Facial Expression Recognition from Video using Geometric Features

Yacine Yaddaden[1], Mehdi Adda[2], Abdenour Bouzouane[1], Sebastien Gaboury[1] & Bruno Bouchard[3]

[1]Université du Québec à Chicoutimi (UQAC), Canada
[2]Université du Québec à Rimouski (UQAR), Canada
[3]IEEE Senior, ACM Professional Specialist in Artificial Intelligence
{yacine.yaddaden1, abdenour_bouzouane, sebastien_gaboury}@uqac.ca, mehdi_adda@uqar.ca

Keywords: Facial Expressions, Emotion Recognition, Human Computer Interaction, Machine Learning, Data Mining.

Abstract

Human Computer Interaction has a significant impact in different fields of Information and Communication Technology. It is mainly due to the importance of interaction between human beings and the technologies they are using. Therefore, facial expression recognition has been widely used to enhance this interaction and make it more natural. Most of the proposed methods are based on images and even if they showed good performances, they do not match the real interaction model of people. Video is a rich source of information when considering the presence of temporal aspect. In this paper, we propose an approach that uses videos to recognize facial expressions and based on geometric features. We have tested it on a popular dataset and the carried experimentations showed promising results.

1 Introduction

Emergence and spread of Information and Communication Technology (ICT) made Human Computer Interaction (HCI) a real necessity to ease the use of technology and making it more natural for people. There are different ways and *modalities* for interaction that might be defined by two distinct categories: *unimodal* which uses a single source of information such as: speech, body gesture and facial expressions. The second category is the *multimodal* as a combination of two or more modalities aiming to improve the system accuracy by collecting more information. When it comes to designing a unimodal system, the best option is to use facial expressions. Indeed, Mehrabian [6] estimates that facial expressions contribute with 55% in the interaction while the verbal and vocal part contribute with 7% and 38%, respectively. Most of the proposed and existing approaches are based on Ekman [1] works. He considers that human beings can express six basic emotions: happiness, anger, disgust, sadness, fear and surprise. He has also introduced a major contribution that consists in describing facial expressions or emotions based on face muscle movement; it is called Facial Action Coding System (FACS).

Various approaches have been proposed in the last years. Some of them are static-based and recognize an emotion from a single image. Those approaches are simple to develop and yield good performances. However, they do not reflect the reality because an emotion might be split to different stages: onset (beginning), apex (highest intensity) and offset (ending). Dynamic-based approaches provide this missing information by taking into account the temporal aspect and using a sequence of images to describe as specific emotion. Nevertheless, they suffer from few issues such as the computation load and complexity when handling such a large amount of information.

In this paper, we propose a dynamic-based approach for recognizing emotions through facial expressions represented by sequences of images. The proposed approach consists of various steps. *Sequence normalization* that is performed to get the same number of images in each sequence. *Feature extraction* which extracts geometric-based features from each image of the sequence. Geometric-based features consist in facial fiducial points and for the proposed approach, two metrics have been used: *Euclidean distances* and *Variance*. The most relevant and representative features are chosen using *feature selection* that is based on a common statistical metric. The next step consists in *value normalization* used to represent the feature values within a specific interval. The last and the most important task is *classification*. Two different methods have been used, the first one is *Support Vector Machines* (SVM) for its ability to handle nonlinear data and the second one is *K-Nearest Neighbors* (KNN) for its speed and simplicity to be implemented. The choice of the method depends on which aspect is prioritized: complexity and accuracy (SVM) or simplicity and speed (KNN). There are several contributions in this paper. Indeed, we have introduced two different algrithms for sequence normlization and feature selection. Moreover, we have introduced the use of variance as metric to describe the variation of feature vector attributes over time that has as advantage to widely reduce the feature vector size. In order to validate the proposed approach, we used Cohn-Kanade Extended [5], a common and popular dataset of sequence-based facial expressions. The obtained results are promising as they outperform the existing approaches in terms of Recognition Rate (RR).

This paper is organized as follows: Section 2, introduces some facial expression recognition fundamentals and related work. The proposed approach is detailed in Section 3. Section 4 and 5, describe, respectively: the experimental protocol, the validation process and the obtained results. Finally, concluding remarks are presented in Section 6.

2 Related Work

Emotion recognition has always been considered as the best way to humanize machines and computers. It enhances and improves the interaction with them and makes it more natural. In order to design an effective system to recognize emotions, some fundamentals have to be understood. It consists mainly on the chosen modality, in our case facial expressions, and the recognition process related to this modality.

Depending on the type of the input, image or video, the system is classified as *static* or *dynamic*, respectively. The static one uses a single image to recognize the expressed emotion while in a dynamic system, the input consists of a sequence of images. Furthermore, there are two subcategories of video-based system: *frame-based* systems treat each image of the sequence separately and then use a specific vote function to select the detected emotion from a candidate set [12] [11]. *Sequence-based* systems process all the images and extract the needed information in order to perform the recognition. The next processing steps are basically the same in both static and dynamic systems.

Feature extraction extracts specific characteristics and provides a good representation of the input data. For facial expressions, there are three types of features [12] [11]. *Appearance-based* which consists in using texture descriptors in a pixel level. *Geometric-based* that depends on facial fiducial points to compute specific distances. The last type of features is *hybrid-based* and it combines the two previous ones. As, among these features, there are redundant and noisy ones, the *feature selection* is necessary. In this step, the different extracted features are ranked and sorted by priority, relevancy and only the more representative and discriminant ones are kept. The last step consists in the *classification* phase where one or more machine learning methods are used. Finally, a decision related to the expressed emotion is generated.

Based on this system description, various approaches have been proposed with variable performances. We begin by presenting some image-based approaches such as Ou et al. [8] who proposed to exploit Gabor filters coefficients and use them as appearance-based features. Then, they applied *Principal Component Analysis* (PCA) to select the more representative features before performing classification with KNN. Using a different type of features, Yu et al. [15] proposed an approach based on the extraction of the *Active Appearance Model* (AAM) and on which they applied PCA before using the final features to feed an SVM classifier. In the recent years, a new trend in machine learning has emerged. It consists at developing approaches based on *Deep Learning* (DL) [3]. For example, Li et al. [4] and Peng et al. [9] introduced methods based on *Convolutional Neural Network* (CNN). Those approaches are considered as all in one as they perform feature extraction, feature selection and classification using a single block.

As input data, a single image is considered insufficient regarding to the quantity of information it provides. For this reason and in order to fit with the real world interaction, video-based approaches have been introduced. Thereby, Lucey et al. [5] proposed to generate the AAM from the input video and then extract two types of features: *Similarity Normalized Shape* (SPTS) and *Canonical Normalized Appearance* (CAPP). They combined both features as input to an SVM classifier. Wan et al. [13] used geometric-based features that are extracted using *Active Shape Model* (ASM). Then, the different facial fiducial points are tracked over the sequence and then the vector is reduced using PCA before feeding an SVM classifier. Using the same type of features, Saeed et al. [10] proposed to extract the facial fiducial points and normalize them using mean and standard deviation values. They have also computed some distances based on these points and used two different classifiers: KNN and SVM. Mohammadian et al. [7] extracted *Facial Characteristics Points* (FCP) from a sequence of face images and reduced the feature vector by applying PCA and Linear Discriminant Analysis (LDA) before using a combination of SVM and *Hidden Markov Model* (HMM) in the classification stage.

It is noteworthy that video-based approaches are complex as the amount of information provided by the input data is far greater compared to the single image signal. There is also the time parameter which is taken into account and increases complexity. The main challenge consists at reducing this complexity by developing approaches that are both simple to implement, fast and computationally efficient. It is the ultimate goal of the approach we are presenting in the present work.

3 The Proposed Approach

Based on the different issues related to image-based facial expression recognition, we propose a simple approach using image sequences. We designed each one of the different building blocks according to its impact on the whole system. To measure the efficiency of our approach, we have used as input a popular dataset of facial expression videos. We have noticed that most of effective approaches of facial expressions recognition are geometric-based and that is why the first step consists in extracting the facial fiducial points. Those FCPs are exploited for sequence normalization step. Two types of features are extracted. The first one uses Euclidean distance. The second feature type represents the variation of each FCP over the time using a single value to reduce the computation load. For feature selection, we adapted a method proposed by Yaddaden et al. [14] that was initially used for static input. They have used variance to rank the feature vector attributes before selecting the most representative. Finally, for the classification part, we proposed to use two different classifiers depending on the priority: complexity and accuracy (SVM) or simplicity and speed (KNN). Before performing the classification, we normalized the value of the data aiming to increase the RR.

In the Figure 1 are represented the different building blocks of our approach. The process is repeated until the best list of features is found which provides the highest accuracy.

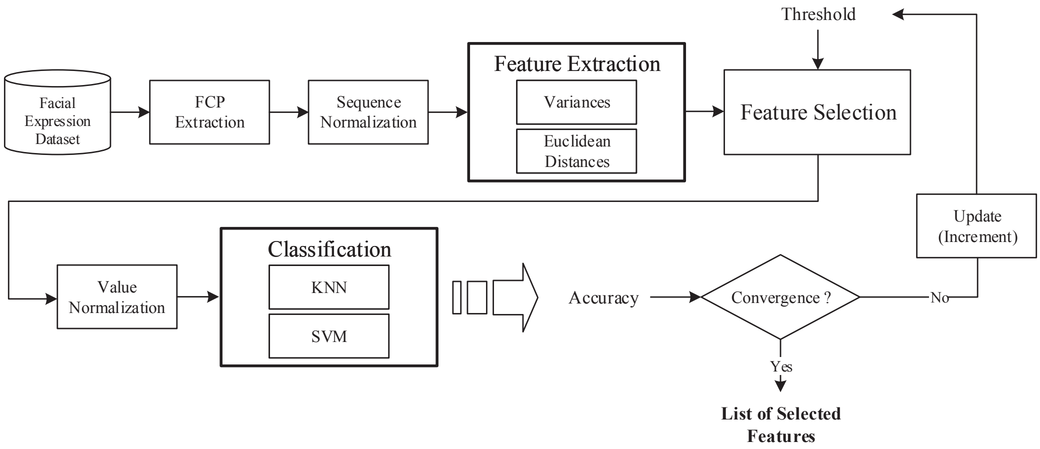

Figure 1. Diagram of the proposed approach.

3.1 Facial Fiducial Point Extraction

For each frame of the input sequence, we extract the different facial fiducial points. To that end, various methods, such as ASM and AAM have been proposed. In our case, we chose to use the relatively new approach proposed by Kazemi and Sullivan [2]. Its speed, accuracy and computational efficiency were our main motivation to choose this technique. The process of FCP extraction using this method is described by the following expression (1):

$$\hat{S}^{t+1} = \hat{S}^{t} + r_t(I, \hat{S}^{t}) \quad (1)$$

Basically, the method uses a predefined estimation of the face shape and the intensities of sparse set of pixels in the neighborhood of the first FCP. The current estimation of the face shape is adjusted, iteratively, until convergence. In the expression (1), $S = \{P_1, P_2, ..., P_N\} \in \mathbb{R}^{2N}$ represents the FCP vector where P_i is the coordinate (x, y) of each facial fiducial point. The adjustment of the face shape is performed by a regression function $r_t()$ which takes as argument the current face shape estimation $\hat{S}^t$ and the input image I. The function $r_t()$ consists in a cascade of decision trees trained using the gradient boosting approach. This operation is repeated until convergence and perfect fit of the face shape. In Figure 2, we can see the extracted facial fiducial points. This operation is also performed for each image of the different sequences.

Figure 2. (a) Original sequence (b) Sequence with FCPs.

3.2 Sequence Normalization

After extracting all the FCPs from the different image sequences, the next step is to normalize the number of frames for each sequence. In fact, based on our experiments, the best performances are achieved when every image sequence contains the same number of frames. To do that, three main steps are performed: 1) define the number of images per sequence, 2) duplicate images in shorter sequences, and finally 3) reduce the number of images in longer sequences. The number of images per sequence is generally set to the average value computed using all sequence lengths of the dataset. In the case of the dataset we used, this number is equal to 18.

While it is relatively simple to duplicate images in shorter sequences, it is trickier to select images to delete in longer sequences without losing useful information. In order to do that, we have used a specific method that we have developed and described by the Algorithm 1. It is run for each long sequence until the defined number of images per sequence is reached. To do that, we begin by computing the current number of images in the targeted sequence S and store it in a variable N_{curr}. If it is greater than the defined threshold T, then we extract all the facial fiducial points from the image I and store them in the array F. After that, a conditional loop performs the following steps. Using F, a new array containing the variances between each two consecutive images of the sequence S is computed and the results are stored in V. Remove the image corresponding to the lowest value of the variance. Update the value of the N_{curr}. Those three steps are repeated until reaching the target number of images per sequence defined before with T.

3.3 Feature Extraction

From each image of a sequence, we extracted the 68 facial fiducial points using Kazemi and Sullivan [2] technique. Then, we compute two different types of geometric-based features: the first type consists in all the possible Euclidean distances which might be extracted from the FCPs. According to the number of extracted FCPs, the number of possible Euclidean

Algorithm 1: Image Reduction

Data: Dataset of images D_{orig} & Threshold $T = 18$
Result: New sequence normalized dataset D_{new}
begin
  foreach *Sequence S in D_{orig}* **do**
    Compute current number of images N_{curr};
    if $N_{curr} > T$ **then**
      foreach *Image I in S* **do**
        Extract facial fiducial points from I;
        Store result in array F;
      end
      repeat
        Based on F compute the variances;
        Store result in array V;
        Order decreasingly V;
        Remove image with lower variance in V;
        Update N_{curr};
      until $N_{curr} == T$;
    end
  end
end

distances equals to $n = 2278$. From Figure 3 we notice that the representation of the sequence feature vector is defined by a concatenation of each image feature vector. The final one has an important size $m \times n = 18 \times 2278 = 41004$ where m is the number of images per sequence. The second type of feature consists in replacing each distance-based vectors of each sequence using a statistical measure as described in Figure 3. It represents the dispersion and variance of data. It is defined by the following expression (2):

$$V_i^2 = \frac{1}{m}\sum_{j=1}^{m}(D_j^i - \bar{D}^i)^2 \qquad (2)$$

D_j^i represents the value of the i^{th} Euclidean distance in the j^{th} face image of the sequence where $i = 1, ..., n = 2278$ and $j = 1, ..., m = 18$. $\bar{D}^i$ is the average value of the i^{th} Euclidean distance over the m images of the sequence. V_i^2 represents the computed variance.

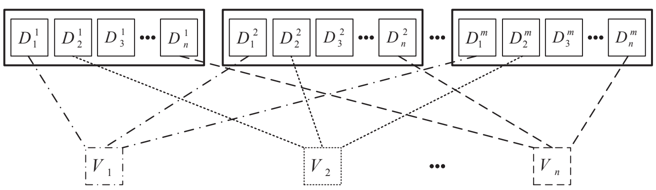

Figure 3. Description of the feature vectors.

We note that the size of the variance-based feature vector is 18 times smaller than the distance-based one. It is an important parameter to take into account if the proposed approach targets implementation in an environment with limited computational resources.

3.4 Feature Selection

From the previous section, we notice that the size for the feature vector is large. Therefore, it increases the computational time and resource consumption. Feature vectors might also contain noisy and redundant values which may reduce the performances. For these reasons, we propose to use a specific feature selection. Yaddaden et al. [14] have proposed a method used on images-based inputs. We have adapted it for sequence-based inputs. Thus, two distinct algorithms have been developed to select distance-based and variance-based features, respectively.

Algorithm 2: Feature Selection for Distances

Data: Dataset of distances D_{orig} & Threshold $T_{att} = 10$
Result: New dataset of distances D_{new}
begin
  Initialize final array of FCP Id F_{id};
  foreach *Emotion group E_{group} in D_{orig}* **do**
    Initialize temporary array of FCP Id F_{tmp};
    foreach *Sequence S in E_{group}* **do**
      Compute the variance V for each FCP in S;
      Update F_{tmp} order according to V values;
    end
    Update F_{id} according to the resultant F_{tmp};
  end
  Apply Threshold T_{att} on the final F_{id};
  Extract distances from D_{orig} according to F_{id};
  Store the distances in D_{new};
end

In the Algorithm 2, we create and initialize an array of FCP F_{id}. Then, for each emotion, we create another array of FCP F_{tmp}. For each sequence S corresponding to a specific emotion, we compute the variance for each FCP and store them in V. After that, we update the order of F_{tmp} according to the values of V. We update F_{id} regarding to F_{tmp}. Finally, we apply the threshold T and extract the corresponding distances according to F_{id}.

In the Algorithm 3, we begin by creating an array of FCP F_{id}. Then, we form a different vector from each emotion V_{emo}. For each FCP Id in F_{id}, we compute the variance V using V_{emo}. The result are stored in V_{id}. The array F_{id} is sorted decreasingly according to V_{id} before applying the threshold T on it. Finally, we extract the corresponding variances from V_{orig} according to F_{id}

For both features, several selections are performed according to $S = \{2\%, 4\%, 6\%, ..., 100\%\}$ where S corresponds to the percentage of held features. The adequate value of S is defined by performing classification until reaching convergence in terms of accuracy.

3.5 Value Normalization & Classification

Classification allows to recognize unlabled facial expressions. However, this operation is sensitive to the data value range.

Algorithm 3: Feature Selection for Variances

Data: Dataset of variances V_{orig} & Threshold $T_{att} = 10$
Result: New dataset of variances V_{new}
begin
  Initialize final array of FCP Id F_{id};
  Form vectors V_{emo} from V_{orig} for each emotion;
  foreach Id *in* F_{id} **do**
    Compute variance V from V_{emo} according to Id;
    Store result in an array V_{id};
  end
  Order decreasingly F_{id} according to value of V_{id};
  Apply Threshold T_{att} on the final F_{id};
  Extract variances from V_{orig} according to F_{id};
  Store the variances in V_{new};
end

Therefore, we propose to apply a value normalization in order to increase the accuracy. The used technique is called Min-Max Normalization that provides a linear transformation changing the range of data to a predefined one. This operation is described by the following expression (3):

$$A_{norm} = \left(\frac{A - A_{min}}{A_{max} - A_{min}} \right) \times (R_{max} - R_{min}) + R_{min} \quad (3)$$

Where A is the current value to normalize and A_{norm} is the resultant value. The actual range of values of the input data is computed and represented by $[A_{min}, A_{max}]$. There are only two parameters to be defined manually and they describe the interval of values $R = [R_{min}, R_{max}]$. In our case, they are $R_{min} = 0, R_{max} = 1$.

For our approach, we propose the use of two *supervised* learning techniques. It implies the presence of two different stages: *training* and *validation*. The first one is KNN and it is chosen for its computational efficiency and simplicity of implementation. As it does not generate a model, KNN is considered as an *instance-based* or *lazy learning* method. There are two important parameters to define: the number of nearest neighbors which is, in our case, equal to $K = 1$ and the second parameter is the distance metric. In our case, the Cosine distance is used as it yields better results compared to the classical Euclidean distance. It is defined by the following expression (4):

$$D_{Cosine} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} y_i^2}} \quad (4)$$

The second classifier we use is SVM as it can handle nonlinear data by performing transformation using a specific function called *kernel*. There are different types of kernels: Gaussian (RBF), Polynomial, Linear, etc. In our case, the best results are achieved using the Linear kernel. SVM performs training by generating the parameters of the maximum-margin hyperplane in order to separate two distinct classes based on a training set. The SVM classifier is considered as a binary classifier but our approach enables to recognize six different emotions. Therefore, we construct a global SVM classifier from six binary ones adopting One-vs-All strategy.

4 Experimentation

In this section, we present the dataset we used for our experimentations, the experimental protocol and finally the obtained results.

In order to validate our approach, we used the Cohn-Kanade Extended (CK+) [5] dataset, a common and widely used video facial expression dataset. It contains 593 sequences with 309 labeled with the six basic emotions: happiness (HA), anger (AN), disgust (DI), sadness (SA), fear (FE) and surprise (SU). The sequences consist of 10 to 60 frames in terms of length. They are recorded from 210 adults ageing between 18 to 50 years. The dataset is also diversified: 69% female, 81% Euro-American, 13% Afro-American and 6% other groups. The images are digitized into either 640×490 or 640×480 pixels resolution with 8-bit gray scale and 24-bit color values.

To verify the efficiency of our approach, we perform two different comparisons. The first one concerns the features we propose to use (distance-based and variance-based) combined with the two classifiers: SVM and KNN. The second evaluation consists in comparing the best obtained RR with existing approaches. We adopt a common validation strategy which is 10-folds cross-validation.

Table 1. RR by emotions with distance-based features

	HA	AN	DI	SA	FE	SU	Avg.
KNN	90%	53%	61%	47%	60%	95%	68%
	94%	91%	69%	67%	77%	97%	**77%**
SVM	97%	91%	93%	77%	85%	96%	90%
	99%	91%	97%	90%	87%	97%	**90%**

Table 1 shows the RR by emotion for the experiment using distance-based features. It can be noticed that the best average value, that consists in an arithmetic mean, is reached when there is no normalization (Second line) for KNN but the values are almost the same for SVM. There is an important difference between the values obtained with KNN and those obtained with SVM ($\approx 13\%$). This might be explained by the fact that KNN cannot handle too many attributes. We also notice that the lowest values are obtained when recognizing Sad (SA) and Fear (FE) emotions and this is due to the reduced number of available sequences related to these emotions in the CK+ dataset.

Table 2. RR by emotions with variance-based features

	HA	AN	DI	SA	FE	SU	Avg.
KNN	97%	73%	95%	68%	62%	99%	**82%**
	96%	68%	92%	45%	53%	96%	75%
SVM	99%	91%	91%	77%	73%	98%	**88%**
	93%	77%	88%	57%	72%	94%	80%

Table 2 shows that the best results are achieved when using normalization (First line). Unlike the first experimention, it can be noticed that the difference between KNN and SVM in terms of RR is lower ($\approx 6\%$).

Table 3. Comparison with existing approaches using CK+ dataset

Input Signal	Feature Type	Method	Classifiers	RR
Image-based	Appearance-based	Ou et al. [8]	KNN + PCA	80.00%
	Geometric-based	Yu et al. [15]	SVM + PCA	75.50%
	Appearance-based	Li et al. [4]	Deep Learning	83.00%
		Peng et al. [9]		66.90%
Sequence-based	Hybrid-based	Lucey et al. [5]	SVM	83.32%
	Geometric-based	Wan et al. [13]		80.00%
		Saeed et al. [10]		83.00%
		Mohammadian et al. [7]	SVM + HMM	83.90%
		Our approach	KNN	**88.34%**
			SVM	**92.54%**

From Table 3, we notice that our approach whether it uses SVM or KNN classifiers outperforms the existing methods. The best RR value with the SVM classifier is reached with distance-based features and equals to 92.54% with 37723 attributes but with variance-based we reached 91.80% with only 1503. With KNN classifier, we have reached 81.6% with 820 attributes when using distance-based features. When using variance-based ones, we reached a better RR of 88.34% with 1594 attributes. Globally, we conclude that the use of variance-based features is more suitable regarding to the RR and feature vector size. There is only a difference of ($\approx 3.46\%$) when using KNN and SVM combined with variance-based features.

5 Conclusion

We presented in this paper an approach to recognize emotions through facial expressions using image sequences. It is based on two feature types: distance-based and variance-based. Although the first feature type is more accurate, it generates bigger feature vectors when used with videos. Therefore, we proposed to represent the temporal variations using variance. The approach offers good generalization performances as it uses SVM classifier. Furthermore, we plan to improve the whole system.

References

[1] Paul Ekman. Are there basic emotions? *Psychol Rev*, 99(3):550–553, July 1992.

[2] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, pages 1867–1874, 2014.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[4] Wei Li, Min Li, Zhong Su, and Zhigang Zhu. A deep-learning approach to facial expression recognition with candid images. In *MVA*, pages 279–282. IEEE, 2015.

[5] 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 *CVPRW*, pages 94–101. IEEE, 2010.

[6] Albert Mehrabian. *Communication without words*, pages 51–52. 2 edition, 1968.

[7] Amin Mohammadian, Hassan Aghaeinia, and Farzad Towhidkhah. Video-based facial expression recognition by removing the style variations. *IET Image Processing*, 9(7):596–603, 2015.

[8] Jun Ou, Xiao-Bo Bai, Yun Pei, Liang Ma, and Wei Liu. Automatic facial expression recognition using gabor filter and expression analysis. In *ICCMS*, volume 2, pages 215–218. IEEE, 2010.

[9] Xianlin Peng, Zhaoqiang Xia, Lei Li, and Xiaoyi Feng. Towards facial expression recognition in the wild: A new database and deep recognition system. In *CVPRW*, pages 93–99, 2016.

[10] Anwar Saeed, Ayoub Al-Hamadi, Robert Niese, and Moftah Elzobi. Frame-based facial expression recognition using geometrical features. *Advances in Human-Computer Interaction*, 2014:4, 2014.

[11] CP Sumathi, T Santhanam, and M Mahadevi. Automatic facial expression analysis a survey. *International Journal of Computer Science and Engineering Survey*, 3(6):47, 2012.

[12] Yingli Tian, Takeo Kanade, and Jeffrey F Cohn. Facial expression recognition. In *Handbook of face recognition*, pages 487–519. Springer, 2011.

[13] Chuan Wan, Yantao Tian, and Shuaishi Liu. Facial expression recognition in video sequences. In *WCICA*, pages 4766–4770. IEEE, 2012.

[14] Yacine Yaddaden, Abdenour Bouzouane, Mehdi Adda, and Bruno Bouchard. A new approach of facial expression recognition for ambient assisted living. In *PETRAE*, page 14. ACM, 2016.

[15] Kaimin Yu, Zhiyong Wang, Markus Hagenbuchner, and David Dagan Feng. Spectral embedding based facial expression recognition with multiple features. *Neurocomputing*, 129:136–145, 2014.