A new pipeline for the recognition of universal expressions of multiple faces in a video sequence

Latifa Greche, Mohamed Akil, Rostom Kachouri, Najia Es-Sbai

A new pipeline for the recognition of universal expressions of multiple faces in a video sequence

Latifa Greche [a] · **Mohamed Akil**[b] · **Rostom Kachouri**[b] · **Najia Es-sbai**[a]

Abstract Facial Expression Recognition (FER) is a crucial issue in human-machine interaction. It allows machines to act according to facial expression changes. However, acting in real time requires recognizing the expressions at video speed. Usually, the video speed differs from one device to another. However, one of the standard settings for shooting videos is 24 fps. This speed is considered as the low-end of what our brain can perceive as fluid video. From this perspective, to achieve a real-time FER the image analysis must be completed, strictly, in less than 0.042 second no matter how the background complexity is or how many faces exists in the scene. In this paper, a new pipeline has been proposed in order to recognize the fundamental facial expressions for more than one person in real world sequence videos. First, the pipeline takes as input a video and performs a face detection and tracking. Regions of Interest (ROI) are extracted from the detected face in order to extract the shape information when applying the Histogram of Oriented Gradient (HOG) descriptor. The number of features yield by HOG descriptor is reduced by means of a Linear Discriminant Analysis (LDA). Then, a deep data analysis was carried out, exploiting the pipeline, for the objective of setting up the LDA classifier. The analysis aimed at proving the suitability of the decision rule selected to separate the facial expression clusters in the LDA training phase. To conduct our analysis, we used ChonKanade(CK+) database and F-measure as an evaluation metric to calculate the average recognition rates. An automatic evaluation over time is proposed, where labelled videos is utilized to investigate the suitability of the pipeline in real world condition. The pipeline results showed that the use of HOG descriptor and the LDA gives a high recognition rate of 94.66%. It should be noted that the proposed pipeline achieves an average processing time of 0.018 second, without requiring any device that speeds up the processing.

Keywords Facial expression recognition · Histogram of Oriented Gradient · Linear Discriminant Analysis

Latifa Greche
E-mail: Latifa.greche@usmba.ac.ma

Mohamed Akil
E-mail: mohamed.akil@esiee.fr

[a] Laboratoire énergies renouvelables et systèmes intelligents, Faculté des Sciences et Techniques, USMBA, Fès, Maroc. · [b]Laboratoire d'Informatique Gaspard-Monge, A3SI, ESIEE Paris, CNRS, Université Paris-Est, Marne-la-Vallée, France

1 Introduction

The early studies and analysis of facial expressions were studied only in the psychological field where Charles Darwin [1] demonstrated for the first time that despite the differences between individuals and cultures, six facial expressions are innate and common among human beings, which are joy, fear, sadness, surprise, anger and disgust. With the emergence of new technologies and robots, the psychologist provided the machine learning and computer science community with labeled databases [2–4] which contain a set of face images of universal expressions or a set of micro-expressions [5] information. Facial expression recognition has many applications in security, human-machine interaction, and medicine. It is an active area of research where different solutions have been proposed over the years. Generally, proposed solutions can be grouped in two categories.
The first category uses a video sequence to build dynamic models for facial expressions. This category is based on approaches that extracts spatio-temporal features [6–9] from the face over time. Although, this first category provides more information that is useful for

expression analysis, the dynamic aspect of those models requires considering the neutral expression as reference to predict the exhibited expressions [10]. Therefore, the dependency to the neutral expression limits the use of dynamic models in real world scenes where the expression is unpredictable.

The second category of solutions extracts particular features from single images. Such solutions get rid of the dependency to the neutral expression. In Shan et al. [11] authors investigate the Local Binary pattern (LBP [12]) descriptor for low-resolution FER. Some modified versions of LBP were proposed for expression analysis like the Local Tenary Patterns (LTP) [13] and Compound Local Binary pattern (CLBP) [14]. Uçar et al. [15] proposed curvelet transform and online sequential extreme learning machine (OSELM) with radial basis function (RBF) hidden node having an optimal network structure. In the state-of-the-art, a shape descriptor has been used in recent FER studies which is HOG descriptor. It was first proposed by Dalal et al. [16] for human detection and then studied as part of FER. D. McDuff et al. [17] present a real-time FER toolkit that automatically recognizes the expressions of multiple people simultaneously. Authors extract in that work HOG features from regions of interest defined by a set of landmark points delimiting the face components like the eyebrows, the eyes and mouth. Then, the Support Vector Machine (SVM) classifier is trained on 10000 images, collected from around the world over the Internet [18] after filming volunteers while they are watching videos that stimulate their emotions. However, the landmark points have been manually localized in the images. For the real time recognition the toolkit user must be connected to a server in order to exchange and preprocess the frames instantly. The HOG descriptor has been studied and compared to many descriptors in some recent studies. P. Carcagni et al. [19] presented with details a comprehensive study of the HOG extractor in FER problem. Where, HOG descriptor has been compared to LBP, CLBP, and Spatial Weber's Local descriptor (SWLD). The descriptors' comparison has been realized with an SVM classifier that measures the recognition rate related to each descriptor. The authors showed that HOG features are more relevant than the LBP, CLBP and SWLD features. K. Lekdioui et al. [20] compared the HOG descriptor to other descriptors like the LBP, LTP, and the pair combinations of the descriptors which are HOG-LBP, HOG-LTP, and LBP-LTP. The performance of recognition has been measured using the SVM classifier. The descriptors' evaluation demonstrated that the association of LTP (a texture extractor) and HOG (a shape descriptor) is the appropriate method that describes facial expressions. Through these studies and comparisons, the authors showed that HOG is one of the useful shape and form descriptor of facial expressions if compared to the texture descriptors. Therfore, our interest goes towards using the HOG descriptor because of its capability to discriminate the shape and the changes of facial expressions, by computing the distribution of edge directions related to the facial traits, on the one hand. On the other hand its capability to get rid of problems related to the variation in shadowing and illumination [16].

Automatic facial expression recognition is a composite task that involves the expression classification stage in addition to the feature extraction stage. Some studies [19–21] put emphasis on searching the relevant features to extract from facial expression data. This paper will focus on analysing the hidden structure of facial expression data, after being encoded by HOG features, for the objective of gaining high FER rates and reducing the processing time until reaching a real-time FER that fits the video speed of 24 fps. In fact, it is difficult to decide which classifier is relevant for the facial expression classes before the hidden structure of data classes is analysed and understood. Visualizing the hidden structure of data classes can be helpful to investigate the suitability of different classification methods. However, it is challenging to visualize the data structure when the subjects within the data are represented by more than three features. To solve this problem some dimensionality reduction methods are proposed like the Principal Component Analysis (PCA) [22] and the LDA [23]. Though the both methods can reduce thousands of features extracted from each face image into only two or three features, there are differences in the manner they transform the dataset when reducing the number of features. As the PCA ignores the class labels during the search of a low-dimension set of axes that summarize data, reducing our data using this reduction method will certainly cause an information lose related to the clusters' separability within the data. The LDA, however, focuses more on finding a low-dimensional set of axes in which the data classes separability increases. It has been widely used. However, in facial expression recognition issue and related to HOG features the LDA has seldom been addressed. M.H. Sidiqi et al. [24] proposed curvelet transform [25] to extract features from images. The features are then reduced by employing linear discriminant analysis (LDA). Finally, the hidden Markov model (HMM) is used to recognize the expressions. M.H Sidiqi et al. [26] used a Stepwise Linear Discriminant Analysis (SWLDA) [27] that focuses in selecting the features from the images. For recognition, the hidden conditional random field model is applied. J. Wang et al. [28] investigated the usefulness of 3D fa-

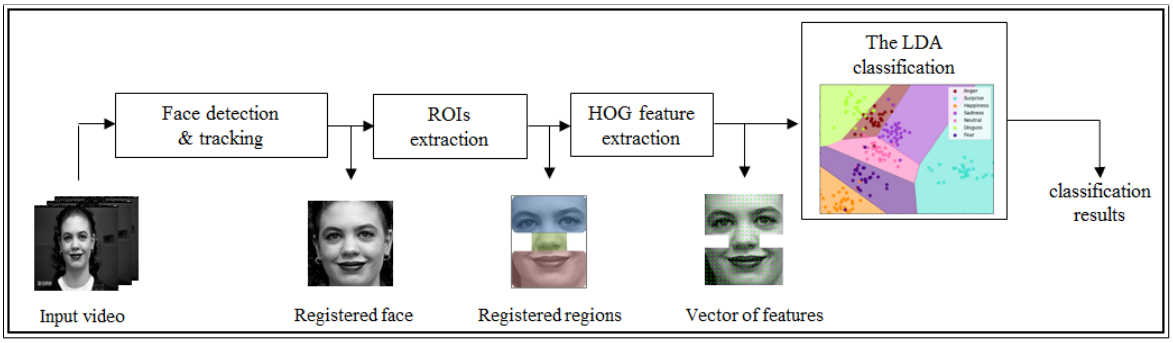


Fig. 1 The overall view of the proposed pipeline

cial geometric shapes to represent and recognize facial expressions, where the LDA classifier has been used to evaluate the performance of recognition under different head poses.

Usually, the LDA is used for dimensionality reduction rather than for classification. In fact, the choice of whether to use the LDA only for reduction or for classification too depends on the relationships between the classes when the original data is transformed into a new set of low-dimensional data. In other words, the presence of significant overlaps between the classes, in the scatter diagram of the transformed data, excludes the LDA from the option of being used as a classifier. However, when the data are separable, the LDA makes the class separation easy with less overlap and can be exploited as a supervised classification method if combined with the suitable decision rule allowing the correct classification of the test set. As the main aim is to reach an effective recognition, we are interested in reducing the number of HOG features, related to each subject, with maximizing the between-class variation and minimizing the within-class variation in the reduced data. For the above mentioned purpose the LDA has been used for dimensionality reduction. Furthermore, our preliminary analysis [29] for exploring facial expression data, by means of multiple feature descriptors and classifiers, showed that the LDA can be used also as a powerful classifier for HOG data. This conclusion was reached after automatically tuning the parameters of a preselected set of image descriptors and classifiers and comparing the recognition rates of their pair combinations (extractor-classifier). B. Hariharan et al [30] showed as well that the use of the whitened HOG descriptor and the LDA classifier instead of the SVM classifier allows building a competitive object detection system that is significantly faster not only at the level of the training time but also the detection time.

The work in this study consists then of adapting the processing chain, that was essentially used to explore images and to work on real world videos.

Making FER systems reliable is challenging for practical real-time application. To address this challenge, this paper proposes the following contribution, which is a real-time pipeline based on a processing chain that extracts the shape of the face from images and makes an extreme reduction of the extracted features with preserving the essential needed information for recognition. The pipeline takes as input a video and returns the expression of individuals whom frontal face is correctly detected in the frames. In addition to the previously mentioned contribution, an automatic evaluation technique has been carried out in order to evaluate the pipeline using labelled videos instead of labelled images. Evaluating the pipeline using real world videos allows enhancing the pipeline functioning, Particularly when the frames where the false alarms are detected are registered and added to the training set. This would help the pipeline to make a self-learning while reviewing new images.

The remainder of this paper is divided as follows, in Section 2, a new pipeline to automatically recognise the expressions of multiple faces has been proposed. Then, in Section 3 a set of analysis has been carried out for the objective of setting up the LDA classifier and investigating the pipeline suitability in real world application. Finally, Before concluding and proposing some future works in Section 5, a discussion and comparison of the pipeline results to the state-of-the-art has been presented in Section 4.

2 The methodology overview

The recognition of facial expression requires a pipeline that involves different processing steps. The pipeline in figure 1 has been used in this work: the first step detects the faces within the frame and then the registered one passes through the ROIs extraction step in order to register the most expressive parts of the face. This preliminary operation allows extracting the top region

of the face, including the eyes, the eyebrows, and the forehead, the nose region, and the mouth region. Therefore, relevant information like the shape and the wrinkles can be described by applying the HOG extractor. Finally ,for the recognition of the appeared emotions, the vector of features extracted by HOG is classified by the LDA classifier. The pipeline operating steps are detailed in the following subsections.

2.1 Face detection and tracking

As people spend time looking straight at their device screen while working, texting, reading or watching videos, the interest goes towards detecting and processing frontal faces in the frames provided by the front-camera of those devices. In this study, static images and video frames are scanned by the fast Viola and Jones face detector [31] in order to find out faces at multiple scales and positions. Detected faces are then tracked over time using an accurate method which is the Kanade-Lucas-Tomasi (KLT [32]) method. Usually, the tracker is automatically executed once the face detector finds the face in any frame. It extracts the feature points, which are the Harris corners illustrated in figure 2, from the face and tracks them from one frame to another till the face disappears from the scene.
The most important role of the tracker is to reduce the image processing time by reducing the image area where the face detector will search for faces. The time is further reduced by skipping some video frames when recalling the face detector. For this, after measuring the pipeline speed while skipping certain number of frames every time the pipeline re-runs, we decided to make the face detector works after every 10 frames. The interesting point here is even if the face detector skips 10 frames, the face tracker operates to mantain the face tracking from one frame to another without skipping any frame, which keeps the pipeline processing works at the real time frame rate. However, The challenge lies in finding new faces. In other words, if a new face appears in the scene, just after calling the face detector, the pipeline will detect and process the new face only after skipping 10 frames. Analysing this point in terms of time, the skip of 10 frames in the case of devices having a shooting video of 30 fps means waiting 0.33 seconds before updating the face detection stage to be able to search new faces, knowing that the tracking is maintained for the already detected ones during the 0.33 seconds. When running the pipeline, the effect of the delay of 0.33 seconds is not visually observable as it is related only to the first face detection and once the face is detected the expression will be analysed from one frame to another till the face disappears.

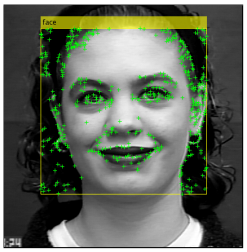


Fig. 2 The feature points extracted from the face and used for the tracking

2.2 ROIs extraction

Once the face tracking is ensured in the video, the ROIs containing transient features like the wrinkles, the forehead lines, the frown lines, and the laugher lines and permanent features like the forehead with the eyes, the nose, and the mouth are extracted from each face window. Thus, the irrelevant face parts are eliminated while facial regions containing transient and permanent features that are prone to changes are selected.
As long as the face is detected in its frontal position, the ROIs can be localized easily and rapidly by using geometric method [33] rather than using detectors for facial components like the eyes, the nose, and the mouth detectors. The geometric method is based on the fact that human faces have similar geometric configuration and there is a proportionality between the face components dimension, such as the proportionality between the face length and the position of the eyes or the mouth. Thus, the three ROIs have been localized using this method as follows:

- The face window having L x L dimensions is divided into 6 x 6 sub-regions as shown in figure 3.
- For each ROI, specific sub-regions are selected. For example, the region of the eyes is bounded by sub-regions (2-5) horizontally and (2-3) vertically.

Emotions stimulate initially the face muscles. Therefore, the movement of the muscles appears through two types of face shape changes involved when a new emotion is occurring. The first change is related to the movements of permanent features. The second change concerns the transient features that appear only when

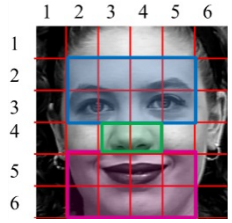


Fig. 3 The face partitioning and ROIs extraction

the emotion is occurring. The best way to describe facial expression is to extract the shape of the face components. By shape, we mean the relative positional information that can be formed while computing particular information like the local distribution of the edge directions [34]. For this reason, the powerful shape descriptor which is HOG extractor has been used to construct the proposed pipeline.

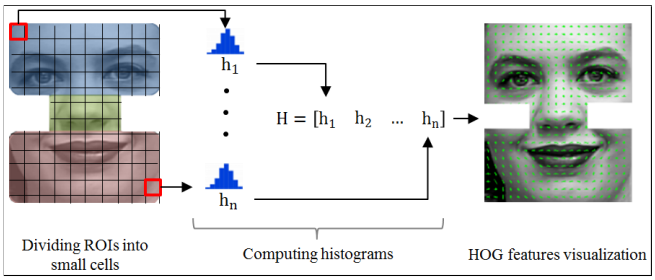


Fig. 4 HOG feature extraction from the ROIs

2.3 HOG descriptor

Allows extracting the local shape from the regions of interest as illustrated in figure 4. It calculates the distribution of local edge directions in small cells of size 8×8 pixels. At each pixel, the horizontal g_x and the vertical g_y gradients are calculated using 1-D centred derivative masks $[-1\,0\,1]$ and $[-1\,0\,1]^T$. A gradient orientation $\theta_{(x,y)} = \arctan \frac{g_x}{g_y}$ is used to vote into eight corresponding orientation histogram bins equally spaced between $0 - 360^o$. Then, the votes are weighed by a gradient magnitude $M_{(x,y)} = \sqrt{g_x^2 + g_y^2}$ and accumulated over pixels of each cell to construct one resulting histogram h_i. Histograms calculated over blocks are normalized with an overlap of 50% to provide better invariance to the change of illumination. Finally, all histograms h_i are concatenated in order to construct the final vector $H = [h_1, h_2 \ldots, h_n]$.
Once the HOG descriptor extracts the vector of features, the pipeline final step classifies that vector. However, before going through the classification step, a training phase is required in order to construct the LDA classifier.

2.4 LDA training

The training and the classification can only be achievable in two distinct phases. The scheme of the training phase is illustrated in figure 5. The scheme consists of the same above mentioned pipeline preprocessing steps in addition to the LDA reduction step. However, no face tracking was needed for the face detection step since labelled images were used instead of video sequences. The labelled images were collected from two different databases in order to diversify the subjects within the training set. The ChonKanade Image Database (CK+) [4] is specifically acquired for facial expression analysis. This choice has been made according to the following reasons: the diversity of samples within the same classes in terms of gender, as well as ethnicity and age; the necessity to compare the performance of our proposed pipeline to the ones of the state-of-the-art. The CK+ contains image sequences of people with seven facial expressions (joy, surprise, anger, fear, disgusts, sadness and neutral state). Each sequence begins with the neutral expression and ends with an expressive face. To increase the diversity of our training set, which is essentially based on CK+ images, the Yale Face [2] image database was also used. It contains 165 images of 15 subjects viewed under 11 different observation conditions. Where, subjects express different expressions like joy, sadness, boredom, surprise, and wink under different condition of illumination or with partial occlusions.
The scheme takes as input the training set that was labelled according to the universal expressions of emotion. Then, the images pass through the three preprocessing steps which are the face detection, the ROIs and the HOG feature extraction steps. At the output of the HOG descriptor, the vector of features extracted from each face image is stored within the matrix $data_{m,n}$ represented as follows:

$$data_{m,n} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m,1} & h_{m,2} & \cdots & h_{m,n} \end{pmatrix} \quad (1)$$

Where $data_{m,n}$ is the resulting labelled data after extracting HOG features from the images, the rows refer the vector related to each image within the database, n is the number of features extracted from the ROIs, and m is the number of images.
LDA reduction [23]: when the resulting matrix $data_{m,n}$ is given, the number of variables n can be reduced. The idea is to transform the matrix $data_{m,n}$ having

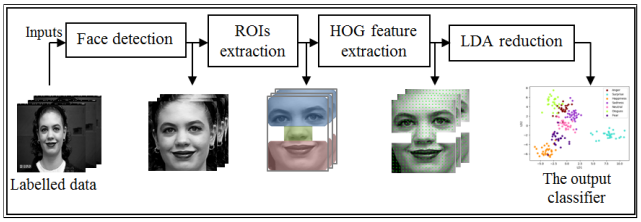


Fig. 5 The training phase of the LDA classifier

n variables into a data, having two variables and representable in a scatter diagram, allowing a deeper vision of data classes distribution in low-dimensional space. This enables us to understand the geometric relationship between the data classes in the training phase, so that those classes can be separated effectively and thereby construct an appropriate classifier for facial expression classes.

So, to reduce the resulting matrix $data_{m,n}$ that is composed of n=768 variables, which are the HOG features $h_{i,j}$, the LDA has been used. Applying the LDA on the resulting matrix creates a new set of features where the number of the variables n is reduced into two variables. Therefore, creating the new two-dimensional space of data involves computing various statistics about the resulting matrix like computing the covariance matrix which can tell us about the scatter within the data. To compute this scatter the covariance is multiplied by the probability of the classe P_C (that is the number of subjects inside the class divided by the total number of subjects in the data). Summing the scatter values related to each class gives us the within-class scatter (S_W). With this measure we can know whether images belonging to the same class are tightly clustered together. As the within-class scatter gives us the information about how much the class images are tightly clustered, we need also to get the inter-class information in order to know if data classes are easy to separate. For this, the between-class scatter (S_b), which refers the distance between the classes, has been computed. Generally, getting a small value of S_W and a large value of S_b means that the data is easy to separate into classes. The calculation of S_W and S_b are the following [35] :

$$S_w = \sum_{classes C} \sum_{i \varepsilon C}^{M} P_C.(h_i - \mu_C)(h_i - \mu_C)^T \quad (2)$$

$$S_b = \sum_{classes C} (\mu_C - \mu)(\mu_C - \mu)^T \quad (3)$$

Where $\mu_C = \frac{1}{M}\sum_{i=1}^{M} h_{(i,j)}$ *and* $\mu = \frac{1}{m}\sum_{i=1}^{m} h_{(i,j)}$ *are the mean vector in each class and* μ *is the overall mean, respectively.* M *and* C *are the size of samples in each class and the class, respectively.*

Making $S_W^{-1}S_b$ as large as possible is the key to reduce the thousands of data variables into two discriminant variables that contain almost all the classification information embedded in the original images. Those variables are computed by resolving the generalised eigenvectors of the matrix $S_W^{-1}S_b$ in the equation 4 and selecting the eigenvectors v that have the maximal eigenvalues λ:

$$Av = \lambda v \quad (4)$$

Where $A = S_W^{-1}S_b$ *is the between-class scatter divided by the within-class scatter assuming that* S_W^{-1} *exists.*

The two selected eigenvectors, having maximal eigenvalues, constitute the columns of the two variables of the reduced data. Those two variables are then plotted in the scatter diagram of figure 6. This figure represents the spread of 248 face expressions clustered into seven classes after being transformed and reduced by the LDA, where the axes LDA 1 represents the first eigenvector and the axes LDA 2 represents the second eigenvector.

The produced diagram shows that the classes can cluster separately. However, there's an overlap of samples at the boundaries of the seven classes like the class of happy faces with scared faces. Therefore, we must construct our classifier by separating the data clusters with the adequate decision rule. In this context, the pipeline has been exploited to test multiple decision rules over the produced clusters. For that different classification techniques have been associated with the LDA in which the LDA decision rule has been replaced by the decision rule related to other classifiers like the k-NN, the Naïve Bayes, the SVM, and the Binary Tree. Then, the recognition rate has been measured for each combination test, so that the LDA decision rule can be chosen according to the combination having the maximum recognition rate.

After finishing the training and fixing the decision rules of the LDA classifier, the last pipeline step uses this classifier to recognize expressions in real world videos.

2.5 Classification

At this step when a new face image with the vector of features X is projected in a particular location in the

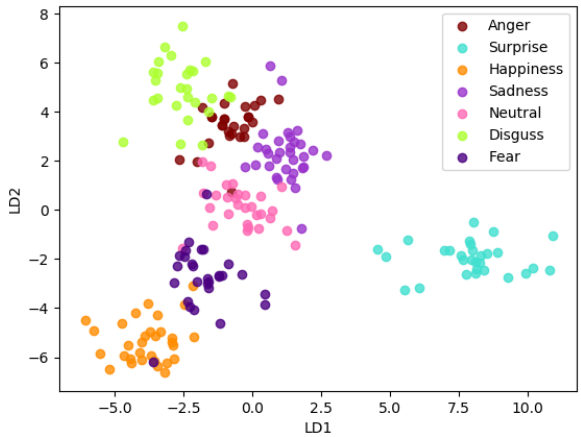


Fig. 6 The scatter diagram of the CK+ database after extracting HOG features and reducing dimension by the LDA

scatter diagram, the LDA classifier relies on a multivariate Gaussian to estimate the probability that the facial expression X belongs to a particular class (called also the posterior probability $P(C = c_i/X = x)$), where c_i refers the ith class and $i\epsilon\{1, 2, 3, 4, 5, 6, 7\}$ refers the class number. The posterior probability is computed using the Bayes rule witch is equal to the probability of X knowing the class $P(X = x/C = c_i)$ multiplied by the probability $P(C = c_i)$ of the class c_i and divided by the sum of the probabilities of the facial expression X under all the possible classes. So, the posterior probability is defined as:

$$P(C = c_i/X = x) = \frac{P(X = x/C = c_i)P(C = c_i)}{\sum_{j=1}^{7} P(X = x/C = c_j)P(C = c_j)} \quad (5)$$

As we assume that our data come from a multivariate distribution, the $P(X = x/C = c_i)$ can be computed as follows:

$$P(X = x/C = c_i) = \frac{exp(-\frac{1}{2}(x - \mu_i)^T Co_i^{-1}(x - \mu_i))}{\sqrt{2\pi \mid Co_i \mid}} \quad (6)$$

Where, μ_i is the vector mean and Co_i is the covariance of the classe i.
The requested facial expression X is classified as a class c_i according to the class that gives the maximum posterior probability, given by:

$$P_{max} = \underset{c_i}{\text{argmax}}\{P(C = c_i/X = x)\} \quad (7)$$

After the LDA recognizes the face expression, the pipeline outputs the location of the faces (if there are so many within the frame) as bounding boxes and exhibits the face expression on the top of those bounding boxes. In addition to that, the pipeline records the result of recognition and the face location for the objective of evaluating the pipeline.
On the whole, this section describes the pipeline steps and details the training phase in which further analysis have been realized to set up the LDA classifier.

3 The pipeline evaluation

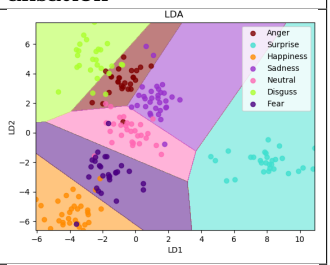
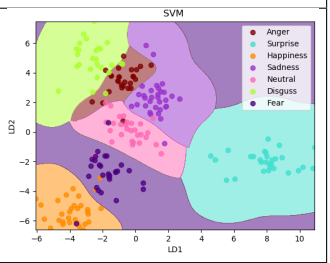
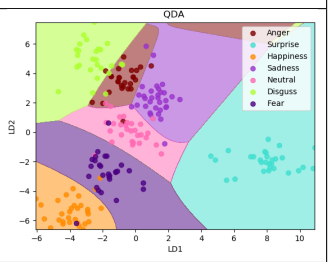
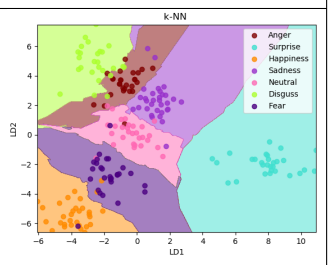
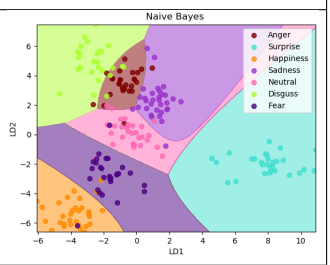
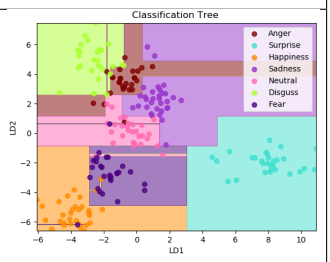
In this section many experimentations and tests have been carried out to evaluate our proposed pipeline functioning when using real world videos. Before the evaluation, first, the pipeline has been exploited to conduct deep analysis on the hidden structure of data and set up the decision boundary of the LDA classifier.

3.1 The LDA evaluation

An effective use of the HOG descriptor in FER requires the use of robust and fast classification method. However, choosing the appropriate classification tools is difficult if one does not have a prior knowledge about the structure of the data classes. By knowing the hidden structure we can understand how the data classes are organized and clustered. The structure means the geometric relationships among the data classes in two-dimensional space. An example of a structure is linear and non-linear relationships among the vectors of the classes. Therefore, understanding the geometric relationships among the data classes will help us figure out the suitable decision boundary that separate the classes.

We need to choose the appropriate decision boundary for the LDA classifier. For this purpose, table 1 summarizes the analysis after testing different decision boundaries to separate the clusters within the LDA classifier during the training. The generalization ability of the hybrid classifiers, presented in the second column of the table, reveals that associating the LDA with the Multivariate Gaussian or Naïve Bayes makes the recognition better with an average recognition rate of 96.44% and 95%, respectively. However, combining the LDA with the SVM, the k-NN, and the Binary Tree doesn't enhance the recognition of the facial expressions at all. The problem with the Quadratic Discriminant Analysis (QDA) and the other classifiers is that even if the decision boundaries between the classes seem to be correct in the scatter diagrams at the third column of the table 1, those methods fail in classifying new requested images and that's because they make wrong decision boundaries in regions where no data points are available. This problem is noticeable in the case of the scatter diagram that represents the SVM decision boundary, in table 1. Looking at the SVM decision boundary, it is obvious that the top right region of the scatter diagram is identified as a part of the sad class while there is no point that belongs to this class in that region. Thus, in this case, if a requested face image is projected within this region the expression of that face will be misclassified. The analysis of table 1 demonstrates that the LDA with the multivariate Gaussian seems to be suitable for the classification of the seven facial expression classes. That's because it can make good decision boundaries by separating, linearly, the seven classes (as shown in figure 7) and can classify, accurately, requested images with a maximal average recognition rate of 96.44%. Once the LDA decision boundary is fixed, the pipeline uses the LDA classifier to recognize the face expressions in videos collected from the MMI database and videos

Table 1 The recognition results and the visualization of the hybrid classifiers

Methods	Recognition rate	Decision boudaries visualisation
The LDA with the Multivariate Gaussian	0.96	
The LDA with the SVM	0.43	
The Quadratic Discriminant Analysis (QDA)	0.80	
the LDA with the k-NN	0.77	
The LDA with the Naive Bayes	0.95	
The LDA with the Binary Tree	0.43	

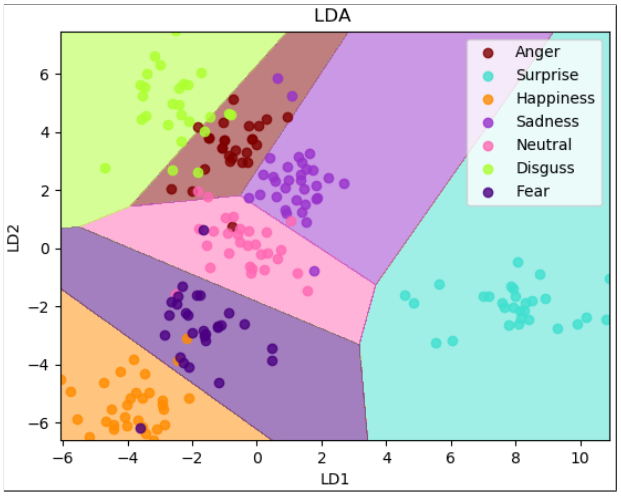


Fig. 7 The scatter diagram showing the linear decision boundaries of the LDA classifier

filmed in real word in the presence of more than one person at the scene.

As it was mentioned in the subsection 2.4 that the training is made using the Yale Face and the CK+ databases, it is worthwile to make a closer look regarding the generalization ability of the LDA classifier. According to the obtained results the LDA is general enough to be representative of those databases and some real world situations as we can see in the Subsection 3.2. The CK+ database is widely used for algorithm development and evaluation of facial expression recognition systems [19,20,36,37] The reason of this is in addition to the diversity of samples within this data, it includes also video sequences that vary in duration (i.e. 10 to 60 frames) and incorporate the onset (which is the neutral frame) to peak formation of the facial expressions. On the other hand, the benefit behind using the Yale Face together with the CK+ database is just to increment the diversity of samples and to add images with partially occluded faces to the experimental dataset. However, we still can't generalize in real world data in complicated situations like occluded and rotated faces, because the training data contain only few samples considering those situations. To make the LDA classifier general enough to be representative of the real world it should be trained using an important quantity of real world data. To make this possible, the pipeline can register the results of recognition including the face image and the expression associated with, so that the false recognition ones can be selected and used to retrain the LDA classifier.

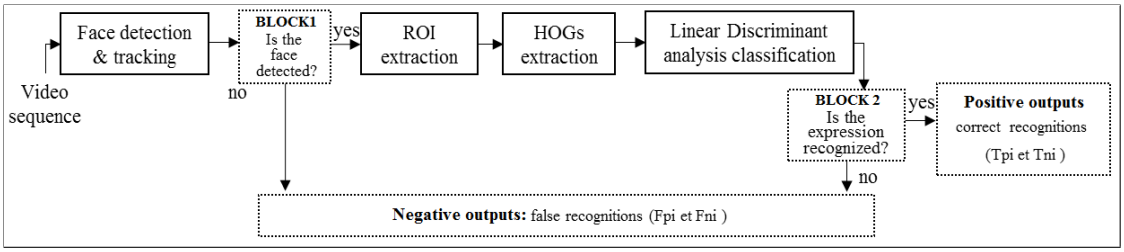


Fig. 8 The automatic evaluation technique of our pipeline using real labelled videos

3.2 The evaluation of the proposed pipeline

The challenge of evaluating the pipeline in real world videos leads us to build the scheme presented in figure 8. The scheme consists of the same pipeline processing elements in which two additional operating blocks have been added. The first block aims at evaluating the face detector and the second one aims at evaluating the LDA classifier. For the automatic evaluation, first the scheme takes as inputs a set of labelled videos that contains frames with faces displaying different transitions between facial expressions. Those videos were collected from the MMI database [3]. The reason for choosing this data is diversity of subjects and the existence of partially occluded faces by hair and glasses in addition to the various lighting conditions. The face detector scans the video frames in order to find out the faces. At the face detector output, the block 1 has been added to ensure that the face window found by the face detector is a true positive.

Block 1: To decide whether the outputs of the face detector are faces, the block 1 compares the face positions that have been manually extracted from the video frames (the reference vector) to those given by the face detector when the same video is being entered then processed by the pipeline. As the face of individuals participating in the MMI videos are in the same position over the frames, it was easy for us to set the face position for all the videos and thereby to construct the reference vectors for the block 1. Therefore, if the face positions that are given by the face detector don't match with the face position in the reference vector then the recognition of the facial expression fails. Then, the output of the Block 1 could be a false positive F_{p_i} (the face detector can't find the face while it exists in the image) or a False Negative F_{n_i} (the face detector estimates are found the face while it doesn't exist). In this case, the outputs of this block F_{p_i} and F_{n_i} will be stored in the false recognition metric. Otherwise, if the face position given by the face detector matches the one in the reference vector, the detected face pass throughout the remaining pipeline steps till arriving at the second block that evaluate the recognition results outputted by the LDA classifier.

Block 2: allows comparing the pipeline outputs with the reference vectors stored in this block.To do this, the reference vector related to each video sequence used in the experimentation has been manually constructed as follows. As almost videos in the within the MMI database begin with a neutral expression that changes over time to achieve a high-intensity expression and then returns to the neutral expression, the reference vector can be constructed by assigning the value 1 to the frames that contain the high-intensity expression and 0 otherwise, as illustrated in figure 9. To avoid assigning manually the label for each frame in video sequences that lasts on an average of 6 seconds with a speed of 30 fps, the solution is to just localize the transition frames as follows: T_1 where the expression changes from the neutral state to the high-intensity expression and T_2 where the expression changes to go back to the neutral state. So, The frames between the transitions T_1 and T_2 will have the same label that is equal to 1 and the ones at the left of T_1 and the right of T_2 are labelled with the value 0. However, for few videos the expression begins and ends with a minimal intensity like the sequence presented in figure 10. In this case, all the frame are assigned into 1. For us a sequence is considered correctly recognized when the frames containing the neutral expression are correctly recognized and the frames containing the basic expression of the video sequence are also correctly recognized by the classifier.

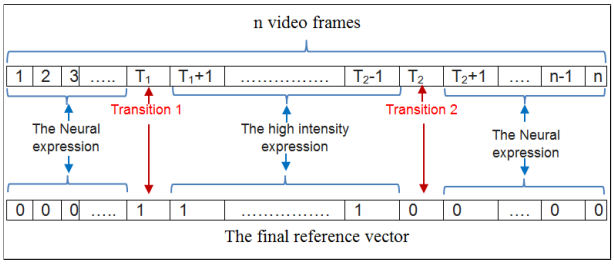


Fig. 9 A simplified scheme showing the construction process of the reference vector

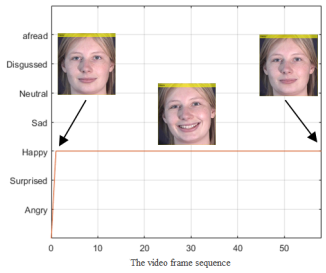


Fig. 10 The recognition results of the video "S036-005.avi" taken from the MMI database

Table 2 The confusion matrix after the pipline evaluation over video sequences.

	jo	su	fe	di	an	sa	ne
Jo	95.7	0	3.1	0	0	0	1.2
Su	1.3	96	1.9	0	0	0	0.8
fe	5.12	7.01	85.09	0	0.08	0.1	2.6
di	0	0	0	89.92	3.58	3.1	3.4
an	0	0	0.84	1.6	91.1	4.02	2.44
sa	0	0	1.3	2.9	8	83.49	4.31
ne	0.3	0.3	5.5	0.9	8.94	10.23	73.83

with jo, su, fe, di, an, sa, and ne refer to joy, surprise, fear, disgust, anger, sadness, and neutrality respectively

Finally, for each video, the reference vector is constructed and stored in the block 2, so that the block can compare the pipeline outputs $\widehat{y}_i$ with the elements of the reference vector y_i as follows:

$$S_i = \begin{cases} y_i & if\ \widehat{y}_i = y_i \\ \widehat{y}_i & if\ \widehat{y}_i \neq y_i \end{cases} \quad (8)$$

Where y_i is the face expression in the reference vector at the i-th frame, $\widehat{y}_i$ is the face expression classified by the LDA at the i-th frame

The block can decide whether the expression is correct or not using the following condition. If $S_i = y_i$ then the resulting S_i is a positive outputs, which means the facial expression is correctly classified (see scheme 8). Otherwise, if $S_i = \widehat{y}_i$ then S_i is a negative outputs, which means the facial expression is misclassified.

On the whole, the positive and negative outputs are used to construct the confusion matrix and compute the f-measure [38] which gives a precise calculation about the average recognition rate. Table 2 reports the resulting confusion matrix computed after the automatic pipeline evaluation. Knowing that the LDA has been trained by the CK+ and Yale Face databases, the tests using MMI videos showed that the pipeline is accurate in recognizing the expression of joy, surprise, and anger. Those three expressions have recognition rates, superior than 90%, which is normal for the joy and surprise expressions as they are the easiest expressions to discriminant[39,40]. However the recognition rates of fear, disgust,and sadness are limited between 90% and 80%. There can be many reasons why certain expressions are not correctly classified. the first reason is the low-intensity of the expressions when it changes from expression to another which makes the recognition particularly difficult at the frames of transitions. Another reason is the similarities in the appearance of some facial traits, like strict lips and when the eyebrows are down, makes confusions between the expressions of disgust, anger, and sadness.

The scheme has been adapted to continuously visualize the change of the expression over time. Figures 11(a) and 10 illustrate the FER in two different videos where the visualisation of the expression transitions has been automatically returned after executing the proposed pipeline. In figure 11 (a), a correct prediction is obtained over the frames containing the basic expression, whereas the prediction is unstable due to the pipeline uncertainty during the permutation of the face expression. The proposed pipeline has been designed to recognize the expression of multiple faces in the case where several people appear in the video scene. An example of expression recognition over four frames extracted from a real video has been presented in Figure 12, where three individuals are filmed using the front camera of a cell phone. After testing the pipeline on some real world videos, it should be noted that the proposed pipeline achieves an average processing time of 0.018 second for

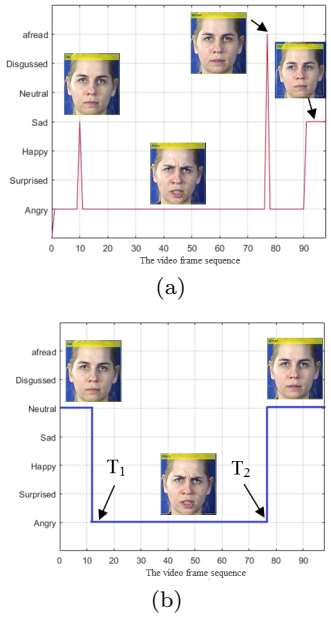


Fig. 11 The graphical representation of the facial expression variation over time in a video. (a) The recognition results of the video "S001-100.avi" taken from the MMI database. (b) The reference vector of the same video "S001-100.avi".

one face. This means that pipeline is applicable at least for devices having a shooting video of 24 fps. However, the more the faces are multiplied in the scene the more the time is consumed by the pipeline. Indeed, the processing time varies according to many variables which are the face size and the scene background complexity. Meaning that in the case of scenes containing faces having large dimension or containing many moving objects in the background, the pipeline consumes a lot of time in analysing the scene frames.

4 Discussion

For fair comparison with the state-of-the-art, many protocols must be respected which limits some choices regarding the database, the manner we split it, and the metrics we use during the pipeline evaluation. For this reason, the pipeline results (reported in section 3) have been compared with the results of the existing works that use in common: the CK+ database, the 10-fold or LOO cross-validation techniques, and the recognition metric F-measure. Based on those protocols, the comparison with the state-of-the-art have been done using two studies [20,19] in which the HOG descriptor has been used to extract facial expression features. Table 3 reports the comparison result demonstrating that our proposed pipeline provides a high average recognition rate of 96.44 and processing time of 18 ms. The pipeline computational cost, in terms of time, has been measured considering an average of 100 frames using Intel core i7 processors, CPU @ 2.20GHz 2.2GHz, and RAM of 16.0 Go.

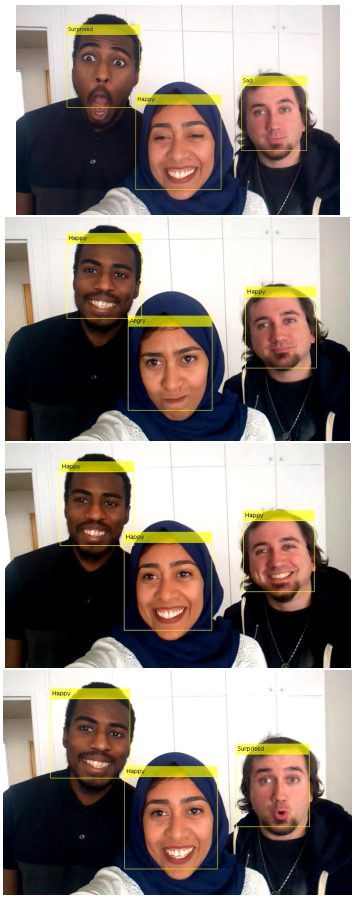


Fig. 12 The resulting recognition after applying the proposed pipeline on a real world video filmed by the front camera of a cell phone

Table 3 Performance comparison of our approach vs recent State-of-the-Art approaches (with using CK+ data, 10-fold, and F-measure)

	Carcagni et al. [19]	**Lekdioui et al. [20]**	**Proposed pipeline**
Av	94.1	96.06	96.44
time	43.38 ms	236.18 ms	18 ms

Av: is the average recognition rate (the F-measure)
time: is the whole preprocessing time of the systems
ms: refers the millisecond unit

Lekdioui et al. [20] propose a FER system that employs an automatic technique for finding the proper cropping of facial components. Their experimental analysis showed that the performance of HOG descriptor associated with a texture descriptor, like the LTP, and an SVM classifier provide a recognition rate of 96.06%. However, the recognition time demonstrated that this FER system have an expensive computational cost in terms of time and thereby can be applied only to static images.
Carcagni et al. [19] carried out a comprehensive study of the HOG descriptor as a part of the FER issue, showing that a proper tuning of HOG parameters can lead to a powerful representation of facial expressions and thus an effective recognition with an average of 94.1%. The simplicity of the model, which is based on HOG descriptor and the SVM classifier, have allowed to apply the pipeline in real-world operating conditions with a preprocessing time of 43.38 milliseconds. However, with the analysis carried out in subsection 3.1 and the pipeline evaluation in subsection 3.2 we emphasize that a potential reduction of HOG features by the LDA can speed up the FER with maintaining high recognition rates.

5 Conclusion

The need for building practical facial expression recognition system, lead us to propose a pipeline after making several data analysis. The pipeline first extracts

three ROIs isolating some face areas like the forehead with the eyes, the nose, and the mouth. The ROIs are then processed in order to extract the shape information using a HOG feature extractor. An extreme reduction of HOG feature has been realized by the LDA. For classification, the following classification techniques including k-NN, the Naïve Bayes, the SVM, the Binary Tree, and the pair combination of these classifiers with the LDA have been tested.

An automatic evaluation technique has been proposed to compute the pipeline performance of recognition when many expression transitions are present in the videos. Already trained by CK+ video sequences and Yale Face database, the pipeline evaluation using MMI video sequences showed that the proposed pipeline can be applied in real-time (i.e. at the frame rate of 24 fps) with an average processing time of 0.18 millisecond.

However, the implementation of the pipeline has highlighted several points that should be improved, namely: the adaptation of the pipeline in order to recognize further expressions such as recognizing the driver fatigue, to survey the state of patients and to recognize the micro-expressions. It is necessary to run this pipeline, constantly, in real scenes either if the facial expression is a standard expression or not. Parallelization is also needed in order to ensure fast facial expression recognition in the presence of dozens of faces in the video scene.

References

1. Charles Darwin and Phillip Prodger. *The expression of the emotions in man and animals*. Oxford University Press, USA, 1998.
2. A Georghiades, P Belhumeur, and D Kriegman. Yale face database. *Center for computational Vision and Control at Yale University, http://cvc. yale. edu/projects/yalefaces/yalefa*, 2:6, 1997.
3. Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *2005 IEEE international conference on multimedia and Expo*, page 5. IEEE, 2005.
4. Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010.
5. Paul Ekman and Wallace V Friesen. Measuring facial movement. *Environmental psychology and nonverbal behavior*, 1(1):56–75, 1976.
6. Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):915–928, 2007.
7. Yi Ji and Khalid Idrissi. Automatic facial expression recognition based on spatiotemporal descriptors. *Pattern Recognition Letters*, 33(10):1373–1380, 2012.
8. Muhammad Hameed Siddiqi, Rahman Ali, Adil Mehmood Khan, Eun Soo Kim, Gerard Junghyun Kim, and Sungyoung Lee. Facial expression recognition using active contour-based face detection, facial movement-based feature extraction, and non-linear feature selection. *Multimedia Systems*, 21(6):541–555, 2015.
9. Xijian Fan and Tardi Tjahjadi. A spatial-temporal framework based on histogram of gradients and optical flow for facial expression recognition in video sequences. *Pattern Recognition*, 48(11):3407–3416, 2015.
10. Fadi Dornaika, Elena Lazkano, and Basilio Sierra. Improving dynamic facial expression recognition with feature subset selection. *Pattern Recognition Letters*, 32(5):740–748, 2011.
11. Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and vision Computing*, 27(6):803–816, 2009.
12. Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.
13. Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6):1635–1650, 2010.
14. Faisal Ahmed, Hossain Bari, and Emam Hossain. Person-independent facial expression recognition based on compound local binary pattern (clbp). *Int. Arab J. Inf. Technol.*, 11(2):195–203, 2014.
15. Ayşegül Uçar, Yakup Demir, and Cüneyt Güzeliş. A new facial expression recognition based on curvelet transform and online sequential extreme learning machine initialized with spherical clustering. *Neural Computing and Applications*, 27(1):131–142, 2016.
16. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
17. Daniel McDuff, Abdelrahman Mahmoud, Mohammad Mavadati, May Amr, Jay Turcot, and Rana el Kaliouby. Affdex sdk: a cross-platform real-time multi-face expression recognition toolkit. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 3723–3726. ACM, 2016.
18. Thibaud Senechal, Daniel McDuff, and Rana Kaliouby. Facial action unit detection using active learning and an efficient non-linear kernel approximation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–18, 2015.
19. Pierluigi Carcagnì, Marco Del Coco, Marco Leo, and Cosimo Distante. Facial expression recognition and histograms of oriented gradients: a comprehensive study. *SpringerPlus*, 4(1):645, 2015.
20. Khadija Lekdioui, Rochdi Messoussi, Yassine Ruichek, Youness Chaabi, and Raja Touahni. Facial decomposition for expression recognition using texture/shape descriptors and svm classifier. *Signal Processing: Image Communication*, 58:300–312, 2017.
21. Zhengyou Zhang, Michael Lyons, Michael Schuster, and Shigeru Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Automatic Face and*

Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, pages 454–459. IEEE, 1998.
22. Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.
23. Peter N Belhumeur, João P Hespanha, and David J Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Technical report, Yale University New Haven United States, 1997.
24. Muhammad Hameed Siddiqi, Rahman Ali, Muhammad Idris, Adil Mehmood Khan, Eun Soo Kim, Min Cheol Whang, and Sungyoung Lee. Human facial expression recognition using curvelet feature extraction and normalized mutual information feature selection. *Multimedia Tools and Applications*, 75(2):935–959, 2016.
25. Min Tang and Feng Chen. Facial expression recognition and its application based on curvelet transform and pso-svm. *Optik-International Journal for Light and Electron Optics*, 124(22):5401–5406, 2013.
26. Muhammad Hameed Siddiqi, Rahman Ali, Adil Mehmood Khan, Young-Tack Park, and Sungyoung Lee. Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields. *IEEE Transactions on Image Processing*, 24(4):1386–1398, 2015.
27. Dean J Krusienski, Eric W Sellers, Dennis J McFarland, Theresa M Vaughan, and Jonathan R Wolpaw. Toward enhanced p300 speller performance. *Journal of neuroscience methods*, 167(1):15–21, 2008.
28. Jun Wang, Lijun Yin, Xiaozhou Wei, and Yi Sun. 3d facial expression recognition based on primitive surface feature distribution. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1399–1406. IEEE, 2006.
29. Greche Latifa, Akil Mohamed, Es-sbai Najia, and Kachouri Rostom. A novel tool for automatic exploration of feature extraction and classication methods: A case of study in facial expression recognition. *The manuscript is submitted for publication.*
30. Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. In *European Conference on Computer Vision*, pages 459–472. Springer, 2012.
31. Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *null*, page 734. IEEE, 2003.
32. Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.
33. Shi-Hong Jeng, Hong Yuan Mark Liao, Chin Chuan Han, Ming Yang Chern, and Yao Tsorng Liu. Facial feature detection using geometrical face model: an efficient approach. *Pattern recognition*, 31(3):273–282, 1998.
34. Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in neural information processing systems*, pages 831–837, 2001.
35. Stephen Marsland. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC, 2011.
36. SL Happy and Aurobinda Routray. Automatic facial expression recognition using features of salient facial patches. *IEEE transactions on Affective Computing*, 6(1):1–12, 2014.
37. Marian Bartlett, Gwen Littlewort, Tingfan Wu, and Javier Movellan. Computer expression recognition toolbox. In *2008 8th IEEE international conference on automatic face & gesture recognition*, pages 1–2. IEEE, 2008.
38. Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.
39. Philipp Michel and Rana El Kaliouby. Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 258–264. ACM, 2003.
40. Montse Pardàs and Antonio Bonafonte. Facial animation parameters extraction and expression recognition using hidden markov models. *Signal Processing: Image Communication*, 17(9):675–688, 2002.