Image and Vision Computing 34 (2015) 27–41

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Editor's choice article

Simultaneous high-dimensional clustering and feature selection using asymmetric Gaussian mixture models[☆]

Tarek Elguebaly [a,*], Nizar Bouguila [b]

[a] Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada, H3G 1T7
[b] The Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada, H3G 1 T7

ARTICLE INFO

Article history:
Received 9 February 2014
Received in revised form 28 August 2014
Accepted 31 October 2014
Available online 3 December 2014

Keywords:
Asymmetric Gaussian distribution
Mixture modeling
Expectation-maximization (EM)
Rival penalized EM (RPEM)
Feature selection
Model selection
Minimum message length (MML)
Scene categorization
Facial expression recognition

ABSTRACT

Finite mixture models are broadly applied in a wide range of applications concerning density estimation and clustering due to their sound mathematical basis and to the interpretability of their results. Indeed, they permit the incorporation of domain knowledge which allows the provision of better insight into the nature of the clusters and then uncovers application-specific desirable patterns that the practitioner is looking for. However, most of the works done on mixture models, when applied to computer vision tasks, assume that per-component data follow a mixture of Gaussians which may not hold as data are generally non-Gaussian (for instance, it is well-known that the distribution of natural images is highly non-Gaussian). The effect of the Gaussian mixture is analogous to the deployment of Euclidean or Mahalanobis type distances for discrimination purposes. Thus, this mixture cannot be applied efficiently in several applications involving asymmetric shapes. In this paper, we overcome this problem by using the asymmetric Gaussian mixture (AGM) model. The AGM can change its shape to model non-symmetrical and heavy tailed real world data which make it a good choice for modeling data with outliers. Modern computer vision applications generally generate complex high-dimensional data and usually, some features are noisy, redundant, or uninformative which may affect the speed and also compromise the accuracy of the used learning algorithm. Therefore, this paper addresses also the problem of unsupervised feature selection when considering AGM models. We propose two approaches for learning the resulting statistical framework. The first approach is based on the minimization of a message length objective and the second one considers rival penalized competitive learning. Our extensive simulations and experiments involving two challenging tasks namely visual scene categorization and facial expression recognition indicate that the method developed in this paper is efficient and has merits.

1. Introduction

Mathematical models in general and statistical approaches in particular have been widely used for the development of useful computer vision, signal and image processing algorithms [1–3]. Many of these approaches are based on finite mixture models (i.e. a weighted sum of distributions) which have been the topic of extensive research in the past [4] and have been applied in several applications such as content-based images categorization and retrieval [5]. In the field of finite mixtures, Gaussian mixture model (GMM) has been widely considered, studied and used [6–8]. However, the Gaussian assumption is rarely justified and met in practice [9] and this is especially true in the case of natural images as shown by several studies and research works [10]. Gaussian density has several drawbacks such as its symmetry around the mean and the rigidity of its shape, which prevent it from having a good approximation to data with outliers. Therefore, in order to model data with different shapes many researchers have considered the generalized Gaussian density (GGD) [11,12]. The GGD is able to model data with various shapes thanks to its shape parameter that controls the tail of the distribution: the larger the value of this parameter is, the flatter is the distribution; the smaller is, the more peaked is the distribution. Despite the higher flexibility that GGD offers, it is still a symmetric distribution inappropriate to model non-symmetrical data. In this article, we suggest the consideration of the asymmetric Gaussian distribution (AGD) capable of modeling heavy and short tailed data [13,14]. The AGD uses two variance parameters for left and right parts of the distribution, which allow it not only to approximate a large class of statistical distributions but also to include the asymmetry. An important part of the mixture modeling problem concerns learning the model parameters and determining the number of consistent components (M) which best describes the data.

Concerning parameters estimation, the most popular approach is perhaps the one based on the maximization of the likelihood function through the expectation maximization (EM) framework [15]. It is well-known that the EM algorithm needs an appropriate predefined

[☆] Editor's Choice Articles are invited and handled by a select rotating 12 member Editorial Board committee. This paper has been recommended for acceptance by Dr. Todorovic.

[*] Corresponding author. Tel.: +1 5148482424; fax: +1 5148483171.
E-mail addresses: t_elgue@encs.concordia.ca (T. Elguebaly), nizar.bouguila@concordia.ca (N. Bouguila).

http://dx.doi.org/10.1016/j.imavis.2014.10.011
0262-8856/© 2014 Elsevier B.V. All rights reserved.

number of clusters. Therefore, in the past decades, a lot of research has been devoted to the automatic selection of the number of clusters which best describes a given data set and a lot of selection criteria have been proposed such as Akaike's information criterion (AIC), minimum description length (MDL), Laplace empirical criterion (LEC), and minimum message length (MML) [4,16]. In particular, the MML criterion has been shown to outperform the majority of existing selection criteria. Thus, we shall consider it in this work by comparing it to another approach based on the rival penalized competitive learning (RPCL) algorithm which has received a lot of attention [17,18]. The RPCL algorithm allows automatic selection of the number of clusters during learning via penalizing the rival in competition. Its basic idea is that for each input not only the winner of the input sample is updated to adapt to the input, but also its rival is de-learned by a smaller de-learning rate. Many experiments have shown that the RPCL can indeed automatically select the correct cluster number by gradually driving extra seed points far away from the input data set. However, its performance is sensitive to the selection of the de-learning rate, such that if it is not well selected, the RPCL may completely break down. In order to overcome this problem, the rival penalized controlled competitive learning (RPCCL) was introduced in [19]. This algorithm sets the de-learning rate at the same value as the learning rate, then dynamically adjust it based on the relative distance of the winner to the rival and the current input, respectively. In Ref. [20], the rival penalized EM (RPEM) algorithm was proposed for mixture-based clustering. The RPEM learns the model parameters by making the mixture components compete with each other at each time step; this can be done by not only updating the winning density component parameters to adapt to the input but also all rivals's parameters are penalized with the strength proportional to the corresponding posterior density probabilities. Therefore, the RPEM is able to automatically select an appropriate number of densities by fading out the redundant densities from a density mixture which can save the computing time. Thus, we propose to use the RPEM algorithm to perform model selection and parameters learning together in a single step for the AGM model.

Modern computer vision application generates high-dimensional vectors. Handling data defined in high-dimensional feature spaces is a difficult problem [21]. Theoretically, the more information we have about each pattern, the better a learning algorithm is expected to perform. However, in many cases, some features can be noisy or uninformative which can degrade clustering efficiency [22]. Thus, in order to achieve a good performance of data modeling, irrelevant features have to be discarded. An accurate feature selection (FS), the task of choosing the best feature subset, allows the improvement of the understandability, scalability, and accuracy of the resulting learned models that generalize better to unseen data. Indeed, several recent studies have shown that selecting relevant features allows more meaningful modeling results [23,24]. However, the problem is challenging especially in unsupervised settings because of the absence of class labels that could guide the selection process [25]. Therefore, there have been only few feature selection techniques that have been applied in mixture-based clustering [26–28] since the aim is to identify simultaneously two inter-related unknowns that are optimal feature subset and optimal number of clusters. In this article, and following recent approaches (see, for instance [26–28]), we perform unsupervised feature selection approach by casting it as an estimation problem, thus avoiding any combinatorial search. For each feature, we associate a relevance weight which measures the degree of its dependence with class labels.

The remainder of this paper is organized as follows: After the introduction we first describe our AGM model and then detail our feature selection approach in Section 2. In Section 3, we address the issue of identifying the model's order using the minimum message length approach. In Section 4, we integrate the concept of feature saliency into the RPEM algorithm for the AGM model. The subsequent Section 5 demonstrates some computer simulation and experimental results on challenging applications. Finally, the paper closes with a summary of the work and concluding remarks.

2. The AGM model

Formally we say that a D-dimensional random variable, the image feature vector in our case, $\vec{X} = [X_1, \ldots, X_D]^T$ follows a M-component mixture distribution if its probability density function can be written in the following form:

$$p\left(\vec{X}|\Theta\right) = \sum_{j=1}^{M} p_j p\left(\vec{X}|\xi_j\right) \tag{1}$$

where ξ_j is the set of the parameters of the jth component, $\{p_j\}$ are the mixing proportions which must be positive and sum to one, $\Theta = \{p_1, \ldots, p_M, \xi_1, \ldots, \xi_M\}$ is the complete set of parameters fully characterizing the mixture, $M \geq 1$ is the number of components in the mixture. For the AGM, each component density $p\left(\vec{X}|\xi_j\right)$ is an asymmetric Gaussian distribution (AGD):

$$p\left(\vec{X}|\xi_j\right) = \prod_{d=1}^{D} \sqrt{\frac{2}{\pi}} \frac{1}{\left(\sigma_{l_{jd}} + \sigma_{r_{jd}}\right)} \times \begin{cases} exp\left[-\frac{\left(X_d - \mu_{jd}\right)^2}{2\sigma^2_{l_{jd}}}\right] & \text{if } X_d < \mu_{jd} \\ exp\left[-\frac{\left(X_d - \mu_{jd}\right)^2}{2\sigma^2_{r_{jd}}}\right] & \text{if } X_d \geq \mu_{jd} \end{cases} \tag{2}$$

where $\xi_j = \left(\vec{\mu}_j, \vec{\sigma}_{l_j}, \vec{\sigma}_{r_j}\right)$ is the set of the parameters of component j where $\vec{\mu}_j = \left(\mu_{j1}, \ldots, \mu_{jD}\right)$, $\vec{\sigma}_{l_j} = \left(\vec{\sigma}_{l_{j1}}, \ldots, \vec{\sigma}_{l_{jD}}\right)$, and $\vec{\sigma}_{r_j} = \left(\vec{\sigma}_{r_{j1}}, \ldots, \vec{\sigma}_{r_{jD}}\right)$ are the mean, the left standard deviation, and the right standard deviation of the D-dimensional AGD, respectively. The AGD is chosen to be able to fit, in analytically simple and realistic way, symmetric or non-symmetric data by the combination of the left and right variances.

Let $\mathcal{X} = \left(\vec{X}_1, \ldots, \vec{X}_N\right)$ be a set of N independent and identically distributed vectors, assumed to arise from a finite AGM model with M components. Thus, its likelihood function can be expressed as follows:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{M} p_j p\left(\vec{X}_i|\xi_j\right) \tag{3}$$

We introduce stochastic indicator variables, $Z_i = (Z_{i1}, \ldots, Z_{iM})$, one for each observation, whose role is to encode to which component the observation belongs. In other words, Z_{ij}, the unobserved or missing vector, equals 1 if $\vec{X}_i$ belongs to class j and 0, otherwise. The complete-data likelihood for this case is then:

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left(p_j p\left(\vec{X}_i|\xi_j\right)\right)^{Z_{ij}}. \tag{4}$$

where $Z = \{Z_1, \ldots, Z_N\}$. Taking the logarithm of Eq. (4) we can get the complete data log-likelihood by:

$$logp(\mathcal{X}, Z|\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{ij} log\left[p_j p\left(\vec{X}_i|\xi_j\right)\right] \tag{5}$$

The expectation-maximization (EM) algorithm is the main framework to find the maximum likelihood estimate of the parameters of a probabilistic data generation process characterized by the presence of incomplete (or missing) data in general and the parameters of an underlying finite mixture model in particular.

It is noteworthy that the previous model in Eq. (1) supposes actually that the D features have the same importance and carry pertinent information which is not generally the case, since many of which can be irrelevant for the classification task. In order to take into account the potential presence of irrelevant features, it is possible to represent irrelevant features by background Gaussian distribution with parameter $\overrightarrow{\lambda} = \left\{\overrightarrow{\eta}, \overrightarrow{\delta}\right\}$ for all classes, where $\overrightarrow{\eta} = (\eta_1, \ldots, \eta_D)$ and $\overrightarrow{\delta} = (\delta_1, \ldots, \delta_D)$ represent the mean and standard deviation of the Gaussian distribution, respectively. We adopt the feature relevancy approach suggested in Ref. [26] in the case of the finite Gaussian mixture, because it is suitable for unsupervised learning. The main idea is to consider the dth feature as irrelevant if its distribution is independent of the class labels and can follow our common Gaussian density $p(X_d|\lambda_d)$. Then, the mixture density in Eq. (1) can be written as:

$$p\left(\overrightarrow{X}|\Theta, \overrightarrow{\lambda}, \overrightarrow{\varphi}\right) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} p\left(X_d|\xi_{jd}\right)^{\varphi_d} p(X_d|\lambda_d)^{1-\varphi_d} \tag{6}$$

where $\lambda_d = (\eta_d, \delta_d)$ and $\overrightarrow{\varphi} = [\varphi_1, \ldots, \varphi_D]^T$ is a set of binary parameters, such that $\varphi_d = 1$ if the dth feature is relevant and $\varphi_d = 0$, otherwise. Note that, $\{\varphi_d\}$ can be considered as missing variables. Thus, the resulting model can be given by Ref. [26]:

$$p\left(\overrightarrow{X}|\Theta_M\right) = \sum_{j=1}^{M} p_j \prod_{d=1}^{D} \left[\omega_d p\left(X_d|\xi_{jd}\right) + (1-\omega_d) p(X_d|\lambda_d)\right] \tag{7}$$

where $\Theta_M = \left\{\Theta, \overrightarrow{\omega}, \overrightarrow{\lambda}\right\}$ is the complete set of parameters fully characterizing the mixture. We suppose that not all the features of an observation are important, through the weight relevancy of these features. That is, the weight is denoted as $\omega = [\omega_1, \ldots, \omega_D]^T$ with $0 \leq \omega_d \leq 1$, where ω_d represents the probability that the dth feature is relevant to all the clusters ($\omega_d = p(\varphi_d = 1)$). Therefore, the irrelevant features have little contribution to a given cluster in the subspace, thus their distributions are common to all the clusters in this case. We finally note that the previous model is reduced to the one in Eq. (1) when all the features are considered as relevant.

3. Learning via EM and MML

First, we suppose that the number of mixture components M is known and we use the EM algorithm to estimate the model's parameters. Then, we use the MML criterion to choose the optimal number of classes M.

3.1. Maximum likelihood estimation of the mixture parameters

By treating M as known, we can derive the following EM algorithm for parameters estimation:

- Expectation step:

$$h\left(j|\overrightarrow{X}_i, \Theta_M\right) = \frac{p_j \prod_{d=1}^{D} \zeta_{ijd}}{\sum_{j=1}^{M} p_j \prod_{d=1}^{D} \zeta_{ijd}} \tag{8}$$

where $\zeta_{ijd} = \omega_d p(X_{id}|\xi_{jd}) + (1-\omega_d) p(X_{id}|\lambda_d)$

- Maximization step:

$$p_j^{new} = \frac{\sum_{i=1}^{N} h\left(j|\overrightarrow{X}_i, \Theta_M\right)}{N} \tag{9}$$

$$\mu_{jd}^{new} = \frac{\sum_{i=1}^{N} \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} h\left(j|\overrightarrow{X}_i, \Theta_M\right) X_{id}}{\sum_{i=1}^{N} \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} h\left(j|\overrightarrow{X}_i, \Theta_M\right)} \tag{10}$$

$$\sigma_{l_{jd}}^{new} = \sigma_{l_{jd}}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jd}}^2}\right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jd}}}\right)\right] \tag{11}$$

$$\sigma_{r_{jd}}^{new} = \sigma_{r_{jd}}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jd}}^2}\right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jd}}}\right)\right] \tag{12}$$

$$\eta_d^{new} = \frac{\sum_{i=1}^{N} \left[\sum_{j=1}^{M} \frac{(1-\omega_d) p(X_{id}|\lambda_d)}{\zeta_{ijd}} h\left(j|\overrightarrow{X}_i, \Theta_M\right)]\right] X_{id}}{\sum_{i=1}^{N} \sum_{j=1}^{M} \frac{(1-\omega_d) p(X_{id}|\lambda_d)}{\zeta_{ijd}} h\left(j|\overrightarrow{X}_i, \Theta_M\right)} \tag{13}$$

$$\delta_d^{2^{new}} = \frac{\sum_{i=1}^{N} \left[\sum_{j=1}^{M} \frac{(1-\omega_d) p(X_{id}|\lambda_d)}{\zeta_{ijd}} h\left(j|\overrightarrow{X}_i, \Theta_M\right)\right] (X_{id}-\eta_d)^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} \frac{(1-\omega_d) p(X_{id}|\lambda_d)}{\zeta_{ijd}} h\left(j|\overrightarrow{X}_i, \Theta_M\right)} \tag{14}$$

$$\omega_d^{new} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} h\left(j|\vec{X}_i, \Theta_M\right)}{N} \tag{15}$$

where $\mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)$ is the model's complete data log-likelihood $\mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)$, and we have

$$\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jd}}} = \sum_{i=1}^{N} \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} h\left(j|\vec{X}_i, \Theta_M\right) \vartheta_{l_{ijd}}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jd}}^2} = \sum_{i=1}^{N} \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} h\left(j|\vec{X}_i, \Theta_M\right) \vartheta_{l_{ijd}} \left[\rho_{l_{ijd}} + \vartheta_{l_{ijd}} \frac{(1-\omega_d) p(X_{id}|\lambda_d)}{\zeta_{ijd}}\right]$$

$$\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jd}}} = \sum_{i=1}^{N} \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} h\left(j|\vec{X}_i, \Theta_M\right) \vartheta_{r_{ijd}}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jd}}^2} = \sum_{i=1}^{N} \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} h\left(j|\vec{X}_i, \Theta_M\right) \vartheta_{r_{ijd}} \left[\rho_{r_{ijd}} + \vartheta_{r_{ijd}} \frac{(1-\omega_d) p(X_{id}|\lambda_d)}{\zeta_{ijd}}\right]$$

where

$$\vartheta_{l_{ijd}} = \begin{cases} \frac{\left(X_{id}-\mu_{jd}\right)^2}{\sigma_{l_{jd}}^3} - \frac{1}{\left(\sigma_{l_{jd}} + \sigma_{r_{jd}}\right)} & \text{if } X_{id} < \mu_{jd} \\ -\frac{1}{\left(\sigma_{l_{jd}} + \sigma_{r_{jd}}\right)} & \text{if } X_{id} \geq \mu_{jd} \end{cases} ; \quad \vartheta_{r_{ijd}} = \begin{cases} -\frac{1}{\left(\sigma_{l_{jd}} + \sigma_{r_{jd}}\right)} & \text{if } X_{id} < \mu_{jd} \\ \frac{\left(X_{id}-\mu_{jd}\right)^2}{\sigma_{r_{jd}}^3} - \frac{1}{\left(\sigma_{l_{jd}} + \sigma_{r_{jd}}\right)} & \text{if } X_{id} \geq \mu_{jd} \end{cases}$$

$$\rho_{l_{ijd}} = \begin{cases} \frac{1}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)^2} - \frac{3\left(X_{id}-\mu_{jd}\right)^2}{\sigma_{l_{jd}}^4} & \text{if } X_{id} < \mu_{jd} \\ \frac{1}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)^2} & \text{if } X_{id} \geq \mu_{jd} \end{cases} \quad \rho_{r_{ijd}} = \begin{cases} \frac{1}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)^2} & \text{if } X_{id} < \mu_{jd} \\ \frac{1}{\left(\sigma_{l_{jd}}+\sigma_{r_{jd}}\right)^2} - \frac{3\left(X_{id}-\mu_{jd}\right)^2}{\sigma_{r_{jd}}^4} & \text{if } X_{id} \geq \mu_{jd} \end{cases}$$

Note that the gradients with respect to $\sigma_{l_{jd}}$ and $\sigma_{r_{jd}}$ are non-linear, therefore we have decided to use the Newton–Raphson method, based on the first and second gradients of $\mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)$, for estimation.

3.2. Model selection using MML

Generally, the maximum likelihood estimate favors higher values for *M* which leads to overfitting. Therefore, a model selection criterion is needed to estimate the number of components of a mixture model. The MML approach is based on evaluating statistical models according to their ability to compress a message containing the data (minimum coding length criterion). High compression is obtained by building a short code for your data. Therefore, in the case of MML, the optimal number of classes in the mixture is obtained by minimizing the following cost function [29]:

$$MessLens \approx -log p(\theta_M) + \frac{c}{2}\left(1 + log\frac{1}{12}\right) + \frac{1}{2} log\left|I(\Theta_M)\right| - log p(\mathcal{X}|\theta_M) \tag{16}$$

where $p(\theta_M)$, $I(\Theta_M)$, and $p(\mathcal{X}|\theta_M)$ denote the prior distribution, the Fisher information matrix, and the likelihood, respectively. The constant $c = M + D + 3DM + 2D$ represents the total number of parameters, and |. | denotes the determinant. Note that the information matrix of the model is very difficult to obtained analytically, therefore, we assume the independence of the different groups of parameters, which allows the factorization of both $p(\theta_M)$ and $|I(\Theta_M)|$. Furthermore, we approximate the Fisher information $|I(\Theta_M)|$ using the complete likelihood which assumes labeled observations. Additionally, since we have no knowledge about the parameters, we adopt the uninformative Jeffrey's prior for each group of parameters as prior distribution. From this, we obtain the following objective:

$$MessLens \approx \frac{c}{2}\left(1 + log\frac{1}{12}\right) + \frac{c}{2}(logN) + \frac{3M}{2}\sum_{d=1}^{D} log\omega_d + \frac{3D}{2}\sum_{j=1}^{M} log p_j + \sum_{d=1}^{D} log(1-\omega_d) - log p(\mathcal{X}|\theta_M) \tag{17}$$

which we minimize under the constraints $0 < p_j \leq 1, 0 < \omega_d \leq 1$, and $\sum_{j=1}^{M} p_j = 1$ in a manner similar to the one followed in Ref. [26]. In order to use the MML approached the EM algorithm undergoes a minor modification in the calculation of the mixing proportions p_j and the feature relevancy ω_d:

$$p_j^{new} = \frac{max\left(\sum_{i=1}^{N} h\left(j|\vec{X}_i, \Theta_M\right) - \frac{3D}{2}, 0\right)}{\sum_{j=1}^{M} max\left(\sum_{i=1}^{N} h\left(j|\vec{X}_i, \Theta_M\right) - \frac{3D}{2}, 0\right)} \tag{18}$$

$$\omega_d^{new} = \frac{max\left(\sum_{i=1}^{N}\sum_{j=1}^{M} a_{ijd} - \frac{3M}{2}, 0\right)}{max\left(\sum_{i=1}^{N}\sum_{j=1}^{M} a_{ijd} - \frac{3M}{2}, 0\right) + max\left(\sum_{i=1}^{N}\sum_{j=1}^{M} b_{ijd} - 1, 0\right)} \tag{19}$$

where

$$a_{ijd} = h\left(j|\vec{X}_i, \Theta_M\right) \frac{\omega_d p\left(X_{id}|\xi_{jd}\right)}{\zeta_{ijd}} \tag{20}$$

$$b_{ijd} = h\left(j|\vec{X}_i, \Theta_M\right) \frac{(1-\omega_d) p(X_{id}|\lambda_d)}{\zeta_{ijd}} \tag{21}$$

3.3. The complete learning algorithm

The following script summarizes the main steps of the algorithm used for the AGM parameters estimation and model selection

```
1. Initialize $\Theta_M$:
   • The feature relevancy is set to $\omega_d = 0.5$.
   • The number of parameters $M = M_{max} = 10$.
   • The AGM parameters $\Theta$ are initialized using the Fuzzy C-means. Note that, we initialized both the left and right standard deviations with the standard deviation values obtained from the Fuzzy C-means.
   • Perform the common Gaussian density $\vec{\lambda}$ parameters estimation to cover the whole data.
2. Implement the EM + MML approach
   While $M < M_{max}$ do {
   (a) While not converged do {
       i. Perform E-Step according to Eq. (8)
       ii. Perform M-Step according to Eqs. (10) to (14), (18) and (19).
       iii. If $p_j = 0$, Then the $j^{th}$ component is eliminated.
       iv. If $\omega_d = 0$, Then the $(X_{id}|\xi_{jd})$ is eliminated.
       v. If $\omega_d = 1$, Then the $p(X_{id}|\lambda_d)$ is eliminated.
       }End While
   (b) Calculate the associated message length Eq. (17).
   (c) Remove the component $j$ with smallest $p_j$.
   }End While
3. Return the model parameters with the smallest message length.
```

4. Learning via RPEM

The main learning purpose is to estimate the parameters Θ_M from N observations, denoted as $\mathcal{X} = \left[\vec{X}_1, \ldots, \vec{X}_N\right]$. Then, the rival penalized EM (RPEM) algorithm [20,30] is developed from the maximum weighted likelihood framework via maximizing the following weighted likelihood:

$$Q(\Theta_M, \mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{M}\left(\Theta_M, \vec{X}_i\right) \tag{22}$$

with

$$\begin{aligned}\mathcal{M}\left(\Theta_M, \vec{X}_i\right) = & \sum_{j=1}^{M} g\left(j|\vec{X}_i, \Theta_M\right) ln\left\{p_j \prod_{d=1}^{D}\left[\omega_d p\left(X_{id}|\xi_{jd}\right) + (1-\omega_d) p(X_{id}|\lambda_d)\right\}\right] \\ & - \sum_{j=1}^{M} g\left(j|\vec{X}_i, \Theta_M\right) ln h\left(j|\vec{X}_i, \Theta_M\right)\end{aligned}$$

where $g\left(j|\vec{X}_i, \Theta_M\right)$ is a designable weight, which can embody the model selection in the learning process, satisfying the two constraints: $\sum_{j=1}^{M} g\left(j|\vec{X}_i, \Theta_M\right) = 1$ and $\forall j g\left(j|\vec{X}_i, \Theta_M\right) = 0$ if $h\left(j|\vec{X}_i, \Theta_M\right) = 0$. Thus, the weight $g\left(j|\vec{X}_i, \Theta_M\right)$ can be expressed by:

$$g\left(j|\vec{X}_i, \Theta_M\right) = (1+\varepsilon) I\left(j|\vec{X}_i, \Theta_M\right) - \varepsilon h\left(j|\vec{X}_i, \Theta_M\right) \tag{23}$$

where ε is a small positive quantity which we took as 1. Moreover,

$$I\left(j|\vec{X}_i, \Theta_M\right) = \begin{cases} 1 & \text{if } j = c \\ 0 & \text{if } j \neq c \end{cases} \tag{24}$$

and $c = arg\,max_{\{1 \leq j \leq M\}}\ h\left(j|\vec{X}_i, \Theta_M\right)$. Thus, the feature weighted RPEM (FW-RPEM) algorithm for the AGM can be implemented in the following steps:

1. Initialize Θ_M:
   - The feature relevancy is set to $\omega_d = 0.5$.
   - The number of parameters $M = M_{max} = 10$.
   - The AGM parameters Θ are initialized using the Fuzzy C-means algorithm. Note that, we initialized both the left and right standard

deviations with the standard deviation values obtained from the Fuzzy C-means.

2. Perform the common Gaussian density $\overrightarrow{\lambda}$ parameters estimation:

$$\eta_d = \frac{1}{N}\sum_{i=1}^{N} X_{id} \tag{25}$$

$$\delta_d^2 = \frac{1}{N}\sum_{i=1}^{N} (X_{id}-\eta_d)^2 \tag{26}$$

3. Repeated until convergence for each $\overrightarrow{X}_i, i = [1, \ldots N]$
   - Expectation step

$$\begin{aligned} h\left(j|\overrightarrow{X}_i, \Theta_M\right) &= \frac{p_j \prod_{d=1}^{D} \zeta_{ijd}}{\sum_{j=1}^{M} p_j \prod_{d=1}^{D} \zeta_{ijd}} \\ g\left(j|\overrightarrow{X}_i, \Theta_M\right) &= \begin{cases} 2-h\left(j|\overrightarrow{X}_i, \Theta_M\right) & \text{if } j = c \\ -h\left(j|\overrightarrow{X}_i, \Theta_M\right) & \text{if } j \neq c \end{cases} \end{aligned} \tag{27}$$

   - Maximization step

$$\beta_j^{new} = \beta_j^{old} + \gamma_\beta \frac{\partial \mathcal{M}\left(\Theta_M, \overrightarrow{X}_i\right)}{\partial \beta_j}\bigg|_{\Theta_M^{old}} = \beta_j^{old} + \gamma_\beta\left(g\left(j|\overrightarrow{X}_i, \Theta_M\right) - p_j^{old}\right) \tag{28}$$

$$\mu_{jd}^{new} = \mu_{jd}^{old} + \gamma \frac{\partial \mathcal{M}\left(\Theta_M, \overrightarrow{X}_i\right)}{\partial \mu_{jd}}\bigg|_{\Theta_M^{old}} = \mu_{jd}^{old} + \gamma g\left(j|\overrightarrow{X}_i, \Theta_M\right) \frac{\omega_d^{old}}{\zeta_{ijd}} \frac{\partial p\left(X_{id}|\xi_{jd}^{old}\right)}{\partial \mu_{jd}} \tag{29}$$

$$S_{l_{jd}}^{new} = S_{l_{jd}}^{old} + \gamma \frac{\partial \mathcal{M}\left(\Theta_M, \overrightarrow{X}_i\right)}{\partial S_{l_{jd}}}\bigg|_{\Theta_M^{old}} = S_{l_{jd}}^{old} + \gamma g\left(j|\overrightarrow{X}_i, \Theta_M\right) \frac{\omega_d^{old}}{\zeta_{ijd}} \frac{\partial p\left(X_{id}|\xi_{jd}^{old}\right)}{\partial S_{l_{jd}}} \tag{30}$$

$$S_{r_{jd}}^{new} = S_{r_{jd}}^{old} + \gamma \frac{\partial \mathcal{M}\left(\Theta_M, \overrightarrow{X}_i\right)}{\partial S_{r_{jd}}}\bigg|_{\Theta_M^{old}} = S_{r_{jd}}^{old} + \gamma g\left(j|\overrightarrow{X}_i, \Theta_M\right) \frac{\omega_d^{old}}{\zeta_{ijd}} \frac{\partial p\left(X_{id}|\xi_{jd}^{old}\right)}{\partial S_{r_{jd}}} \tag{31}$$

$$\begin{aligned} &\omega_d^{new} = \omega_d^{old} + \gamma_\omega \frac{\partial \mathcal{M}\left(\Theta_M, \overrightarrow{X}_i\right)}{\partial \omega_d}\Bigg|_{\Theta_M^{old}} = \omega_d^{old} + \gamma_\omega \sum_{j=1}^{M} g\left(j|\overrightarrow{X}_i, \Theta_M\right) \frac{\upsilon_{ijd}}{\zeta_{ijd}} \\ &\mathit{if} \quad \omega_d > 1 \quad \mathit{then} \quad \omega_d = 1 \\ &\mathit{if} \quad \omega_d < 0 \quad \mathit{then} \quad \omega_d = 0 \end{aligned} \tag{32}$$

where

$$\begin{aligned} \frac{\partial p\left(X_{id}|\xi_{jd}^{old}\right)}{\partial \mu_{jd}} &= p\left(X_{id}|\xi_{jd}^{old}\right)\kappa_{ijd}; \quad \frac{\partial p\left(X_{id}|\xi_{jd}^{old}\right)}{\partial S_{l_{jd}}} \\ &= p\left(X_{id}|\xi_{jd}^{old}\right)\tau_{l_{ijd}}; \frac{\partial p\left(X_{id}|\xi_{jd}^{old}\right)}{\partial S_{r_{jd}}} = p\left(X_{id}|\xi_{jd}^{old}\right)\tau_{r_{ijd}} \end{aligned} \tag{33}$$

$$S_{l_{jd}} = \frac{1}{\sigma_{l_{jd}}^2} \quad S_{r_{jd}} = \frac{1}{\sigma_{r_{jd}}^2}; \quad p_j = \frac{exp\left(\beta_j\right)}{\sum_{j=1}^{M} exp\left(\beta_j\right)} \quad \mathit{for} \quad 1 \le j \le M \tag{34}$$

$$\upsilon_{ijd} = p\left(X_{id}|\xi_{jd}\right) - p(X_{id}|\lambda_d); \quad \kappa_{ijd} = \begin{cases} S_{l_{jd}}\left(X_{id}-\mu_{jd}\right) & \text{if } X_{id} < \mu_{jd} \\ S_{r_{jd}}\left(X_{id}-\mu_{jd}\right) & \text{if } X_{id} \ge \mu_{jd} \end{cases} \tag{35}$$

$$\tau_{l_{ijd}} = \begin{cases} \frac{1}{2}\left[\frac{S_{r_{jd}}^{1/2}}{S_{l_{jd}}\left(S_{l_{jd}}^{1/2} + S_{r_{jd}}^{1/2}\right)} - \left(X_{id}-\mu_{jd}\right)^2\right] & \text{if } X_{id} < \mu_{jd} \\ \frac{S_{r_{jd}}^{1/2}}{2S_{l_{jd}}\left(S_{l_{jd}}^{1/2} + S_{r_{jd}}^{1/2}\right)} & \text{if } X_{id} \ge \mu_{jd} \end{cases} \tag{36}$$

$$\tau_{r_{ijd}} = \begin{cases} \frac{S_{l_{jd}}^{1/2}}{2S_{r_{jd}}\left(S_{l_{jd}}^{1/2} + S_{r_{jd}}^{1/2}\right)} & \text{if } X_{id} < \mu_{jd} \\ \frac{1}{2}\left[\frac{S_{l_{jd}}^{1/2}}{S_{r_{jd}}\left(S_{l_{jd}}^{1/2} + S_{r_{jd}}^{1/2}\right)} - \left(X_{id}-\mu_{jd}\right)^2\right] & \text{if } X_{id} \ge \mu_{jd} \end{cases} \tag{37}$$

Note that, we have used the inverse variances (S_l and S_r) in order to ensure that the learning of the model standard deviations are more stable. Also, the learning rates γ_β, γ, and γ_ω are taken as 0.0001, 0.001, and 0.0001, respectively.

5. Experimental results

In this section, the effectiveness of the two proposed frameworks for learning the AGM model with feature selection are tested on two real-world applications namely scene categorization and facial expression recognition.

5.1. Scene categorization

One of the most impressive feats of the human visual system is how rapidly, accurately and comprehensively it can recognize and understand a complex scene [31]. This remarkable ability, known as “visual recognition”, has recently drawn considerable interest and has been successfully applied in various applications such as the automatic understanding of images, object recognition, image database browsing and content-based image annotation, suggestion and retrieval [32–40]. In this section, we build a method for recognizing scene categories by imitating the human perception. Thus, our approach can be divided into three main components: feature extraction, image representation, and scene classification. In feature extraction, we normalize the images then we represent each image by a collection of local image patches. In addition, we scan local image patches and extract their low level feature vectors. Then, we use the bag of visual words (BOW) approach to have an overall representation for each image [41]. The last step in our image representation step is to apply a probabilistic Latent Semantic Analysis (pLSA) to the obtained histograms to represent each image by a D-dimensional vector where D is the number of latent aspects [42]. Our final goal is to classify the overall image to its right group using our AGM model.

The remainder of this section is organized as follows, first we represent some related works for scene classification. Then, we introduce the databases used in this application. Later, we describe the three components of our algorithm. Finally, we evaluate the performance of our algorithm for scene classification.

5.1.1. Related works

Classifying images into semantic types of scenes [43–51] is a classical image understanding problem. Automatic techniques for recognizing scenes have an enormous impact for improving the performance of other computer vision applications such as browsing, retrieval and object recognition. However, scenes classification is not an easy task owing to their variability, ambiguity, and the wide range of illumination and scale conditions that may apply. In Ref. [52], the authors presented a stratified approach to both binary (outdoor–indoor) and multiple

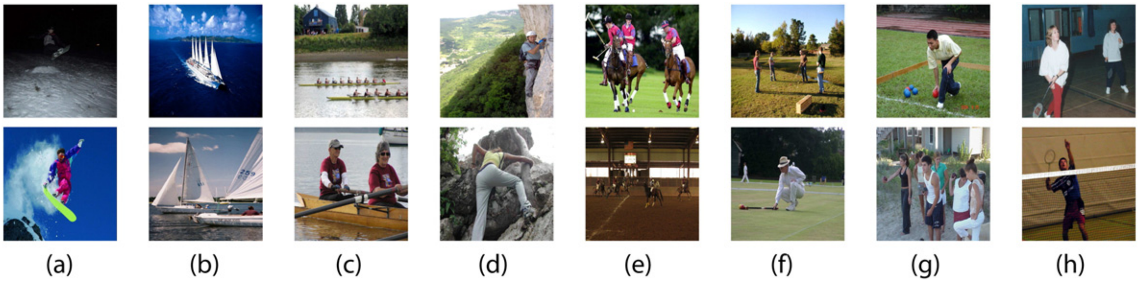

Fig. 1. Sample images from the UIUC sports event data set; (a) Snow-boarding, (b) Sailing, (c) Rowing, (d) Rock-climbing, (e) Polo, (f) Croquet, (g) Bocce, and (h) Badminton.

category of scene classification. Their idea was to learn mixture models for 20 basic classes of local image content based on color and texture information. Then, they applied these models to the test image in order to produce 20 probability density response maps (PDRMs) indicating the likelihood that each image region was produced by each class. Later, they extracted some very simple features from those PDRMs, and used them to train a bagged LDA classifier for 10 scene categories. In Ref. [43], the authors used a simplified low-level feature set to predict multiple semantic scene attributes that are integrated probabilistically to obtain a final indoor/outdoor scene classification. An initial indoor/outdoor prediction is obtained by using support vector machines for classifying computationally efficient, low-dimensional color and wavelet texture features. Furthermore, they used the same low-level features to explicitly predict the presence of semantic features including grass and sky. The semantic scene attributes are then integrated using a Bayesian network designed for improved indoor/outdoor scene classification. In Ref. [53], the authors proposed a hierarchical generative model that classifies the overall scene, recognizes and segments each object component, as well as annotates the image with a list of tags. Visually relevant objects are represented by regions and patches, while visually irrelevant textual annotations are influenced directly by the overall scene class. However, these methods use manually annotated patches which may be time consuming and unpractical. The authors in Ref. [42] proposed the use of pLSA to discover object categories in images using the bag-of-words document representation. Then, they used the nearest neighbor classifier for scenes classification. Our research efforts are focused on extending this last approach in order to construct a robust system capable of classifying images into different scenes.

5.1.2. Scene databases

In this section, we test our approach on two well-known data sets. Our first dataset contains 1579 diverse scene images from 8 categories [53]: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). This data set is very challenging because most images have highly cluttered and diverse background, and object classes are highly diverse. In addition, object sizes and poses are very different in each image. The second data set contains 15 categories of natural scenes [54,55]: highway (260 images), inside of cities (308 images), tall buildings (356 images), streets (292 images), suburb residence (241 images), forest (328 images), coast (360 images), mountain (374 images), open country (410 images), bedroom (174 images), kitchen (151 images), livingroom (289 images), office (216 images), store (315 images), and industrial (311 images). The major sources of the pictures in the data set include the COREL collection, personal photographs, and Google image search. The average size of each image is approximately 250 × 300 pixels. Figs. 1 and 2 show example images from the two data sets under consideration.

5.1.3. Feature extraction and image representation

Representing an image by a collection of local image patches of certain size has become very popular and achieved certain success in visual recognition, image retrieval, scene modeling/categorization, etc., due to its robustness to occlusions, geometric deformations and illumination variations. In addition, the strategy of dense sampling has been shown to provide better performance than interest points for scene classification [54]. Furthermore, SIFT descriptor is robust to illumination, clutter and scale changes. Therefore in this paper we use dense SIFT descriptors of 16 × 16 pixel patches computed over a grid with spacing of 8 pixels.

Inspired by the huge success of the bag of words (BOW) method in scenes classification, we decided to employ it in order to represent each image by a feature vector [41]. In order to build the dictionary of the bag of words, also known as codebook, we use a K-means algorithm to cluster our training-set descriptors in a vocabulary of V visual words. Then, for each SIFT point in an input image, the nearest neighbor in the vocabulary is calculated; based on this statistics a V-dimensional feature vector is built collecting the number of points in the image that can be approximated by each of the V visual words. Thus, each image can be represented as a frequency histogram over the V visual words. Then, we apply the pLSA model to the bag of visual words representation which allows the description of each image as a D-dimensional vector, where D is the number of aspects (or learned topics). In order to evaluate the sensitivity of our approaches for the chosen V and D, we have applied our algorithms for various V and D values. From Fig. 3, we notice that the classification accuracy does not change much after using fifty topics because our feature selection approach identified most of the extra features as irrelevant. Therefore, we fixed V and D for the rest of the evaluations to 900 and 50, respectively.

5.1.4. Scene classification and results

The goal of classification is to estimate the most likely scene class. While classification might be easy for human beings it is very hard for machines. Recently, several classification methods were introduced

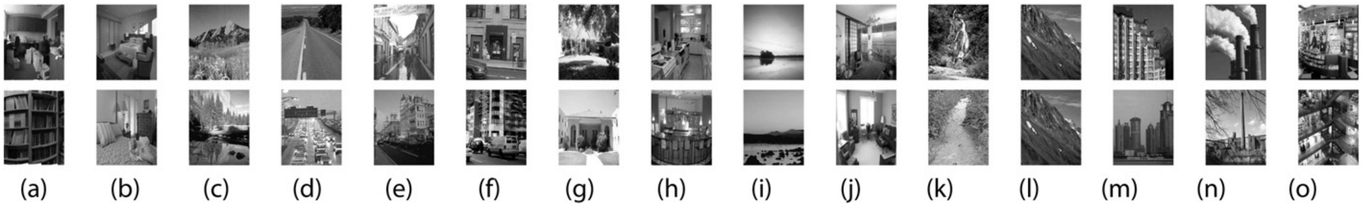

Fig. 2. Sample images from the 15 categories data set; (a) Office, (b) Bedroom, (c) Open country, (d) Highway, (e) Street, (f) inside-city, (g) Suburb-residence, (h) kitchen, (i) Coast, (j) Living-room, (k) Forest, (l) Mountain, (m) Tall-buildings, (n) Industrial, and (o) Store.

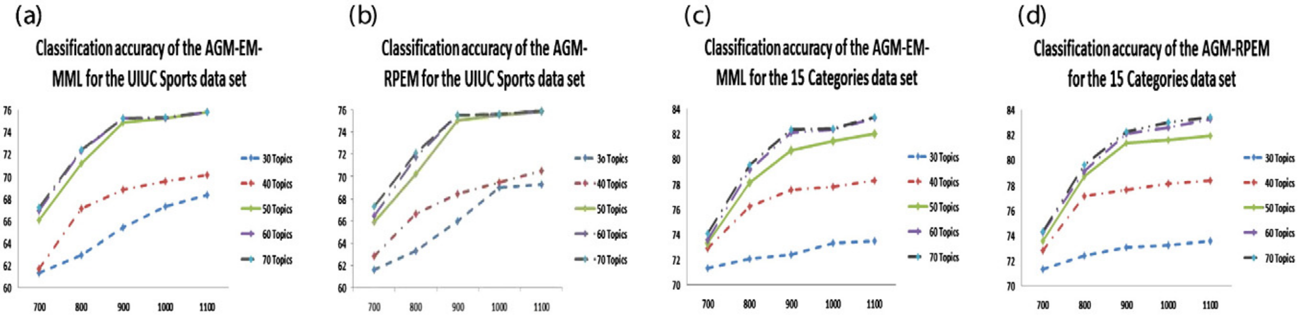

Fig. 3. Classification accuracy for different aspects and codebook sizes. (a) Classification accuracy by the AGM-EM-MML for the UIUC sports event data set, (b) Classification accuracy by the AGM-RPEM for the UIUC sports event data set, (c) Classification accurate by the AGM-EM-MML for the 15 categories data set, and (d) Classification accuracy by the AGM-RPEM for the 15 categories data set.

and most of them fall into two broad classes: deterministic and probabilistic classification. Deterministic approaches classify each image to one of a number of classes. This is done by considering some metric that defines the distance between classes and by defining the class boundaries. On the other hand, the probabilistic method classifies each image by calculating its probabilities of belonging to each class of interest. We believe that a probabilistic classification approach is more suitable because of its robustness to measurement error and its effectiveness in identifying similar characteristics from supervised training images. Therefore, we use our AGM to model the training images of each class. Then, for each input image, we calculate its likelihood of being generated from each class. Finally, we classify each image to the class that maximizes more its likelihood. Fig. 4 shows the average number of relevant features for both datasets under consideration when the RPEM and the EM-MML are used for learning the AGM parameters.

For the UIUC sports event data set, we have used a color SIFT descriptor in order to incorporate color information. The only difference between the color SIFT and the regular gray SIFT is the number of input channels (one versus three). Therefore, we represented each image using the HSV (Hue, Saturation, and Value) color space. For each event class, 70 randomly selected images are used for training and 60 are used for testing. Note that, we evaluated the performance of the proposed algorithm by running it 20 times, as well as using the *RPEM* and the *EM* + *MML* for the unsupervised learning of the AGM parameters. The confusion matrices calculated by the *AGM* + *EM* + *MML* and the *AGM* + *RPEM* are shown in Tables 1 and 2, respectively. Note that, we found the maximum classification variance under the 20 runs for each category to be 3.33% and 5.00% for the *AGM* − *RPEM* and the *AGM* − *EM* − *MML*, respectively.

Moreover, we can notice a lower accuracy for Bocce and Croquet classes. However, we found that both classes have lower classification rates for all the other methods under consideration which is mainly due to the inefficiency of the extracted features to discriminate between these two classes and the rest. Excluding the pre-processing time of feature detection and visual vocabulary formation, it takes about 5 and 7 min to train the AGM models on the training images using RPEM and EM-MML, respectively (Matlab implementation on an Intel Core i7-3610QM CPU with 2.3 GHz).

For the 15 scene category data set, we are using 100 images per class for training and the rest for testing as suggested in Refs. [54,55]. Note that, we evaluated the performance of the proposed algorithm by running it 20 times, as well as using the *RPEM* and the *EM* + *MML* for the unsupervised learning of the AGM parameters. The confusion matrices calculated by the *AGM* + *EM* + *MML* and the *AGM* + *RPEM* are shown in Tables 3 and 4, respectively. Note that, we found the maximum classification variance under the 20 runs for each category to be 1.97% and 2.16% for the *AGM* − *RPEM* and the *AGM* − *EM* − *MML*, respectively. Excluding the pre-processing time of feature detection and visual vocabulary formation, it takes about 11 and 13 min to train the AGM models on the training images using RPEM and EM-MML, respectively (Matlab implementation on an Intel Core i7-3610QM CPU with 2.3 GHz).

In order to evaluate the performance of both of our algorithms, we compare them with a number of state-of-the-art approaches:

- GMM-EM-MML [26] and GMM-RPEM [30]: We use the same proposed method with the Gaussian mixture model for classification. Note that, we use the *EM* + *MML* and the *RPEM* for the unsupervised learning of the GMM parameters in the case of the GMM-EM-MML and GMM-RPEM, respectively.
- GIST [56]: A spatial envelope that represents the dominant spatial structure of a scene is proposed then for classification the *K* nearest neighbors (KNN) classifier is used. It is noteworthy that each image is divided into 16 sub-images or blocks, then, a Gabor filter with 4 scales and 8 orientations is applied to create a 512 feature vector representing the gist of the scene. Finally, Principal Component

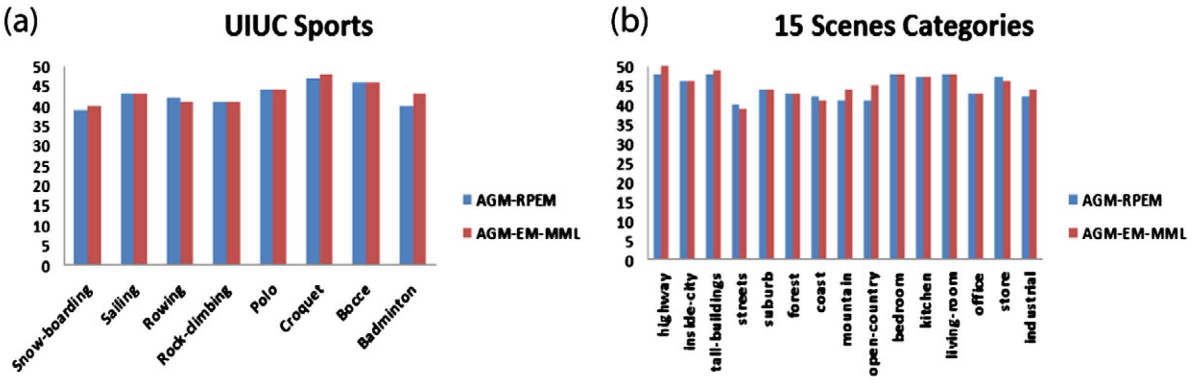

Fig. 4. Average number of relevant features for the AGM model. (a) For the UIUC sports event data set, (b) For the 15 categories data set.

Table 1
The confusion matrix of the AGM-EM-MML for the UIUC sports event data set.

	Snow-boarding	Sailing	Rowing	Rock-climbing	Polo	Croquet	Bocce	Badminton
Snow-boarding	**0.81**	0.02	0.00	0.02	0.00	0.01	0.01	0.13
Sailing	0.04	**0.83**	0.05	0.00	0.00	0.00	0.00	0.08
Rowing	0.00	0.00	**0.79**	0.00	0.03	0.13	0.04	0.01
Rock-climbing	0.02	0.01	0.00	**0.91**	0.00	0.00	0.00	0.06
Polo	0.00	0.00	0.05	0.00	**0.67**	0.09	0.19	0.00
Croquet	0.00	0.00	0.11	0.00	0.09	**0.54**	0.26	0.00
Bocce	0.00	0.00	0.04	0.00	0.16	0.22	**0.56**	0.02
Badminton	0.03	0.00	0.00	0.06	0.00	0.00	0.00	**0.91**

Table 2
The confusion matrix of the AGM-RPEM for the UIUC sports event data set.

	Snow-boarding	Sailing	Rowing	Rock-climbing	Polo	Croquet	Bocce	Badminton
Snow-boarding	**0.81**	0.02	0.00	0.02	0.00	0.01	0.01	0.13
Sailing	0.04	**0.83**	0.05	0.00	0.00	0.00	0.00	0.08
Rowing	0.00	0.00	**0.79**	0.00	0.03	0.13	0.04	0.01
Rock-climbing	0.02	0.01	0.00	**0.91**	0.00	0.00	0.00	0.06
Polo	0.00	0.00	0.05	0.00	**0.67**	0.09	0.19	0.00
Croquet	0.00	0.00	0.11	0.00	0.09	**0.54**	0.26	0.00
Bocce	0.00	0.00	0.04	0.00	0.16	0.22	**0.56**	0.02
Badminton	0.03	0.00	0.00	0.06	0.00	0.00	0.00	**0.91**

Analysis (PCA) is used to reduce the feature vector size to 80 and the KNN classifier is used for classification.

- Hierarchical [54]: The image is represented by a collection of local regions denoted as codewords using evenly sampled grid spaced at 10 × 10 then each patch is described by a 128-dimensional SIFT vector. Furthermore, the codebook is learned by performing K-means algorithm. Then, latent dirichlet allocation (LDA) algorithm is used to identify the different themes of the image. Finally, each image is classified to the class having the highest likelihood. Note that, the 20 highest themes for each category were used for classification.
- Probabilistic [57]: A probabilistic model for jointly modeling the image, its class label, and its annotations is introduced. For feature extraction 128-dimensional SIFT region descriptors selected by a sliding grid of size 5 × 5 were used. Then, the k-means algorithm is used to learn a codebook of 240 codewords. Moreover, image annotation is performed by choosing the terms that occurred more than 8 times. Then, two multi-class supervised topic modeling (sLDA) are learned on labeled and annotated images.
- SPM [55]: Spatial pyramid matching (SPM) partitions the image into increasingly fine sub-regions, then computes histograms of local features found inside each sub-region. Therefore, it represents an extension of the orderless bag-of-feature image representation. In this method, SIFT descriptors of 16 × 16 pixel patches computed over a grid with spacing 8 pixels were used to build a codebook of 400 words. Furthermore, a three level pyramid (three resolutions) is used to count the features that fall in each spatial bin. Finally, a multi-class SVM is used for classification.
- BOW [41]: This method is similar to our method except the difference of Gaussian (DoG) detector with SIFT descriptor is used as input for the bag of words method and SVM is applied for classification.
- Scene-objects [53]: An integrative model is used to classify events by integrating scene and object categorization. This method assumes that objects and scenes are independent given the event category and that there is no spatial relationship between objects. Furthermore, it extracts as much information as possible by extracting 12 × 12 patches by a sampling grid of size 10 × 10. Then, for each of the 200,000 extracted patches, a 128-dim SIFT vector is used to represent it. Then, a codebook of 300 visual words is built by applying K-means on the extracted SIFT vectors. To represent the geometry or layout information each pixel is given one of the three geometry labels namely: ground plane, vertical structure and sky. Note that, each patch is assigned to a geometry membership by the major vote of the pixels within.
- MLE-Scene [58] and MM-Scene [58]: This method presents a well-balanced joint latent topic discovery and prediction model estimation. It employs the maximum likelihood estimation (MLE-Scene) and the

Table 3
The confusion matrix of the AGM-EM-MML for the 15 scenes categories data set.

	Highway	Inside-city	Tall-buildings	Streets	Suburb	Forest	Coast	Mountain	Open-country	Bedroom	Kitchen	Living-room	Office	Store	Industrial
Highway	**0.86**	0.00	0.00	0.03	0.00	0.00	0.07	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.01
Inside-city	0.00	**0.81**	0.01	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.04	0.05
Tall-buildings	0.00	0.02	**0.91**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07
Streets	0.08	0.02	0.00	**0.90**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Suburb	0.00	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Forest	0.00	0.00	0.00	0.00	0.00	**0.95**	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.01	0.00
Coast	0.01	0.00	0.00	0.00	0.00	0.00	**0.82**	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.01
Mountain	0.00	0.00	0.00	0.00	0.00	0.03	0.00	**0.89**	0.08	0.00	0.00	0.00	0.00	0.00	0.00
Open-country	0.06	0.00	0.00	0.00	0.00	0.00	0.16	0.07	**0.71**	0.00	0.00	0.00	0.00	0.00	0.00
Bedroom	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.68**	0.14	0.18	0.00	0.00	0.00
Kitchen	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	**0.69**	0.09	0.08	0.00	0.00
Living-room	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23	0.16	**0.61**	0.00	0.00	0.00
Office	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	**0.93**	0.00	0.00
Store	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.00	**0.75**	0.11
Industrial	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.21	**0.65**

Table 4
The confusion matrix of the AGM-RPEM for the 15 scenes categories data set.

	Highway	Inside-city	Tall-buildings	Streets	Suburb	Forest	Coast	Mountain	Open-country	Bedroom	Kitchen	Living-room	Office	Store	Industrial
Highway	**0.86**	0.00	0.00	0.03	0.00	0.00	0.07	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.01
Inside-city	0.00	**0.82**	0.01	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.04	0.05
Tall-buildings	0.00	0.02	**0.93**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05
Streets	0.08	0.01	0.00	**0.91**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Suburb	0.00	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Forest	0.00	0.00	0.00	0.00	0.00	**0.95**	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.01	0.00
Coast	0.01	0.00	0.00	0.00	0.00	0.00	**0.82**	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.01
Mountain	0.00	0.00	0.00	0.00	0.00	0.02	0.00	**0.91**	0.07	0.00	0.00	0.00	0.00	0.00	0.00
Open-country	0.06	0.00	0.00	0.00	0.00	0.00	0.16	0.07	**0.71**	0.00	0.00	0.00	0.00	0.00	0.00
Bedroom	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.68**	0.14	0.18	0.00	0.00	0.00
Kitchen	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	**0.69**	0.09	0.08	0.00	0.00
Living-room	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23	0.16	**0.61**	0.00	0.00	0.00
Office	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	**0.93**	0.00	0.00
Store	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.00	**0.75**	0.11
Industrial	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.21	**0.65**

max-margin learning (MM-Scene) to learn an upstream scene model. Starting from a set of seed points, an algorithm for constructing Voronoi tessellations with respect to a distance defined by the Ultrametric Contour Map (UCM) is used to segment the input images into small regions based on color, brightness and texture homogeneity. For each region, color, texture and location features are extracted and quantized into 120 codewords, respectively. Similarly, the SIFT features extracted from the small patches within each region are quantized into 300 SIFT codewords. Furthermore, GIST features [56] are used in order to have a global representation. Finally, two scene models trained with max-margin and MLE are used for classification.

- OB-SVM [59]: The object Bank method represents an image as a scale-invariant response map of a large number of pre-trained generic object detectors and uses the SVM algorithm to classify it. Note that, for this application 200 object detectors at 12 detection scales and 3 spatial pyramid levels were used resulting in an object filter map of 50,400 grids.

Table 5 shows the average classification rates for the methods under consideration.

From both evaluations, we can conclude that our method with both learning algorithms can achieve good results for the task of scenes categorization. Also, we found that the RPEM approach achieves higher results than the EM-MML.

5.2. Facial expression recognition

The interpretation of face images has been the topic of extensive research in the past [60,61]. Indeed, the face is the most important object in the field of human information processing. In particular, it allows the extraction of rich information about human emotion which plays a crucial role in human communications, cognitive processes, neural, social interaction and psychological studies [62–65]. For this reason, facial expression analysis and recognition have been the topic of extensive research (see, for instance, [66–72]) due to the increased demand for emotion analysis, human behavior interpretation, biometrics, and image retrieval to ensure security and safety. The main goal of theses researches is to create a computer system capable of automatically detecting the emotional state of any person. Thus, face expression recognition (FER) is based on applying machine vision and pattern recognition algorithms on both still images and/or image video sequences in order to extract emotional content from visual patterns of a person's face. Despite the progress made in recognizing facial expressions, reliable and accurate automatic FER is still an evolving research subject taking into account the expressiveness of the human face and to the subtlety, complexity and variability of facial expressions [73,74]. In this section we present a novel system for facial expressions recognition that considers both shape and texture information to represent facial emotions. Research has been mainly based on distinguishing between seven kinds of fundamental expressions (Anger, Disgust, Fear, Joy, Sadness, Surprise with the Neutral expression) [75,76]. We are mainly interested in recognizing these seven facial expressions, also. Fig. 5 shows a sample image from each of the seven facial expressions.

Normally, any FER system consists of two main components namely pre-processing and processing. In the pre-processing part, detection and location of faces in a cluttered scene are carried out, then a normalization process aiming to align these extracted face images is performed. On the other hand, the processing part aims to extract specific facial features from the pre-processed output images, then, uses them to recognize different facial expressions. Thus, our algorithm can be divided into three main parts: face detection and normalization, feature selection, and facial expression recognition.

5.2.1. Face detection and normalization

Face acquisition, the first part of any FER system, is concerned with identifying and locating human faces under realistic environment regardless of their positions, scales, orientations, poses, and illumination. In [77], the authors used a retinally connected neural network to examine small windows of an image and decide whether each window contains a face. The work in Ref. [78] presented a novel approach for face detection with the use of support vector machines (SVM). In this section, we use the method introduced in [79] for its small computational time and high detection accuracy. The method can be summarized in the following algorithm:

1. Get the integral image, where the integral image at location (x, y) contains the sum of the pixels above and to the left of (x, y).
2. Compute the rectangle features (Haar features) using the integral image from step (1).

Table 5
The average accuracies (%) for both data sets.

	GMM-EM-MML	GMM-RPEM	GIST	Hierarchical	Probabilistic	SPM	BOW	Scene-Objects	MLE-Scene	MM-Scene	OB-SVM	AGM-EM-MML	AGM-RPEM
UIUC event	69.51%	69.76%	63.88%	–	66.00%	71.57%	69.25%	73.40%	69.87%	71.70%	76.30%	74.87%	75.08%
Categories	74.78%	74.21%	74.00%	65.20%	–	81:20%	73.50%	–	–	–	80.90%	80.69%	81.38%

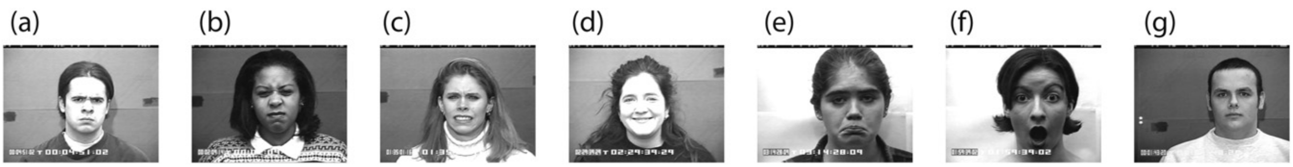

Fig. 5. Sample images from the seven facial expressions; (a) Anger, (b) Disgust, (c) Fear, (d) Joy, (e) Sadness, (f) Surprise, and (g) Neutral.

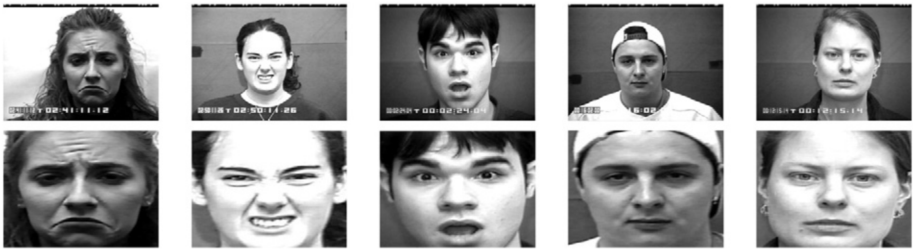

Fig. 6. Sample images and the detected faces.

3. Use AdaBoost learning to select a small number of critical visual features.
4. Apply a cascade of weak classifiers for the final face detection.

Note that in step (2) the three rectangular features used are: 1 — *Two-rectangle feature* which computes the difference between the sum of the pixels within two rectangular regions, 2 — *Three-rectangle feature* which calculates the sum within two outside rectangles subtracted from the sum in a center rectangle, and 3 — *Four-rectangle feature* which computes the difference between diagonal pairs of rectangles. The importance of these features is that they can be evaluated at any scale and location in a few operations. Using the above method we can easily and accurately identify all faces in the input image data set, as shown in Fig. 6. Then, we normalize all faces detected to 140 × 106 for all images.

5.2.2. Feature selection

In order to decrease the dimensionality of the detected faces, they are usually represented in terms of low-level feature vectors in lower dimensional feature space [80,81]. Extracting facial feature is an important step and essential requirement in automated facial expression recognition and has been widely studied in the literature [82–88,75,89]. The extracted features affect generally the capability of recognition methods. In facial feature extraction for expression analysis, there are two common approaches: geometric feature-based methods and appearance-based methods. Geometric facial feature approaches extract the shape and locations of facial components (such as mouth, eyes, or nose) to form a feature vector that represents the face geometry. In Ref. [90], the authors have implemented a geometric feature-based method and applied it for facial expression recognition. However, geometric feature-based methods require accurate and reliable facial feature detection and tracking, which is hard to accommodate in many situations. Appearance-based methods apply image filters to either the whole face or specific regions in a face image to extract a feature vector [91,92]. However, appearance-based methods usually convolve face images with a bank of Gabor filters to extract multi-scale and multi-orientational coefficients, which is time and memory intensive. In this work, we use local binary pattern (LBP) features introduced for texture analysis [93,94] due to their tolerance against illumination changes and their computational simplicity. The main idea was to label the pixels of an image by thresholding the 3 × 3-neighbourhood of each pixel with the center value and considering the result as a binary number, then, the histogram of the labels can be used as a texture descriptor. Fig. 7 shows an illustration of the basic LBP operator.

Later the operator was extended to use uniform patterns which are LBP that contain at most two bitwise transitions from 0 to 1 or vice versa. For example 00000000, 00010000, and 01000100 are considered uniform patterns. The advantage of using uniform patterns is that uniform patterns account for a bit less than 90% of all patterns while can be represented by only 59 bins [94]. Hence, using only uniform patterns and labeling all remaining patterns with a single label can decrease our feature vectors greatly. The only disadvantage of using the LBP histograms is that it only encodes the occurrences of the micro-patterns without any indication about their locations. In order to overcome this issue, we divided the face image into 48 regions, each of 17 × 17 pixels as shown in Fig. 8. It is known that useful information for expression classification lie in regions such as eye and mouth regions. Thus, we are only interested with the LBP for these regions as illustrated in Fig. 8.

5.2.3. Facial expression recognition

In order to differentiate between different facial expressions, various machine learning methods were introduced to perform facial expression recognition using facial features such as support vector machine (SVM), linear discriminant analysis (LDA), and linear programming technique (see, for instance, [95–97]). In this section, we will use our AGM model for this purpose. Our idea is to use training data from the seven expressions to build a classifier. Thus, we group all the training images coming from the same class together, then apply the face recognition and feature selection methods in order to represent each image in the training set by 14 LBP histograms. Then, we grouped all the LBP histograms coming from the same region for each emotion together and modeled them with the AGM. Note that, we used 14 AGMs for each emotion (an AGM per region). Then, in the classification stage, we use the majority vote to classify the data into the right facial expression.

5.2.4. Results

In order to evaluate our approach we have used the Cohn–Kanade database [98] because it is one of the most widely used test-beds for

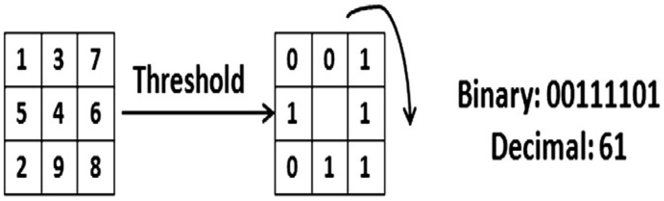

Fig. 7. The basic LBP operator.

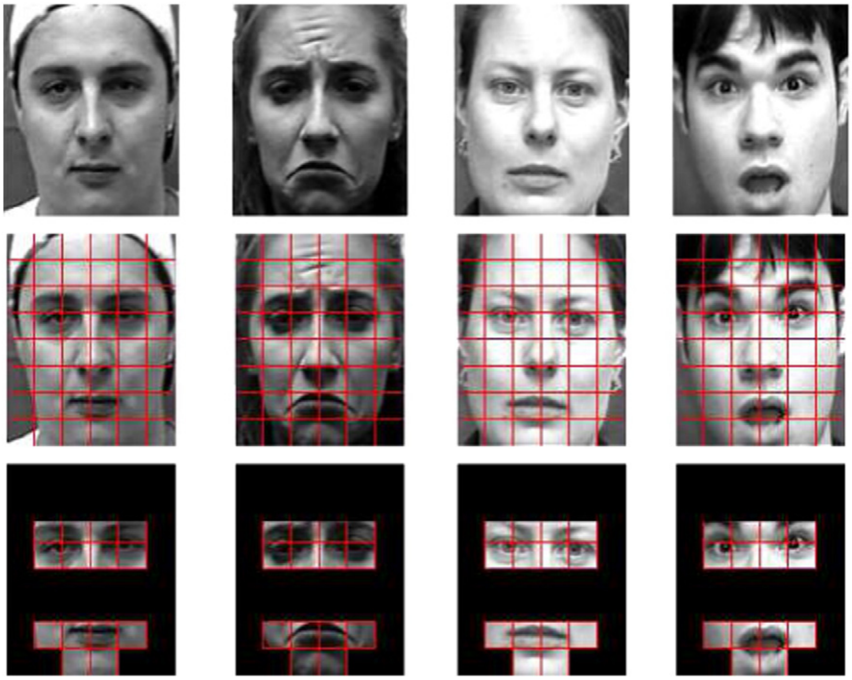

Fig. 8. LBP method for facial expression feature selection; first row represents the face images, second row illustrates face images after division to 48 regions, third row represents the important regions for facial expression recognition.

facial expression algorithm development and evaluation. This database consists of 97 university students between the ages of 18 and 50 years, of which 69% are female, 13% are African-American, 81% are Euro-American, and 6% are from other groups. Subjects were instructed to perform a series of 23 facial displays, six of which were based on description of prototypic emotions, where each display began and ended with a neutral face. In addition, image sequences were digitized into 640 × 490 pixel arrays with 8-bit precision for grayscale values.

For our experiments, we selected 320 image sequences where each comes from one of the six basic emotions. The sequences come from 96 subjects, with 1–6 emotions per subject. For each sequence, the neutral face and three peak frames were used for expression recognition, resulting in 1280 images (108 Anger, 120 Disgust, 99 Fear, 282 Joy, 126 Sadness, 225 Surprise and 320 Neutral). To evaluate the performance of our algorithms (*RPEM* and *EM* + *MML*) we adopted a 10 cross validation testing scheme repeated for 10 times in our experiments. The confusion matrices for the Cohn–Kanade data set for our algorithms are shown in Tables 6 and 7.

Note that, we found the maximum classification variance under the 10 runs for each category to be 1.56% and 1.42% for the $AGM - RPEM$ and the $AGM - EM - MML$, respectively. Furthermore, the time complexity of our approaches, excluding pre-processing time of feature detection, takes about 14 and 17 min to train the AGM models on the training images using RPEM and EM-MML, respectively (Matlab implementation on an Intel Core i7-3610QM CPU with 2.3 GHz). In order to evaluate the performance of both of our algorithms, we compare them with a number of state-of-the-art approaches:

- GMM-EM-MML [26] and GMM-RPEM [30]: We use the same proposed method with the Gaussian mixture model for classification. Note that, we use the *EM* + *MML* and the *RPEM* for the unsupervised learning of the GMM parameters for the GMM-EM-MML and GMM-RPEM, respectively.
- TAN [90]: A tree-augmented naive-Bayes classifier is trained on some geometric facial features (eyebrows, eyelids and mouth).
- LBP-χ^2, LBP-SVM, LBP-PolySVM, LBP-RBFSVM, and LBP-LP [72,99]: First, each face image is divided into 42 (6 × 7) regions of 18 × 21 pixels each. Then, the extended LBP features are extracted from each region. Finally, the χ^2 is used for template matching in the case of LBP-χ^2. On the other hand, linear SVM, polynomial SVM, SVM with Radial Basis Function (RBF) kernel, and linear programming technique are used for classification in the case of the LBP-SVM, LBP-PolySVM, LBP-RBFSVM, and LBP-LP, respectively.
- GW-SVM, GW-PolySVM, and GW-RBFSVM [91,92]: The facial images were converted into a Gabor magnitude representation using a bank of 40 Gabor filters. Then, linear SVM, polynomial SVM, and SVM with Radial Basis Function (RBF) kernel are used for classification in the case of the GW-SVM, GW-PolySVM, and GW-RBFSVM, respectively.
- PCA-LDA-NN, and PCA-LDA-SVM [80,81]: The training faces are first projected into a PCA subspace to have a number of eigenvectors that ranges from 405-431. Then, an LDA is trained to recognize different

Table 6
The confusion matrix of the AGM-EM-MML for the Cohn–Kanade data set.

	Anger	Disgust	Fear	Joy	Sadness	Surprise	Neutral
Anger	**84.1%**	2.9%	0.00%	0.00%	7.9%	0.00%	5.1%
Disgust	1.4%	**95.7%**	2.9%	0.00%	0.00%	0.00%	0.00%
Fear	0.00%	0.00%	**81.0%**	11.2%	5.7%	0.00%	2.1%
Joy	0.00%	0.00%	1.5%	**95.1%**	0.00%	0.00%	3.4%
Sadness	3.9%	0.00%	0.00%	0.00%	**73.4%**	1.7%	21.0%
Surprise	0.00%	0.00%	0.4%	0.00%	0.8%	**97.5%**	1.3%
Neutral	0.3%	0.1%	0.00%	1.2%	3.6%	0.00%	**94.8%**

Table 7
The confusion matrix of the AGM-RPEM for the Cohn–Kanade data set.

	Anger	Disgust	Fear	Joy	Sadness	Surprise	Neutral
Anger	**85.8%**	1.7%	0.00%	0.00%	7.6%	0.00%	4.9%
Disgust	1.4%	**95.7%**	2.9%	0.00%	0.00%	0.00%	0.00%
Fear	0.00%	0.00%	**81.2%**	11.3%	5.7%	0.00%	1.8%
Joy	0.00%	0.00%	1.5%	**95.1%**	0.00%	0.00%	3.4%
Sadness	3.8%	0.00%	0.00%	0.00%	**73.6%**	1.8%	20.8%
Surprise	0.00%	0.00%	0.4%	0.00%	0.8%	**97.5%**	1.3%
Neutral	0.3%	0.1%	0.00%	1.2%	3.6%	0.00%	**94.8%**

Table 8
The average accuracies (%) for the Cohn–Kanade data set.

GMM-EM-MML 85.4%	GMM-RPEM 85.4%	LBP-χ^2 79.1%	TAN 73.2%	LBP-LP 82.3%
LBP-SVM 88.1%	LBP-PolySVM 88.1%	LBP-RBFSVM 88.9%	GW-SVM 86.6%	GW-PolySVM 86.6%
GW-RBFSVM 86.8%	PCA-LDA-NN 73.4%	PCA-LDA-SVM 79.2%	AGM-EM-MML 88.8%	AGM-RPEM 89.1%

expressions. Finally, a nearest neighbor classifier or support vector machine (SVM) classifier is used for classification in the case of PCA-LDA-NN, and PCA-LDA-SVM, respectively.

Table 8 shows the average classification rates for the methods under consideration. From both evaluations, we can conclude that our method with both learning algorithms can achieve good results for the task of facial expression recognition. Also, we found that the RPEM approach achieves slightly higher results than the EM-MML for the asymmetric Gaussian mixture.

6. Conclusion

Four of the critical issues that arise when clustering and modeling objects using finite mixtures are: (1) determination of what features best discriminate among the different clusters, (2) choice of the probability density functions, (3) estimation of the mixture parameters and (4) automatic determination of the number of mixture components. Toward objects clustering goal, we have proposed in this paper two unified statistical learning frameworks based on finite AGM models that tackle all these problems simultaneously. The first learning algorithm is based on the optimization of a message length objective and the second one learns the AGM model via an RPEM technique which allows simultaneous parameters estimation and model selection. Also, for both algorithms, we tackled the problem of noisy and uninformative features by determining a set of relevant features for each data cluster. The merits of the proposed work have been shown through a complicated computer vision application, involving high-dimensional feature vectors and large number of classes, namely scenes categorization. Moreover, we have tackled an important research topic in computer vision which is determining the emotional state of the face regardless of its identity. We have focused on the recognition of some basic expressions. From experimental results, we can observe the higher accuracy of the AGM model when compared to the GMM. Having an extra parameter, the AGM was able to increase the GMM accuracy by 3% and 4% for facial expression recognition and scene categorization, respectively. This is due to its asymmetrical property and its ability to model different shapes which represents a good choice for high dimensional data. Future works could be devoted to the extension of this work for the recognition of other expressions which may have applications in automated behavioral analysis, video conference, human–machine interface, face recognition and animation, and affective computing, etc. Indeed, human emotion is composed of thousands of expressions. Future works could be devoted also to the development of a Bayesian or variational approach to learn the proposed model and its application to other image and signal processing applications.

Acknowledgment

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank the anonymous referees and the associate editor for their comments.

References

[1] A.K. Jain, Advances in mathematical models for image processing, Proc. IEEE 69 (1981) 502–528.

[2] W. Choi, K. Shahid, S. Savarese, Learning context for collective activity recognition, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3273–3280.

[3] M.J. Black, D.J. Fleet, Probabilistic detection and tracking of motion boundaries, Int. J. Comput. Vis. 38 (2000) 231–245.

[4] G.J. McLachlan, D. Peel, Finite Mixture Models, Wiley, New York, 2000.

[5] N. Bouguila, W. ElGuebaly, Discrete data clustering using finite mixture models, Pattern Recogn. 42 (2009) 33–42.

[6] G. McLachlan, D. Peel, Mixfit: an algorithm for the automatic fitting and testing of normal mixture models, Proc. of the International Conference on Pattern Recognition (ICPR), IEEE, vol. 1, 1998, pp. 553–557.

[7] J. Goldberger, H. Greenspan, J. Dreyfuss, Simplifying mixture models using the unscented transform, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 1496–1502.

[8] R.P. Browne, P.D. McNicholas, M.D. Sparling, Model-based learning using a mixture of mixtures of Gaussian and uniform distributions, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 814–817.

[9] M.R. Whiteley, B.M. Welsh, M.C. Roggemann, Limitations of gaussian assumptions for the irradiance distribution in digital imagery: nonstationary image ensemble considerations, J. Opt. Soc. Am. A 15 (1998) 802–810.

[10] A. Hyvärinen, P.O. Hoyer, Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces, Neural Comput. 12 (2000) 1705–1720.

[11] M.S. Allili, N. Bouguila, D. Ziou, Finite general gaussian mixture modeling and application to image and video foreground segmentation, J. Elec. Imaging 17 (2008) 1–13.

[12] T. Elguebaly, N. Bouguila, Bayesian learning of finite generalized gaussian mixture models on images, Signal Process. 91 (2011) 801–820.

[13] P.N. Bennett, Using asymmetric distributions to improve text classifier probability estimates, Proc. of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), ACM, 2003, pp. 111–118.

[14] T. Elguebaly, N. Bouguila, Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection, Mach. Vis. Appl. 25 (2014) 1145–1162.

[15] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley, New York, 1997.

[16] N. Bouguila, D. Ziou, High-dimensional unsupervised selection and estimation of a finite generalized dirichlet mixture model based on minimum message length, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1716–1731.

[17] L. Xu, A. Krzyzak, E. Oja, Rival penalized competitive learning for clustering analysis, IEEE Trans. Neural Networks 4 (1993) 636–649.

[18] L. Xu, Rival penalized competitive learning, finite mixture, and multisets clustering, Proc. of the IEEE International Joint Conference on Neural Networks (IJCNN), IEEE, 1998, pp. 2525–2530.

[19] Y. ming Cheung, Rival penalization controlled competitive learning for data clustering with unknown cluster number, Proc. of the 9th International Conference on Neural Information Processing (ICONIP), vol. 1, IEEE, 2002, pp. 467–471.

[20] Y.-M. Cheung, Maximum weighted likelihood via rival penalized em for density mixture clustering with automatic model selection, IEEE Trans. Knowl. Data Eng. 17 (2005) 750–761.

[21] A. Kolcz, X. Sun, J.K. Kalita, Efficient handling of high-dimensional feature spaces by randomized classifier ensembles, Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002, pp. 307–313.

[22] E. Bart, S. Ullman, Cross-generalization: learning novel classes from a single example by feature replacement, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2005, pp. 672–679.

[23] K. Tieu, P. Viola, Boosting image retrieval, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2000, pp. 228–235.

[24] C.-Y. Tsai, C.-C. Chiu, Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm, Comput. Stat. Data Anal. 52 (2008) 4658–4672.

[25] Y. Kim, W.N. Street, F. Menczer, Feature selection in unsupervised learning via evolutionary search, Proc. of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD), ACM, 2000, pp. 365–369.

[26] M.H.C. Law, M.A.T. Figueiredo, A.K. Jain, Simultaneous feature selection and clustering using mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 26 (2004) 1154–1166.

[27] Y. Li, M. Dong, J. Hua, Simultaneous localized feature selection and model detection for gaussian mixtures, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 953–960.

[28] S. Boutemedjet, N. Bouguila, D. Ziou, A hybrid feature extraction selection approach for high-dimensional non-gaussian data clustering, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 1429–1443.

[29] C.S. Wallace, Statistical and Inductive Inference by Minimum Message Length, Springer-Verlag, 2005.

[30] Y.-M. Cheung, H. Zeng, Feature weighted rival penalized em for gaussian mixture clustering: automatic feature and model selections in a single paradigm, Proc. of the International Conference on Computational Intelligence and Security (CIS), IEEE, 2006, pp. 1018–1028.

[31] L. Fei-Fei, A. Iyer, C. Koch, P. Perona, What do we see in a glance of a scene? J. Vis. 7 (2007) 1–29.

[32] Z. Su, H.-J. Zhang, S. Li, S. Ma, Relevance feedback in content-based image retrieval: Bayesian framework, features subspaces, and progressive learning, IEEE Trans. Image Process. 12 (2003) 924–937.

[33] S. Boutemedjet, D. Ziou, N. Bouguila, A graphical model for content based image suggestion and feature selection, in: J.N. Kok, J. Koronacki, R.L. de Mántaras, S. Matwin, D. Mladenic, A. Skowron (Eds.), PKDD, volume 4702 of LNCS, Springer, 2007, pp. 30–41.

[34] A. Singhal, J. Luo, W. Zhu, Probabilistic spatial context models for scene content understanding, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, IEEE, 2003, pp. I-235–I-241.

[35] Y. LeCun, F.J. Huang, L. Bottou, Learning methods for generic object recognition with invariance to pose and lighting, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, Vol. 2, IEEE2004 (pp. II–97–104).

[36] Y. Wu, I. Kozintsev, J.Y. Bouguet, C. Dulong, Sampling strategies for active learning in personal photo retrieval, Proc. of the IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2006, pp. 529–532.

[37] N. Rasiwasia, J.C. Pereira, E. Coviello, G. Doyle, G.R.G. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, Proc. of the 18th International Conference on Multimedia (MM), ACM, 2010, pp. 251–260.

[38] A. Gupta, L.S. Davis, Beyond nouns: exploiting prepositions and comparative adjectives for learning visual classifiers, in: D.A. Forsyth, P.H.S. Torr, A. Zisserman (Eds.), ECCV (1), volume 5302 of LNCS, Springer, 2008, pp. 16–29.

[39] S. Savarese, J. Winn, A. Criminisi, Discriminative object class models of appearance and shape by correlatons, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, IEEE, 2006, pp. 2033–2040.

[40] A. Rocha, S. Goldenstein, Progressive randomization: seeing the unseen, Comput. Vis. Image Underst. 114 (2010) 349–362.

[41] G. Csurka, C. Bray, C. Dance, L. Fan, Visual categorization with bags of keypoints, Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision (ECCV), Springer, 2004, pp. 1–22.

[42] A. Bosch, A. Zisserman, X. Muñoz, Scene classification via pLSA, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), ECCV (4), volume 3954 of LNCS, Springer, 2006, pp. 517–530.

[43] N. Serrano, A. Savakis, J. Luo, Improved scene classification using efficient low-level features and semantic cues, Pattern Recogn. 37 (2004) 1773–1784.

[44] S. Kumar, M. Hebert, A hierarchical field framework for unified context-based classification, Proc. of the IEEE International Conference on Computer Vision (ICCV), IEEE, 2005, pp. 1284–1291.

[45] J. van Gemert, J.-M. Geusebroek, C.J. Veenman, A.W.M. Smeulders, Kernel codebooks for scene categorization, in: D.A. Forsyth, P.H.S. Torr, A. Zisserman (Eds.), Proc. of the 10th European Conference on Computer Vision (ECCV), volume 5304 of LNCS, Springer, 2008, pp. III.696–III.709.

[46] C. Galleguillos, A. Rabinovich, S. Belongie, Object categorization using co-occurrence, location and appearance, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2008, pp. 1–8.

[47] C. Galleguillos, S.J. Belongie, Context based object categorization: a critical survey, Comput. Vis. Image Underst. 114 (2010) 712–722.

[48] X. Zhou, K. Yu, T. Zhang, T.S. Huang, Image classification using super-vector coding of local image descriptors, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), ECCV (5), volume 6315 of LNCS, Springer, 2010, pp. 141–154.

[49] J. Batlle, A. Casals, J. Freixenet, J. Mart, A review on strategies for recognizing natural objects in colour images of outdoor scenes, Image Vis. Comput. 18 (2000) 515–530.

[50] J. Mart, J. Freixenet, J. Batlle, A. Casals, A new approach to outdoor scene description based on learning and top-down segmentation, Image Vis. Comput. 19 (2001) 1041–1055.

[51] A. Quattoni, A. Torralba, Recognizing indoor scenes, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 413–420.

[52] L. Lu, K. Toyama, G.D. Hager, A two level approach for scene recognition, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2005, pp. 688–695.

[53] L.-J. Li, L. Fei-Fei, What, where and who? Classifying event by scene and object recognition, Proc. of the IEEE International Conference in Computer Vision (ICCV), IEEE, 2007, pp. 1–8.

[54] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2005, pp. 524–531.

[55] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2006, pp. 2169–2178.

[56] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, Int. J. Comput. Vis. 42 (2001) 145–175.

[57] C. Wang, D.M. Blei, L. Fei-Fei, Simultaneous image classification and annotation, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), IEEE, 2009, pp. 1903–1910.

[58] J. Zhu, L.-J. Li, L. Fei-Fei, E.P. Xing, Large margin learning of upstream scene understanding models, Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc., 2010, pp. 2586–2594.

[59] L.-J. Li, H. Su, E.P. Xing, L. Fei-Fei, Object bank: a high-level image representation for scene classification and semantic feature sparsification, Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc, 2010, pp. 1378–1386.

[60] G.J. Edwards, A. Lanitis, C.J. Taylor, T.F. Cootes, Statistical models of face images — improving specificity, Image Vis. Comput. 16 (1998) 203–211.

[61] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The feret evaluation methodology for face-recognition algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1090–1104.

[62] H. Yamada, Visual information for categorizing facial expression of emotions, Appl. Cogn. Psychol. 7 (1993) 257–270.

[63] M.S. Bartlett, J.C. Hager, P. Ekman, T.J. Sejnowski, Measuring facial expressions by computer image analysis, Psychophysiology 36 (1999) 253–263.

[64] H. Kobayashi, F. Hara, Facial interaction between animated 3d face robot and human beings, Proc. of the IEEE International Conference on Systems, Man, and Cybernetic (SMC), vol. 4, IEEE, 1997, pp. 3732–3737.

[65] R.J.R. Blair, J.S. Morris, C.D. Frith, D.I. Perrett, R.J. Dolan, Dissociable neural response to facial expressions of sadness and anger, Brain 122 (1999) 883–893.

[66] H. Li, P. Roivainen, R. Forchheimer, 3-D motion estimation in model-based facial image coding, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1993) 545–555.

[67] D. Terzopoulos, K. Waters, Analysis and synthesis of facial image sequences using physical and anatomical models, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1993) 569–579.

[68] Y. Yaccob, L. Davis, Recognizing facial expressions by spatio-temporal analysis, Proc. of the 12th IAPR International Conference on Pattern Recognition (ICPR), vol. 1, IEEE, 1994, pp. 747–749.

[69] Y. Yacoob, L. Davis, Computing spatio-temporal representations of human faces, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 1994, pp. 70–75.

[70] S. Kimura, M. Yachida, Facial expression recognition and its degree estimation, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 1997, pp. 29–300.

[71] J.J.-J. Lien, T. Kanade, J.F. Cohn, C.-C. Li, Detection, tracking, and classification of action units in facial expression, Robot. Auton. Syst. 31 (2000) 131–146.

[72] C. Shana, S. Gong, P.W. McOwan, Facial expression recognition based on local binary patterns: a comprehensive study, Image Vis. Comput. 27 (2009) 803–816.

[73] G. Littlewort, J. Whitehill, T. Wu, I.R. Fasel, M.G. Frank, J.R. Movellan, M.S. Bartlett, The computer expression recognition toolbox (cert), Proc. of the Ninth IEEE International Conference on Automatic Face and Gesture Recognition (FG), IEEE, 2011, pp. 298–305.

[74] B. Fasel, J. Luettin, Automatic facial expression analysis: a survey, Pattern Recogn. 36 (2003) 259–275.

[75] Z. Zhang, Feature-based facial expression recognition: sensitivity analysis and experiments with a multilayer perceptron, Int. J. Pattern Recognit. Artif. Intell. 13 (1999) 893–911.

[76] M. Pantic, L.J.M. Rothkrantz, Expert system for automatic analysis of facial expressions, Image Vis. Comput. 18 (2000) 881–905.

[77] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1996) 23–38.

[78] Y. Li, S. Gong, J. Sherrah, H. Liddell, Support vector machine based multi-view face detection and recognition, Image Vis. Comput. 22 (2004) 413–427.

[79] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2004) 137–154.

[80] Y. Tian, Evaluation of face resolution for expression analysis, The 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), IEEE, 2004, p. 82.

[81] Y. Tian, L. Brown, A. Hampapur, S. Pankanti, A. Senior, R. Bolle, Real world real-time automatic recognition of facial expression, Proc. of the IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS), IEEE2003.

[82] K.-M. Lam, H. Yan, Locating and extracting the eye in human face images, Pattern Recogn. 29 (1996) 771–779.

[83] P.S. Penev, J.A. Atick, Local feature analysis: a general statistical theory for object representation, Netw. Comput. Neural Syst. 7 (1996) 477–500.

[84] R. Brunelli, T. Poggio, Face recognition: features versus templates, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1993) 1042–1052.

[85] D. Terzopoulos, K. Waters, Analysis of facial images using physical and anatomical models, Proc. of the IEEE Third International Conference on Computer Vision (ICCV), IEEE, 1990, pp. 727–732.

[86] H. Wu, T. Yokoyama, D. Pramadihanto, M. Yachida, Face and facial feature extraction from color image, Proc. of the Second International Conference on Automatic Face and Gesture Recognition (AFGR), IEEE, 1996, pp. 345–350.

[87] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, Proc. of the Third International Conference on Automatic Face and Gesture Recognition (AFGR), IEEE, 1998, pp. 200–205.

[88] A.L. Yuille, P.W. Hallinan, D.S. Cohen, Feature extraction from faces using deformable templates, Int. J. Comput. Vis. 8 (1992) 99–111.

[89] X. Xie, K.-M. Lam, Gabor-based kernel PCA with doubly nonlinear mapping for face recognition with a single face image, IEEE Trans. Image Process. 15 (2006) 2481–2492.

[90] I. Cohen, N. Sebe, A. Garg, L. Chen, T. Huang, Facial expression recognition from video sequences: temporal and static modeling, Comput. Vis. Image Underst. 91 (2003) 160–187.

[91] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, J. Movellan, Recognizing facial expression: machine learning and application to spontaneous behavior, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2005, pp. 568–573.

[92] M.S. Bartlett, G. Littlewort, I. Fasel, R. Movellan, Real time face detection and facial expression recognition: development and application to human computer interaction, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), IEEE, 2003, p. 53.

[93] T. Ojala, M. Pietikainen, D. Harwood, A comparative study of texture measures with classification based on featured distributions, Pattern Recogn. 29 (1996) 51–59.

[94] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 971–987.

[95] J. Lien, T. Kanade, J. Cohn, C.-C. Li, Automated facial expression recognition based on FACS action units, Proc. of the Third International Conference on Automatic Face and Gesture Recognition (AFGR), IEEE, 1998, pp. 390–395.

[96] X. wen Chen, T.S. Huang, Facial expression recognition: A clustering-based approach, Pattern Recogn. Lett. 24 (2003) 1295–1302.
[97] C.-F. Chuang, F.Y.-C. Shih, Recognizing facial action units using independent component analysis and support vector machine, Pattern Recogn. 39 (2006) 1795–1798.
[98] T. Kanade, J. Cohn, Y. Tian, Comprehensive database for facial expression analysis, Proc. of the International Conference on Automatic Face and Gesture Recognition (AFGR), IEEE, 2000, p. 46.
[99] X. Feng, M. Pietikainen, A. Hadid, Facial expression recognition with local binary patterns and linear programming, Pattern Recognit. Image Anal. 15 (2005) 546–548.