Automatic Facial Expression Recognition Using Statistical-Like Moments

Roberto D'Ambrosio, Giulio Iannello, and Paolo Soda

Integrated Research Center, Università Campus Bio-Medico di Roma,
Via Alvaro del Portillo, 00128 Roma, Italy
{r.dambrosio,g.iannello,p.soda}@unicampus.it

Abstract. Research in automatic facial expression recognition has permitted the development of systems discriminating between the six prototypical expressions, i.e. anger, disgust, fear, happiness, sadness and surprise, in frontal video sequences. Achieving high recognition rate often implies high computational costs that are not compatible with real time applications on limited-resource platforms. In order to have high recognition rate as well as computational efficiency, we propose an automatic facial expression recognition system using a set of novel features inspired by statistical moments. Such descriptors, named as statistical-like moments extract high order statistic from texture descriptors such as local binary patterns. The approach has been successfully tested on the second edition of Cohn-Kanade database, showing a computational advantage and achieving a performance recognition rate comparable than methods based on different descriptors.

1 Introduction

Nowadays reliable and intuitive interfaces are one of the major factor influencing acceptance of technological devices. Indeed some researches aim at developing smart human computer interfaces (HCIs) that even could understand user emotion. One way to achieve this goal consists in recognizing facial expressions, one of the most important human communication modality.

The study of facial expressions, started with Darwin and led to a facial expressions encoding by Ekman and Friesen in the 1971. They described elementary facial movements (AUs) roughly corresponding to movement of facial muscles and represented all possible faced expressions as a combination of such AUs. They also proposed six prototypical expressions that are the same among all ethnicity, i.e. anger, disgust, fear, happiness, sadness and surprise.

Facial expression recognition has long been competence of medical doctors and anthropologists but, the necessity of intuitive HCIs stimulate researches on automatic facial expression recognition (AFER) in computer vision and pattern recognition areas.

Researches efforts have been finalized to find the best descriptors and classifiers to recognize facial expressions in images and videos. In this area Gabor

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part I, LNCS 6978, pp. 585–594, 2011.

energy filters (GEFs) and local binary patterns (LBPs) provide the best performance. Since the development of real time AFER systems on resource limited devices, i.e. mobile devices, requires that features extraction, selection and classification are characterized by low computation costs, a trade-off between computation costs and recognition rate is an issue still open.

This work presents an AFER system classifying facial expressions into six prototypical expressions. We define a set of features which effectively describes facial expression in frontal view video sequences. They are based on statistical-like moments computed on LBP transformed images. These descriptors reduce the amount of data conveying expressive information and the classification time.

The paper is organized as follow: next section presents an overview of most significant works on AFER. Section 3 discusses system architecture, whereas section 4 present the experimental protocol. I section 5 we discuss the results, and section 6 provides concluding remarks.

2 Background

Reviews of researches carried out until 2009 presented the fundamental approaches for AFER systems development, pointing out research targets and limits [1,2,3]. First systems were inadequate to face real problems due to the exiguous amount of images and videos available to train the classifiers. More recently, the availability of larger databases of facial spontaneous expressions has permitted to develop AFER systems which may be applied to real world situations.

We present now a review of most important AFER systems providing best performance among existing literature. As described below, such systems are based on different methods for both features extraction and selection as well as sample classification.

In [4] the authors present an automatic system which detects faces, normalizes facial images and recognizes 20 facial actions. In the recognition step, features are the outputs of a bank of Gabor energy filters and classifiers are SVM or an Adaboost algorithm.

In [5] different features are tested, i.e. GEFs, box filter (BF), hedge orientation histogram (EOH), BF+EOH and LBPs. In order to test the performance achieved with such descriptors, GentelBoost and SVM are employed as classification methods. Results show that GEFs provides good performance with SVM.

In [6] the authors present a framework for facial expression recognition based on encoded dynamic features. In the first step the system detects faces in each frame, normalize them and computes Haar-like features. Dynamic features units are calculated as a set of Haar-like features located at the same position along a temporal window. The second step consists in a coding phase which analyses each dynamic features distribution generating the corresponding codebook. Such a codebook is used to map a dynamic features into a binary pattern. The last step applies a classifier based on the AdaBoost algorithm.

The system presented in [7] carries out a preprocessing step in which images alignment is manually performed to realign the common regions of the face by the identification of mouth and eyes's coordinates. Then, images are scaled and the first frame of each input sequence is subtracted from the following frames to analyse the changes in facial expression over time. Such an image is named as *delta image*. The system extracts local features from delta images applying Independent component analysis (ICA) over the PCs space. Then, the features classes are separated by fisher linear discriminant analysis. Finally, each local feature is compared with a vector belonging to a codebook generated by a vector quantization algorithm from features of training set samples.

In [8] the authors present a comprehensive study for AFER applications investigating LBP-based features for low resolution facial expression recognition. Furthermore, they propose Boosted-LBP as the most discriminative LBP histograms selected by AdaBoost, reducing the number of features processed in the classification stage.

In [9] the authors use LBP-based features computed using spatio-temporal information, which are named as LBP-TOP. Boosted Haar features are used for automatic coarse face detection and 2D Cascade AdaBoost is applied for localizing eyes in detected faces. The positions of the two eyes, determined in the first frame of each sequence, define the facial area used for further analysis in the whole video sequence. Authors try to achieve real-time performance proposing multi-resolution features, computed from different sized blocks, different neighbouring samplings and different sampling scales of LBPs. AdaBoost is used for features selection, whose training is performed either by one-against-one or all-against-all classes strategies.

In [10] the authors combine the strengths of two-dimensional principal component analysis (2DPCA) and LBP operators for feature extraction. Before LBP computation, authors apply a filter for edge detection aiming at lowering the sensitivity to noise or changes in light conditions of LBP operators, although such operators have proven their robustness to those issue [11]. Finally, the concatenated features are passed as input to a decision direct acyclic graph based multi-class SVM classifier.

In [12] the authors test LBP-based features, histogram of oriented gradient and scale invariant feature transform to characterize facial expressions over 5 yaw rotations angle from frontal to profile views.

In [13] the prototypical expressions are classified at 5 different poses using multi-scale LBP and local gabor binary patterns (LGBP). LGBPs utilize multi-resolution spatial histogram combined with local intensity distributions and spatial information. A SVM is used as classifier. Experiments suggest, that facial expressions recognition is largely consistent across all poses, but the optimal view is dependent on the data and features used.

This review shows that considerable effort has been devoted to find the best combinations of preprocessing steps, descriptors, and classifier to achieve the task of discriminate between the six prototypical expressions. Nevertheless the issue is still open, especially if high recognition performance must be obtained respecting real time and computational constraints.

3 System Architecture

This section first presents an overview of the proposed system, and then focuses on the feature vectors we tested.

The system can be divided into four blocks performing: face and eye detection, image normalization, features extraction, and sample classification (Fig.1).

In the first module each frame of the input video sequence is processed by Haar-like features cascades to find a region of interest containing the face. Then the frame is cropped obtaining the facial image. Next, employing appropriate Haar-cascades, we locate eyes and compute morphological information, such as the distance between the eyes and the degree of head rotation. Since subjects often close their eyes while are showing the facial expressions, an eye tracking algorithm is used to estimate current eyes positions using information on eyes position and speed in previous frames.

In the normalization module the facial image is converted into grey-scale and its histogram is equalized to make the system more robust to light variation. To reduce the computational complexity each facial image is rescaled to 96x96 pixels maintaining a distance between the eyes of 48 pixels. Furthermore to improve system's performance the rescaled facial images are aligned on the basis of eyes position in each frame.

The third module processes facial image with LBP operators and extracts several descriptors, such as histograms and statistical-like moments, used to built the feature vector of the sample (in subsection 3.1).

The fourth module classifies the samples as discussed in section 4.

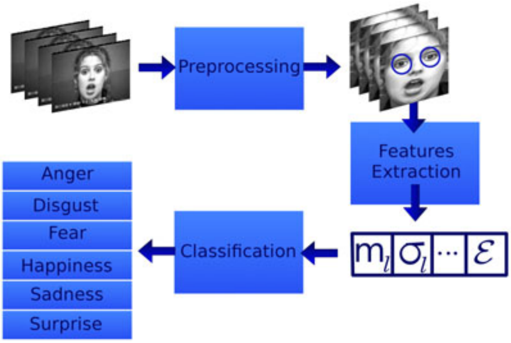

Fig. 1. System architecture

3.1 Feature Vectors

The following two paragraphs describe features commonly used in the literature, whereas the third paragraph presents the novel set of features applied to this work.

I_{LBP} Histograms. LBP operators are grey-scale invariant statistical primitives showing good performance in texture classification [11]. Defining g_c the grey value of the central pixel of the neighbourhood and g_p the grey value of the equally spaced pixels on a circle of radius R (with $p = 1, 2, ..., P-1$), LBP operators are defined as:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{1}$$

where $s(g_p - g_c)$ is 1 if $g_p - g_c \geq 0$ and 0 otherwise. In our work we use $P = 8$ and $R = 1$. In the following we denoted as I_{LBP} the LBP transformed image computed applying the LBP operators to a grey scale image I.

Features vectors from I_{LBP} are generally obtained collecting several histograms $\mathbf{H}^{\mathbf{w_i}}_{\mathbf{LBP}}$, each one computed on a region w_i of I_{LBP}. Dividing I_{LBP} into n regions, the corresponding feature vector of I_{LBP} is given by:

$$\mathbf{F}_{LBP} = \{\mathbf{H}^{\mathbf{w_i}}_{\mathbf{LBP}}\}_{i=1}^{n} \tag{2}$$

Since, $\mathbf{H}^{\mathbf{w_i}}_{\mathbf{LBP}}$ is composed of 256 bins [11], $\mathbf{F}_{LBP}$ is composed of $(256 \times n)$ elements.

$I_{LBP^{u2}}$ Histograms. A more compact way to describe texture information of an image is to use LBP^{u2} operators. Indeed, information, on about 90%, of the vast majority of all 3x3 patterns obtained when $R = 1$ is captured by a limited number of patterns. $LBPs^{u2}$ work as templates for microstructures such as bright and dark spots, flat areas, and edges. Formally, they are defined by a uniformity measure U corresponding to the number of pattern spatial transitions i.e. bitwise 0/1 changes in binary code. Considering only patterns having U value of at most 2, the following descriptors may be introduced:

$$LBP^{u2}_{P,R} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & if \quad U(LBP_{P,R}) \leq 2 \\ P+1 & otherwise \end{cases} \tag{3}$$

LBP^{u2} can be effectively described by histograms with 59 bins [11]. Hence, the corresponding feature vector is composed of $(59 \times n)$ elements:

$$\mathbf{F}_{LBP^{u2}} = \{\mathbf{H}^{\mathbf{w_i}}_{\mathbf{LBP^{u2}}}\}_{i=1}^{n} \tag{4}$$

Statistical-like moments. A larger feature vector requires larger time to extract features, to train the classifier and to label test samples. For these reasons, we propose to represent LBP transformed images using very few descriptors. Given I_{LBP} histograms, we may extract statistical descriptors since they are able to catch texture information [14]. In this respect, we consider the mean (m), the variance (σ^2), the skewness (ς), the kurtosis (χ), and the energy (ϵ).

Note that skewness and kurtosis are defined only if $\sigma^2 \neq 0$.

As reported above, features may be extracted not only from the whole I_{LBP}, but also from subregions w_i. In this case, we may find regions with a uniform pixel value, where $\sigma^2 = 0$ and, hence, skewness and kurtosis cannot be computed. In order to overcome this limitation, we introduce in the following a new set of descriptors, named as *statistical-like moments*, inspired by statistical descriptors. To this aim, let us introduce the following notations:

- w is a subregion of I_{LBP};
- n_k is the number of pixels having grey value k in I_{LBP};
- $n_k(w)$ is the number of pixels having grey value k in w;
- S is the number of pixels in I_{LBP};
- $S(w)$ is the number of pixels in w.
- $\mu_n(w)$ is the nth statistical moments of w.

We first introduce the following descriptor named as *local mean* and denoted by $m_l(w)$:

$$m_l(w) \triangleq \sum_{k=0}^{L-1} r_k \frac{n_k(w)}{S} \tag{5}$$

where r denote a discrete variable representing pixel value in the range $[0, L-1]$. Similarly, we define the nth *local moments* as:

$$\mu_{l_n}(w) \triangleq \sum_{k=0}^{L-1} (r_k - m_l(w))^n \frac{n_k(w)}{S} \quad n = 0, 1, \dots \tag{6}$$

The following relationship between our *local moments* and the conventional statistical moments computed in w holds:

$$\mu_{l_n}(w) = \sum_{i=0}^{n} \binom{n}{i} \mu_{n-i}(w)(m(w) - m_l(w))^{n-i} \phi(w) \quad if n > 0 \tag{7}$$

where $\mu_n(w)$ is the nth conventional statistical moment and $m(w)$ is the conventional mean, both in subregion w, and $\phi(w) = S(w)/S$.

Using such quantities we define two further descriptors inspired to skewness and kurtosis and respectively denoted as $\varsigma_l(w)$ and $\chi_l(w)$. They are given by:

$$\varsigma_l(w) = \frac{\mu_{l_3}(w)}{\sigma_l^3(w)}, \qquad \chi_l(w) = \frac{\mu_{l_4}(w)}{\sigma_l^4(w)} \tag{8}$$

where $\sigma_l(w)$ is:

$$\sqrt{\sum_{k=0}^{L-1} (r_k - m_l(w))^2 \frac{n_k(w)}{S}} = \sqrt{\mu_{l_2}(w)} \tag{9}$$

Since it can be easily shown that $\sigma_l(w)$ is different from zero providing that pixels in region w are not all 0, $\varsigma_l(w)$ and $\chi_l(w)$ are always defined.

Using previous descriptors, the features vector is built computing from each subregion the following quantities:

$$\mathbf{f}_l^{w_i} = \{m_l(w_i), \sigma_l(w_i), \varsigma_l(w_i), \chi_l(w_i), \epsilon\} \quad i = 1, \ldots, n \quad \forall w \in I \tag{10}$$

and collecting them into the final vector which is composed of $(5 \times n)$ elements only:

$$\mathbf{F}_l = \{\mathbf{f}_l^{w_i}\}_{i=1}^{n} \tag{11}$$

4 Experimental Protocol

For our experiment we used the second edition of Cohn-Kanade database (CK+) [15] arranging the sequences in two different sets.

One set is composed of 282 video sequences corresponding to 876 images of 95 different subjects containing prototype expressions. According to a general practice reported in the literature, we select the last three frame of each video sequence corresponding to the peak of expression. In the following this set is referred to $\mathbf{D}_p$.

The second set is composed of 4409 images obtained selecting in the 282 video sequences all the frames except the first three ones. In the follow we referred to this set as $\mathbf{D}_s$.

While $\mathbf{D}_p$ permits to test the capability of our system to detect expressions at the maximum of the expressions intensity, $\mathbf{D}_s$ permits to test our system in a scenario near to the real situation where people rarely show expression of maximum intensity.

As already mentioned, features extraction approaches based on both $\mathbf{F}_{LBP}$ and $\mathbf{F}_{LBP^{u2}}$ as well as the proposed statistical-like moments $\mathbf{F}_l$ divide facial image into regions. In the literature, a common approach divides the image into n non-overlapping regions [8,9,10]. Hence, we extract $\mathbf{F}_{LBP}$, $\mathbf{F}_{LBP^{u2}}$ and $\mathbf{F}_l$ dividing facial image into $9, 16, 25, 36, 49$ and 64 non overlapping squared regions. An alternative method may use a shifting window providing partially overlapped subregions. Although this approach cannot be applied to $\mathbf{F}_{LBP}$ and $\mathbf{F}_{LBP^{u2}}$ since their long feature vectors would greatly increase the computational load, the reduced size of our $\mathbf{F}_l$ descriptors enable us to test the shifting window approach. In this case tested windows sizes are 12, 14,16, 20, 24 and 32 pixels. In the rest of paper, the computation of $\mathbf{F}_l$ using a shifting window is denoted as $\mathbf{F}_{l_w}$.

All feature vectors are given to a SVM classifier with an RBF kernel, performing 10-fold cross validation and averaging out the results. We performed a grid search on SVM parameter space, where C ranges from 1 up to 10^4, and γ ranges from 10^{-1} up to 10^{-5}.

Image processing and features extraction stages have been implemented in C++, while classifier training and testing have been performed using Weka-3.0 [16]. Experiments have ran on Hp xw8600 workstation with 8-core $3.00\,GHz$ Intel(R) Xeon(R) CPU and $8.00\,GB$ RAM.

Table 1. Recognition rate (%) measured considering only peak of expression ($\mathbf{D}_p$) and the whole video sequence ($\mathbf{D}_s$). $\mathbf{F}_{LBP}$, $\mathbf{F}_{LBP^{u2}}$, $\mathbf{F}_{stat}$ and $\mathbf{F}_l$ were computed on non overlapping regions. $\mathbf{F}_{l_w}$ was computed on shifting window.

	$\mathbf{D}_p$						$\mathbf{D}_s$					
Features	Side in pixels of squared regions						Side in pixels of squared regions					
	32	24	20	16	14	12	32	24	20	16	14	12
$\mathbf{F}_{LBP}$	98.2	99.1	99.1	99.3	99.1	99.1	95.5	96.2	96.6	96.7	96.6	97.3
$\mathbf{F}_{LBP^{u2}}$	97.9	99.1	99.1	99.4	99.1	99.2	95.4	96.1	96.5	97.0	97.3	97.3
$\mathbf{F}_l$	97.5	97.7	97.1	97.3	97.9	96.0	95.1	96.8	96.8	97.0	96.6	96.5
$\mathbf{F}_{l_w}$	98.7	99.1	99.2	98.9	98.7	94.5	97.7	98.2	98.0	97.5	97.7	96.6

5 Results and Discussion

Table 1 shows the recognition rates achieved using different features on both $\mathbf{D}_p$, and $\mathbf{D}_s$, whereas table 2 shows elapsed time to train and test the classifier on each sample of $\mathbf{D}_p$ (times concerning $\mathbf{D}_s$ convey equivalent information and are not reported for the sake of brevity). As reported above $\mathbf{F}_{LBP}$, $\mathbf{F}_{LBP^{u2}}$ and $\mathbf{F}_l$ are computed from squared, non-overlapping subregions whereas $\mathbf{F}_{l_w}$ are computed using a shifting window. To provide a deeper insight into recognition results, table 3 reports the confusion matrices corresponding to the best and the worst accuracy of $\mathbf{F}_{l_w}$ on $\mathbf{D}_p$. Such best and worst results are achieved using the shifting window with side of 20px and 12px wich are reported in each tabular as the first and second entry, respectively. Other confusion matrices are omitted for brevity.

Turning our attention to results achieved on $\mathbf{D}_p$ (i.e. expression recognition in peak condition) we notice that $\mathbf{F}_l$ descriptors require lower training and test times than LBP-based descriptors (Table 2), whereas their performance is only slightly penalized (Table 1). In particular, focusing on test times, which are the most relevant in real time scenarios, $\mathbf{F}_l$ is 15 and 10 times faster than $\mathbf{F}_{LBP}$, $\mathbf{F}_{LBP^{u2}}$, respectively. Nevertheless, if shifting windows are used, i.e. $\mathbf{F}_{l_w}$, a better trade-off is reached. With respect of LBP-based descriptors, on the one hand, performance are completely comparable and, on the other hand, training and test times are still remarkably low. Indeed, $\mathbf{F}_{l_w}$ is 4 and 3 times faster than $\mathbf{F}_{LBP}$ and $\mathbf{F}_{LBP^{u2}}$, respectively.

Turning our attention on expressions recognition on $\mathbf{D}_s$ (i.e. the whole video sequences) we observe that $\mathbf{F}_l$ performance are comparable with those achieved using $\mathbf{F}_{LBP}$ and $\mathbf{F}_{LBP^{u2}}$. Furthermore, best performance among all tested configurations are achieved by $\mathbf{F}_{l_w}$ descriptors, attaining a recognition rate equal to 99.2%.

This analysis reveals that the introduction of more compact descriptors increases the flexibility in choosing the trade-off between classification performance and computational costs. In particular, our statistical-like moments computed using a shifting window remarkably reduce the computation time and improve classification performance when expressions vary in intensity during a video sequences. This observation suggests that our descriptors may be a good solution for real problems where it is likely to be difficult to isolate peak expression frames.

Table 2. Training and testing time of different descriptors on $\mathbf{D}_p$

Features	Time	Side of squared regions					
		32	24	20	16	14	12
$\mathbf{F}_{LBP}$	train(s)	9.2	12.6	13.6	18.1	18.7	22.6
	test(ms)	11.0	16.9	17.2	23.8	24.4	29.2
$\mathbf{F}_{LBP^{u2}}$	train(s)	4.5	9.1	11.7	12.4	15.1	17.4
	test(ms)	6.1	9.3	14.8	15.3	18.7	21.9
$\mathbf{F}_{l}$	train(s)	0.2	0.6	1.4	2.4	3.6	3.4
	test(ms)	0.4	1.7	2.1	1.4	1.4	1.9
$\mathbf{F}_{l_w}$	train(s)	2.1	3.6	3.6	5.6	6.8	14.5
	test(ms)	3.5	2.1	3.4	7.3	8.8	12.2

Table 3. Confusion matrix corresponding to the best and the worst accuracy of $\mathbf{F}_{l_w}$ on $\mathbf{D}_p$ reported in each tabular as the first and second value, respectively

	Anger	Disgust	Fear	Happiness	Sadness	Surprise
Anger	125-114	0-0	0-0	0-0	0-0	1-12
Disgust	0-1	161-147	1-0	0-0	0-0	0-14
Fear	1-0	0 - 0	65-60	0-2	0-0	0-4
Happiness	0-0	0-0	1-0	197-188	0-0	0-10
Sadness	0-0	2-1	0-0	0-0	76-73	0-4
Surprise	1-0	0-0	0-0	0-0	0-0	245-246

6 Conclusions

In this paper we have presented an automatic facial expression recognition system able to conveying facial expression information using a compact set of descriptors based on statistical properties of LBP transformed images histograms. The small number of elements composing the resulting feature vectors permits us to significantly reduce the classification times making our system suited to real time applications on resource-limited platforms, such as mobile devices. Performance of our descriptors has been compared with that achieved by LBP-based descriptors in their basic configuration. Future works will be directed towards the use of features selection algorithms, such as Adaboost. This will permit us to compare performance of our selected feature vectors with performance achieved by boosted-descriptors known in literature and employed in person-independent system.

Acknowledgements. This work has been (partially) carried out in the framework of the ITINERIS2 project, Codice CUP F87G10000050009, under the financial support of Regione Lazio (Programme "Sviluppo dell'Innovazione Tecnologica nel Territorio Regionale, Art. 182, comma 4, lettera c), L.R. n. 4, 28 Aprile 2006).

References

1. Fasel, B., Luettin, J.: Automatic facial expression analysis: a survey. Pattern Recognition 36(1), 259–275 (2003)
2. Pantic, M., Rothkrantz, L.J.M.: Automatic analysis of facial expression: the state of art. IEEE Trans. on Pattern Analysis and Machine Intelligence (2000)
3. Liu, S.S., Tian, Y.T., Li, D.: New research advances of facial expression recognition. In: Proc. of the Eighth Int. Conf. on Machine Learning and Cybernetics (2009)
4. Bartlett, M.S., Littlewort, G.C., Frank, M.G., Lainscsek, C., Fasel, I.R., Movelland, R.: Automatic recognition of facial actions in spontaneous expressions. Journal of Multimedia (2006)
5. Whitehill, J., Littlewort, G., Fasel, I., Bartlett, M., Movellan, J.: Toward practical smile detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 31, 2106–2111 (2009)
6. Yang, P., Liu, Q., Metaxas, D.N.: Boosting encoded dynamic features for facial expression recognition. Pattern Recognition Letters 30(2), 132–139 (2009), Video-based Object and Event Analysis
7. Uddin, M. Z., Lee, J.J., Kim, T.S.: An enhanced indipendent component-based human facial expression recognition from video. IEEE Trans. on Consumer Electronics (2009)
8. Shan, C., Gong, S., McOwan, P.W.: Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing 27(6), 803–816 (2009)
9. Zhao, G., Pietikinen, M.: Boosted multi-resolution spatiotemporal descriptors for facial expression recognition. Pattern Recognition Letters 30(12), 1117–1127 (2009), Image/video-based Pattern Analysis and HCI Applications
10. Lin, D.-T., Pan, D.-C.: Integrating a mixed-feature model and multiclass support vector machine for facial expression recognition. Integr. Comput.-Aided Eng. 16, 61–74 (2009)
11. Ojala, T., Pietikäinen, M.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
12. Hu, Y., Zeng, Z., Yin, L., Wei, X., Zhou, X., Huang, T.S.: Multi-View Facial Expression Recognition. In: 8th Int. Conf. on Automatic Face and Gesture Recognition (2008)
13. Moore, S., Bowden, R.: Local binary patterns for multi-view facial expression recognition. Computer Vision and Image Understanding 115(4), 541–558 (2011)
14. Gonazalez, C., Woods, E.: Digital Image Processing. Prentice-Hall, Englewood Cliffs (2001)
15. Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: Proc. of the Fourth IEEE Int. Conf. on Automatic Face and Gesture Recognition, pp. 46–53 (2000)
16. Hall, M., Frank, E.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)