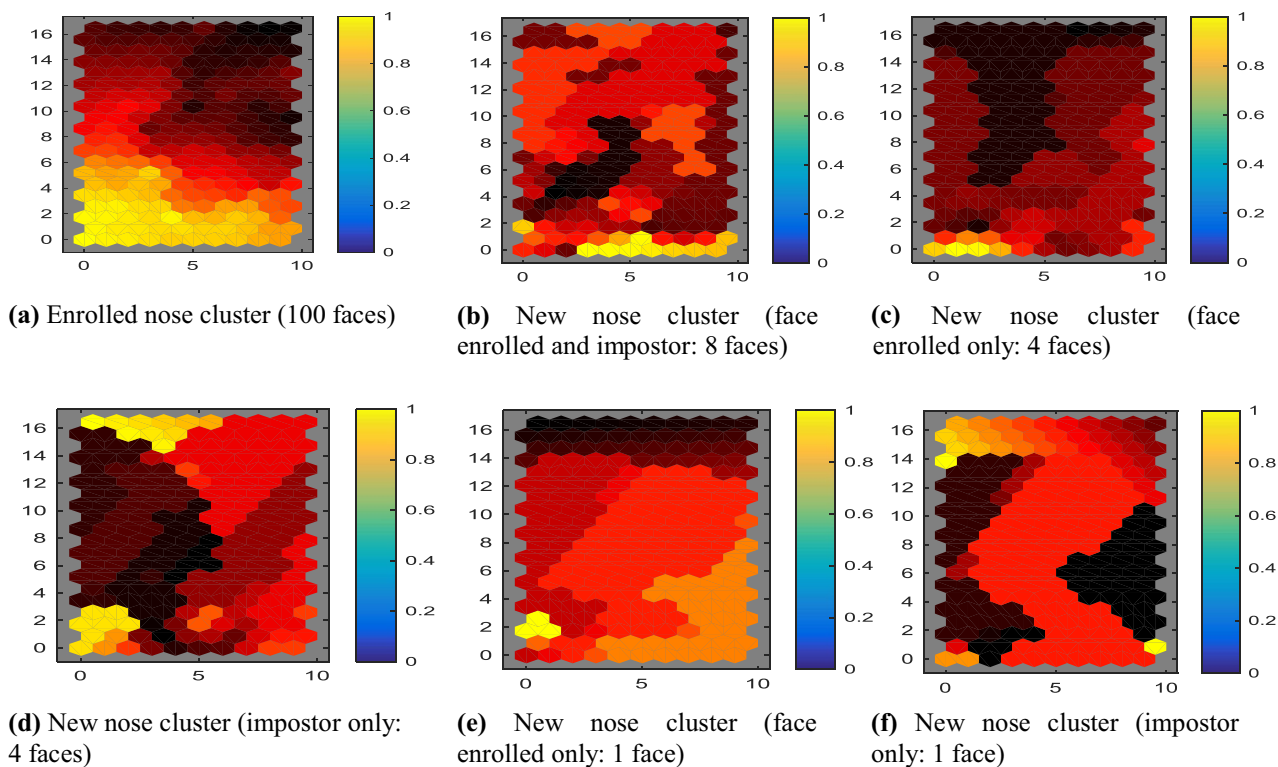


Fig. 20 Nose component planes for enrolled and new (for identity enrolled and impostor) dataset

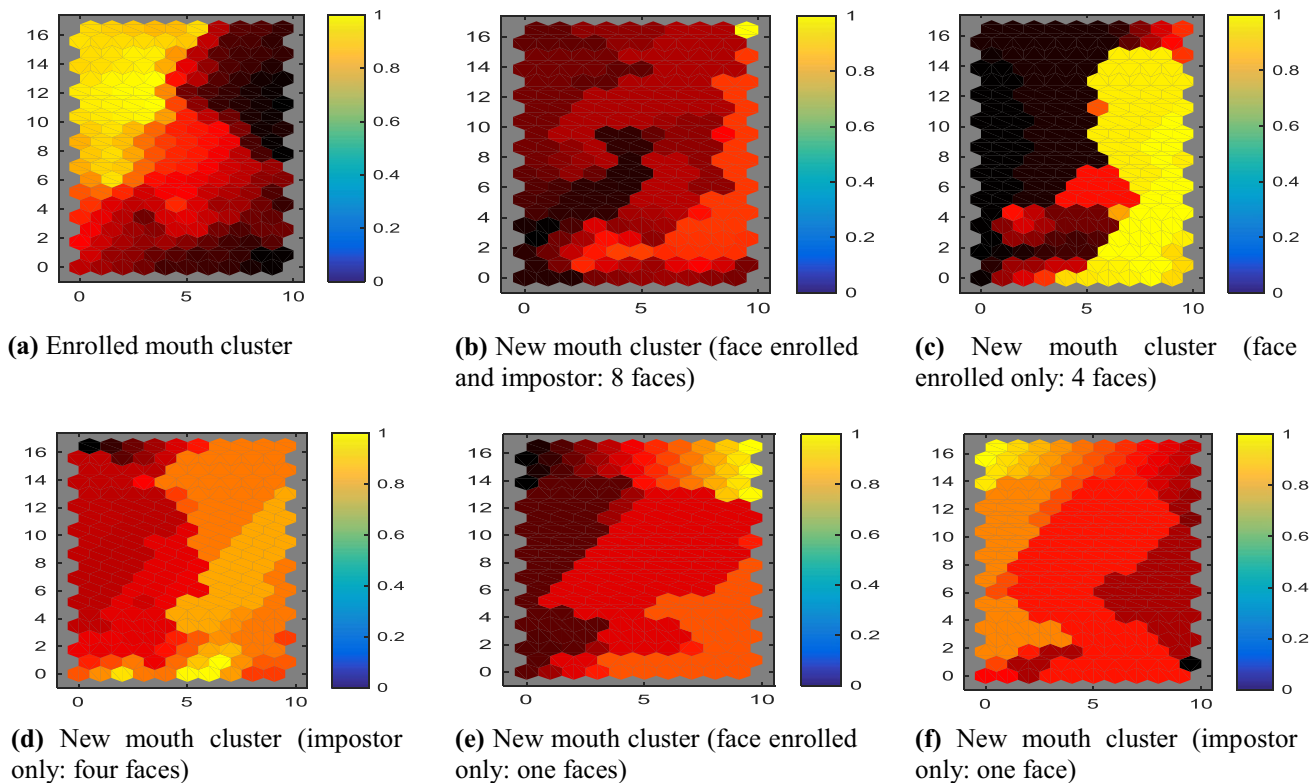


Fig. 21 Mouth component planes for enrolled and new (for identity in enrolled and impostor) dataset

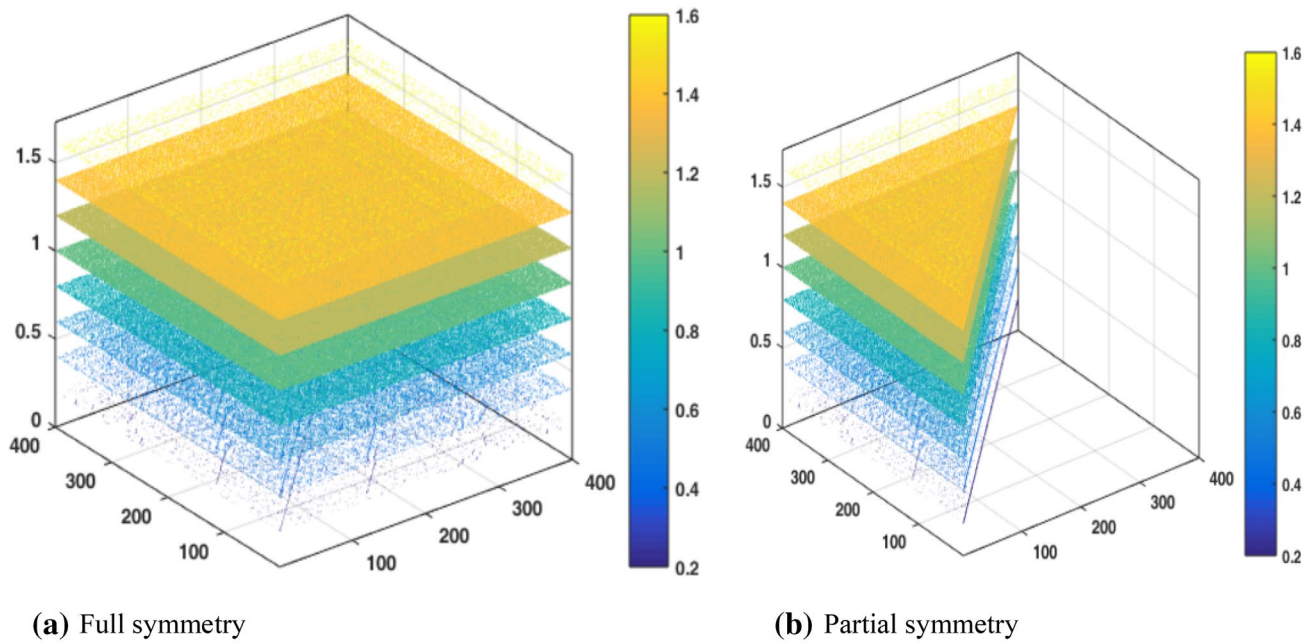


Fig. 22 3D contour plots of unnormalized image Euclidean distance matrix

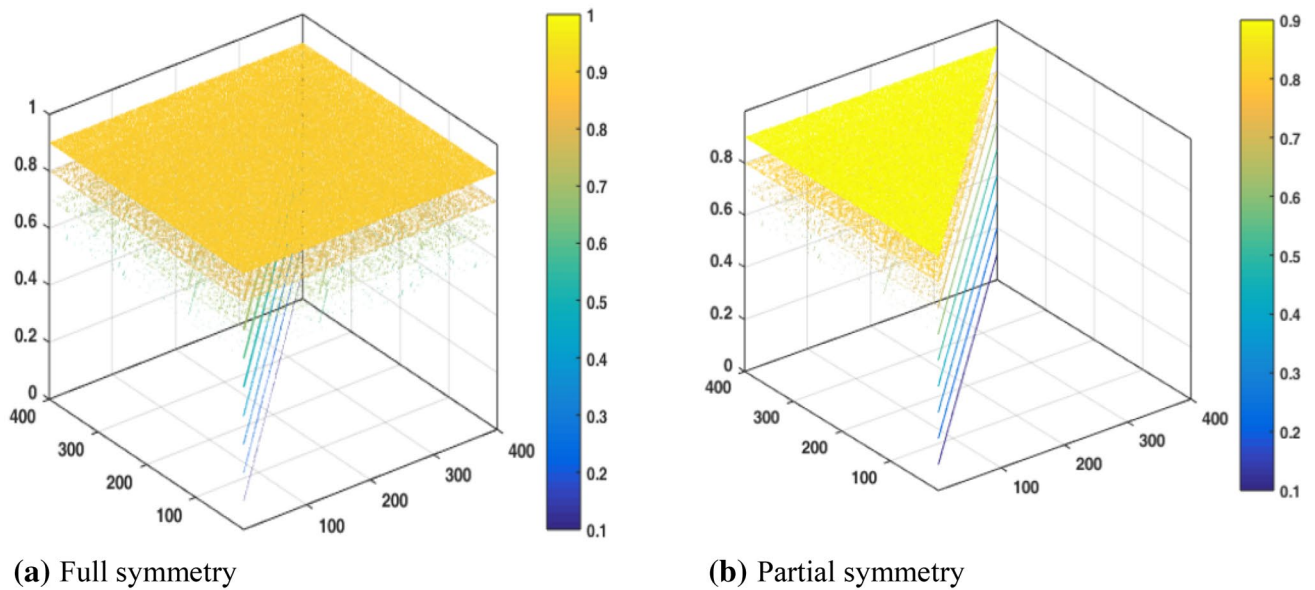


Fig. 23 3D contour plots of normalized image Euclidean distance matrix

achieved recognition accuracies of 96.81% and 96.55%, superior to state-of-the-art methods. Furthermore, the use of hybrid methodologies such as SOM and convolution neural network has proven to compare favorably with other methods; For instance, the use of the Karhunen–Loève transform in place of SOM and a multilayer perceptron in place of a convolution neural network yielded error rates of 5.3% versus 3.8%, whereas the multilayer perceptron performed very

poorly (40% versus 3.8%). Sayeed et al. [65] accomplished face detection using the Adaboost learning algorithm, and individually split up the faces into five segments: forehead, eyes, nose, mouth, and chin. Each segment was considered as a separate image, and eigenface (PCA) features of each segment were computed. The segmental Euclidean distance classifier was then used to match the test image with the stored one. They achieved an 88% success rate on the CG

Table 4 SVM classification for LPCA dataset

Dataset	Classification accuracy			
	Expt. database	CK+	JAFFE	MMI
LPCA (new face dataset {face enrolled + impostor} versus enrolled face dataset)	91.70	79.60	82.50	87.00
LPCA (new face dataset {face enrolled only} versus enrolled face dataset)	96.20	82.30	85.40	92.40
LPCA (new face dataset {impostor only} versus enrolled face dataset)	95.20	80.50	83.20	89.70

Table 5 SVM classification results for UIED dataset

Dataset	Classification accuracy			
	Expt. database	CK+	JAFFE	MMI
UIED (new face dataset {face enrolled + impostor} versus enrolled face dataset)	94.40	80.10	84.30	91.30
UIED (new face dataset {face enrolled only} versus enrolled face dataset)	97.30	85.50	88.60	93.20
UIED (new face dataset {impostor only} versus enrolled face dataset)	96.60	82.70	86.90	92.00

Table 6 SVM classification results for NED dataset

Dataset	Classification accuracy			
	Expt. database	CK+	JAFFE	MMI
NIED (new face dataset {face enrolled + impostor} versus enrolled face dataset)	99.70	86.00	90.20	94.60
NIED (new face dataset {face enrolled only} versus enrolled face dataset)	99.90	89.50	93.00	96.00
NIED (new face dataset {impostor only} versus enrolled face dataset)	99.80	87.20	92.70	94.90

database created from the databases of the California Institute and Georgia Institute. However, the performance of this approach on the ORL database using the same features was only 70%. Mixture-distance and raw images have been found to reduce the error rate by up to 73.90% [48]. Abbas et al. [1] also found that the cluster method with the squared Euclidean distance method produced a higher recognition rate than the Euclidean distance method, giving a human face recognition rate of 98%, compared with 95% when using the city-block distance method.

This paper, in addition to localizing the facial features—a case similar to Sayeed et al. [65], introduces a global (redundancy) facial domain or superset to guarantee search optimality. We define a cross-validation procedure—a technique used to evaluate predictive models by partitioning the original sample into a training set to train the model and a test set to evaluate it. Our training and test samples correspond to 85% and 15% of the original dataset, respectively. We adopt the radial basis function (RBF or Gaussian) kernel, commonly used in SVM classification, regarded as the squared Euclidean distance between two feature vectors, and can work without prior knowledge of the problem [29]. This Gaussian kernel is given as

$$K(\vec{x}, \vec{z}) = e^{-(\vec{x}-\vec{z})^2/2\delta^2}, \quad (37)$$

where \vec{x} and \vec{z} are feature vectors and δ is the kernel parameter. RBF kernels have proven excellent in facial classification [3, 81]. It acts as a prior filter, similar to a low-pass filter (an algorithm used in speech processing to eliminate the effects of environmental noise)—as it selects the smoothest solution from a range of supplied inputs. The classification accuracy is adopted as the performance metric for discrimination, given by

$$CLAcc(X_i) = \frac{C_{TPI}}{C_{TPI} + C_{FNI}}, \quad (38)$$

where C_{TPI} is the number of true-positive instances and C_{FNI} is the number of false-negative instances.

Tables 4, 5, and 6 compare the SVM classification results for the LPCA, unnormalized image Euclidean (UIED), and normalized image Euclidean (NIED) datasets, respectively, for the experiment and benchmark databases (CK+, JAFFE, and MMI). We found that the NIED dataset produced the least overall classification errors for the respective test cases, confirming our hypothesis that feature redundancy and NIED are indeed useful extraction techniques for efficient face recognition. Furthermore, the recognition rate for the experiment database was higher than the benchmark databases, as diverging facial expressions further reduced the

recognition accuracy [33, 34]. This implies that more images were easily recognized and classified in the experiment database—as this database contained more duplicates of same faces, compared with the benchmark databases, which contained few duplicates but with different variations in facial features. Hence, the greater the variations in facial posture, the wider the dissimilarity measure and the more difficult the recognition process.

5 Contributions to knowledge

This paper has made the following modest contributions to the growing field of face recognition:

Application of feature redundancy to facial recognition: Feature redundancy has flourished in the field of software engineering and fault tolerance, but not much has been heard (of it) in the field of facial recognition. This paper exploited feature redundancy to achieve an optimal facial features search within the domain of salient features. The computationally expensive nature of this approach was lightened by using the Viola–Jones algorithm—which concentrates only on the salient features—by eliminating unnecessary background details;

Feature dimension localization: This paper adopted localized dimension training as opposed to global dimension training—to eliminate unusual outlier effects inherent in most recognition algorithms, and ultimately reverse the “curse of dimensionality”;

Facial noise reduction: A 2D convolution operator (Gaussian blur) was used to “blur” the facial images—for removal of noise and unwanted detail (e.g., illumination effects). This method contributed to improving the recognition accuracy;

Use of normalized Euclidean distance: Euclidean distance is a widely explored metric in various fields of research and facial recognition. Due to its expensive computational cost, this paper adopted the normalized Euclidean distance matrix to distribute the facial features into ordered clusters with similar values, as well as to reduce the recognition search space by almost half. In this way, the recognition processes was quickened and the recognition accuracy improved;

Induced scalability: Given the success of this work in improving the facial recognition accuracy to above 99%, there are good prospects for this framework. Certainly, the proposed framework could be integrated into larger facial recognition systems, and is scalable to other recognition tasks when modified.

6 Conclusions and future work

Recent progress in face recognition technology has witnessed a series of standardizations and the application of robust techniques to leverage recognition tasks. Also, current trends in face recognition accuracy indicate that recognition errors have reduced dramatically, with appreciable false-acceptance and false-rejection rates. This paper has demonstrated the use of redundant features for efficient face recognition. Machine-learning techniques were applied to avert the curse of dimensionality and reduce the search space for fast solution convergence, and the use of the normalized image Euclidean distance of the image map greatly reduced the search space for faster recognition. Image variations can degrade recognition performance, as evidenced in the slightly worse performance results achieved for the benchmark facial expression databases. Although we still achieved high recognition rates for the benchmark databases using our framework, further improvements are required to handle severe facial variations.

Acknowledgements This research was supported by the Tertiary Education Trust Fund (TETFund) research grant (reference no. TETFUND/DESS/UNI/UJO/RP/VOL.III). We acknowledge the anonymous reviewers, whose critical comments improved the quality of this paper.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest.

References

1. Abbas EI, Safi ME, Rijab KS (2017) Face recognition rate using different classifier methods based on PCA. In: Proceedings of international conference on current research in computer science and information technology (ICCSIT), Bangladesh, pp 37–40
2. Adini Y, Moses Y, Ullman S (1997) Face recognition: the problem of compensating for challenges in illumination direction. *IEEE Trans Pattern Anal Mach Intell* 19(7):721–732
3. Agarwal V, Bhanot S (2018) Radial basis function neural network-based face recognition using firefly algorithm. *Neural Comput Appl* 30(8):2643–2660
4. Ahlawat D, Nehra V (2017) An efficient hybrid PC-SIFT-based feature extraction technique for face recognition. *Int J Signal Imaging Syst Eng* 10(5):237–247
5. Auffarth B, López M, Cerquides J (2010). Comparison of redundancy and relevance measures for feature selection in tissue classification of CT images. In: Proceedings of IEEE 10th international conference on data mining (ICDM), Sydney, Australia, pp 248–262
6. Balaban S (2015) Deep learning and face recognition: the state of the art. In: SPIE Defense + Security, 94570B-94570B. International Society for Optics and Photonics

7. Baron RJ (1981) Mechanisms of human facial recognition. *Int J Man Mach Stud* 15(2):137–178
8. Bayhan MS, Gökmen M (2008) Scale and pose invariant real-time face detection and tracking. In: *Proceedings of 23rd international symposium on computer and information sciences*, Istanbul, Turkey, pp 1–6
9. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Trans Pattern Anal Mach Intell* 19(7):711–720
10. Benaim M, Fort JC, Pages G (1998) Convergence of the one-dimensional Kohonen algorithm. *Adv Appl Probab* 30:850–869
11. Biedl TC, Grimm C, Palios L, Shewchuk JR, Verdonschot S (2016) Realizing farthest-point Voronoi diagrams. In: *Proceedings of the 28th Canadian conference on computational geometry (CCCG)*, Vancouver
12. Bledsoe WW, Chan H (1965). A man-machine facial recognition system-some preliminary results. Technical report PRI 19A, Panoramic Research, Palo Alto, CA
13. Brunelli R (2009) *Template matching techniques in computer vision: theory and practice*. Wiley Publishing, Hoboken
14. Brunelli R, Poggio T (1993) Face recognition: features versus templates. *IEEE Trans Pattern Anal Mach Intell* 15(10):1042–1052
15. Cagnoni S, Poggio A (1999) A modified modular eigenspace approach to face recognition. In: *10th international conference on image analysis and processing*, Venice, Italy, pp 490–495
16. Chellappa R, Wilson CL, Sirohey S (1995) Human and machine recognition of faces: a survey. *Proc IEEE* 83(5):705–741
17. Colmenarez AJ, Huang TS (1997) Face detection with information-based maximum discrimination. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, San Juan, Puerto Rico, USA, pp 782–787
18. Cottrell M, Fort JC, Pagès G (1998) Theoretical aspects of the SOM algorithm. *Neurocomputing* 21(1):119–138
19. Craw I, Tock D, Bennett A (1992) Finding face features. In: *Proceedings of 2nd European conference on computer vision*, Springer, London, pp 92–96
20. de Carvalho TB, Sibaldo MA, Tsang R, Cavalcanti GD, Tsang J, Sijbers J (2016) Pixel clustering for face recognition. In: *Proceedings of 5th Brazilian conference on intelligent systems (BRACIS)*, pp 121–126
21. DeMers D, Cottrell GW (1993) Nonlinear dimensionality reduction. In: Hanson D, Cowan J, Giles L (eds) *Advances in neural information processing systems 5*. Morgan Kaufmann, San Mateo, CA, pp 580–587
22. Er MJ, Wu S, Lu J (1999) Face recognition using radial basis function (RBF) neural networks. In *38th conference on decision and control*, Phoenix, Arizona, USA, pp 2162–2167
23. Farooq F, Ahmed J, Zheng L (2017) Facial expression recognition using hybrid features and self-organizing maps. In: *IEEE international conference on multimedia and expo (ICME)*, Hong Kong, pp 409–414
24. Fischler MA, Elschlager RA (1973) The representation and matching of pictorial structures. *IEEE Trans Comput* 100(1):67–92
25. Haykin S (1999) *Neural networks: a comprehensive foundation*. Prentice-Hall International, Upper Saddle River
26. Howell JA, Buxton H (1995) Invariance in radial basis function neural networks in human face classification. *Neural Process Lett* 2(3):26–30
27. Huttenlocher DP, Klanderman GA, Rucklidge WJ (1993) Comparing images using the Hausdorff distance. *IEEE Trans Pattern Anal Mach Intell* 15(9):850–863
28. Isa NAM, Salamah SA, Ngah UK (2009) Adaptive fuzzy moving K-means clustering algorithm for image segmentation. *IEEE Trans Consum Electron* 55(4):2145–2153
29. Janssen B (2008) *Support vector machines for binary classification and its applications*. Lecture notes, Indian Institute of Technology, Kharagpur
30. Jirka V, Féder M, Pavlovičová J, Oravec M (2014) Face recognition system with automatic training samples selection using self-organizing map. In: *Proceedings of 56th international symposium, ELMAR (ELMAR)*, Zadar, Croatia, pp 1–4
31. Kahraman F, Kurt B, Gokmen M (2007) Robust face alignment for illumination and pose invariant face recognition. In: *IEEE conference on computer vision and pattern recognition*. IEEE publishers
32. Kasar MM, Bhattacharyya D, Kim TH (2016) Face recognition using neural network: a review. *Int J Secur Appl* 10(3):81–100
33. Khan SA, Hussain A, Usman M (2018) Reliable facial expression recognition for multi-scale images using Weber local binary image based cosine transform features. *Multimed Tools Appl* 77(1):1133–1165
34. Khan SA, Ishtiaq M, Nazir M, Shaheen M (2018) Face recognition under varying expressions and illumination using particle swarm optimization. *J Comput Sci* 28:94–100
35. Khana SA, Usman M, Riaz N (2015) Face recognition via optimized features fusion. *J Intell Fuzzy Syst* 28(2015):1819–1828
36. King I, Xu L (1995) Using global PCA generated receptive fields for face recognition. In: *Proceedings of the world congress on neural network*, Washington, DC, USA, vol 2, pp 542–545
37. Kirby M, Sirovich L (1990) Application of the Karhunen–Loeve procedure for the characterization of human faces. *IEEE Trans Pattern Anal Mach Intell* 12(1):103–108
38. Kitani EC, Hernandez EDM, Thomaz CE, da Silva LA (2010) Visual interpretation of self organizing maps. In: *2010 11th Brazilian symposium on neural networks (SBRN)*, pp 37–42
39. Klontz JC, Klare BF, Klum S, Jain AK, Burge MJ (2013) Open source biometric recognition. In: *Proceedings of 6th IEEE international conference on biometrics: theory, applications and systems (BTAS)*, pp 1–8
40. Kohonen T (1998) The self-organizing map. *Neurocomputing* 21:1–6
41. Kohonen T (2014) *MATLAB implementations and applications of the self-organizing map*. Unigrafia Oy, Helsinki
42. Kour A (2015) Face recognition using template matching. *Int J Comput Appl* 115(8):10–13
43. Kumar D, Rai CS, Kumar S (2005) Face recognition using self-organizing map and principal component analysis. In: *Proceedings of international conference on neural networks and brain*, Beijing, China
44. Lades M, Vorbruggen JC, Buhmann J, Lange J, Von ser Mallburg C, Wurtz RP, Konen W (1997) Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans Comput* 42:300–310
45. Lanitis A, Taylor CJ, Cootes TF (1997) Automatic interpretation and coding of face images using flexible models. *IEEE Trans Pattern Anal Mach Intell* 19(7):743–756
46. Lanitis V, Taylor CJ, Cootes TF (1995) An automatic face identification system using flexible appearance models. *Image Vis Comput* 13(5):393–401
47. Lawrence S, Giles CL, Tsoi AC, Back AC (1997) Face recognition: a convolutional neural network approach. *IEEE Trans Neural Netw* 8(1):98–113
48. Lawrence S, Yianilos P, Cox I (1997) Face recognition using mixture-distance and raw images. In: *Proceedings of IEEE international conference on systems, man, and cybernetics, computational cybernetics and simulation*, Florida, USA, vol 3, pp 2016–2021
49. Lefebvre G, Garcia C (2008) A probabilistic self-organizing map for facial recognition. In: *Proceedings of 19th international conference on pattern recognition, ICPR*, Florida, pp 1–4

50. Leung TK, Burl MC, Perona P (1995) Finding faces in cluttered scenes using random labeled graph matching. In: Proceedings of 5th IEEE international conference on computer vision. IEEE Publishers, pp 637–644
51. Lew MS (1996) Information theoretic view-based and modular face detection. In: Proceedings of 2nd international conference on automatic face and gesture recognition, Killington, VT, USA, pp 198–203
52. Li J, Chen G, Chi Z (2002) A fuzzy image metric with application to fractal coding. *IEEE Trans Image Process* 11(6):636–643
53. Lipočak Z, Lončarić S (1999) A scale-space approach to face recognition from profiles. *Comput Anal Images Patterns* 243–250
54. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The extended Cohn–Kanade Dataset (CK+): a complete facial expression dataset for action unit and emotion-specified expression. In: Proceedings of IEEE workshop on CVPR for human communicative behavior analysis, San Francisco, CA, pp 94–101
55. Lyons ML, Akemastu S, Kamachi M, Gyoba J (1998) Coding facial expressions with Gabor wavelets. In: Proceedings of 3rd IEEE international conference on automatic face and gesture recognition, Nara, Japan, pp 200–205
56. Moeslund TB (2012) Introduction to video and image processing: undergraduate topics in computer science. Springer, London
57. Moghaddam B, Pentland A (1997) Probabilistic visual learning for object recognition. *IEEE Trans Pattern Anal Mach Intell* 19(7):696–710
58. Nixon MS, Aguado AS (2008) Feature extraction and image processing. Academic, Cambridge
59. Osuna E, Freund R, Girosi F (1997) Training support vector machines: an application to face detection. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, San Juan, Puerto Rico, USA
60. Parkhi OM, Vedaldi A, Zisserman A (2015) Deep face recognition. In: Proceedings of British machine vision conference, pp 1–12
61. Phillips PJ, Hyeonjoon M, Rizvi SA, Rauss PJ (2000) The FERET evaluation methodology for face recognition algorithms. *IEEE Trans Pattern Anal Mach Intell* 22(10):1090–1104
62. Radovic M, Ghalwash M, Filipovic N, Obradovic Z (2017) Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. *BMC Bioinform* 18(9):1–14
63. Sadeghi AA (2001) Convergence in distribution of the multidimensional Kohonen algorithm. *J Appl Probab* 38(1):136–151
64. Sakai T, Nagao M, Kanade T (1972) Computer analysis and classification of photographs of human faces. Kyoto University, pp 2–7
65. Sayeed F, Hanmandlu M, Ansari AQ (2011) Face recognition using segmental euclidean distance. *Def Sci J* 61(5):431–442
66. Scholkopf B, Smola AJ, Bernhard A (1998) Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput* 10(5):1299–1319
67. Senior AW, Bolle RM (2002) Face recognition and its applications: biometric solutions for authentication in an e-world. Kluwer Academic, Dordrecht
68. Shermina J (2011) Illumination invariant face recognition using discrete cosine transform and principal component analysis. In: International conference on emerging trends in electrical and computer technology (ICETECT), pp 826–830
69. Simard P, LeCun Y, Denker JS (1993) Efficient pattern recognition using a new transformation distance. In: Hanson S, Cowan J, Giles C (eds) *Advances in neural information processing systems*. Morgan Kaufman, San Mateo, pp 50–58
70. Sinha P, Balas B, Ostrovsky Y, Russell R (2006) Face recognition by humans: nineteen results all computer vision researchers should know about. *Proc IEEE* 94(11):1948–1962
71. Stonham TJ (1986) Practical face recognition and verification with WISARD. In: *Aspects of face processing*. Springer, Dordrecht, pp 426–441
72. Swets DL, Weng JJ (1996) Using discriminant eigenfeatures for image retrieval. *IEEE Trans Pattern Anal Mach Intell* 18(8):831–836
73. Taigman Y, Yang M, Ranzato MA, Wolf L (2014) Deepface: closing the gap to human-level performance in face verification. In: IEEE international conference on computer vision and pattern recognition, Columbus, USA
74. Tan X, Chen S, Zhou ZH, Zhang F (2005) Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft k-NN ensemble. *IEEE Trans Neural Netw* 16(4):875–886
75. Tayyab M, Zafar MF (2009) Face detection using 2D-discrete cosine transform and back propagation neural network. In: Proceedings of international conference on emerging technologies, Islamabad, pp 35–39
76. Thomaz CE, Feitosa RQ, Veiga A (1998) Design of radial basis function network as classifier in face recognition using eigenfaces. In: Proceedings of 5th Brazilian symposium on neural networks, pp 118–123
77. Tsai CC, Cheng WC, Taur JS, Tao CW (2006) Face detection using eigenface and neural network. In: Proceedings of IEEE international conference on systems, man, and cybernetics, Taipei, Taiwan, pp 4343–4347
78. Turk M, Pentland AP (1991) Eigenfaces for recognition. *J Cogn Neurosci* 3(1):71–86
79. Valstar M, Pantic M (2010) Induced disgust, happiness and surprise: an addition to the MMI facial expression database. In: Proceedings of 3rd international workshop on emotion (satellite of LREC): Corpora for Research on Emotion and Affect, Valletta, Malta, pp 65–70
80. Verma RC, Schmid C, Mikolajczyk K (2003) Face detection and tracking in video by propagating detection probabilities. *IEEE Trans Pattern Anal Mach Intell* 25(10):1215–1228
81. Verma RN, Jain K, Rizvi MA (2015) Efficient face recognition method using RBF kernel and genetic algorithm. In: Proceedings of international conference on computer, communication and control (IC4), Indore, India, pp 1–5
82. Vesanto J, Himberg J, Alhoniemi E, Parhankangas J (1999) Self-organizing map in MATLAB: the SOM toolbox. In: Proceedings of MATLAB DSP conference, Espoo, Finland
83. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR), Hawaii, USA
84. Viola P, Jones MJ (2004) Robust real-time face detection. *Int J Comput Vis* 57(2):137–154
85. Voitovetsky I, Guterman H, Cohen A (1997) Unsupervised speaker classification using self-organizing maps (SOM). In: Proceedings of 7th IEEE workshop on neural networks for signal processing, Amelia Island, FL, USA, pp 578–587
86. Wang L, Zhang Y, Feng J (2005) On the Euclidean distance of images. *IEEE Trans Pattern Anal Mach Intell* 27(8):1334–1339
87. Wang YQ (2014) An analysis of the Viola–Jones face detection algorithm. *Image Process Line* 4:128–148
88. Yang MH (2002) Kernel eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods. In: IEEE international conference on face and gesture recognition, Washington, pp 215–220
89. Yegnanarayana B (1999) *Artificial neural networks*. Prentice-Hall of India, New Delhi

90. Yoshitomi Y, Miyaura T, Tomito S, Kimura S (1997) Face identification using thermal image processing. In: 6th IEEE international workshop on robot and human communication, Sendai, Japan, pp 374–379
91. Yu L, Liu H (2003) Efficiently handling feature redundancy in high-dimensional data. In: Proceedings of the 9th ACM SIGKDD international conference on knowledge discovery and data mining, Washington, DC, pp 685–690
92. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. *J Mach Learn Res* 5(2004):1205–1224
93. Zafeiriou S, Zhang C, Zhang Z (2015) A survey on face detection in the wild: past, present and future. *Comput Vis Image Underst* 138:1–24
94. Zayed AI (1996) Handbook of function and generalized function transformations. CRC Press, Boca Raton
95. Zhao W (2000) Discriminant component analysis for face recognition. In: Proceedings of 15th international conference on pattern recognition, vol 2, pp 818–821
96. Zhao W (1999) Robust image based 3D face recognition. Ph.D. thesis, University of Maryland, College Park, USA
97. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A (2003) Face recognition: a literature survey. *ACM Comput Surv (CSUR)* 35(4):399–458

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.