International Journal on Artificial Intelligence Tools
Vol. 22, No. 1 (2013) 1250029 (30 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0218213012500297

UNSUPERVISED DISCOVERY OF VISUAL FACE CATEGORIES

SHICAI YANG
Institute of Systems Engineering, Southeast University, Nanjing, China
shicai.yang@gmail.com

GEORGE BEBIS
Department of Computer Science and Engineering, University of Nevada, Reno, USA
bebis@cse.unr.edu

MUHAMMAD HUSSAIN
Computer Science Department, College of Computer and Information Sciences,
King Saud University, Riyadh 11543, Saudi Arabia
mhussain@ksu.edu.sa

GHULAM MUHAMMAD
Computer Engineering Department, College of Computer and Information Sciences,
King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia
ghulam@ksu.edu.sa

ANWAR M. MIRZA
Computer Engineering Department, College of Computer and Information Sciences,
King Saud University, Riyadh 11543, Saudi Arabia
anwar.m.mirza@gmail.com

Received 30 January 2012
Accepted 10 May 2012
Published

Human faces can be arranged into different face categories using information from common visual cues such as gender, ethnicity, and age. It has been demonstrated that using face categorization as a precursor step to face recognition improves recognition rates and leads to more graceful errors.[1] Although face categorization using common visual cues yields meaningful face categories, developing accurate and robust gender, ethnicity, and age categorizers is a challenging issue. Moreover, it limits the overall number of possible face categories and, in practice, yields unbalanced face categories which can compromise recognition performance. This paper investigates ways to automatically discover a categorization of human faces from a collection of unlabeled face images without relying on predefined visual cues. Specifically, given a set of face images from a group of known individuals (i.e., gallery set), our goal is finding ways to robustly partition the gallery set (i.e., face categories). The objective is being able to assign novel images of the same individuals (i.e., query set) to the correct face category with high accuracy and robustness. To address the issue of face category discovery, we represent faces using local features and apply unsupervised learning (i.e., clustering). To categorize faces in novel images, we employ nearest-neighbor algorithms or learn the separating boundaries between face categories using supervised learning (i.e., classification). To improve face categorization robustness, we allow face categories to share local features as well as to overlap. We demonstrate the performance of the proposed approach through extensive experiments and comparisons using the FERET database.

Keywords: Face categorization; face recognition; local features; clustering, classification.

1. Introduction

There has been increased interest in employing different types of biometrics to robustly and reliably identify people in images and video. Face recognition is a key biometric technology with a wide range of potential applications both in government and private sectors. Despite of significant progress in the field of face recognition over the last decade,[2–4] building robust and reliable face recognition systems, especially in unconstrained environments, is still a challenging issue. Current research efforts involve extracting more powerful features and using high-resolution images, thermal imaging, and 3D models. A notable weakness of traditional face recognition systems, however, is that they represent all faces in a common lower dimensional space using dimensionality reduction techniques (e.g., Principal Component Analysis (PCA)[5] or Linear Discriminant Analysis (LDA)[6]) or local features (e.g., Local Binary Pattern (LBP)[7] or Weber Local Descriptor (WLD)[8]). On the other hand, human faces can be arranged into different face categories, for example, using information from various visual cues such as gender, ethnicity, and age. This information could be exploited to improve face recognition performance by developing category-specific face representation and recognition schemes.

There exists significant cognitive evidence supporting that humans utilize information from various visual cues for face recognition. It is well known, for example, that people are more accurate at recognizing faces of their own race than faces of other races (i.e., “other race effect”).[9,10] Recently, Phillips *et al.*[11] analyzed the other-race effect on face recognition algorithms using the results of the 2006 Face Recognition Vendor Test (FRVT). They found that a Western algorithm (i.e., made by fusing eight algorithms from Western countries) recognized Caucasian faces more accurately than East Asian faces, while an East Asian algorithm (i.e., made by fusing five algorithms from East Asian countries) recognized East Asian faces more accurately than Caucasian faces. Apparently, each algorithm exploits race-specific features to improve recognition performance within its own race category. Other studies have found that humans judge the gender of adults and children using feature sets derived from the appropriate face age category, rather than applying features derived from another age category or from a combination of age categories.[12] In a related study, it was demonstrated that human face recognition can benefit from employing gender information.[13]

Motivated by cognitive evidence, we believe that significant gains in recognition performance can be achieved by using face categorization as a precursor step to face recognition. First, category-specific features could be extracted for representing faces within different face categories more efficiently, thereby optimizing the recognition process. This is in contrast to traditional face recognition systems which represent faces using a common representation scheme, despite of significant differences among different face categories. Second, face categorization could be used to guide search within the most promising region of the face space during recognition, that is, the region containing faces from same face category as the unknown face. Additional performance improvements can be achieved using fast indexing schemes (e.g., nearest neighbor search algorithms) within each face category, as in traditional face and object recognition systems.[14,15] Finally, face categorization could yield more graceful errors, that is, incorrect matches involving faces from the same face category than faces from widely different face categories. We have investigated the effect of face categorization on recognition performance in a earlier study by using gender, ethnicity, and age information to manually categorize faces into different face categories (e.g., male, Asian, between 20 and 40 years old).[1] Although we did not explicitly address the issue optimizing face representation within each face category, we demonstrated that face categorization does improve recognition performance, increases recognition speed, and yields more graceful errors.

In this paper, the focus of our work is on automating face categorization; optimizing recognition performance within face categories will be addressed in future research. Specifically, guiding search in the appropriate region of face space during recognition would rely on our ability to categorize faces in the correct face category accurately and robustly. This requires addressing several important issues. First, we need to address the issue of determining an appropriate set of face categories. Once the face categories have been determined, a classifier can be trained to learn the separating boundaries between them. Then, categorizing faces in novel images during recognition becomes a classification problem. In this case, classification accuracy would depend both on the number and “quality” of face categories. Therefore, determining the face categories and learning to categorize faces in novel images are interrelated issues. Another important issue that needs to be addressed is tolerating face categorization errors. Obviously, if face categorization fails, then recognition will fail too since matching will consider faces from the wrong face category. Therefore, incorporating a mechanism to recover from categorization errors would be essential in order to preserve recognition accuracy.

The most intuitive way to define the face categories is by using information about gender, ethnicity, and age. Although using gender, ethnicity, and age information could yield quite meaningful face categories, it limits the number of possible face categories. Moreover, developing highly accurate and robust gender, ethnicity, and/or age classifiers from face images has been quite difficult.[16–18] Many times, for example, faces from different gender, ethnicity, and/or age groups share common features (e.g., both male and female faces might have long hair), making it difficult to classify them unambiguously using visual information alone. In other cases, faces might share features from multiple groups (e.g., cross-racial faces), making it almost impossible to classify them unambiguously. An important issue is also the lack of representative examples in every face group which is important for training the face classifiers. Many publicly available face databases, for example, contain a relatively large number of "white" faces but a much lower number of faces from other groups.

This paper investigates ways to automatically discover a categorization of faces using "bags-of-local-features" (BoF)[19,20] for face represention and unsupervised learning (i.e., K-means and hierarchical clustering[21,22]) for discovering the face categories. Both "sparse" and "dense" Scale Invariant Feature Transform (SIFT) features[15] have been investigated for building BoFs face representations. For comparison purposes, we have also experimented with other types of local features including Histogram of Oriented Gradient (HOG),[23] Local Binary Pattern (LBP),[7] and Weber Local Descriptor (WLD).[8] To categorize faces in novel images, we have considered both nearest-neighbor algorithms (i.e., k-Nearest-Neighbor (k-NN) and Approximate Nearest Neighbor (ANN)) and supervised learning to learn the separating boundaries between face categories (i.e., Support Vector Machines (SVMs)[24]). It should be mentioned that choosing an appropriate face representation scheme for face categorization is a critical issue. Although one might consider traditional face representation schemes previously introduced in face recognition literature (e.g., eigenface representations based in Principal Component Analysis (PCA)[5]), they might not be quite appropriate for face categorization. The reason is that these schemes were designed to capture face details which are important for recognition purposes. However, these details might be redundant or even irrelevant in the context of face categorization and more generic face representation schemes might work better. We have investigated this issue by comparing BoFs with eigenface representations.

The main advantage of our approach is that it does not rely on determining the face categories using a fixed, predefined set of visual cues which suffers from several problems as previously discussed. In contrast, we discover the face categories using unsupervised learning. The face categories obtained using the proposed methodology might not have a clear physical meaning besides exhibiting intra-class similarities; for example, faces of different gender, ethnicity, and age might all belong to the same face category. This not quite important, however, since the overall goal of face categorization is to improve face recognition performance. To tolerate face categorization errors, we allow face categories to overlap, that is, face images of the same individual can be associated with multiple face categories. Although overlapping face categories will not decrease the search space as much compared to using disjoint categories, this helps to increase face categorization accuracy which is critical for subsequent face recognition. We report experimental results and comparisons using the FERET database.[25]

The rest of the paper is organized as follows: Section 2 provides an extensive review of previous methods that related to our approach. In Section 3, we provide an overview of the proposed methodology. Section 4 discusses face feature extraction methodology and provides a brief review of the local features used in our experiments. Section 5 discussed the face representation scheme and provides a brief review of BoFs. Section 6 presents our approach for discovering the face categories while Section 7 presents our approach for categorizing faces in novel images. Section 8 provides a brief summary of the FERET dataset. Section 9 presents our experimental results and comparisons. Finally, Section 10 presents our conclusions and plans for future research.

2. Background

Partitioning the search space in order to improve retrieval accuracy and time has been investigated before, for example, in character recognition.[26] It has also been investigated in biometrics, for example, in fingerprint recognition where fingerprint classification is applied first to divide fingerprints into different classes.[27] Given a query fingerprint, the nearest class is found first. Then, the query is matched only against fingerprints within that class. In Refs. 28 and 29, it was argued that an effective way to improve retrieval in large biometric databases is by using binning and pruning methods that perform a coarse level classification of the query before performing exhaustive matching. In this context, they proposed using one biometric to bin or hash another biometric. For instance, a less distinctive but very fast to process biometric such as hand geometry can be used to index a more distinctive but slow to process biometric such as fingerprint and/or face. Other modalities, such as signature, can be used to prune the templates in each bin.

Partitioning the face space for improving face recognition performance has received limited attention. The most related approaches to our approach are Refs. 30–33. In Ref. 31, the face space was partitioned using an LDA-based criterion for maximum cluster separability. During recognition, a two stage approach was used. In the first stage, the closest match between the query and each face group was found using group-specific LDAs. In the second stage, a joint LDA space was used to find the best match between the query and the closest matches found in stage one. In Ref. 32, clustering was used to partition the face space for faster retrieval. Given a query, the main idea is using a simple but less accurate similarity metric, based on the face clusters, to retrieve very fast the most feasible face matches in the face database. Potential matches are then compared with the query using a slower but more accurate metric. Specifically, clustering was performed using the Expectation-Maximization (EM) algorithm with an entropy-based constraint to penalize unbalanced clusters. To measure the similarity between faces and cluster centers, an expensive but robust metric based on the Probabilistic Mapping with Local Transformations (PMLT)[34] was used. For efficient indexing and retrieval, each target face was represented by a "characterization" vector which contains the probabilities of assigning the face to each of the face clusters. This is equivalent to projecting a face to the space of face categories. Given a query, its characterization vector is compared to the characterization vectors of the target images to find the nearest matches. Promising matches are then compared with the query in more detail using the PMLT metric. Experimental results reported illustrate speedups by a factor of six or seven without significant degradation in performance. In Ref. 30, K-means clustering was used to partition the face database with the goal of reducing the search space during recognition. For face representation, global features based PCA and LDA were investigated with PCA features outperforming LDA features. To reduce clustering errors during recognition while keeping the search space small, the authors varied the number of face groups and experimented with retrieving the P nearest face groups. Their results show retrieving approximately 30% of the groups yields almost perfect face categorization. In Ref. 33, Genetic Algorithms (GAs) were use to cluster face images into two layers using a fitness involving an LDA-based distance measure and a term for penalizing unbalanced clusters.

Using visual common visual cues to partition the face space has been investigated in Ref. 35. Specifically, gender and age were used to prune the search space and aid the recognition process. First, a random forest classifier was used to classify a face with respect to gender and age. During recognition, unknown faces were only compared against known faces having the same gender and being within the same age group. Experiments were performed using a simple face recognition algorithm, based on histogram intersection, while faces were represented using LBP features which were extracted from Gabor phase and magnitude. The experimental results obtained illustrate that discriminative cues based in gender and age can improve face recognition performance in terms of time, accuracy, and graceful degradation.

Related but different approaches from ours have been reported in Refs. 36 and 37. In Ref. 36, an unsupervised fuzzy learning algorithm was used to cluster face images into groups containing images from the same individual only. Different performance indicators, such as the partition coefficient and the entropy coefficient, were used to evaluate class homogeneity. Then, a multi-layer neural network was trained, using the cluster prototypes, to perform face recognition by learning the separating boundaries between the face groups. Experimental results were reported on a very small data set only. In Ref. 37, the problem of face pose variation was addressed by using a tree-like structure to group faces having similar pose. During recognition, unknown faces were assigned first to the group that matches their pose; then, recognition was performed within this group only.

A few interesting approaches have also been reported in the literature where the search space is partitioned adaptively for each query. This is different from our approach where the partitioning of the face space is performed off-line. These methods, however are computationally expensive and might not be appropriate for real-time face recognition, In Ref. 38, K-means was applied in PCA space to partition the gallery set. Given a query, first it is projected into the PCA space and then the nearest face cluster is found. To identify the query, LDA was used where the LDA space was built using the query and the images in the nearest cluster only. An extension of this approach has been presented in Ref. 39 using an iterative face clustering scheme. In the first iteration, an LDA space is built from the gallery set and K-means is applied in that space to partition the gallery set into face groups. Given a query, first it is projected in the LDA space and then the nearest face groups are found. In the second iteration, a new LDA space is built using the images from the face groups selected in the previous iteration. Then, this process is repeated until only one nearest face group is selected. A new LDA space is built from this group and the most similar face to the query is selected to identify the query. Promising results have been reported using this idea; however, the computational cost of this approach is rather high since new LDA spaces have been computed for each.

Discovering a set of face categories from unlabelled face images is also related to previous work on face clustering which has many applications in image/video indexing and content analysis (i.e., identify significant characters, scenes, and events). The key objective is grouping together images or video frames of individuals who might appear in photo collections or video sequences using unsupervised techniques. Typically, face detection is applied first to find all faces present in an image. Then, clustering, involving spatio-temporal constraints in the case of videos, is applied to establish the face clusters. Two main issues need to be addressed in face clustering: a metric for evaluating similarity between faces and an algorithm for clustering similar faces. In Ref. 40, a face clustering method was proposed using K-means for clustering and Hidden Markov Models (HHMs) for representing the cluster centers. In Ref. 41, faces of the same person detected in contiguous frames were represented in a subspace which is invariant to a desired group of transformations (e.g., affine transformations). To measure similarity between subspaces, the Joint Manifold Distance (JMD) was employed. Face clustering was then performed using an agglomerative clustering strategy based on JMD. In a related approach,[42] face sequences were not matched directly but divided first into subsequences, each containing faces of similar pose. Face subsequences were then clustered using graph partitioning and domain knowledge constraint propagation. In Ref. 43, Haar-like features are used to represent faces along with a similarity metric based on mutual information and fuzzy c-means for face clustering. In Ref. 44, SIFT features were used for computing face similarity and a hierarchical average linkage algorithm for face clustering. A spectral clustering algorithm using a new distance metric that is robust to outliers was proposed in Ref. 45.

Finally, our work has similarities to methods dealing with the problem of discovering visual object categories from sets of unlabelled images. In Ref. 46, a "bag-of-features" approach was used to represent objects and probabilistic Latent Semantic Analysis (pLSA)[47] or Latent Dirichlet Allocation,[48] statistical approaches previously used in unsupervised topic discovery in text, for clustering. In Ref. 49, SIFT features along with an improved agglomerative clustering method were used for building object categories. In Ref. 50, each image was decomposed into a set of SIFT features. Every set was then treated as a node in a graph where edges between nodes were weighted based on how well the corresponding feature sets can be aligned. Spectral clustering was applied next to extract a set of dominant object categories. Further processing leads to a refined set of object categories. In later studies,[51,52] powerful statistical models were used (e.g., hierarchical Latent Dirichlet Allocation (hLDA)) to organize collections of images into a tree-like structure, leading to a hierarchy of visual object categories.

3. Method Overview

Given a collection of unlabeled face images, our goal is to produce a partition of the face images into a set of face categories. Figure 1 illustrates the main steps of the proposed approach. First, the face categories are discovered by partitioning the face space using clustering such that similar faces are grouped together. Once the face categories have been established, a classifier can be trained to learn the separating boundaries between them. Alternatively, nearest neighbor techniques can be used to find the closest face category. During recognition, unknown faces are assigned to the closest face category using classification or nearest neighbor techniques. Once the closest face category has been retrieved, recognition be performed within this category only. To optimize recognition performance, category-specific features can be used for recognition within each face category. In this paper, we have only addressed the issues of discovering the face categories (block 1) and categorizing faces in novel images (block 2). The problem of optimizing recognition performance within each category (block 3) has been left for future research.

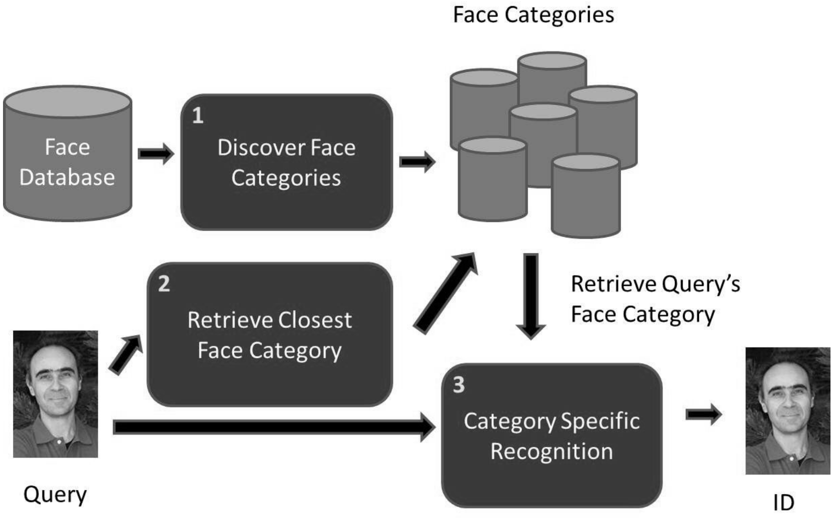

Face Categories
Face Database
1
Discover Face Categories
2
Retrieve Closest Face Category
Retrieve Query's Face Category
3
Category Specific Recognition
Query
ID

Fig. 1. Main steps in face categorization and recognition.

It should be mentioned that since the overall goal of our work is to improve face recognition performance by guiding search into the most promising region of the face space, the proposed face categorization approach assumes a "closed" universe", that is, a set of images from a group of known individuals. Therefore, face categorization is optimized for this predefined set of individuals; including new individuals in the dataset would require updating face categorization to optimize its performance.

4. Local Feature Extraction for Face Description

The human face is a non-rigid object whose appearance can change dramatically under different conditions. Using an appropriate set of features to describe the intrinsic attributes of the human face is important for developing a robust scheme for face category discovery and categorizing faces in novel images. Past research in face recognition has emphasized local features computed from geometric relations between facial features (e.g., distances between eyes, nose, and mouth).[53] Due to difficulties in detecting facial features reliably, efforts in the field shifted to using global features, mainly by employing subspace methods. Subspace methods use dimensionality reduction; two classical examples are the methods of Eigenfaces[5] and Fisherfaces[6] while a more recent example is the method of Sparse Face Representations.[54] Global features based on subspace methods have demonstrated good success; however, they cannot capture reliably subtle and refined discriminative features of the face which might be important for face categorization. Moreover, they require face alignment and are not very robust to variations caused by changes in facial pose and expressions, occlusion, and illumination.

Recently, a new powerful type of local features has been proposed in the object recognition literature. Typically, they are extracted around interest points and a small neighborhood around the interest point is used to compute a descriptor. Local features have the ability to overcome the drawbacks of global features and have shown to perform well with unconstrained object images. A comprehensive review of interest point detectors and descriptors can be found in Refs. 55 and 56. SIFT features[15] are among the most popular local features in the literature. Recent work has shown that SIFT features can be reliably detected and matched across different examples of an object under varying viewpoints, poses, or lighting conditions.

Due to considerable success of SIFT features in object recognition and classification, we have adopted them here for face representation. We summarize below the main ideas of SIFT features and provide a brief review of using SIFT features in face processing. For completeness, we provide brief reviews of HOG, LBP, and WLD features which were used in our comparisons.

4.1. *Scale Invariant Feature Transform* (*SIFT*)

SIFT represents a method for extracting distinctive invariant features from images.[15] SIFT features are invariant to image scale and rotation and have shown to provide robust matching across a substantial range of affine distortion, 3D viewpoint change, noise addition, and change in illumination. Due to being highly distinctive, a single feature can be correctly matched with high probability against a large database of features. As a result, SIFT features have shown to be very powerful for general object detection and recognition under considerable amounts of occlusion. SIFT contains four main steps: (1) Scale-space extrema detection, (2) Accurate keypoint localization, (3) Orientation assignment, and (4) Keypoint descriptor. First, interest points or keypoints are identified as local maxima or minima of Difference-of-Gaussian (DoG) images across scales. To better localize the keypoints, both in space and scale, a detailed model is fit to each candidate location. To account for image rotation, a gradient orientation histogram is computed in the neighborhood of each keypoint and the dominant orientation is assigned to the keypoint. Finally, a descriptor is computed for each keypoint using the image gradients in a region around the keypoints, measured at the selected scale. Typically, a SIFT descriptor is vector of 128 values although its dimensionality can vary depending on various parameter values. It has been argued that computing SIFT descriptors at interest point locations only is not effective when dealing with low texture objects as it yields a relatively low number of features. Faces contain regions of low texture which might affect the number of SIFT features detected but also the distribution of SIFT features over the face which might be non-uniform. Figure 2(a) shows an example of SIFT features detected from different views of the same person. As it can be observed, a relatively small number of features have been detected. One way to increase the number of SIFT features is by adjusting certain thresholds within the SIFT algorithm, however, there are still face regions containing very few SIFT features (see Figure 2(b)). To address this issue, we have also experimented with extracting descriptors at regular image grid points using the method proposed in Ref. 57. We refer to this type of features as *dense* SIFT features; SIFT features extracted at interest point locations will be referred as *sparse* SIFT features.

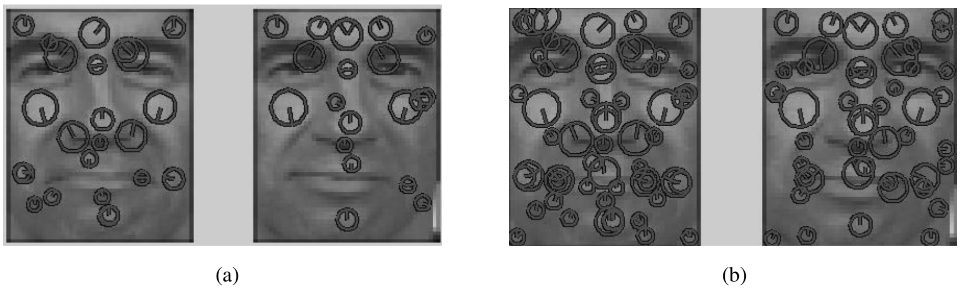


Fig. 2. (a) SIFT features extracted from different views of the same person; (b) a higher number of SIFT features can be detected by relaxing thresholds within the SIFT algorithm.

It should be mentioned that although SIFT features have been very popular in object detection and recognition, their use in face processing has been rather limited. In Ref. 58, SIFT features were used for face authentication. The key idea was computing face similarity using SIFT features from face sub-regions. A related approach was investigated in Ref. 59 where stable face sub-regions were determined using clusters of SIFT features. Matching was performed by combining both global and local similarities using the face sub-regions. In Ref. 60, the performance of SIFT for face recognition was analyzed. Based on the analysis, two modified SIFT descriptors were proposed in order to increase the number of SIFT features and to better handle SIFT features detected at high scales and near face boundaries. In Ref. 61, a quantitative and qualitative analysis of interest point detectors and descriptors in the context of face detection and localization was performed. Emphasis was given on detectors that respond well inside the face region and close to facial features such as mouth, eyes, and nose. In Ref. 59, face-specific SIFT features were proposed for face recognition while in Ref. 62, local features were shown to have superior performance for face recognition in unconstrained environments. In

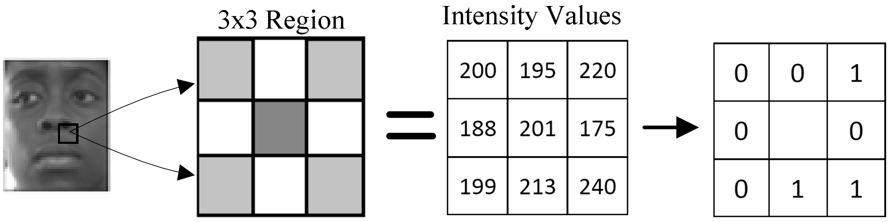
Ref. 63, SIFT and Speeded Up Robust Features (SURF) features were employed for face recognition using both aligned and non-aligned faces. To improve face recognition performance, SIFT and SURF features were computed on a dense grid instead at the interest point locations. In a related approach,[64] dense SIFT features and BoFs were used for face recognition.

4.2. *Histogram of Oriented Gradients* (*HOG*)

HOG features were proposed in the context of human detection.[23] They are extracted by counting occurrences of edge orientations in localized portions of an image. First, the image is divided into small cells and a histogram of edge orientations is computed for each cell. Then, the HOG descriptor is formed by combining the normalized histograms.

4.3. *Local Binary Pattern* (*LPB*)

LBP features were first introduced in Ref. 65 and used in Ref. 7 for face recognition. The basic LBP operator assigns a binary number (i.e., label) to each pixel *p* by thersholding the values of its surrounding pixels in a 3 × 3 neighborhood. The output of thersholding is either 1 or 0 depending on whether the neighbor's value is greater or less that the value of *p*. The label assigned to *p* is obtained by concatenating the thresholded neighbor values as shown in Figure 3. The LBP representation is generated by dividing the image into a grid of windows and computing histograms of the LBP values within each window. These histograms are then concatenated to form the final representation.



200	195	220
188	201	175
199	213	240

0	0	1
0		0
0	1	1

Fig. 3. An example illustrating LBP.

4.1. *Weber Local Descriptor* (*WLD*)

WLD[8] represents an image as a histogram. A WLD feature contains two components: differential excitation and gradient orientation (see Figure 4). The differential excitation component represents local salient patterns in an image. Given a subwindow centered at pixel p, it is computed by taking the ratio between the sum of intensity differences of p against its neighboring pixels and the intensity of p. The differential excitation and dominant orientation components can be computed at each image location or at locations corresponding to a coarser grid. The WLD histogram is computed from the WLD features.

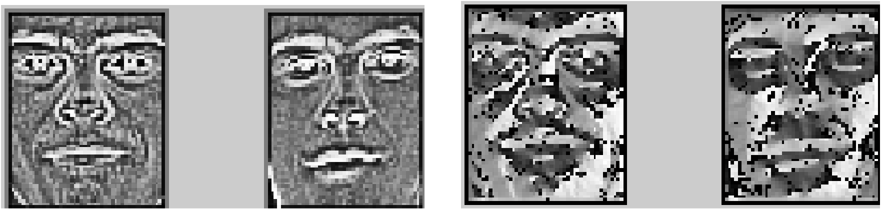


Fig. 4. An example illustrating WLD: (a) differential excitation; (b) gradient orientation.

5. Face Representation

Recently, a powerful object representation scheme has been proposed in the literature, known as BoFs.[19,20] BoFs is based on histograms of quantized local features and has its origins in textual information retrieval.[48] We review below the BoFs method and discuss some important issues.

5.1. *Bags of Features* (*BoFs*)

The main idea of BoFs is representing objects as an orderless collection of quantized local image features, ignoring spatial relationships between features. Generating a BoFs image representation requires building a "visual" vocabulary first. This is performed by extracting local features from a set of training image and applying clustering to group similar features together. The cluster centers obtained through this processes are referred to as "visual" words. The combination of the visual words found forms the visual vocabulary. Images can be represented in terms of visual words in two steps. First, local features are assigned to the closest word in the "visual" vocabulary; this is a quantization step which is referred to as label assignment. Second, each image is represented by a histogram representing the frequency of visual words in the image. We refer to the histogram as the BoFs representation of the image. BoFs can be used compute similarity between images or, given a query, to retrieve the most similar images in an image database. Moreover, object classes (e.g., faces, cars) can be modeled by training a classifier using BoFs to learn the separating boundaries between object categories.[19] Despite its simplicity and low computational complexity, BoFs has been surprisingly successful in various applications including object/scene detection, classification, and retrieval. A comprehensive survey of BoFs and its variations can be found in Ref. 66.

From the designer's point of view, several important issues need to be addressed in each step of the BoFs method. The first issue is the choice of local image features to be extracted. Although SIFT features[15] have been the most popular choice, other types of features could be used as well.[56] Another issue is the choice of the clustering method to be used for building the visual vocabulary. K-means[67] has been the most popular choice although more sophisticated clustering methods could be used too.[21,22] The size of the visual vocabulary has a great effect on the performance method and needs to be optimized. Choosing an appropriate similarity measure for determining similarity between features and cluster centers during clustering represents another issue. Finding the closest cluster centers when clustering local features to build the visual vocabulary or finding the closest visual word when quantizing the local features to compute the BoFs representation requires choosing efficient nearest-neighbor algorithms, for example, k-d trees in lower dimensions or ANNs in higher dimensions. Finally, choosing an appropriate distance measure and an efficient algorithm to compute nearest neighbors are issues that must be addressed when comparing BoF representations for object classification and retrieval.

5.2. *Using BoFs for face representation*

To represent faces using BoFs, a “visual” vocabulary needs to be built first. Figure 5 provides an overview of the various components involved in our face categorization approach where the solid line shows the steps performed during training (i.e., build the “visual” vocabulary (box 1), represent face images in the gallery set using BoFs (box 2), and discover the face categories (box 3)) while the dashed line shows the steps performed during testing (i.e., represent faces in novel views using BoFs (box 2) and categorize them (box 4)).

To build a visual vocabulary, first we extract SIFT features from a set of training face images; then, we apply the steps outlined in the previous section. Figure 6 illustrates the process in more detail and corresponds to box 1 in Figure 5. The visual words obtained reduce the size of the feature space and allow capturing a larger variability of local face image structure than individual features. At the same time,

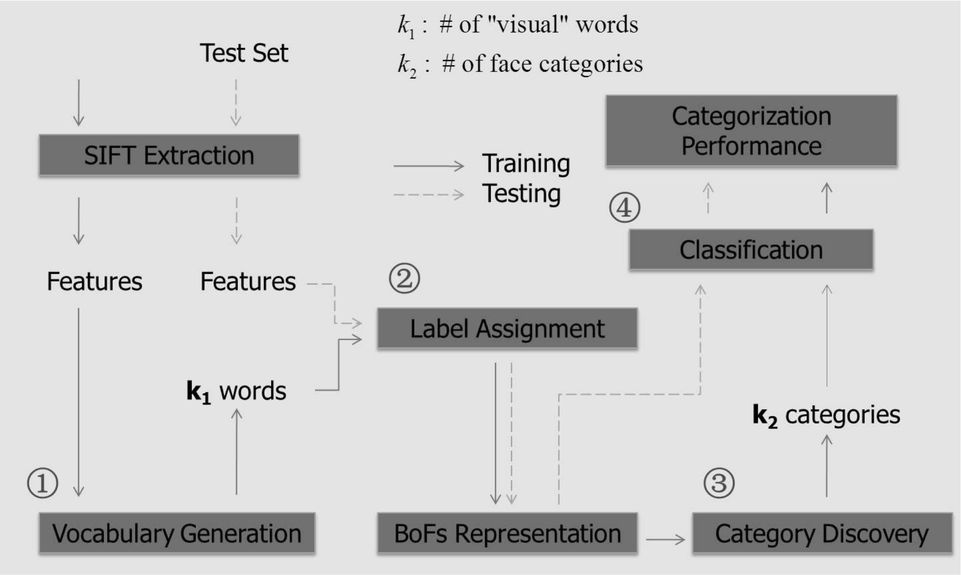


Fig. 5. Mains steps for face category discovery (solid line) and categorization of faces in novel images (dashed line).

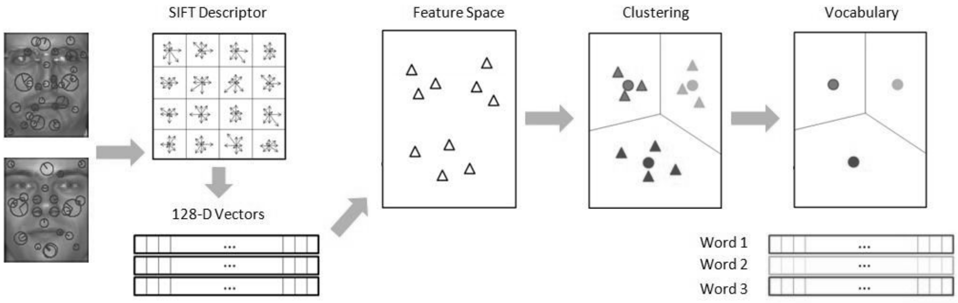
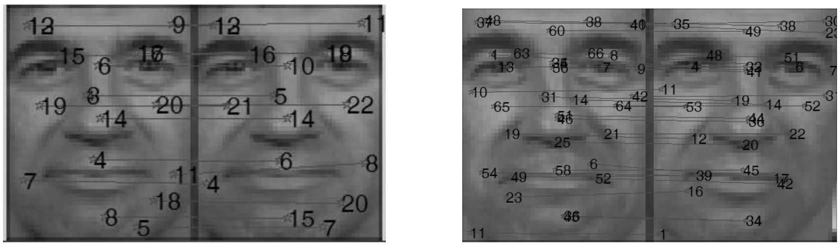


Fig. 6. Steps involved in building the visual vocabulary.

they focus on face characteristics which re-occur in novel face images and therefore generalize over new face instances. To extract a set of “visual” words, we have used the popular K-means method with the Euclidean distance for measuring similarity between SIFT features. It should be mentioned that several other clustering algorithms and distance measures have been investigated in the literature for building a “visual” vocabulary,[66] however, they are more time consuming. The size of the vocabulary is an important parameter that needs to be optimized. In the context of face categorization, it should be large enough to distinguish between important differences among faces. At the same time, it should not be too big to avoid considering irrelevant face variations such as noise. We have experimented with varying the number of clusters in K-means to determine an optimum vocabulary size. Since K-means is affected by initialization, we run it several times using different random initializations. A brief discussion of K-means and its properties is provided in the next section.

Once the “visual” vocabulary has been built, faces can be represented using BoFs. First, a “visual” word is assigned to each SIFT feature; we refer to this process as SIFT feature quantization or label assignment. Then, a histogram is built representing the frequency of “visual” words in the face (i.e., box 2 in Figure 5). Using a common “visual” vocabulary to represent faces in different face categories allows face categories to “share” features, improving face categorization robustness. Quantizing SIFT features extracted from faces in the training set versus faces in novel image is performed differently. SIFT features from training images are simply assigned the “visual” word corresponding to the cluster center that contains them. Although the same process could be used for SIFT features extracted from faces in novel images, it is more robust to assign them the “visual” word associated with their closest SIFT feature(s) in the training set. This requires searching a large set of SIFT features which could be very time intensive. Numerous methods have been proposed for efficiently searching large sets of data (e.g., k-d trees), however, these methods are not effective in high-dimensional spaces. Here, we employ ANNs.[52]

In general, the robustness of SIFT features as well as the tolerance provided by the label assignment step, makes BoFs an attractive face representation scheme. Figure 7(a) illustrates that many SIFT features between different views of the same person can be matched (the original SIFT features are shown in Figure 2(a)). Figure 7(b) shows an even higher number of SIFT matches due to increasing SIFT feature detections (see Figure 2(b). Since similar SIFT features will be assigned the same "visual" word due to SIFT quantization, the greater the number of SIFT features that can be matched, the greater similarity between BoFs representations. This justifies using dense SIFT features as well.



(a) (b)

Fig. 7. Matching SIFT features between different views of the same person. Increasing the number of SIFT features detected increases the matches and promotes similarity between corresponding BoFs. In (a) 15 SIFT features were matched while in (b), 36 SIFT features were matched.

6. Discovering Face Categories

To discover an appropriate set of face categories, we apply unsupervised learning (clustering) on a training set of face images represented by BoFs (i.e., box 3 in Figure 5). Clustering combines sets of objects into classes based on a measure of similarity, such that similar objects are placed in the same cluster, while dissimilar objects are placed in different clusters. Therefore, the face categories obtained through the proposed methodology might not have a physical meaning such as when grouping faces using gender, ethnicity, and race information. In fact, the face categories obtained through clustering contain faces of mixed gender, ethnicity and age. However, this is not an issue since the objective is applying categorization for improving recognition performance. To tolerate face categorization errors, one might consider retrieving the "k" closest face categories instead of a single (i.e., closest) category. This, however, would increase recognition time. Here, we have opted for a different strategy which allows for face categories to overlap. Specifically, we do not force different images of the same individual to be all assigned to the same face category. In contrast, different images of the same individual could be assigned to different face categories. This is particularly useful when face images of a given person are close to the boundary between two or more face categories. In this case, large face variability may not be fully handled by the distance measure used and faces images might end up into different face categories. In general, we have found that this strategy reduces intra-category variations, improving categorization performance.

In discovering the face categories using clustering, two issues need to be addressed: (i) the distance measure for estimating similarity and (ii) the clustering algorithm. Here, we have used Euclidean distance for computing the distance between histograms although other distance measures could be used.[69] Comprehensive surveys of clustering algorithms can be found in Refs. 21, 22 and 67. In general, clustering techniques can be divided into two main categories: *partitional* and *hierarchical*. The hierarchical approach is to divide the data into clusters, and then subdivide each cluster again and again, until a certain criterion is met. The partitional approach, on the other hand, is to divide the data into a desired number of clusters in one step. We have experimented with both types of clustering methods in this study. Of course, the question of what is a good clustering method depends on the context of the application. Here, we are interested in dividing faces into different face categories such that categorizing faces in novel images can be done with high accuracy and robustness. Therefore, our criterion for judging the quality of clustering is based on the accuracy of face categorization in novel images. Next, we provide a brief review of K-means and hierarchical clustering.

6.1. *K-means clustering*

K-means is an iterative clustering algorithm which is initialized randomly by K seed points for the clusters. Each iteration consists of two steps. First, a new partition is generated by assigning each point to its closest cluster center. Then, cluster centers are updated by computing the empirical mean in each partition. Iterations continue until the partition stabilizes. From a theoretical point of view, K-means attempts to find a partition which minimizes the sum of squared errors between the empirical mean of each cluster and the points in that cluster. This is known to be an NP-hard problem. In practice, K-means converges to a local optimum within a few iterations. K-means is frequently used because of its computational simplicity. Its time complexity is $O(NK\mathit{l}d)$ when clustering N data points of d dimensions with K cluster centers and l iterations. However, the clustering solution obtained might be suboptimal when the number of outliers is large. Moreover, the solution depends on the number of clusters and random initialization. To deal with this issue, we run K-means multiple times using different random initializations. A recent review of K-means and its extensions can be found in Ref. 67.

6.2. *Hierarchical clustering*

Hierarchical clustering organizes data into a hierarchical structure; a dendogram is typically used to visualize results where the root represents the whole dataset, leaves represent data points and intermediate nodes the extent that points similar to each other. Most hierarchical clustering algorithms are variants of the single-link and complete-link for measuring closeness between clusters. In each case, closeness is measured as the minimum/maximum distance between any pair of points in each of the two clusters. Here, we used the single-link approach although more powerful but also time consuming measures could be used.[22] Cutting the tree at different levels produces different partitions. In this study, we used agglomerative clustering which performs hierarchical clustering in a "bottom-up" fashion; this is in contrast to K-means that works in a "top-down" fashion.

Hierarchical clustering algorithms using simple similarity measures suffer from robustness to noise and outliers. Their computational complexity is at least O(N^2).

7. Face Categorization in Novel Images

Assuming that the face categories have been established, we treat the problem of assigning faces in novel images to face categories as a classification problem (i.e., box 4 in Figure 3). Categorization accuracy would depend both on the quality of the clusters obtained as well as the performance of the classifier employed. We have experimented with different classification methods to assess the performance of face categorization. First, an SVM classifier[24] was used to learn the separating boundaries between face categories. Since SVM is a binary classifier while our problem is a multi-class classification problem (i.e., multiple face categories), we have applied SVMs by using a one-versus-all strategy. The next method used was a k-Nearest Neighbor (kNN) classifier[70] which performs classification by finding the k nearest neighbors in the training set. Since kNN could be computationally expensive, we have also investigated using ANN[52] for classification. Next, we provide a brief review of each classifier.

7.1. *K-Nearest Neighbor* (*kNN*)

Searching for the closest match (i.e., nearest neighbor) in some space is a common problem in computer vision and many other fields. Nearest neighbor (NN) classifiers work by first finding the nearest neighbor to a query; then, the class of the nearest neighbor is used to determine the class of the query. Typically, however, improved results can be obtained by taking more than one neighbors into account; this is called kNN classification. There are different ways to determine the class of the query using the k nearest neighbors. The most common approach, which was also used here, is to assign the majority class among the nearest neighbors to the query. kNN has high memory and computational complexity requirements, especially when the number of training examples and data dimensionality are high. A recent review on kNN classifiers and their extensions can be found in Ref. 70.

7.2. *Approximate Nearest Neighbors* (*ANN*)

When dealing with data of low dimensionality, efficient indexing techniques, such as k-d trees, can be used to find nearest neighbors efficiently. However, when data have dimensionality, linear search is often the only choice for solving the nearest neighbor problem. Linear search is very simple, however, it too expensive to be practical. ANN algorithms have shown to provide significant speedups with only minor loss in accuracy. The idea is to sacrifice time for accuracy, that is, to save computations at the expense of accuracy (i.e., non-optimal neighbors). A plethora of ANN algorithms have been proposed in the literature. Here, we use the method in Ref. 68 which involves automatic algorithm selection and parameter selection using a cross-validation approach.

7.3. *Support Vector Machines* (*SVMs*)

SVMs are binary classifiers which have been shown to be an attractive and more systematic approach to learning linear or nonlinear decision boundaries.[24] Their key characteristic is their mathematical tractability and geometric interpretation. Given a set of points, which belong to either of two classes, SVM finds the hyper-plane leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyper-plane. This is equivalent to performing structural risk minimization to achieve good generalization. Assuming examples from two classes:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l), x_i \in R^N, y_i \in \{-1, +1\}$$

finding the optimal hyper-plane implies solving a constrained optimization problem using quadratic programming. The optimization criterion is the width of the margin between the classes. The discriminate hyper-plane is defined as:

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i k(x - x_i) + b$$

where $k(x - x_i)$ is a kernel function and the sign of $f(x)$ indicates the membership of x. Constructing the optimal hyper-plane is equivalent to finding all the nonzero α_i. Any data point x_i corresponding to a nonzero a_i is a support vector of the optimal hyper-plane. Suitable kernel functions can be expressed as a dot product in some space and satisfy the Mercer's condition.[24] By using different kernels, SVMs implement a variety of learning machines (e.g., a sigmoidal kernel corresponding to a two-layer sigmoidal neural network while a Gaussian kernel corresponding to a radial basis function (*RBF*) neural network). The Gaussian radial basis kernel is given by

$$k(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right)$$

The Gaussian kernel has been used in this study (i.e., our experiments have shown that the Gaussian kernel outperforms other kernels in the context of our application).

8. Dataset

To test our approach, we used the FERET database,[25] which contains a large number of images acquired during different photo sessions and has a good variety of gender, ethnicity and age groups. The lighting conditions, face orientation and time of capture vary. In this work, we concentrate on frontal face poses coded as $\mathrm{f_a}$ (regular frontal image) or $\mathrm{f_b}$ (alternative frontal image, taken shortly after the corresponding $\mathrm{f_a}$ image). In our evaluations, the $\mathrm{f_a}$ images were used as the gallery set while the $\mathrm{f_b}$ images were used as the query set (i.e., face images in question). All faces were normalized in terms of orientation, position and size prior to experimentation. They were also masked to include only the face region (i.e., upper body and background were cropped out) yielding an image size of 48 × 60 pixels. Figure 8 shows some representative examples.

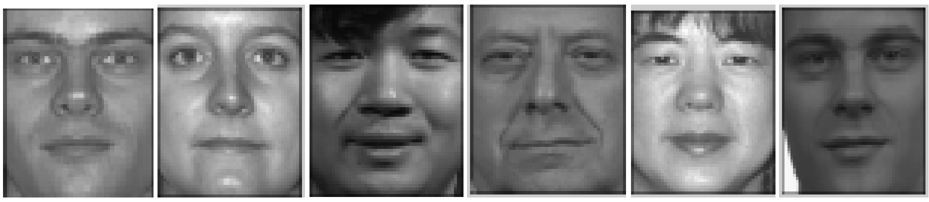


Fig. 8. Example faces from the FERET face database.

9. Experimental Results and Comparisons

We have performed extensive experiments to investigate the performance of the proposed approach as well as the effect of various parameter choices such as type of features, vocabulary size (i.e., denoted as k_1 in our experiments), clustering algorithm for discovering face categories, classification algorithm for face categorization, and number of face categories (i.e., denoted as k_1 in our experiments). In all of our experiments, K-means clustering was used for building the visual vocabulary. Moreover, we have performed comparisons with using global features for face representation. Next, we present our experimental results.

9.1. *Face categorization using local features*

9.1.1. *Vocabulary size*

First, we investigated the effect of vocabulary size on categorization performance using SIFT features and kNN for face categorization. The number of face categories was set to 10 and were found using K-means. As shown in Figure 9, categorization performance increases with vocabulary size. However, increasing vocabulary size increases computation requirements, for example, quantizing SIFT features in novel images becomes more time consuming.

9.1.2. *Local features*

Using different features for face representation will affect the performance of face categorization. We have compared four types of local features: sparse SIFT, dense SIFT, HoG, WLD, and LBP. BoFs was used with SIFT features only since HoG, WLD, and LBP yield a histogram for the whole face image. In the case of BoFs using SIFT, the vocabulary size was set to 1000. For each method, K-means was used to discover the face categories where the number of face categories was set to 10.

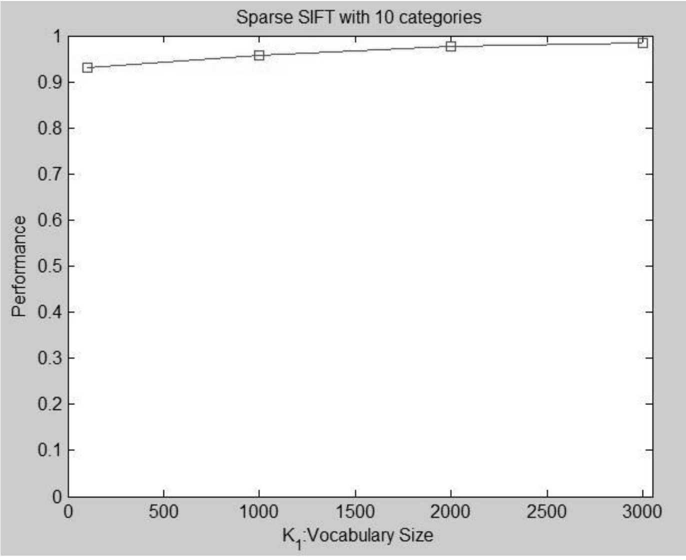


Fig. 9. Categorization performance as vocabulary size increases.

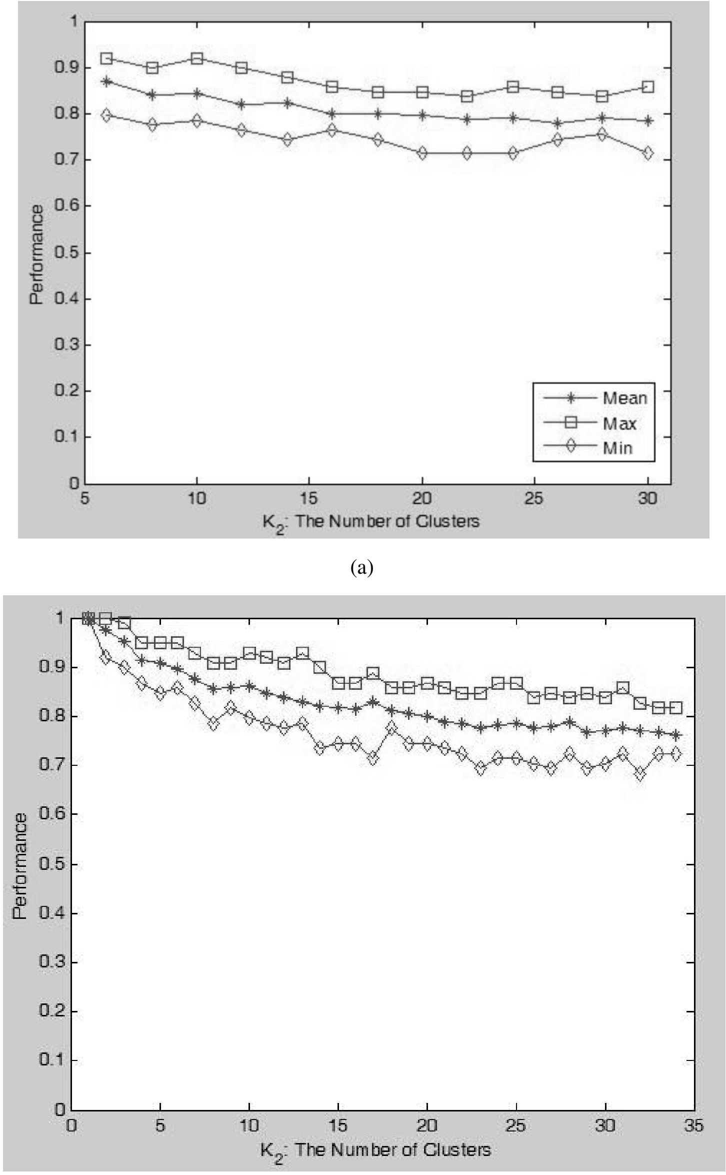
We performed each experiment 20 times, initializing K-means randomly each time. Face categorization was performed using kNN. Table 1 shows the average performance and standard deviation for each method. Our results indicate that SIFT features perform better than HOG and WLD features for face categorization. Moreover, sense SIFT features perform slightly better than sparse SIFT features.

Table 1. Categorization performance using different types of local features.

Feature	**Accuracy (avg)**	**Accuracy (std)**
sparse SIFT	95.65%	0.84%
dense SIFT	96.49%	0.96%
HOG	92.18%	0.59%
WLD	91.78%	1.52%
LBP	89.41%	1.02%

9.1.3. *Clustering algorithms and number of face categories*

In this experiment, we have investigated the effect of the number of face categories on categorization performance. Two different clustering algorithms have been tested: (i) K-means, and (iii) hierarchical clustering. Each experiment was repeated 20 times, each time building the visual vocabulary by initializing K-means randomly. The number of visual words was set to 100 to facilitate experimentation, however, results were observed



(a)

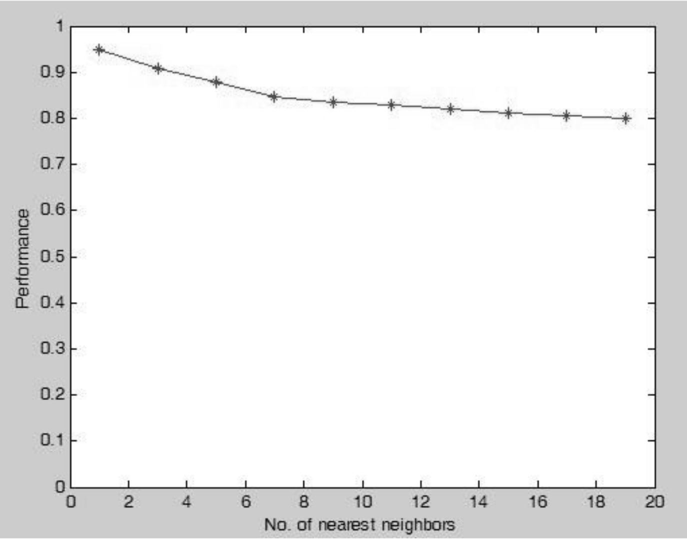
(b)

Fig. 10. Effect of number of face categories on categorization performance; face categories were discovered using (a) K-means clustering and (b) hierarchical clustering.

using larger vocabulary sizes. kNN was used for face categorization. Figure 10 reports the best, worst, and average performance obtained for each clustering algorithm and for different number of face categories. As shown, categorization performance decreases as the number of face categories increases. This is reasonable as it becomes more difficult to separate faces as the number of face categories increases). Although hierarchical clustering performs better when the number of face categories is relatively small (i.e., less than five), its performance degrades as the number of face categories increases. K-means, on the other hand, produces more stable results and achieves its best performance at around 10 face categories.

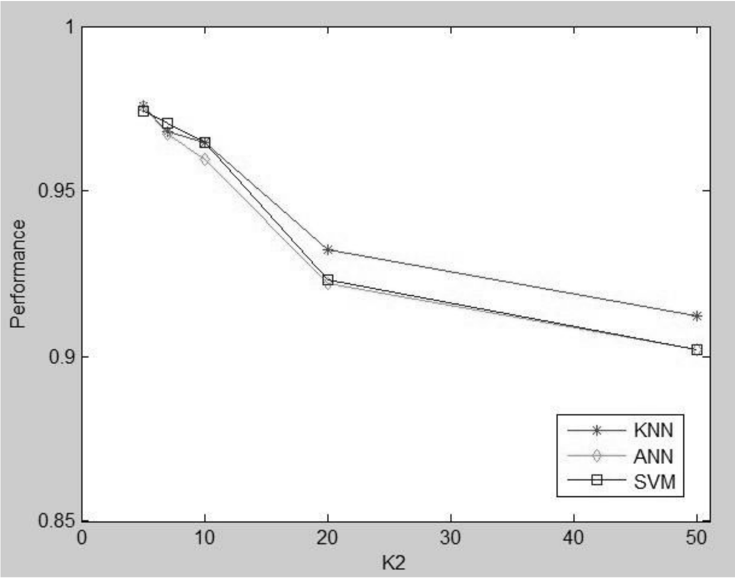
9.1.4. *Classification algorithms*

In this experiment, we have investigated the effect of different classifiers (i.e., kNN, ANN, and SVM) on face categorization performance. The vocabulary size was set to 1,000 while the number of face categories was varied between 5 to 50. To determine the optimum number of nearest neighbors to use in kNN and ANN, we performed experiments using 5 face categories. Best results were obtained using one nearest neighbor; Figure 11(a) shows the results in the case of ANN; similar observations were made in the case of kNN. Figure 11(b) indicates that kNN performs better than ANN and SVM as the number of face categories increase. Their performance drops as the number of face categories increases; this is consistent with the results reported in the previous subsection.



(a)

Fig. 11. (a) Effect of number of nearest neighbors on categorization performance using ANN and (b) categorization performance for each classifiers compared (i.e., kNN, ANN, and SVM) by varying the name of face categories.



(b)

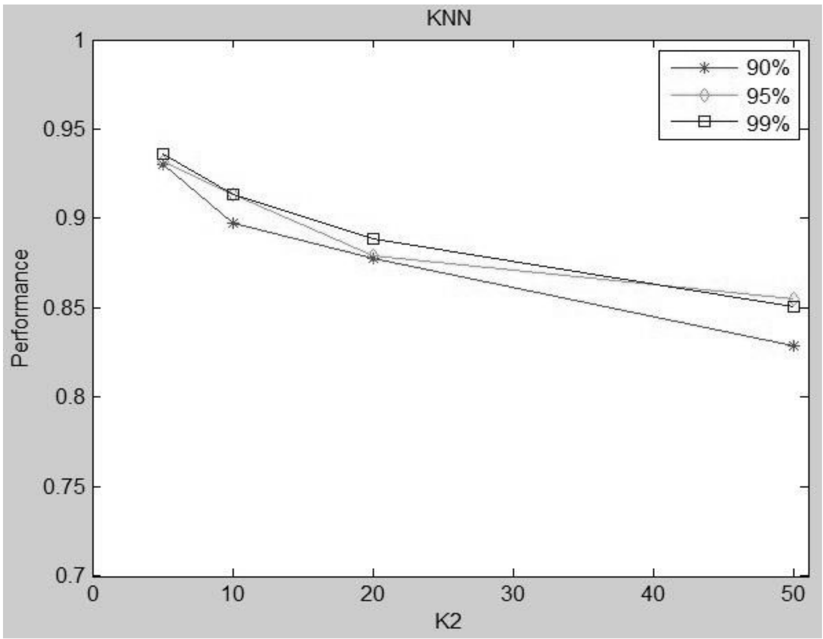
Fig. 11. (*Continued*)

9.2. *Face categorization using global features*

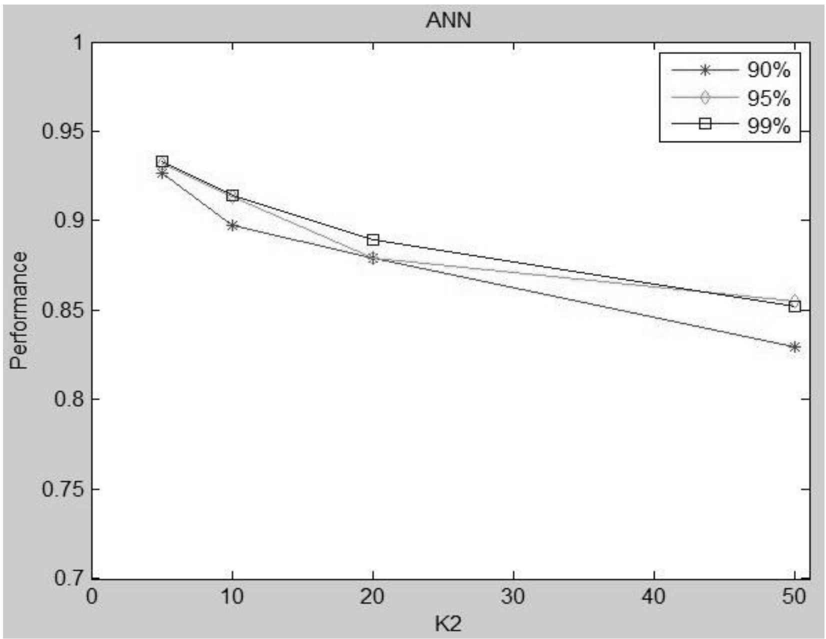
In this section, we report our results on comparing local with global features for face categorization. Dense SIFT features were used in our comparison with the vocabulary size set to 1000. First, we considered PCA features. We experimented with preserving 90%, 95%, and 99% of the information using PCA. Three different classifiers were tested: kNN, ANN, and SVM. Figures 12(a)–(c) illustrated the performance of each classifier by varying the number of face categories and the amount of information preserved in PCA. Best performance was obtained by preserving 99% of the information (i.e., 151 dimensions). Figure 12(d) compares SIFT features with PCA features by varying the number of face categories and keeping 99% of the information in PCA. As it can be observed, SIFT features outperform PCA features in all cases. Among the three classifiers, kNN performed best both for SIFT and PCA features.

9.3. *Summary of results*

In this section, we summarize our experimental results. Vocabulary size is an important parameter in BoFs; our results indicate that face categorization performance increases as vocabulary size increases, reaching close to 99% accuracy with 3,000 “visual” words, assuming 10 face categories. When comparing different local features, dense SIFT features seem to perform best. Moreover, local features outperform global features. When varying the number of face categories, we noticed that categorization accuracy decreases

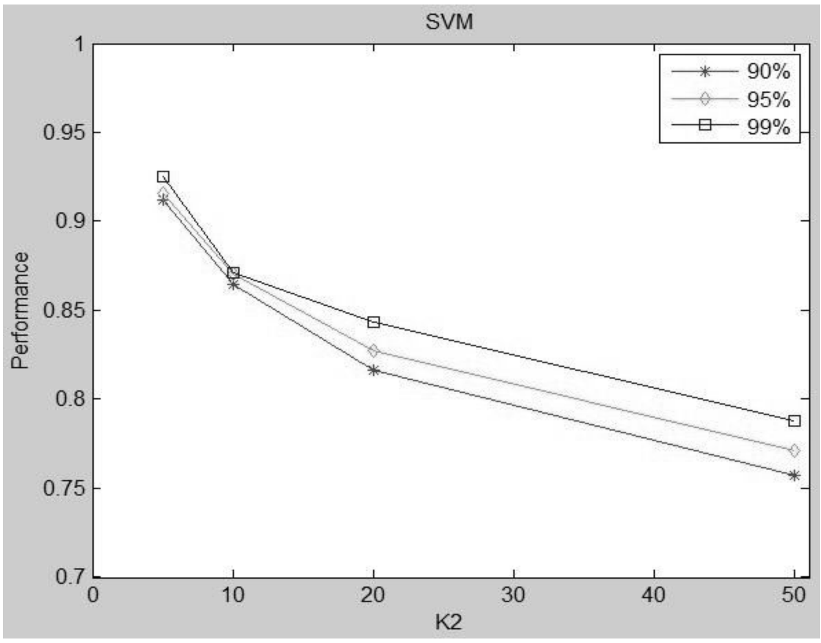


(a)

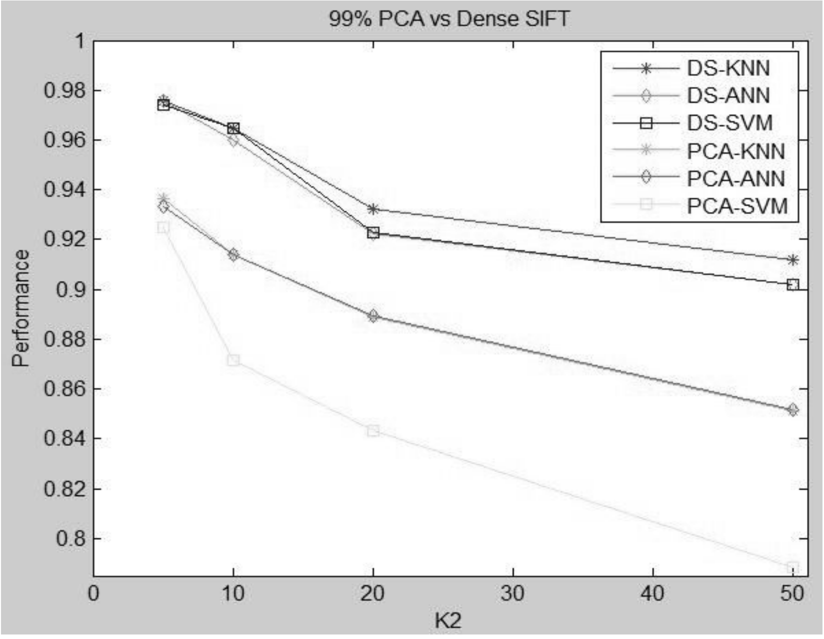


(b)

Fig. 12. Categorization performance as number of categories increases. (a) Three different classifiers were compared: SVM, kNN, and ANN.



(c)



(d)

Fig. 12. (*Continued*)

as the number of face categories increases. We obtained satisfactory results using 10 face categories; however, this is data dependent. In discovering the face categories, hierarchical clustering performed better than K-means when the number of face categories was relatively small. However, K-means outperformed hierarchical clustering when increasing the number of face categories. Finally, kNN (k = 1) gave the best results when categorizing faces in novel images. ANN, which is much more practical than kNN, performed slightly worst than kNN.

10. Conclusions and Future Work

In this paper, we investigated the problem of automatically discovering a categorization of human faces from a collection of unlabeled face images. Our long term objective is to use face categorization as a precursor step to face recognition to improve recognition accuracy and robustness. For face representation, we employed BoFs using SIFT features. To discover the face categories, we investigated unsupervised learning (i.e., clustering) while to categorize faces in novel images we investigated nearest-neighbor algorithms and supervised learning (i.e., classification). We have reported promising experimental results using the FERET database. For future work, we plan to investigate more powerful clustering algorithms for discovering the face categories, for example, some of the methods discussed in Section 2 for discovering object categories. Second, we plan to perform experiments using additional databases such as XM2VTS,[71] Yale-B,[72] BANCA,[73] and LFW.[74] Finally, we plan to integrate face categorization with recognition. In this regards, we will explore ways to optimize recognition within face categories. This can be done using different recognition algorithms within face categories or applying feature selection to customize face representation within each category.[75]

Acknowledgments

This work was supported by the National Plan for Science and Technology, King Saud University, Riyadh, Saudi Arabia under project number 10-INF1044-02. George Bebis is a Visiting Professor in the College of Computer and Information Sciences at King Saud University, Riyadh 11543, Saudi Arabia.

References

1. K. Veropoulos, G. Bebis, and M. Webster, “Investigating the impact of face categorization on recognition performance”, *International Symposium on Visual Computing*, (LNCS, Vol. 3804), 2005.
2. A. Abate, N. Nappi, D. Riccio, G. Sabatino, “2D and 3D face recognition: A survey”, *Pattern Recognition Letters,* Vol. 28, No. 14, pp. 1885–1906, 2007.
3. X. Tan, S, Chen, Z.-H. Zhou, and F. Zhang, “Face recognition from a single image per person: A survey”, *Pattern Recognition*, Vol. 39, pp. 1725–1745, 2006.
4. W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: Literature survey”, *ACM Computing Surveys*, Vol. 35, No. 4, pp. 399–458, 2003.

5. M. Turk and A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, Vol. 3, pp. 71–86, 1991.
6. P. N. Belhumeur *et al.*, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 7, pp. 711–720, 1997.
7. T. Ahonen, A. Hadid, M. Pietikäinen, "Face description with local binary patterns: Application to face recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, pp. 2037–2041, 2006.
8. J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao, "WLD: A robust local image descriptor", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 9, pp. 1705–1720, Sep. 2010.
9. J. Brigham, "The influence of race on face recognition", *Aspects of Face Processing* (H. Ellis, M. Jeeves and F. Newcombe, eds.), pp. 170–177, 1986.
10. A. O'Toole, J. Peterson, and K. Deffenbacher, "An other-race effect for classifying faces by sex", *Perception*, Vol. 25, pp. 669–676, 1996.
11. P. J. Phillips, F. Jiang, A. Narvekar, J. Ayyad, and A. O'Toole, "An other-race effect for face recognition algorithms", *ACM Transactions on Applied Perception*, Vol. 8, No. 2, 2011.
12. Y. Cheng, A. O'Toole, and H. Abdi, "Classifying adults' and children's faces by sex: Computational investigations of subcategorical feature encoding", *Cognitive Science*, Vol. 25, 2001.
13. J. Y. Baudouin and G. Tiberghien, "Gender is a dimension of face recognition", *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 28(2), pp. 362–365, 2002.
14. R.-S. Lin, D. Ross, and J. Yagnik, "SPEC hashing: Similarity preserving algorithm for entropy-based coding", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 848–854, 2010.
15. D. G. Lowe, "Distinctive image features from scale invariant keypoints", *International Journal of Computer Vision*, Vol. 60, pp. 91–110, 2004.
16. G. Guo, Y. Fu, C. R. Dyer, and T. S. Huang, "Image-based human age estimation by manifold learning and locally adjusted robust regression", *IEEE Transactions on Image Processing*, Vol. 17, No. 7, pp.1178–1188, 2008.
17. X. Lu and A. K. Jain, "Ethnicity identification from face images", *SPIE International Symposium on Defense and Security: Biometric Technology for Human Identification*, 2004.
18. Z. Sun, G. Bebis, X. Yuan, and S. Louis, "Genetic feature subset selection for gender classification: A comparison study", *IEEE Workshop on Applications of Computer Vision*, pp. 165–170, Orlando, December 2002.
19. G. Csurka, C. Dance, L. X. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints", *International Workshop on Statistical Learning in Computer Vision* (in conjunction with ECCV), 2004.
20. L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 524–531, 2005.
21. A. Jain, M. N. Murphy, and P. J. Flynn, "Data clustering: A review", *ACM Computing Surveys*, Vol. 31, No. 3, pp. 264–323, 1999.
22. R. Xu and D. Wunsch II, "Survey of clustering algorithms", *IEEE Transactions on Neural Networks*, Vol. 16, No. 3, pp. 645–678, 2005.
23. N. Dalal, and B. Triggs, "Histogram of oriented gradient for human detection", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886–893, 2005.
24. C. Burges, "Tutorial on support vector machines for pattern recognition", *Data Mining Knowledge Discovery,* Vol. 2, No. 2, pp. 955–974, 1998.

25. P. J. Phillips, M. Hyeonjoon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 10, pp. 1090–1104, 2000.
26. G. Bebis, M. Georgiopoulos, G. Papadourakis and G. Heileman, "Increasing classification accuracy using multiple neural network schemes", *Proceedings of the SPIE,* Vol. 1709, pp. 221–231, 1992.
27. D. Maltoni, A, Jain, and S. Prabhakar, *Handbook on Fingerprint Recognition*, 2nd edition, (Springer-Verlag, 2009).
28. A. Mhatre, S. Palla, S. Chikkerur and V. Govindaraju V, "Efficient search and retrieval in biometric databases", *SPIE Defense and Security Symposium*, Vol. 5779, pp. 265–273, 2005.
29. S. Palla, S. Chikkerur, V. Govindaraju, and P. Rudravaram, "Classification and indexing in large biometric databases", *Biometrics Consortium Conference*, 2004.
30. A. Chaari, S. Lelandais, and M. Ahmed, "A new clustering approach for face identification", *IEEE Conference on Image Processing Theory, Tools, and Applications*, 2008.
31. J. Lu and K. N. Plataniotis, "Boosting face recognition on a large-scale database", *IEEE International Conference on Image Processing*, 2002.
32. F. Perronnin and J.-L. Dugelay, "Clustering face images with application to image retrieval in large databases", Biometric Technology for Human Identification II, *Proceedings of the SPIE*, Vol. 5779, pp. 256–264, 2005.
33. D. Zhang, S. Mabu, F. Wen, and K. Hirasawa, "A sequential subspace face recognition framework using genetic-based clustering", *IEEE Congress on Evolutionary Computation*, pp. 394–400, 2011.
34. F. Perronnin, J.-L. Dugelay, and K. Rose, "Deformable face mapping for person identification", *IEEE International Conference on Image Processing*, Vol. 1, pp. 661–664, 2003.
35. G. Mahalingam and C. Kambhamettu, "Can discriminative cues aid face recognition across age?", *IEEE International Conference on Face and Gesture Recognition*, pp. 206–212, 2011.
36. H. Benrachid and A. Bouroumi, "Unsupervised classification and recognition of human faces", *Applied Mathematical Science*, Vol. 5, No. 41, pp. 2039–2048, 2011.
37. M. A. Mangai and N. A. Gounden, "A subspace-based multi-view face clustering and recognition approach", *IEEE International Conference on Communications and Signal Processing*, 2011.
38. H.-C. Liu, C.-H. Su, Y.-H. Chiang, and Y.-P. Hung, "Personalized face verification system using owner-specific cluster-dependent LDA-subspace", *IEEE International Conference on Pattern Recognition*, Vol. 4, pp. 344–347, 2004.
39. M. Kyperountas, A. Tefas, and I. Pitas, "Face recognition via adaptive discriminant clustering", *IEEE International Conference on Image Processing,* pp. 2744–2747, 2008.
40. S. Eickeler, F. Wallhoff, U. Iurgel, G. Rigoll. "Content-based indexing of images and videos using face detection and recognition methods", *IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, 2001.
41. A.W. Fitzgibbon and A. Zisserman. "Joint manifold distance: A new approach to appearance based clustering", *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 26–36, 2003.
42. J. Tao and Y.-P. Tan, "Face clustering in videos using constraint propagation", *IEEE International Symposium on Circuits and Systems*, 2008.
43. N. Vretos, V. Solachildis, and I. Pitas, "A mutual information based face clustering algorithm for movies", *IEEE International Conference on Multimedia and Expo*, 2006.
44. P. Antonopoulos, N. Nikolaidis and I. Pitas, "Hierarchical face clustering using SIFT image features", *IEEE Symposium on Computational Intelligence in Image and Signal Processing*, 2007.

45. B. Palit, R. Nigam, K. Perlmutter, and S. Perlmutter, “Spectral face clustering”, *IEEE International Conference on Computer Vision Workshops*, 2009.
46. J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, “Discovering object categories in image collections”, *MIT AI Lab Memo, AIM-2005–005*, 2005.
47. T. Hofmann, “Probabilistic latent semantic analysis”, *15th Conference on Uncertainty in Artificial Intelligence*, 1999.
48. D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet allocation”, *Journal of Machine Learning Research*, Vol. 3, No. 4–5, pp. 993–1022, 2003.
49. B. Leibe, K. Mikolajcyk, and B. Schiele, “Efficient clustering and matching for object class recognition”, *British Machine Vision Conference*, 2006.
50. K. Grauman and T. Darell, “Unsupervised learning of categories from sets of partially matching image features*”, IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
51. E. Bart, I. Porteous, P. Perona, and M. Welling, “Unsupervised learning of visual taxonomies”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
52. J. Sivic, B. Russell, A. Zisserman, W. Freeman, and A. Efros, “Unsupervised discovery of visual object class hierarchies”, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.
53. R. Brunelli and T. Poggio, “Face recognition: Features versus templates”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 10, pp. 1042–1052, 1993.
54. J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, “Robust face recognition via sparse representation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 2, pp. 1–18, 2009.
55. K. Mikolajczyk and C. Schmid, “Scale and affine invariant interest point detectors”, *International Journal of Computer Vision*, Vol. 60, No. 1, pp. 63–86, 2004.
56. K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1615–1630, 2005.
57. Z. Li and Z. Tang, “Bayesian face recognition using support vector machine and face clustering”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2004
58. M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, “On the use of SIFT features for face authentication”, *Computer Vision and Pattern Recognition Workshop*, 2006.
59. J. Luo, Y. Ma, E. Takikawa, S. Lao, M. Kawade, and B.-L. Lu, “Person specific SIFT features for face recognition”, *IEEE International Conference on Acoustics, Speech and Signal Processing,* Vol. II, pp. 593–596, 2007.
60. C. Geng and X. Jiang, “SIFT feature for face recognition”, *IEEE International Conference on Computer Science and Information Technology*, pp. 553–698, 2009.
61. M. Asbach, P. Hosten, and M. Unger, “An evaluation of local features for face detection and localization”, IEEE *International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 32–25, 2008.
62. R. Verschae, J. Ruiz-del-Solar and M. Correa, “Face recognition in unconstrained environments: A comparative study*”, Workshop on Faces in* ‘*Real-Life*’ *Images: Detection, Alignment, and Recognition* (in conjunction with ECCV), 2008.
63. P. Dreuw, P. Steingrube H. Hanselmann, and H. Ney “SURF-Face: Face recognition under viewpoint consistency constraints”, *British Machine Vision Conference*, 2009.
64. Z. Li, J. Imai, and M. Kaneko “Robust face recognition using Block-based Bag of Words”, IEEE *International Conference on Pattern Recognition*, 2010.
65. T. Ojala, M. Pietkainen, and D. Harwood, “A comparative study of texture measures with classification based on feature distributions”, *Pattern Recognition,* Vol. 29, pp. 51–59, Jan. 1996.

66. S. O'Hara and B. Draper, "Introduction to the bag of features paradigm for image classification and retrieval", in *Image Rochester NY*, Vol. cs.CV, Cornell University Library (publisher), pp. 1–25, 2011.
67. A. Jain, "Data clustering: 50 years beyond K-means", *Pattern Recognition Letters*, Vol. 31, No. 8, pp. 651–666, 2010.
68. M. Muja and D Lowe, "Fast approximate nearest neighbors with automatic algorithmic configuration", *International Conference on Computer Vision Theory and Application*, pp. 331–340, 2009.
69. S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions", *International Journal of Mathematical Models and Methods in Applied Sciences*, Vol. 1, No. 4, 2007.
70. P. Cunningham and S. Jane Delany, "k-Nearest neighbour classifiers", Technical Report UCD-CSI-2007-4, University College Dublin, Ireland, 2007.
71. K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Matire, "XM2VTSDB: The extended M2TVS database", *International Conference on Audio and Video-based Biometric Person Authentication*, 1999.
72. A. Georghiades, P. Belhumeuer, and D. Kriegman, "From few to many: Illumination cone models for face recognition theoretic framework", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, pp. 643–660, 2001.
73. E. Bailly-Baillire, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Marithoz, J. Matas, K. Messer, V. Popovici, F. Pore, B. Ruiz, and J.-P. Thiran. The BANCA database and evaluation protocol, *International Conference on Audio and Video-based Biometric Person Authentication*, pp. 625–638, 2003.
74. G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments", University of Massachusetts, Amherst, Technical Report 07-49, 2007.
75. Z. Sun, G. Bebis, and R. Miller, "Object detection using feature subset selection", *Pattern Recognition*, Vol. 37, pp. 2165–2176, 2004.