WEIGHTED VOTING OF SPARSE REPRESENTATION CLASSIFIERS FOR FACIAL EXPRESSION RECOGNITION

Shane F. Cotter

ECE Department, Union College
Schenectady, NY 12308, U.S.A.
Ph: +1-518-388-6274 Fax: +1-518-388-6789
Email: cotters@union.edu

ABSTRACT

We present a new algorithm for facial expression recognition that is robust to occlusion. The facial image is divided into equal sized regions, and a Sparse Representation Classifier (SRC) classifies the facial expression in each region. These classification decisions must be combined and different voting methods were considered. A weighted voting method where the vote assigned to each class in a region was based on the class representation error led to the best recognition results under a variety of occlusion conditions. The recognition rate of our algorithm remains very high for unoccluded images (95.3% success). With large occluded regions ($\geq$25% of the image), it significantly outperforms an SRC algorithm based on the entire image and a Gabor-based algorithm. Since each subimage problem can be solved independently before combining decisions, processing can be done in parallel leading to a fast SRC based classification decision if implemented on a multi-core system.

1. INTRODUCTION

Facial Expression Recognition (FER) has long been a topic of interest in psychology where correctly identifying a person's emotions can be used in a clinical setting. The Facial Action Coding System (FACS) has been used extensively by psychologists. This system details a method for the recognition of expressions by a human observer and is based on identifying the presence or absence of 46 elementary muscle movements, called action units (AUs) [1]. Based on the set of AUs present, the facial expression is categorized as fear, anger, disgust, sadness, surprise, happiness or neutral.

Automated methods for recognizing facial expressions have been explored over the last several decades and reviews can be found in [2, 3]. Human computer interaction (HCI) represents an important application and more natural interaction will require accurate expression recognition systems. While many methods use the entire facial image to identify the facial expression, recent successful methods have applied techniques to extract localized features from the face image for use in classification. In particular, Gabor filter banks (which are used to approximately model the processing in the primary visual cortex) have been successfully used as a feature extraction method [4, 5, 6]. Other methods such as Independent Component Analysis (ICA) [7] and Localized Non-negative Matrix Factorization (LNMF) [8], which have been successfully applied to FER, have also extracted highly localized features. These methods more closely parallel the identification of localized changes used in FACS for expression identification.

Very frequently, the FER problem is complicated by occlusion of part of the facial image which can be caused by glasses, headwear, facial hair, or hands for example. Under these conditions, recognition methods which use local facial information have an advantage over holistic features. Features extracted from occluded regions will be lost, but the features extracted from unoccluded regions are not affected and may be sufficient to allow accurate classification of the facial expression. The classification decision is most commonly obtained using a nearest neighbor algorithm or support vector machine [9].

Recently, we have proposed the application of a Sparse Recognition Classifier (SRC) [10] to the recognition of facial expressions. We used the entire facial image and showed that excellent recognition performance was obtained when the system was presented with occluded or noisy images [11]. In the SRC method [10], a sparse representation of an unknown test image is formed using training samples whose classes are known. The pixel values (rather than any extracted features) are used to represent each image and the sparse representation is obtained using l_1-norm minimization. The representation is then used *directly* in the classification of the images, i.e, there is no need for a secondary classification stage.

In this paper, we propose splitting the image into a number of smaller subimages which can be classified independently by separate Sparse Representation Classifiers. The expression recognition is therefore based on localized features which, as discussed above, have been shown to improve upon results obtained using whole image features, especially when part of the image is occluded. When a classification decision is obtained from each of the local regions, these decisions must be combined to produce a final decision. We considered a number of different methods of combining the decisions from the local classifiers and found that a weighted voting scheme that uses the representation error for each of the C classes $\{\rho_j\}_{j=1}^{C}$ in each local region gave the best results. This method is termed ρ-Weighted Voting SRC. For large occlusions ($\geq$25% of the image), we show through simulations that this new method significantly outperforms the SRC method based on the entire image and a Gabor-based method. Furthermore, each subimage classification problem is of much smaller size and can be solved more quickly and in parallel. This algorithm could be run on a multi-core processor [12] leading to a much faster decision than an SRC method based on the entire image.

The outline of the paper is as follows. In section 2, we briefly describe the Sparse Representation based Classi-

This work was partially supported by National Science Foundation Award No. 0837458

© EURASIP, 2010 ISSN 2076-1465

fication (SRC) method. In section 3, we detail the separation of the images into local regions and the different voting strategies we used in combining the decisions obtained from the SRC in each local region. Simulations were used to determine the performance of our new method and its performance was compared to SRC on the entire image and a Gabor-based method in section 4. We also examined the difference in computation time between SRC based on the entire image and subimages. We draw some conclusions from our work in section 5.

2. SPARSE REPRESENTATION BASED CLASSIFICATION (SRC)

We consider representing the signal $y \in \mathrm{R}^m$ using vectors from a dictionary $A = \{a_1, a_2, \cdots, a_n\}$ where each $a_k \in \mathrm{R}^m, k = 1, \cdots, n$. A sparse representation of y using elements from A is one which uses very few vectors from A. Using $x \in \mathrm{R}^n$ to denote the weights on each of the dictionary vectors, the sparsest solution is obtained by solving

$$\min \|x\|_0 \text{ s.t. } Ax = y \tag{1}$$

where the l_0 norm counts the number of nonzero entries in x. Obtaining the solution to this problem is NP-hard, but recent work in the area of compressed sensing [13] has shown that if the solution is sufficiently sparse, the solution to (1) can be obtained *exactly* by solving the l_1 norm problem

$$\min \|x\|_1 \text{ s.t. } Ax = y. \tag{2}$$

This problem is convex and can be solved in polynomial time using a variety of methods. The application of sparse representation to classification has been proposed in [10], and we briefly summarize this method in the following paragraphs.

In a classification problem, known samples of the different classes are available and are used to give the training vectors. Each sample can be represented by the raw data (e.g., image or speech data) or a feature vector (e.g., by projecting on principal or independent components derived from the data). The unknown test samples are represented in the same manner as the training samples, and the class of the test sample is derived by finding a representation of the test sample as a linear combination of the available training samples.

Assuming that there are C classes and $n_k, k = 1, \cdots, C$ examples from each class, the training vectors for each class are denoted by $T^k = \{t_j^k\}, j = 1, \cdots, n_k$. Forming a dictionary of training vectors from all available examples as $A = [T_1, T_2, \cdots, T_C]$, and denoting a test vector as $y \in \mathrm{R}^m$, some similarity measure must be used to determine which class of training vectors most closely matches the test vector; this class is then chosen as the unknown test class. The key insight from [10] is that a test sample from a class i should be efficiently representable as a linear combination solely of the training vectors from the class T_i. Hence, given the matrix A of all training vectors, the representation of a vector y from class $i \in \{1, \cdots, C\}$ should ideally produce a solution vector of the form:

$$x_s = [0, \cdots, 0, x_1^i, x_2^i, \cdots, x_{n_i}^i, 0, \cdots, 0]^T \in \mathrm{R}^n. \tag{3}$$

Since the solution is formed by using a small number of training vectors from the large training set, the solution is sparse and can be obtained by solving (2).

In reality, due to noise in the data and correlation between elements from different classes, the solution obtained from (2) will not consist solely of vectors from a single class, i.e., some elements not associated with class i will have a nonzero value in (3). The support of y using the different classes $j = 1, \cdots, C$ in A is given by

$$\widehat{y^j} = T_j x_s^j, \;\; x_s^j = [x_1^j, x_2^j, \cdots, x_{n_j}^j]^T \tag{4}$$

where x_s^j is obtained from the solution of (2) as the weights on the training vectors associated with class j. The classification decision is obtained from the minimum representation error $\rho_j, j = 1, \cdots, C$ as

$$c^* = \arg\min_j \rho_j = \arg\min_j \|y - \widehat{y^j}\|_2. \tag{5}$$

As discussed in section 1, occlusion of part of an image makes the classification problem more difficult, and the framework outlined above needs to be modified to account for this. An augmented matrix is formed as

$$B = [A \;\; F] \tag{6}$$

where the vectors in F are used to represent the corruption or occlusion. As noted in [10], if the images used to form the dictionary A and test vector y are represented by the raw pixel values then the error due to corruption or occlusion can be represented by the identity matrix I. If the degree of corruption is not excessively large, the problem can be solved by using the l_1 norm as outlined above and using the matrix B instead of A:

$$\min \|z\|_1 \;\; \text{s.t. } Bz = y, \text{ where } z = [x \;\; e]^T. \tag{7}$$

When the solution vector z_s is obtained, it is separated into its component parts x_s and e_s. Since the corruption has been isolated to the components e_s, this is subtracted from the test vector in the classification stage and (5) is modified to

$$c^* = \arg\min_j \rho_j = \arg\min_j \|(y - Fe_s) - \widehat{y^j}\|_2. \tag{8}$$

with $\widehat{y^j}$ given by (4).

3. WEIGHTED VOTING ALGORITHM

Using the notation of section 2, we consider the classification of the facial expression of an unknown test image $y \in \mathrm{R}^m$ using training images (with known facial expression) $A = \{a_1, a_2, \cdots, a_n\} \in \mathrm{R}^{m \times n}$. Localized subimages are extracted from the test image giving vectors $y_p, \; p = 1, \cdots, P$ and training images are processed to give training dictionaries $A_p, p = 1, \cdots, P$. An example image and its separation into subimages is shown in figure 3(a). Each vector is of length m/P, which is assumed an integer (in practice, this will always be the case). The augmented dictionary $B_p = [A_p \;\; I_{\{\frac{m}{P} \times \frac{m}{P}\}}]$ is formed to account for occlusion. For each of these local classification problems, the SRC method classifies the unknown test subimage as belonging to one of the C classes. Using the original formulation of the SRC method (given in section 2), the class decision in each region is made based on the class which yields the lowest representation error as given in (8). These independent decisions must

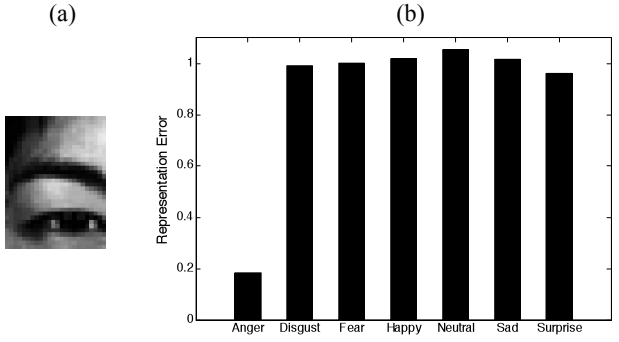

Figure 1: (a) Left eye subimage from figure 3(a) and (b) representation error based on the dictionary training vectors available for each of the classes.

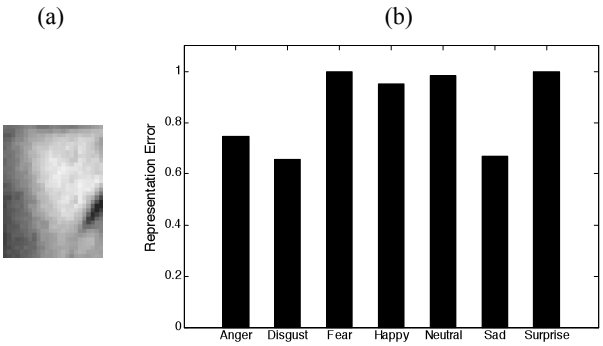

Figure 2: (a) Cheek subimage from figure 3(a) and (b) representation error based on the dictionary training vectors available for each of the classes.

be combined to produce the final classification decision. The problem of combining decisions from different classifiers has received a lot of attention and a summary of different methods can be found in [14].

A simple method of combining the decisions from the different regions is to use a voting scheme. If each classifier has one vote, it votes for the class with the lowest representation error and all other classes receive no vote. The final decision is made by a decision center which sums up the number of votes received by each class and classifies the image as belonging to the class that receives the most votes. This method corresponds to a plurality decision rule and we term this Plurality Voting SRC [14].

We would expect the plurality voting strategy to work well in informative regions with little or no occlusion where the correct class is easily identified and gets the one vote available while all other classes receive no vote. An example is shown in figure 1 where the subimage includes part of the left eye and the representation error obtained for the correct class (Anger) is much less than the other classes. However, in less informative regions, the difference in representation errors between classes is fairly small. Figure 2 shows an example based on the left cheek subimage where the representation error for the 3 top choices is very similar. In this case, the correct class (Anger) does not have the lowest representation error and gets no vote while an incorrect class receives a vote. Similarly, when part of a subimage is occluded, it becomes difficult for any class to represent the presented image and the resulting representation error distribution becomes more similar to that shown in figure 2 than in figure 1. This led us to consider using a weighted voting strategy in combining the classification results from the local classifiers.

A simple modification of the plurality vote is the Borda count method where each class is assigned a weight based on the number of classes which are ranked below it by the classifier [15]. If we consider a subimage y_p, the class which yields the lowest representation error gets a weight of $(C-1)$, the next lowest representation error gets $(C-2)$, etc. For each class j, $j = 1,\cdots,C$, we denote by $N_j^{(k)}$ the number of times over all the subimages $p = 1,\cdots,P$ that the class is ranked in kth position. The total vote obtained by class j, can be stated as

$$N_j = \lambda_1 N_j^{(1)} + \lambda_2 N_j^{(2)} + \cdots + \lambda_C N_j^{(C)} \tag{9}$$

where $\lambda_k, k = 1,\cdots,C$ are the weights assigned to the different rank positions. In the case of the Borda count, the weights are given as $\lambda_k = (C-k)$. The class which gets the greatest number of votes is taken as the output of the classifier which we refer to as the Borda Count Voting SRC:

$$\hat{c}_{Borda} = \arg \max_{j\in\{1,\cdots,C\}} N_j. \tag{10}$$

The Borda count is simple to implement but does not take into account the differences between classifiers [15]. Each subimage classifier is treated equally although it is well known that some parts of the face are more informative for facial expression recognition [1]. In addition, within a region the relative difference in the representation error between different classes is not reflected in the vote assigned to each class, i.e., there is a difference of 1 in the vote which each class receives irrespective of how closely the class is represented. To rectify this, we propose weighting each vote from the different subregions $p = 1,\cdots,P$ by using the representation errors $\rho_j^{(p)}$, $j = 1,\cdots,C$ (following the notation in (5)) that are obtained for each class using the SRC. For a subimage p, we assign a weighted vote to a class j as $1/\rho_j^{(p)}$. The final class decision, which we refer to as ρ-Weighted Voting SRC in the simulation section is obtained as

$$\hat{c}_\rho = \arg \max_{j\in\{1,\cdots,C\}} \sum_{p=1}^{P} 1/\rho_j^{(p)}. \tag{11}$$

We show through simulations that this voting method results in much better performance with large occlusions than using the other voting methods.

4. SIMULATIONS

4.1 Database and Image Preprocessing

We used the JAFFE female expression database in our experiments. There are 213 facial expressions from 10 different people and each person posed for 3 or 4 examples of each of the six basic expressions plus a neutral expression [6]. To account for the slightly different head tilts and head sizes, preprocessing is used to eliminate these differences. Our processing is similar to that used in [5]: each image was rotated so that the eyes were horizontally aligned, and the interocular distance was used to crop out a rectangular area from the original image. The image was resized to 96×72 pixels and histogram equalization was used to increase the local contrast in some areas of the image. The images were divided into 9 subimages as shown in figure 3(a) and each subimage is of size 32×24 pixels.

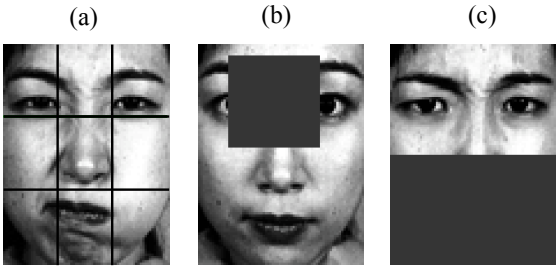

Figure 3: (a) Example subimages derived from facial image, (b) block occlusion of size 40×40 placed randomly, and (c) occlusion of lower half of the facial image.

4.2 Feature Extraction and Classification

Each of the local image regions extracted was converted into a vector by stacking the pixels column by column to give the test images $y_p,\ p = 1, \cdots, P$ and training dictionaries $A_p,\ p = 1, \cdots, P$. The decisions obtained from the SRC method in each local region were combined using the different voting methods we have outlined in section 3.

We compared our results to those obtained using SRC on the entire image [11] and to a Gabor-based method [6]. We include the Gabor-based method as these features have been shown to be robust to block occlusions [16]. 40 Gabor filters given by the following expression were applied to each image to form Gabor jets at each point in the image [6]:

$$\begin{aligned} \psi(x,y,\omega,\theta) &= \frac{1}{2\pi\sigma^2} e^{-(\frac{x'^2+y'^2}{2\sigma^2})} [e^{i\omega x'} - e^{-\frac{\omega^2\sigma^2}{2}}] \qquad (12) \\ \text{where } x' &= x\cos\theta + y\sin\theta \ ; \ y' = -x\sin\theta + y\cos\theta. \end{aligned}$$

(x,y) is the pixel position in the spatial domain, ω the radial center frequency, θ the orientation of Gabor filter, and σ is the standard deviation of the Gaussian function. The parameters were set as $\sigma = \pi/\omega$, with 5 values of ω chosen as $\omega_p = \frac{\pi}{2}(\frac{1}{\sqrt{2}})^{(p-1)}, p = \{1, \cdots, 5\}$, and 8 values of θ given by $\theta_q = (q-1)\pi/8, q = 1, \cdots, 8$. The absolute values of the outputs from all filters were concatenated to form a large vector and this was downsampled by a factor of 64. The classification was done using a simple Nearest Neighbor (NN) method [9].

The solution to (2) which leads to the SRC solution was obtained using the CVX package [17]. Since the dataset is relatively small, recognition results were obtained using a leave one out strategy where in each trial one image from the database is excluded and the remaining images are used in training. The results from all these recognition experiments were averaged to give the recognition rate.

4.3 Occlusion Results

In general, any part of the facial image may be occluded. We simulate this situation by placing a square block of size $W \times W$ pixels randomly over the image. If this block falls outside the more informative regions of the face (the eyes or mouth region) then we expect the occlusion to have less impact on the recognition of the facial expression. However, as the size of the block increases, the eyes or mouth will be partially or fully occluded. For example, a square block of side $W = 40$ is large enough to fully occlude the entire eye or mouth but its position may dictate that it partially occludes more than one feature; an example facial image occluded with a block of side $W = 40$ is shown in figure 3(b).

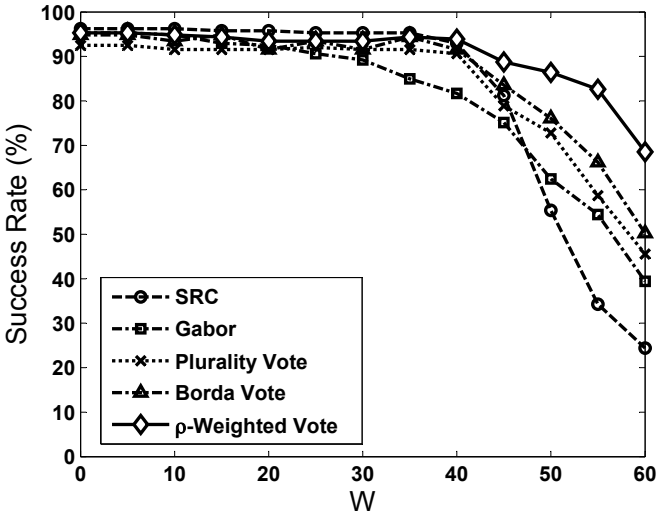

Figure 4: Comparison of recognition performance obtained using SRC Voting Methods (Plurality, Borda Count, and ρ-Weighted) to a Gabor-based method and an SRC method based on the entire image.

The size of the square block W was increased in steps of 5 and the recognition results obtained using each of the different voting methods described in section 3 were compared to the Gabor based features and the SRC method applied to the entire image. The results are plotted in figure 4.

All methods perform very well when there is no occlusion and the success rate of all algorithms is over 94% except for the Plurality Vote which gives a success rate of 92.5%. Over the range $W = 10$ to $W = 35$, the SRC method applied to the entire image performs the best and its performance is slightly better by 1.5% than the ρ-Weighted Voting SRC method at $W = 25$. The success rate of the Gabor-based method drops off over the range $W = 20$ to $W = 40$ and its performance is lower than all of the SRC based methods.

When the occlusion is greater than a size 40×40 block (which comprises $\approx 23\%$ of the image), the success rate of the SRC method based on the entire image drops off quickly and falls below that of the Gabor-based method. The Borda Count Voting SRC method does better than the Plurality Voting SRC method and they both improve on the performance obtained using the Gabor-based method. The ρ-Weighted Voting SRC method emerges as the clear winner when the occlusion size is increased to occupy more of the image, i.e., $W \geq 40$. Even when the occlusion size is increased to $W = 60$ (corresponding to an occlusion of $\approx 52\%$ of the image), the success rate is 68% and much higher than any other method. To illustrate the different performance obtained by the algorithms in a situation where a particular part of each image is occluded, we considered the occlusion of the lower half of the face (as shown in figure 3(c)) and of the upper half of the face. The results obtained using each algorithm are given in table 1. The ρ-Weighted Voting SRC algorithm greatly outperforms all other algorithms on this set of occluded images. As we would expect, the performance is higher when the upper half is unoccluded since the eye regions are more informative than the mouth region. The parts of the image which are occluded do not impact the decision much while the eye regions dominate the voting (as we have illustrated in figures 1 and 2) and this leads to the higher recognition rate

Occlusion	SRC	Gabor+NN	SRC Voting Methods		
			Plurality	Borda	ρ-Weighted
Upper Half of Face	26.8	31.9	24.4	28.2	63.8
Lower Half of Face	41.3	46.5	20.7	29.6	77.0

Table 1: Comparison of recognition performance of SRC Voting Methods (Plurality, Borda Count, and ρ-Weighted) to a Gabor-based method and an SRC method based on the entire image when half of the presented image is fully occluded.

SRC Method	Variables	Constraints	Time Mean(Std. Dev.)
Full Image	14248	6912	343.2 ($\pm$24.7)
Subimage	1960	768	40.6 ($\pm$3.9)

Table 2: Dimensions and computation time (mean and standard deviation) of the l_1-norm problem solved by CVX [17].

obtained using the ρ-Weighted Voting SRC algorithm.

4.4 Computational Considerations

An important consideration outside the performance of an algorithm is how quickly a solution can be generated and its memory requirements. We ran our experiments on a low-end PC (Intel Pentium 4, 3.2 GHz with 3 GB RAM). The original SRC method uses all the image pixels which results in a complex problem when running the l_1-norm optimization while the subimage problem is much more manageable. Using the CVX package [17], the optimization problem to be solved has a number of variables determined by the formation of the matrix $[B \quad -B]$, where B is obtained from (6). The number of constraints is derived from the number of pixels. The full image and subimage problem dimensions are summarized in table 2 along with the time taken to run each method. Indeed, the size of the problem using the entire image requires the use of 64-bit MATLAB running on a 64-bit version of Windows while the subimage problem can be run on a machine which has 32 bit versions of MATLAB and Windows. The ability to use multiple processors, which is an emerging trend in DSP applications [12], would allow the different classification problems in the ρ-Weighted Voting SRC method to be run in parallel. This would lead to a large reduction in the time required to solve the classification problem in comparison to the SRC method using the entire image.

5. CONCLUSION

We have introduced a Sparse Representation Classifier (SRC) algorithm based on subimages extracted from a facial image to recognize facial expressions. The decisions from the different image regions must be combined. We experimented with a number of methods and introduced a voting method where the vote assigned to each class in each region was based on the class representation error obtained. We termed this algorithm the ρ-Weighted Voting SRC method. The recognition rate of our algorithm is very high for unoccluded images giving a recognition rate of 95.3%, and the method performs very well as the occluded region is made larger. We compared our results to an SRC algorithm based on the entire image as well as a Gabor-based algorithm. The performance of our algorithm is particularly impressive when the occlusion size is increased to occlude more than 25% of the image. For example, with a square block occlusion of size 50×50, the recognition rate obtained using ρ-Weighted Voting is 86.4% compared to 62.4% using a Gabor-based method and 55.4% using an SRC method based on all the image pixels. Each subimage problem is independent and can be solved in parallel before the decisions are combined. This subimage based SRC method can therefore classify an image much faster than an SRC method using the entire image if implemented on a multi-core system.

REFERENCES

[1] P. Ekman and W. Friesen, *Facial action coding system (FACS): Manual*, Consulting Psychologists Press, Palo Alto, 1978

[2] M. Pantic and L. Rothkrantz, "Automatic analysis of facial expression: the state of the art", *IEEE Trans. PAMI*, pp. 1424–1445, Dec. 2000

[3] B. Fasel and J. Luttin, "Automatic facial expression analysis: a survey", *Patt. Recog.*, vol. 36, pp. 259–275, 2003

[4] M. Lyons et al., "Automatic classification of single facial images", *IEEE Trans. PAMI*, pp. 1357–1362, Dec. 1999

[5] H. Deng et al., "A new facial expression recognition method based on local Gabor filter bank and PCA plus LDA", *Intl. Jrnl. Info. Tech.*, pp. 86–96, Nov. 2005

[6] M. Lyons et al., "Coding facial expressions with Gabor wavelets", *Proc. AFGSR*, pp. 200–205, Apr. 1998

[7] G.L. Donato et al., "Classifying facial actions", *IEEE Trans. PAMI*, pp. 974-989, Oct. 1999

[8] I. Buciu and I. Pitas. "Application of non-negative and local non-negative matrix factorization to facial expression recognition". *Proc. ICPR*, pp. 288-291, 2004

[9] R. Duda, P. Hart and D. Stork, *Pattern Classification*, 2nd Edition, John Wiley, New York, 2001

[10] J. Wright et al., "Robust face recognition via sparse representation", *IEEE Trans. PAMI*, pp. 210–227, Feb. 2009

[11] S.F. Cotter, "Sparse representation for accurate classification of corrupted and occluded facial expressions", *Proc. ICASSP*, pp. 838-841, Mar. 2010

[12] Y. Chen et al. (Ed), "Signal processing on platforms with multiple cores", *IEEE Sig. Proc. Mag.*, Nov. 2009

[13] E. Candes and M. Wakin, "An introduction to compressive sampling", *IEEE Sig. Proc. Mag.*, pp. 21–30, Mar. 2008

[14] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley, July 2004

[15] T. Ho et al., "Decision combination in multiple classifier systems", *IEEE Trans. PAMI*, pp. 66–75, Jan. 1994

[16] I. Buciu et al., "Facial expression recognition under partial occlusion", *Proc. ICASSP*, pp. 453–456, Mar. 2005

[17] M. Grant and S. Boyd. "CVX: MATLAB software for disciplined convex programming (web page and software)", http://stanford.edu/ boyd/cvx, June 2009