Recognizing Facial Expressions in Videos

Lin Su, Matthew Balazsi

Abstract

Facial expression recognition plays a significant role in human-computer interaction systems, but remains an unsolved problem. In this paper, two competing approaches to detect seven basic facial expressions in videos are compared. One approach used Principle Component Analysis to project the training and testing faces onto the eigenspace, and categorized the testing faces using K nearest neighbor according to Euclidean distance. The other approach extracted Local Gabor Binary Patterns as feature descriptor, and fed the features into Support Vector Machines for categorization. Finally, a data-smoothing algorithm was applied to remove outliers from the predicted results.

1. Introduction

As a strong non-verbal clue to people's emotion states, facial expression recognition enjoys high popularity among researchers. Defined by P. Ekman [1], the seven basic facial expressions are: anger, disgust, fear, sad, happy, surprise, and neutral. Facial expression recognition has long been studied, since it acts as an important component of human-computer interaction systems. For instance, automatic recognition of human emotion states enables the robot to adjust its behavior according to the user's reactions; the video surveillance system in the hospital could set an alarm when it detects abnormal emotion states of the patient; facial expression could also cooperate with micro-expression recognition to conduct lie detection task.

The facial expression recognition task consists of three steps: face detection, facial feature extraction, and facial expression interpretation [2]. A myriad of related studies have been carried out to investigate each step. A good face detector should work well even with occlusion, non-ideal illumination, large head pose, and low resolution. The face detection approach based on integral features by Viola and Jones [3] is the most widely used method in recent years. Variety of facial features has been used for representing faces. Some researchers modeled the geometric structure of the face using fiducial points, which is regarded as geometric features, while others used appearance features, which represents the texture information of the face. Among various appearance features, Local Binary Patterns (LBP) was popular due to its simplicity and computational efficiency [4]. Another popular appearance representation approach is Gabor wavelets. Though it is both time and memory intensive to convolve the images with a filter bank, the performance of Gabor wavelets is impressively good [5]. Moreover, a combination of LBP and Gabor is applied to facial expression recognition recently, which is known as the Local Gabor Binary Patterns (LGBP). Recent studies [6-8] all demonstrated the expressive ability of LGBP when representing the disparity of facial expressions. Principle Component Analysis (PCA) [9] is a holistic feature, which represents the changes spanning the whole face. In terms of facial expression classification, different machine learning techniques have been utilized: Support Vector Machine (SVM) [6-8], AdaBoost [10], Neural Networks and [11] are but some examples. When detecting facial expressions in videos, *Shin et al.* [12] used optical flow to track facial movements and Hidden Markov Model (HMM) to conduct classification. In [13], a feature point tracking approach is compared with dense flow method, and SVM is used as the classifier.

In our study, we will explore how to detect the seven facial expression using two classification methods: the first more naïve approach based on PCA, and the second based on LGBP supported by SVM.

2. Algorithm

2.1. Overview

Our implementation will perform facial expression recognition on an input video. On a frame per frame basis, the algorithm will rescale the image to 320x240 pixels then detect a single face using the Viola Jones [3] framework. With the segmented face, rescaled to 150x150 pixels, we apply two competing methods for expression classification. The first is based on PCA [9], where all training and queried faces are projected onto the principal components' space. Euclidian distance measures the proximity of the queried face to the training faces. The second approach creates a (LGBP) [14] vector for all training and queried images. An SVM [15] classifier is used for categorization.

After the algorithm processes the entire video, it outputs an array of expression labels associated with each frame. This array may contain outliers; it is therefore smoothed for optimal results. The overview can be seen in Figure 1.

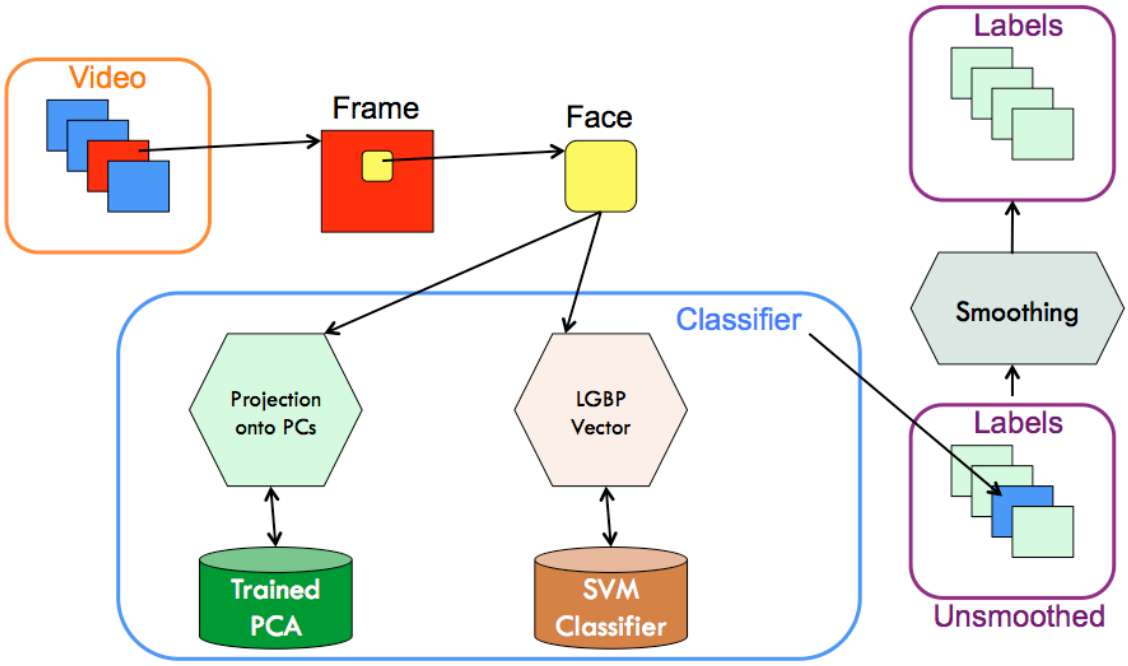


Figure 1: Algorithm Overview

2.2. Face Detection

The first step of our implementation is to segment the face in a given frame. For this, we used a method proposed by Viola and Jones [16]. At increasing scales, the algorithm iteratively detects features in the image, and determines if these can be classified as a face.

Features are computable quantities equal to the sum of pixels in white areas minus the sum of pixels in the gray areas, as seen in Figure 4 [16]. Rectangular areas, or windows, are quickly computed by integral images (Figure 3) [16], and combined in different configurations to constitute a feature (Figure 2) [16].

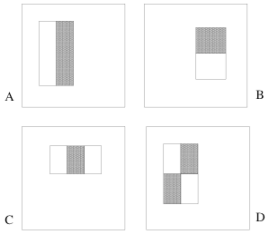


Figure 4: Features in four different configurations

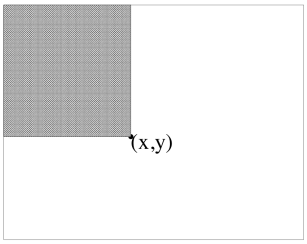


Figure 3: Example of an integral image

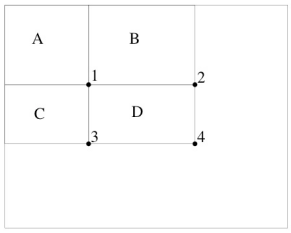


Figure 2: The sum within D can be computed by the integral images at 4 + 1 - (2 + 3)

Despite the efficiency in which we can find features, there are thousands of features associated with each sub window. Prioritizing two configurations (Figure 6) [16] will greatly increase performance. For this optimization, a variation of the Adaboost [17] was used for feature selection and classifier training.

Using the Adaboost approach, a cascade of classifiers is trained for increasingly complex classification. This will quickly reject irrelevant sub-windows before calling upon more complex classifiers.

A Matlab toolbox of the Viola-Jones Face Detector exists, including a cascade trained for faces. It was written by D Kroon from the University of Twente in 2010. Given an image, it will return several possibilities of face areas (Figure 5). The algorithm iteratively decreases the scale by a factor of 1.2, which we modified to 1.15 thus allowing for a more precise search. For our purpose, we systematically selected the smallest possible area returned by the algorithm. To normalize the data, all segmented faces were then rescaled to 150x150 pixels.

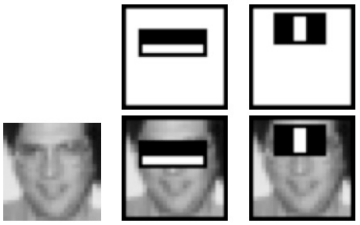


Figure 6: Two features most useful for face detection

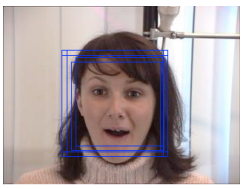


Figure 5: Example of the Viola Jones Algorithm for Matlab

2.3. Principal Components Analysis

Propsed by Pearson in 1901[18], principal components analysis (PCA) projects multidimensional and correlated data onto a linearly uncorrelated data space, called the principal components (PC). The first PC accounts for the largest variance in the data, subsequently decreasing. Projecting all data onto a subset of principal components will reduce the dimensionality of the set, yet may still account for most of the data variance[9].

Given a dataset zero mean dataset X, the principal components are the eigenvectors of its covariance matrix, (Equation 1). For datasets where the dimension is much larger than the number of datapoints, *C* can be large. A simpler solution is to compute the eigenvector of *C'* (Equation 2) which shares the same

eigenvectors as C. The PCs of X can be found by dividing the eigenvectors of C' by their respective eigenvalues.

$$C = XX^T \quad (1)$$
$$C' = X^T X \quad (2)$$

Representing the data through a subset of PCs will reduce dimensionality of the problem. The number of PCs used will depend on the data presented and the objectives of the implementation, trading data complexity to accuracy.

Sirovich and Kirby applied PCA to the face recognition problem [19]. Every pixel represents a dimension of the data and every image a data point. Training images constitute the dataset X, for which the principal components are determined. New queries of images are normalized and projected onto the principal component space. In this way, every face is represented as a vector, and Euclidian distances may be calculated to determine the similarity between two faces. Experimentally dependent, 20 PCs usually suffice for reasonable face detection.

For our purpose, we created the principal component space using 87 images per emotion. New queries were resized to 150x150, normalized to the mean found in training, and projected onto a subset of the PCs. Experimentally, we also found that 20 PCs represented the data reasonably well (Section 4.1). A new query will be compared to the trained data, returning the closest matches for each emotion. Since we are trying to couple the query with an emotion and not an individual, the K top matches for every emotion will be summed for a similarity measure. Experimentally, we determined that $K=1$ yields the most accurate results.

2.4. Local Gabor Binary Patterns

First proposed by *Zhang et al*. [14], Local Gabor Binary Patterns (LGBP) was used as a novel face representation approach. LGBP models each image using a histogram sequence by concatenating the histograms of the local Gabor magnitude binary pattern maps. By combining Gabor and Local Binary Patterns (LBP), LGBP has excellent representation power of the spatial information of the face. The overall framework of LGBP is shown in Figure 7.

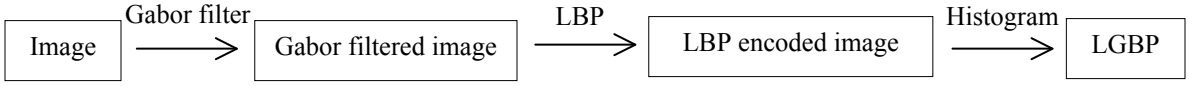


Figure 7: Framework of Local Gabor Binary Patterns

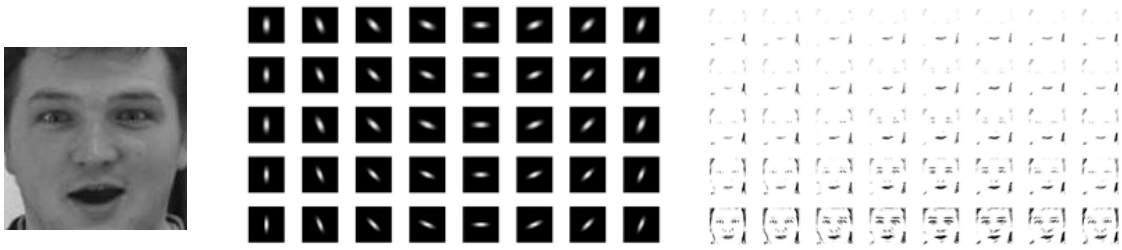
Gabor filter is used for detecting edges, and it has been widely used in computer vision, largely because it's very similar to human visual system. The 2D Gabor filter can be obtained by modulating a Gaussian kernel with a sinusoidal plane wave. The mathematical form of Gabor filter is represented in (Equation 3).

$$gabor\_filter = e^{-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}} e^{i(2\pi\frac{x'}{\lambda}+\varphi)} \quad (3)$$

where $x' = x\cos\theta + y\sin\theta, y' = -x\sin\theta + y\cos\theta$

where λ is the wavelength of the sinusoidal factor, θ is the orientation of the Gabor filter, σ is the Gaussian wavelength, φ is the phase offset, and γ is the spatial aspect ratio.

In this paper, a filter bank of 40 Gabor filters is convolved with each face image. The parameters are: $\lambda = 8, \varphi = \frac{\pi}{2}, \gamma = 1, \theta = \left\{0, \frac{\pi}{8}, \frac{\pi}{4}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{3\pi}{4}, \frac{7\pi}{8}\right\}, \sigma = \{2.5, 2.25, 2, 1.75, 1.5\}$. Figure 8 shows the convolution results of an example image with the filter bank.



(a) (b) (c)

Figure 8: Gabor filters. (a) An example image. (b) A filter bank of Gabor filters, 5 scales, 8 orientations. (c) The Gabor filtered images after convolving (a) with (b).

Afterwards, a Local Binary Pattern operator is applied to the multi-resolution and multi-orientation Gabor filtered images. LBP [20, 21] encodes each pixel value by applying a threshold to the eight neighbors with the center value, and the resulted binary number is transformed into a decimal code. The process is illustrated in Figure 3.

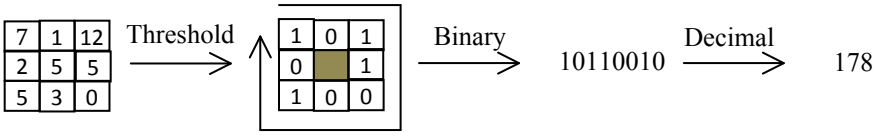
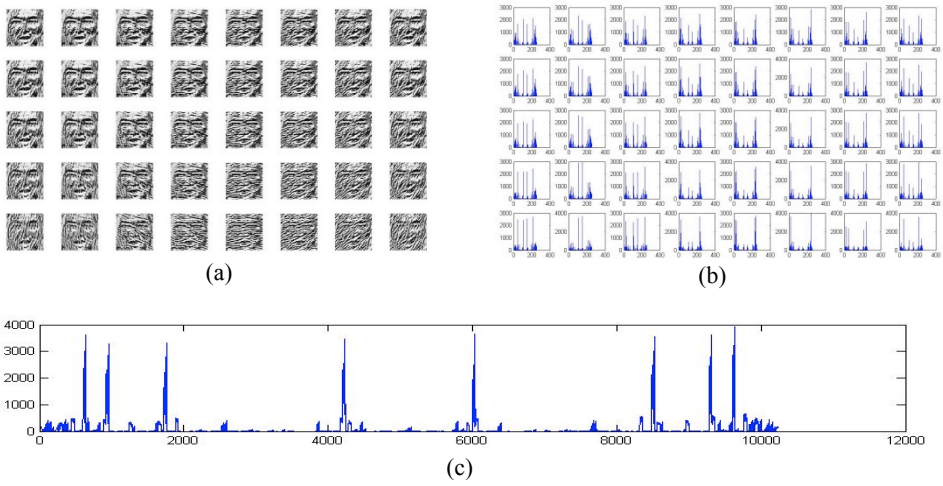


Figure 9: Process of Local Binary Patterns encoding.

Each of the LBP encoded images gives rise to a histogram, which is shown in Figure 4, and all the histograms are concatenated into a single histogram, resulting in a LGBP feature vector (Figure 10).



(a) (b) (c)

Figure 10: (a) LBP encoded images. (b) Histograms of images in (a). (c) LGBP feature vector, which is a concatenation of histograms in (b).

After extracting the LGBP features, Support Vector Machine (SVM) [15] is used as the classifier to conduct facial expression classification. SVM serves as a powerful classifier in face recognition, facial expression recognition and other computer vision areas, mainly due to its ability to deal with high dimensional data and high generalization performance. Since there are seven classes, one-against-all classifiers are trained using a Gaussian kernel with the optimal σ and box constraint chosen by grid search.

2.5. Data Smoothing

The prediction results obtained from either PCA or SVM could possibly contain some outliers. The predicted labels for a video sometimes change rapidly. To solve this problem, a smoothing algorithm is proposed. For the i^{th} label in the video, if its neighbors within $[i-N, i+N]$ are unique, the i^{th} label is set to be neutral expression; otherwise it is set to be the one occurs most frequently in its neighborhood. Through this process, many outliers are smoothed out. Detailed results will be presented in section 4.

3. Methods

3.1. Database

There are many databases available for facial expression recognition research. In earlier studies, subsets of some well-known face recognition databases, such as CMU-PIE [22], FERET [23], BioID [24], are selected and used for facial expression study. However, the subsets are usually small, containing only three or four different facial expressions, and they have no official ground truth. Nowadays a variety of facial expression databases have been built. One of the benchmark datasets is the Japanese Female Facial Expression (JAFFE) [25]. Though JAFFE is the most widely used database, it suffers problems such as small size and lack of illumination changes. C-K+ [26], which is a subset of Cohn-Kanade AU-Coded Facial Expression Database, has a large number of video sequences with facial expression fully FACS coded, and can be used for not only action units analysis but phototypic emotions recognition. MMI facial expression database [26] includes both videos and images, part of which are frame-by-frame AU-coded while others are manually labeled as one of the six basic emotions. Acted Facial Expressions in the Wild (AFEW) [27] collected facial expressions from the movies under unconstrained conditions, which is aimed to mimic the real world environments. Some studies took experiments on 3D facial expression databases as well, such as BU-3DFE [28].

In this paper, the Facial Expressions and Emotions Database (FEED) [29] is used. There are 399 webcam-style videos recorded from 18 subjects with a 100fps camera, and they are labeled into seven facial expressions. The video format is MPG with a resolution of 480*640, while the image format is JPG with a resolution of 240*320. From each video, the most representative frames are selected, as shown in Table 1.

Facial expression	Anger	Disgust	Fear	Happy	Neutral	Sadness	Surprise
Number of frames	127	160	87	185	493	275	322

Table 1: Number of frames selected from FEED

To normalize the number of images per emotion were used for training, we randomly selected 87 images per emotion from the set mentioned in Table 1. Of the 87, a third were not used for training but for immediately testing the performance of our classifiers.

3.2. Testing

3.2.1. Training Set. Whether initializing the PC space or the SVM classifier, a subset of images are not used for training, but reserved for testing. A confusion matrix is formed; where the rows are the expected expression and the columns the classifier's output. With this, the overall accuracy is determined and emotions prone to misinterpretations can be easily visualized.

3.2.2. Video Testing. Three videos per emotion were selected from the FEED video set for further testing, for a total of 21. A single evaluator determined the ground truth for all videos, selecting frames where expressions transitioned.

Frames of the video were sequentially classified, yielding an array of expression labels. Smoothed and raw labels were compared to the ground truth, noting the confusion matrix associated with the video. Frames with errors, such as undetected faces were given the label '0'. After processing all training videos, the confusion matrices were averaged to yield the overall performance. This procedure was repeated for both classification methods.

3.2.3. New Individuals. Five videos were selected from the MMI database for additional testing. Ground truth was created by a single evaluator, and compared against the algorithms' output.

4. Results

4.1. Principal Components Analysis

4.1.1. Training. The first step of establishing the PCA classifier is to determine parameters such as the number of principal components to consider (# PCs) and the number of training images a new query is compared against (*K*). Accuracy is most significantly affected with at least 10 PCs (69%) then stabilizes afterwards (Figure 11). Since we do not restrain the solution for time, we chose to evaluate 20 PCs. Selecting a *K* was trivial because performance linearly decreases with its' value (Figure 12). For this reason, we set *K* to 1.

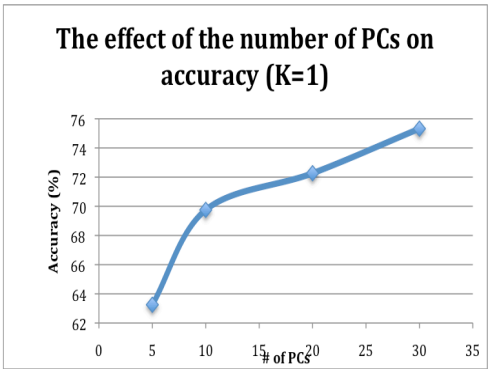


Figure 11: The effect of the number of PCs on accuracy (*K*=1)

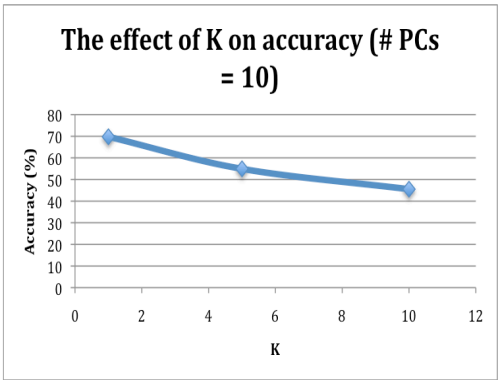


Figure 12: The effect of *K* on accuracy (#PCs = 10)

Our final implementation used 20 principal components and queried 1 image per emotion (*K*=1). The overall success rate was of 72.26% when training. The confusion matrix indicates the most successfully recognized emotions were anger and fear, both with 84.14% accuracy. The emotion with highest failure rates was surprise, at 59.31% success.

4.1.2. Video Testing. The confusion matrix was obtained from all 21 videos. The overall accuracy was 38.09% success when not smoothed, and 40.68% when smoothed. The rate at which errors occurred during the face detection throughout all videos was of 6.29%. With data smoothing, this number was reduced to 0.15%.

For all 21 videos, a timeline was generated comparing the ground truth, unsmoothed, and smoothed output. For brevity's sake, we will only depict the results for a single video named *anger_008_1* (Figure 13). All results can be viewed in the appendix. In this case, we can clearly see some outliers being corrected when smoothed. We can also notice the data smoothing propagating error at frame 10, where the unsmoothed data predicted the correct label, but was smoothed away due to errors in the neighborhood (Figure 13).

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)
Anger	**84.14**	6.90	3.10	0.34	1.03	4.14	0.34
Disgust	3.10	**77.24**	2.07	3.45	1.72	11.38	1.03
Fear	2.41	2.07	**84.14**	0.69	1.72	4.14	4.83
Happy	2.07	10.69	2.07	**70.34**	7.93	3.45	3.45
Neutral	3.45	2.07	3.45	5.17	**69.31**	11.38	5.17
Sadness	5.86	5.86	6.55	1.38	14.83	**61.38**	4.14
Surprise	3.45	2.07	13.10	2.07	10.00	10.00	**59.31**

Table 2 : Confusion matrix for the training phase (PCA)

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**44.01**	7.96	4.04	10.23	14.01	5.52	3.60	10.63
Disgust	29.63	**16.05**	0.93	7.41	9.57	11.42	1.85	23.15
Fear	1.66	0.00	**36.93**	17.84	19.50	5.81	18.26	0.00
Happy	0.00	23.13	0.00	**23.57**	21.82	13.07	7.48	10.93
Neutral	0.32	1.98	8.08	14.96	**41.22**	15.73	13.32	4.37
Sadness	23.27	17.42	6.18	5.07	0.00	**47.20**	0.87	0.00
Surprise	0.00	0.00	51.11	5.56	0.00	0.56	**42.78**	0.00

Table 3: Confusion matrix for all 21 videos tested using PCA, data unsmoothed

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**48.05**	18.59	3.63	11.17	12.39	2.42	3.20	0.54
Disgust	31.79	**16.67**	0.00	7.72	33.33	8.33	2.16	0.00
Fear	1.66	0.00	**42.32**	17.43	17.84	1.66	19.09	0.00
Happy	0.00	23.69	0.00	**25.89**	18.98	25.69	4.60	1.15
Neutral	0.38	0.94	7.23	15.78	**44.28**	19.11	12.28	0.00
Sadness	25.76	12.91	5.81	5.07	0.00	**50.20**	0.25	0.00
Surprise	0.00	0.00	53.89	5.56	0.00	0.00	**40.56**	0.00

Table 4: Confusion matrix for all 21 videos using PCA, data smoothed

---

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)
Anger	**89.27**	5.42	2.19	0.00	1.05	1.67	0.40
Disgust	1.39	**91.57**	0.36	2.49	0.37	2.08	1.74
Fear	0.36	0.96	**91.16**	1.02	1.04	0.65	4.82
Happy	0.00	0.32	0.77	**90.49**	3.63	0.60	4.19
Neutral	1.88	0.64	4.22	2.60	**75.91**	8.50	6.26
Sadness	4.11	5.48	6.64	1.07	12.58	**62.40**	7.72
Surprise	1.01	1.11	7.58	3.07	8.88	2.93	**75.42**

Table 5: Confusion matrix for the training phase (LGBP + SVM)

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**39.15**	1.25	1.25	19.76	0.83	12.08	13.61	12.07
Disgust	0.00	**11.56**	1.88	45.00	0.63	9.37	0.00	31.56
Fear	1.70	0.00	**5.96**	71.49	4.26	11.49	5.11	0.00
Happy	0.00	3.34	0.57	**75.11**	0.00	2.83	7.90	10.24
Neutral	7.56	0.11	1.21	37.44	**19.02**	14.83	17.42	2.42
Sadness	1.55	0.00	1.12	3.00	0.94	**87.71**	5.50	0.19
Surprise	7.95	0.00	0.00	32.95	4.55	21.59	**32.95**	0.00

Table 6: Confusion matrix for all 21 videos using LGBP + SVM, data unsmoothed

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**56.64**	0.00	0.00	16.45	0.00	11.24	15.67	0.00
Disgust	0.00	**14.38**	3.44	42.81	0.00	19.38	19.38	0.63
Fear	0.00	0.00	**5.53**	72.34	3.40	14.89	3.83	0.00
Happy	0.00	2.79	0.00	**92.11**	0.00	0.57	3.97	0.57
Neutral	5.70	0.00	1.09	36.86	**21.99**	15.80	18.56	0.00
Sadness	1.69	0.00	0.00	1.31	1.50	**92.76**	2.75	0.00
Surprise	10.23	0.00	0.00	27.84	3.98	22.73	**35.23**	0.00

Table 7: Confusion Matrix for all 21 videos using LGBP + SVM, data smoothed

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**0.00**	62.37	12.90	1.08	9.68	9.68	4.30	0.00
Disgust	55.93	**3.39**	11.86	0.00	1.69	27.12	0.00	0.00
Fear	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00
Happy	0.81	17.89	0.00	**0.00**	43.90	30.89	6.50	0.00
Neutral	21.89	1.28	26.92	5.90	**4.23**	24.30	15.47	0.00
Sadness	2.73	2.73	6.36	48.18	0.00	**39.09**	0.91	0.00
Surprise	44.44	3.70	29.63	0.00	18.52	3.70	**0.00**	0.00

Table 8: Confusion matrix for 5 MMI videos using PCA, data unsmoothed

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**0.00**	62.37	12.90	1.08	9.68	9.68	4.30	0.00
Disgust	55.93	**3.39**	11.86	0.00	1.69	27.12	0.00	0.00
Fear	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00
Happy	0.81	17.89	0.00	**0.00**	43.90	30.89	6.50	0.00
Neutral	21.89	1.28	26.92	5.90	**4.23**	24.30	15.47	0.00
Sadness	2.73	2.73	6.36	48.18	0.00	**39.09**	0.91	0.00
Surprise	44.44	3.70	29.63	0.00	18.52	3.70	**0.00**	0.00

Table 9: Confustion matrix for 5 MMI videos using PCA, data smoothed

---

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**0.00**	0.00	0.00	100.00	0.00	0.00	0.00	0.00
Disgust	25.42	**0.00**	0.00	5.08	0.00	10.17	59.32	0.00
Fear	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00
Happy	0.00	69.92	0.00	**17.89**	0.00	0.00	12.20	0.00
Neutral	0.00	23.74	0.00	32.99	**0.00**	27.17	16.10	0.00
Sadness	0.00	100.00	0.00	0.00	0.00	**0.00**	0.00	0.00
Surprise	0.00	0.00	0.00	14.81	0.00	0.00	**85.19**	0.00

Table 10: Confusion matrix for 5 MMI videos using LGBP + SVM, data unsmoothed

	Anger (%)	Disgust (%)	Fear (%)	Happy (%)	Neutral (%)	Sadness (%)	Surprise (%)	Error (%)
Anger	**0.00**	0.00	0.00	100.00	0.00	0.00	0.00	0.00
Disgust	18.64	**0.00**	0.00	0.00	0.00	5.08	76.27	0.00
Fear	0.00	0.00	**0.00**	0.00	0.00	0.00	0.00	0.00
Happy	0.00	69.92	0.00	**18.70**	0.00	0.00	11.38	0.00
Neutral	0.00	19.50	0.00	22.73	**25.31**	26.92	5.53	0.00
Sadness	0.00	100.00	0.00	0.00	0.00	**0.00**	0.00	0.00
Surprise	0.00	0.00	0.00	7.41	0.00	0.00	**92.59**	0.00

Table 11: Confusion matrix for 5 MMI videos using LGBP + SVM, data smoothed

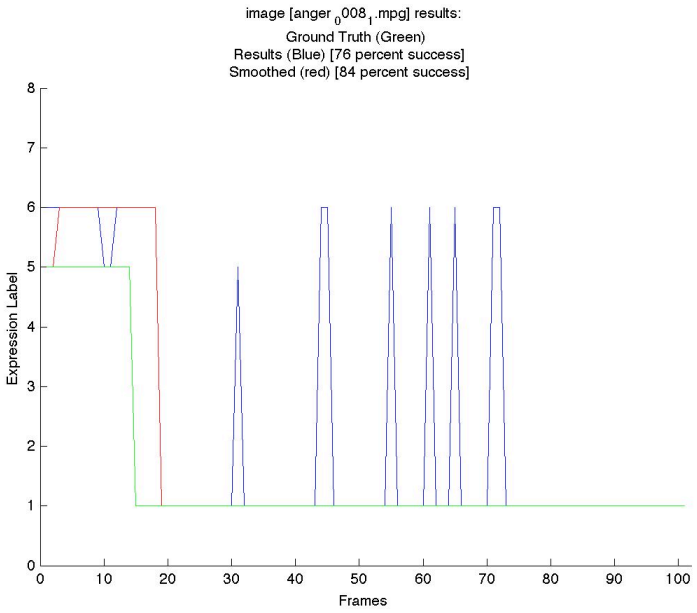


Figure 13: Timeline for video anger_0008_1.mpg, using PCA.

The conversion from index to label is [0: error, 1: anger, 2: disgust, 3: fear, 4: happy, 5: neutral, 6: sadness, 7: surprise]

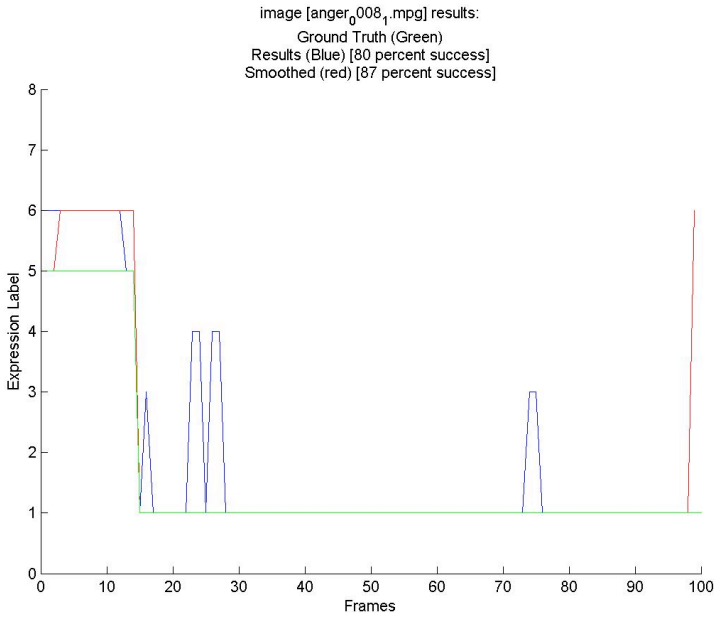


Figure 14: Figure 2: Timeline for video anger_0008_1.mpg, using LGBP.

The conversion from index to label is [0: error, 1: anger, 2: disgust, 3: fear, 4: happy, 5: neutral, 6: sadness, 7: surprise]

4.1.3. Testing on the MMI Database. The confusion matrix was obtained from 5 videos. The overall accuracy was 8.24% success when not smoothed, and 12.24% when smoothed. No errors occurred during face segmentation.

4.2. Local Gabor Binary Patterns

4.2.1. Training. During the training phase of our SVM classifier we obtained a confusion matrix. Overall, this method performed with 81.72% accuracy. Disgust was the most successful expression, 91.57% accurate. Sadness had the most difficulty being recognized, with a success rate of 62.40% (Table 5).

4.2.2. Video Testing. The accuracy for all 21 videos was 34.29% before smoothing, and 40.06% afterwards. With the data smoothing, erroneous frames were reduced to 0.11% from 6.5%. In both original and smoothed cases, sadness was the easiest expression to recognize (87.71% (Table 6) and 92.76% (Table 7) success) whereas fear was the least successful (5.96% (Table 6) and 5.53% (Table 7)).

The timeline for the video *anger_0008_1.mpg* shows similar results than for PCA (Figure 14). In this case, the success rates were slightly better for both the original and unsmoothed data.

4.2.3. Testing on the MMI Database. The overall accuracy was 17.56% success when not smoothed, and 21.80% when smoothed. Once again, no errors occurred during face segmentation.

5. Discussion

5.1. PCA versus LGBP for expression recognition

In our study, we proposed two solutions for facial expression recognition. Both implementations interpreted an expression as a vector dependent on the entire face. In PCA, the vector was based on pixel intensities whereas LGBP correlated edges with expression.

We generally found better results during the training phase compared to testing phases. For PCA, the training phase proved successful 72.26% of the time. However, applying this method to full-length videos reduced the success rates to 38.09% and 40.68% when smoothed. Similarly, the LGBP approach had 81.72% accuracy while training and 34.29% before smoothing, and 40.06% afterwards. This deterioration of performance can be expected because of the randomness of the faces encountered during a full video sequence. A person will transition between expressions, or often have a neutral face despite feeling a certain emotion. The training images on the other hand, were hand picked because they represented a person with a fully expressed face.

Overall, PCA had an accuracy of 38.09% and 40.68% when unsmoothed and smoothed. These values were slightly higher than the LGBP implementations, which succeeded 34.29% and 40.06% of the time. In the following section, we will explore the differences between implementations and their respective results.

5.2. Sources of Error

Neither of the approaches makes assumptions based on face orientation, geometry, or lighting. An idealized situation in both cases would be for the faces to be oriented in the same direction, with all facial features such as the eyes, centered in consistent locations. This assumption was not respected for two reasons. First, we made no attempt to rectify different head poses or orientation, which will throw the important features into different locations of the image. Secondly, we assumed that every person has identical facial geometry. Therefore, even if we centered facial features as mentioned earlier, differences between feature shapes in individuals will not be counted for. In other words, a person with a large nose will yield different results than a smaller one.

Finally, no corrections are made for lighting conditions or perceived skin color. PCA will be severely affected by this, since the method is based on pixel intensity. Videos taken in different lighting conditions, or even if a person is tanned, will affect the projection onto the principal components' space. The LGBP approach does not suffer from this error, since it is based on edges, which is independent of pixel intensity.

Therefore, the overall measurement of an expression is described in (Equation 4). We expect LGBP to perform better than PCA because the $\Delta(lighting)$ term does not affect measurement.

$$Msrmt = Expression + \Delta(Face \quad Orientation) + \Delta(Face \quad Geometry) + \Delta(lighting) \qquad (4)$$

Despite the differing sources of error, both yielded similar success rates when testing with FEED videos. The FEE database controlled lighting and head orientation in a way that minimized these errors. Additionally, training was performed on all individuals, choosing certain frames from most videos. Since there is little variation of lighting and head direction within a video, both implementations were only slightly affected by these errors and yielded similar results. The benefits of the LGBP approach were seen when testing on new individuals, through the MMI databases. Accuracy was increased from 8.24% and 12.24% with PCA to 17.56% and 21.8% with LGBP.

5.3. Importance of Smoothing

A person will express emotions slower than video frame rates. We assume that a person will change emotions with variable transition time, and then hold it for an extended amount of time. For this reason, it is acceptable to use neighboring frames to decide on an expression label. Smoothing the data will remove outliers just as it may propagate errors; however the output makes more sense with respect to time.

The second benefit for smoothing becomes evident when making a decision for hard to interpret frames. During video testing, we kept count of erroneous frames, representing images where faces could not be detected. We saw this result decrease for the better when applying smoothing, because frames with unreliable data used neighborhood data to infer their state.

5.4. Face Detection

Although not the object of our study, the face detection algorithm was reliable overall yet would fail in over 6% of the frames. Since we are working on videos, it is acceptable to miss intermittent frames; however, some instances faces were not found for the majority of a video (87.91% of frames in *disgs_0014_1.mpg*). Increasing the resolution at which the Viola-Jones algorithm searches for features will increase accuracy, but decrease efficiency.

During development, we noticed that the face detector is sensitive to the resolution of the input image. Initial testing was performed on the jpeg frames provided by the FEE database, which are scaled at 320x240 pixels per image. Applying the same face detector on the video frames, which are scaled at 640x480 pixels per image, yielded poorer results. Increasing the image size does not change the number of windows tested in the Viola-Jones algorithm, it merely decreases the granularity of the search, thus lowering the likelihood of finding a solution. It was for this reason that all frames were rescaled to 320x240 before processing.

Although the face detector should be scale independent, we found this quality to be limited. Future iterations of our solution should make this feature more reliable, since it is the precursor to any other step of expression recognition.

5.5. Future Work

In our implementation, we have treated the entire face as a vector and correlated it with an expression. Future implementations can refine the sources of information in order to increase reliability. We have noticed that a subset of features gain importance depending on the emotion. The eyes and mouth are universally the most communicative features. Eyebrows and forehead are unique during periods of anger, and cheeks can help with happiness. Therefore an implementation that would treat facial features independently could increase accuracy. This would also reduce errors due to face orientation and geometry as mentioned earlier (Section 5.2).

Another source of information that could be harnessed in videos is the temporal variations of an individual. We have noticed that surprise is often followed by happiness, or disgust by anger. Keeping track of this prior information may help the algorithm make correct decision.

Finally, an important factor in expressing emotions is in the manner a person moves. When sad, a person tends to move slower than when happy. A person who is surprised tends to hold ones breath, and

resumes when relieved. Extracting shoulder movement to infer breathing rates is an example of temporal information that may make expression recognition more robust.

6. References

1. Ekman, P., *Facial expression and emotion.* American Psychologist, 1993. **48**(4): p. 384-392.
2. Pantic, M., *Machine Analysis of Facial Behaviour : Naturalistic & Dynamic Behaviour 2 . The Process of Automatic Facial Behaviour Analysis.* Philosophical Transactions of the Royal Society of London - Series B: Biological Sciences, 2009. **364**: p. 3505-13.
3. Viola, P. and M. Jones, *Rapid object detection using a boosted cascade of simple features.* Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001, 2001. **1**: p. I-511-I-518.
4. Shan, C., S. Gong, and P.W. McOwan, *Facial expression recognition based on Local Binary Patterns: A comprehensive study.* Image Vision Comput., 2009. **27**(6): p. 803-816.
5. Lajevardi, S.M. and M. Lech. *Facial Expression Recognition Using Neural Networks and Log-Gabor Filters*. in *Computing: Techniques and Applications, 2008. DICTA '08.Digital Image*. 2008.
6. Senechal, T., K. Bailly, and L. Prevost, *Automatic Facial Action Detection Using Histogram Variation Between Emotional States*, in *Proceedings of the 2010 20th International Conference on Pattern Recognition*2010, IEEE Computer Society. p. 3752-3755.
7. Zhang, Z., Z. Zhao, and T. Yuan, *Expression Recognition Based on Multi-scale Block Local Gabor Binary Patterns with Dichotomy-Dependent Weights*

Advances in Neural Networks – ISNN 2009. 2009. **5552**: p. 895-903.

8. Xinghua, S., et al. *Facial expression recognition based on histogram sequence of local Gabor binary patterns*. in *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*. 2008.
9. Jolliffe, I.T., *Principal Component Analysis*2002: Springer-Verlag.
10. Yubo, W., et al. *Real time facial expression recognition with AdaBoost*. in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. 2004.
11. Ying-li, T., *Recognizing Action Units for Facial Expression Analysis.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001. **23**: p. 97-115.
12. Shin, G. and J. Chun, *Spatio-temporal Facial Expression Recognition Using Optical Flow and HMM*

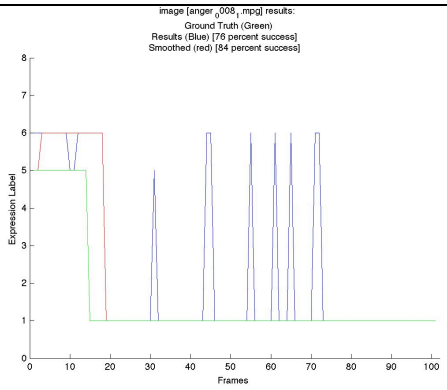
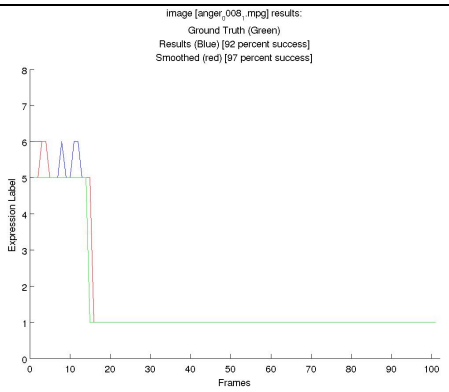
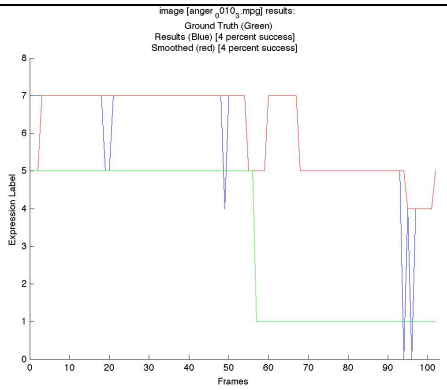
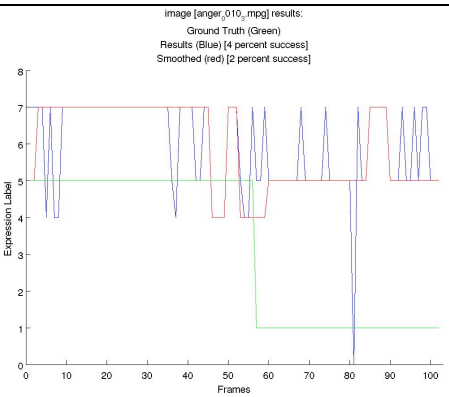
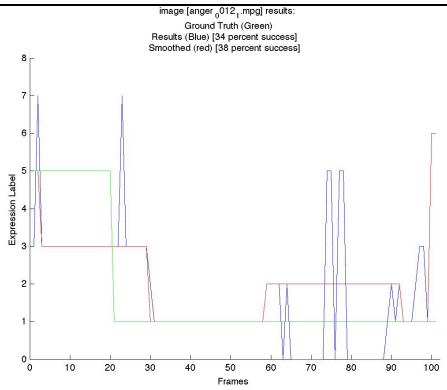
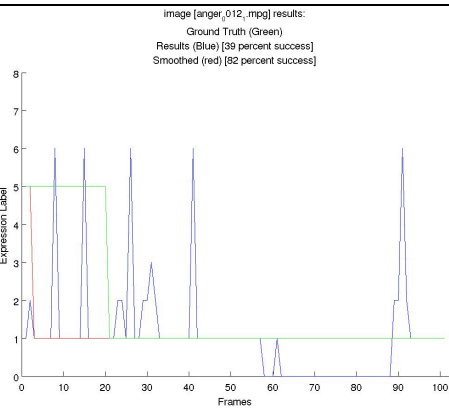
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. 2008. **149**: p. 27-38.

13. S\, A., et al., *Differential optical flow applied to automatic facial expression recognition.* Neurocomput., 2011. **74**(8): p. 1272-1282.
14. Wenchao, Z., et al. *Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition*. in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. 2005.
15. Cristianini, N. and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*2000: Cambridge University Press.
16. Viola, P. and M. Jones. *Robust real-time face detection*. in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. 2001.
17. Freund, Y. and R.E. Schapire, *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* Journal of Computer and System Sciences, 1997. **55**(1): p. 119-139.
18. Pearson, K., *{On lines and planes of closest fit to systems of points in space}.* Philosophical Magazine, 1901. **2**(6): p. 559-572.
19. Sirovich, L. and M. Kirby, *Low-dimensional procedure for the characterization of human faces.* J. Opt. Soc. Am. A, 1987. **4**(3): p. 519-524.
20. Ojala, T., M. Pietikainen, and D. Harwood. *Performance evaluation of texture measures with classification based on Kullback discrimination of distributions*. in *Pattern Recognition, 1994.*

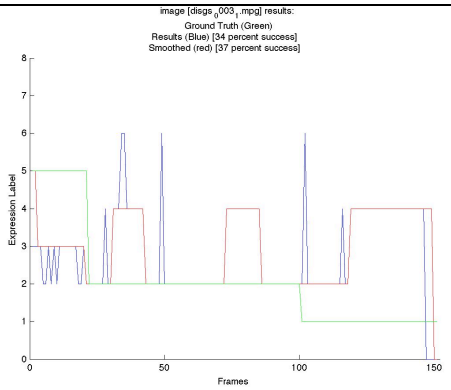
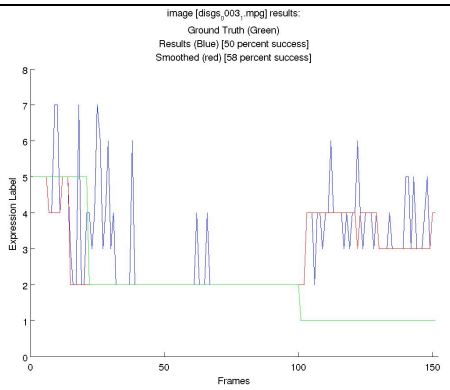
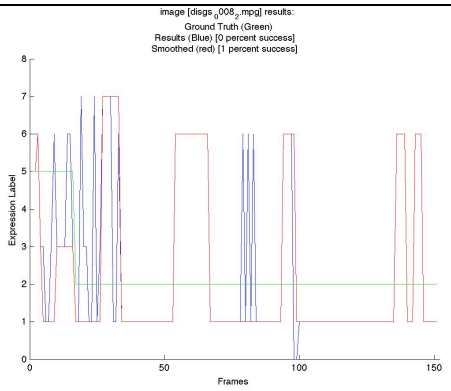
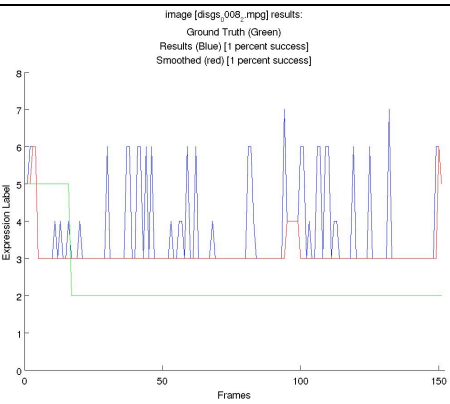
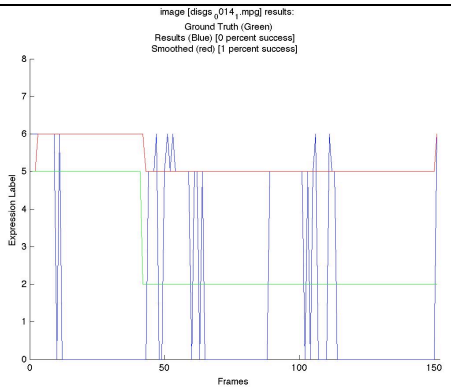
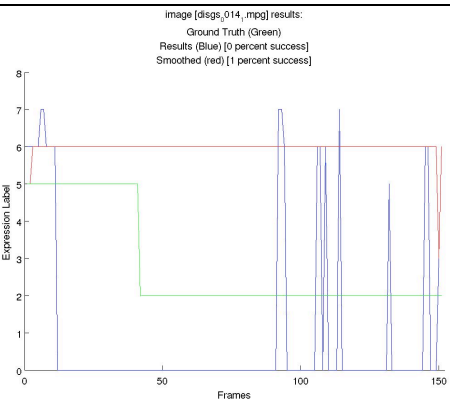
Vol. 1 - Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on. 1994.
21. Ojala, T., M. Pietikäinen, and D. Harwood, *A comparative study of texture measures with classification based on featured distributions.* Pattern Recognition, 1996. **29**(1): p. 51-59.
22. Sim, T., S. Baker, and M. Bsat, *The CMU Pose, Illumination, and Expression Database.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003. **25**(12): p. 1615-1618.
23. Phillips, P.J., et al., *The FERET evaluation methodology for face-recognition algorithms.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2000. **22**(10): p. 1090-1104.
24. Frischholz, R.W. and U. Dieckmann, *BiolD: a multimodal biometric identification system.* Computer, 2000. **33**(2): p. 64-68.
25. Lyons, M., et al. *Coding facial expressions with Gabor wavelets*. in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. 1998.
26. Kanade, T., Y. Tian, and J.F. Cohn, *Comprehensive Database for Facial Expression Analysis*, in *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*2000, IEEE Computer Society. p. 46.
27. Dhall, A., et al. *Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark*. in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. 2011.
28. Lijun, Y., et al. *A 3D facial expression database for facial behavior research*. in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*. 2006.
29. Wallhoff, F., et al. *Efficient Recognition of Authentic Dynamic Facial Expressions on the Feedtum Database*. in *Multimedia and Expo, 2006 IEEE International Conference on*. 2006.

7. Appendix

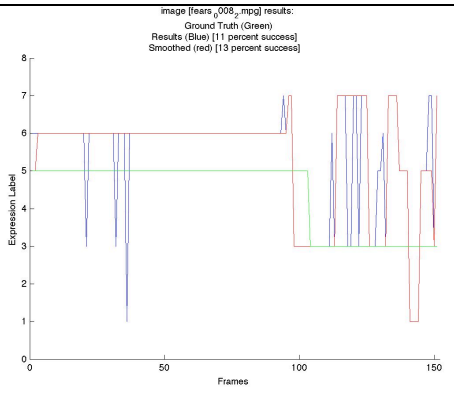
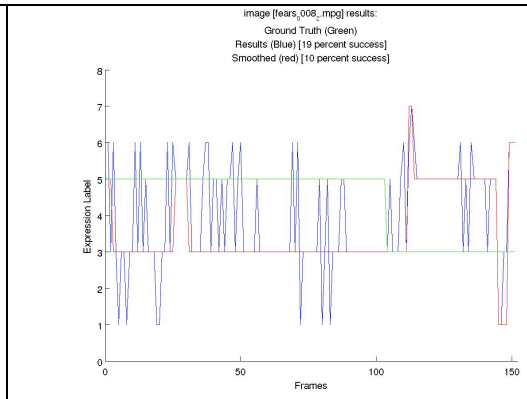
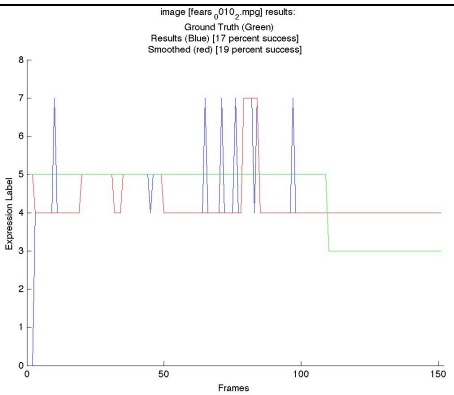
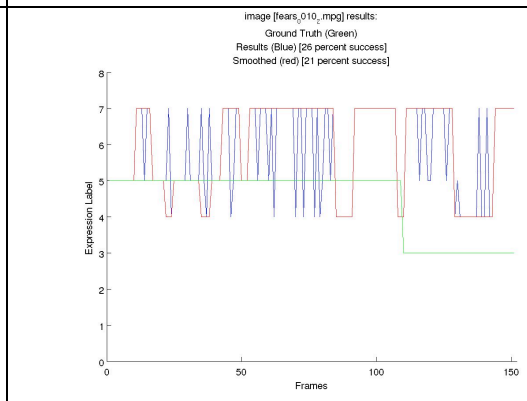
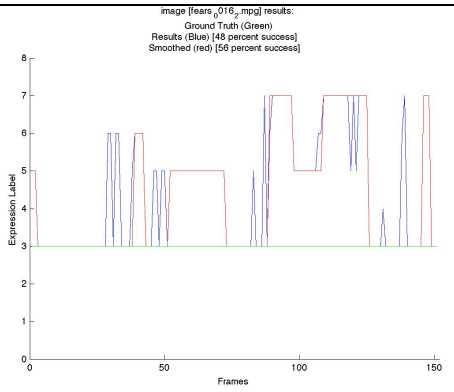
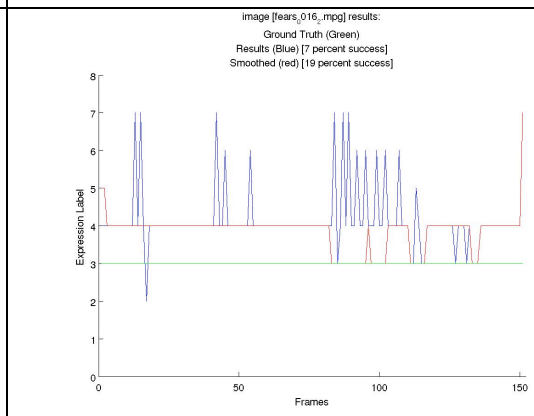
7.1. Anger

PCA	LGBP + SVM
	
	
	

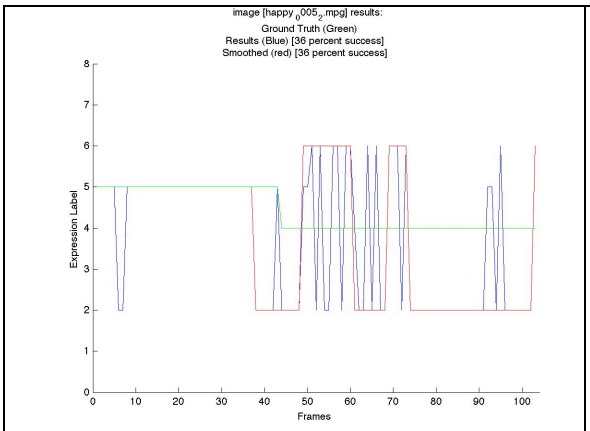
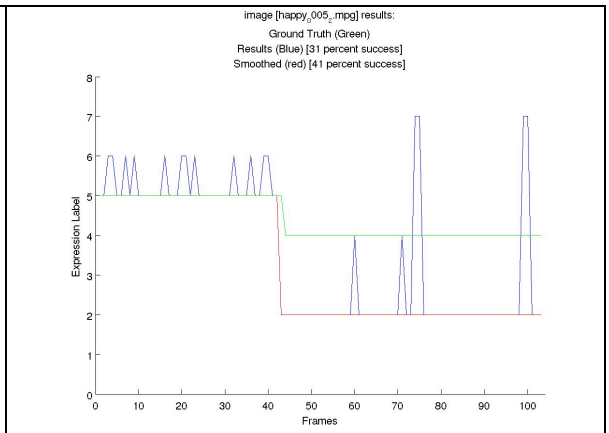
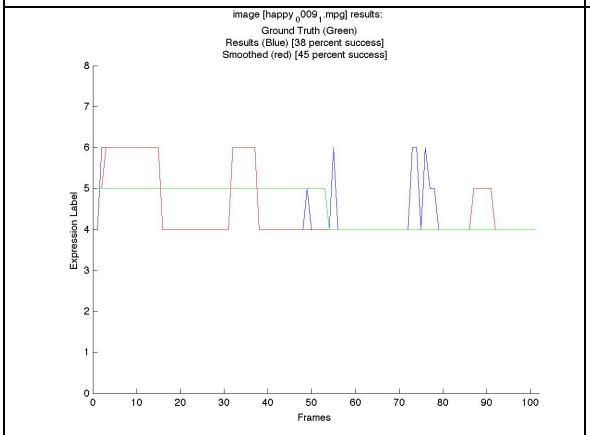
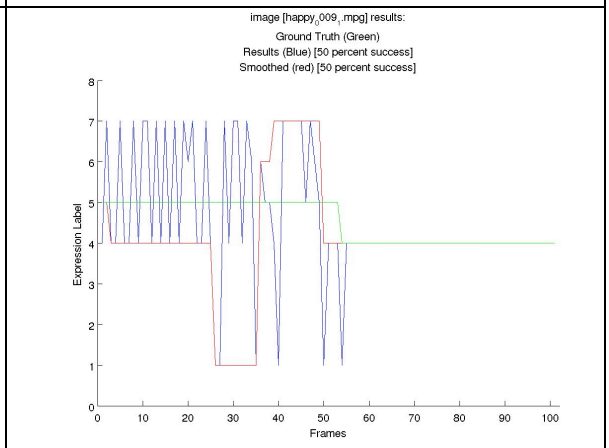
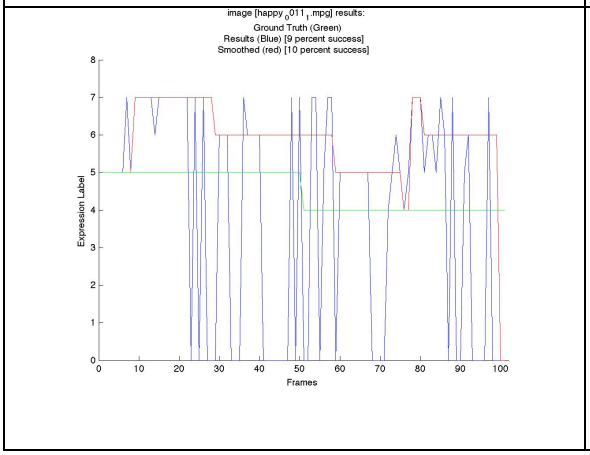
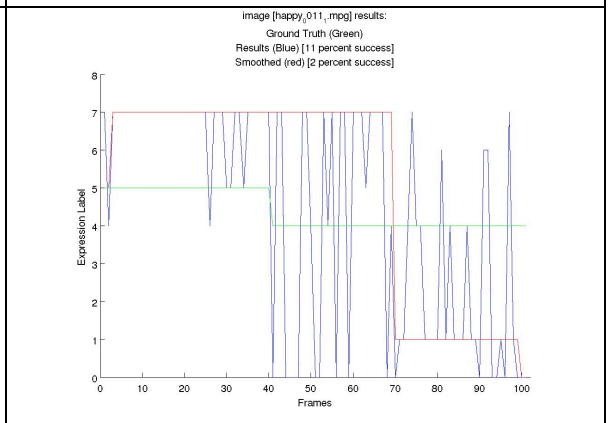
7.2. Disgust

PCA	LGBP + SVM
	
	
	

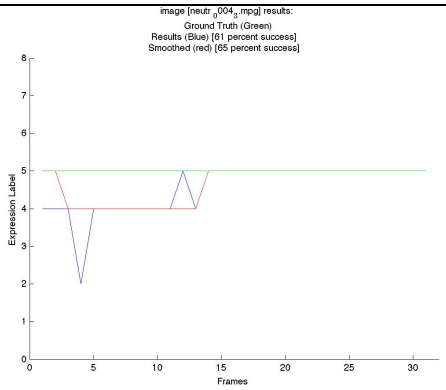
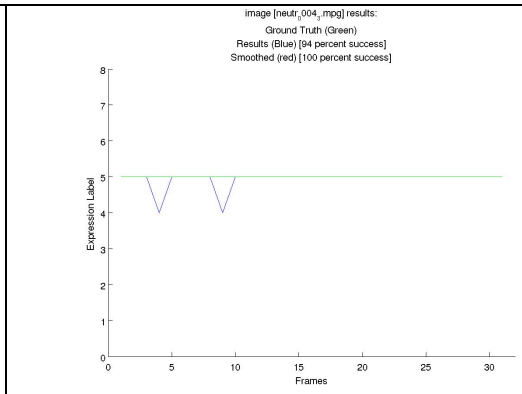
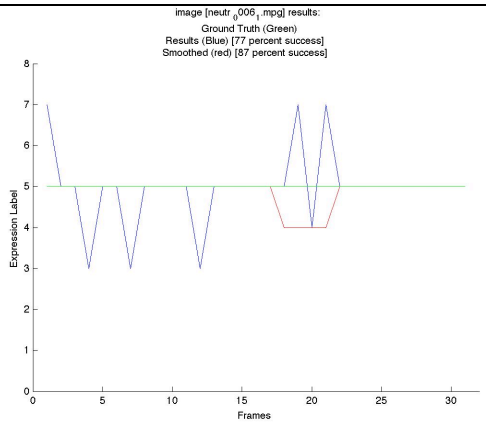
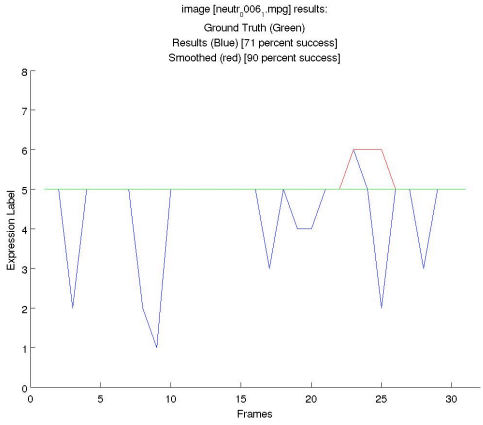
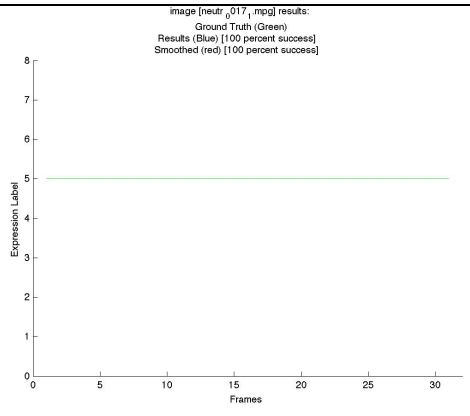
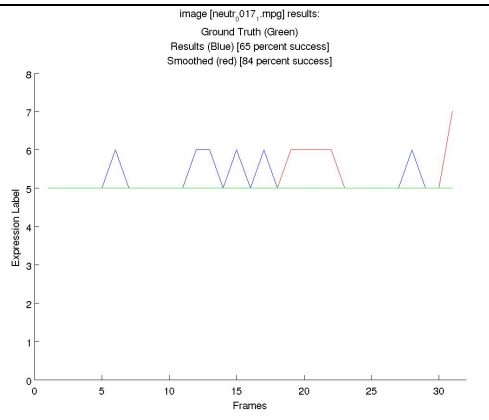
7.3. Fear

PCA	LGBP + SVM
	
	
	

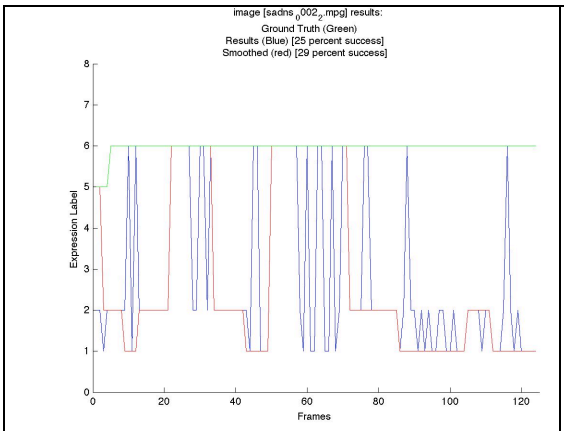
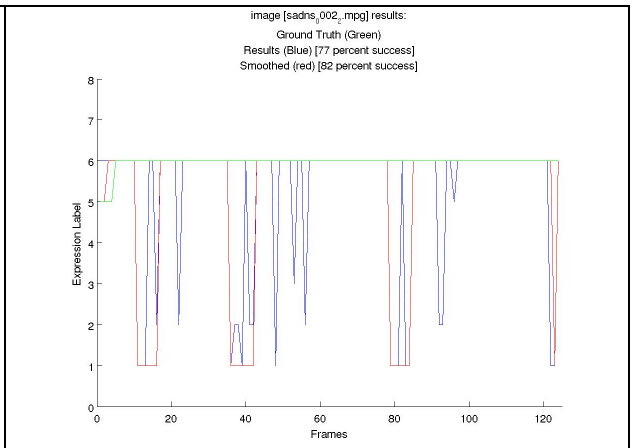
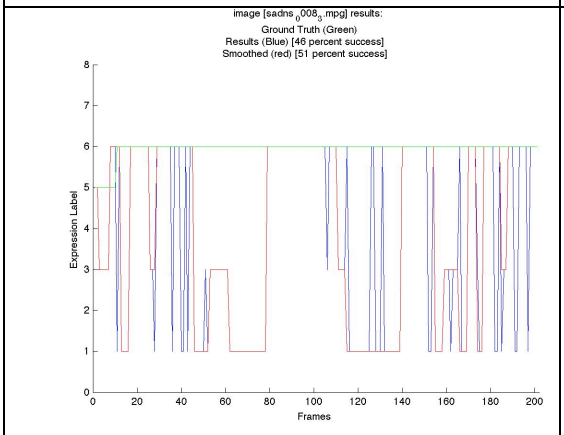
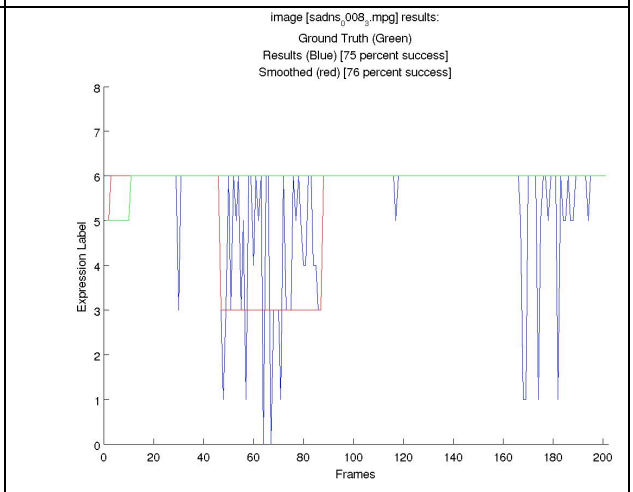
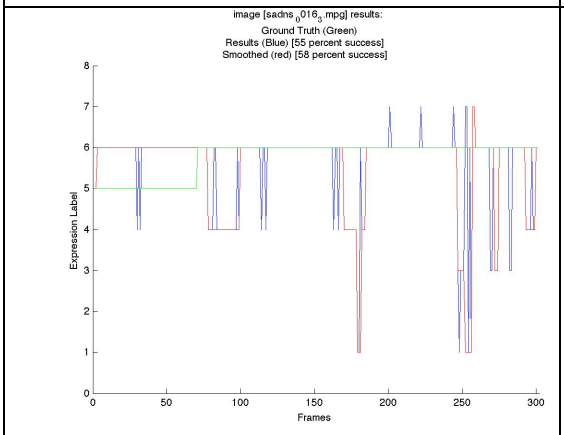
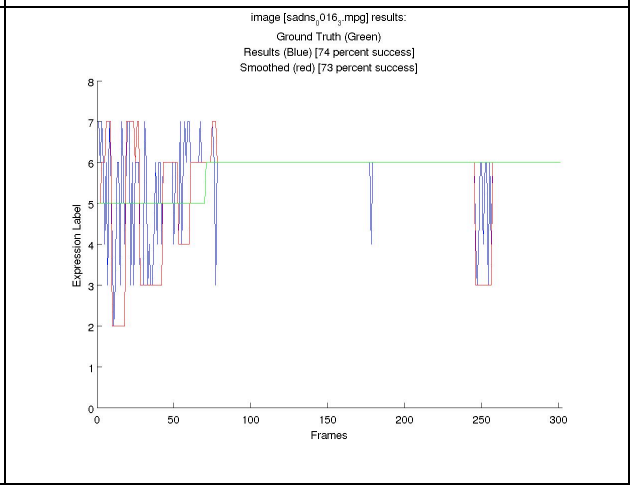
7.4. Happy

PCA	LGBP + SVM
	
	
	

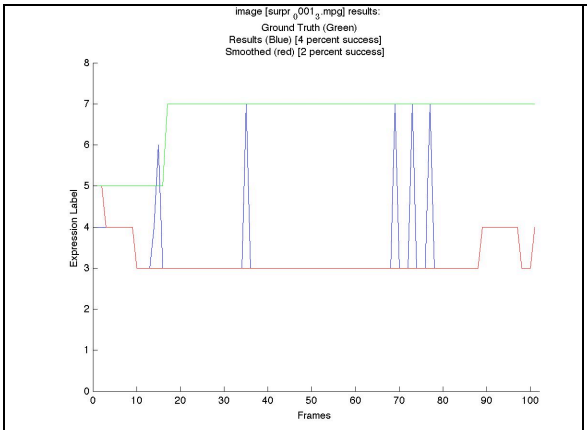
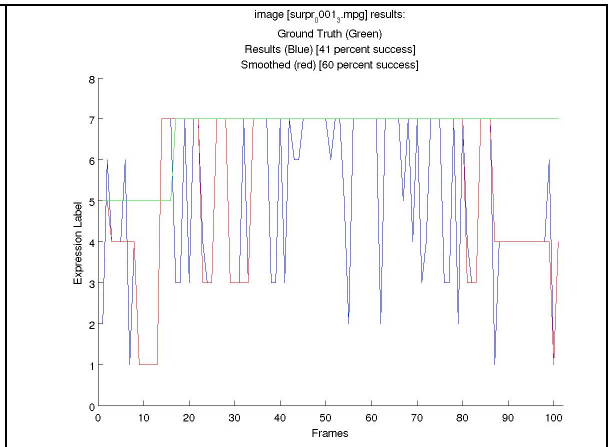
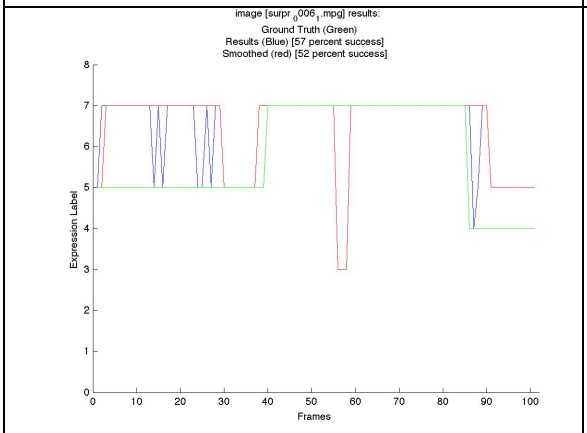
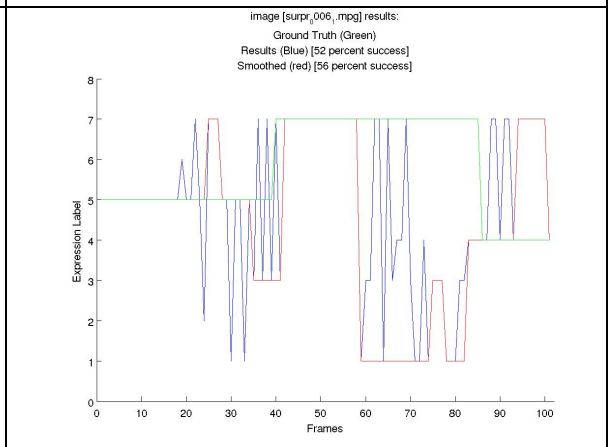
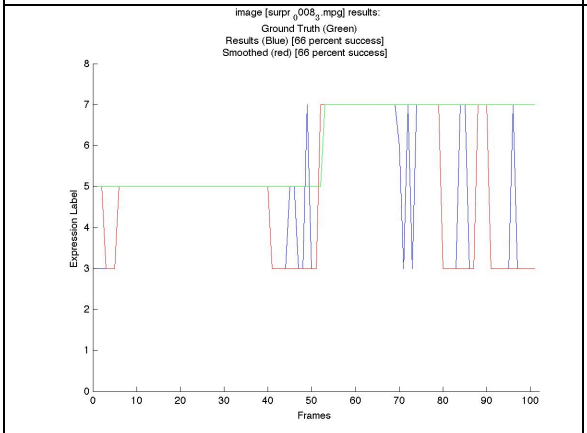
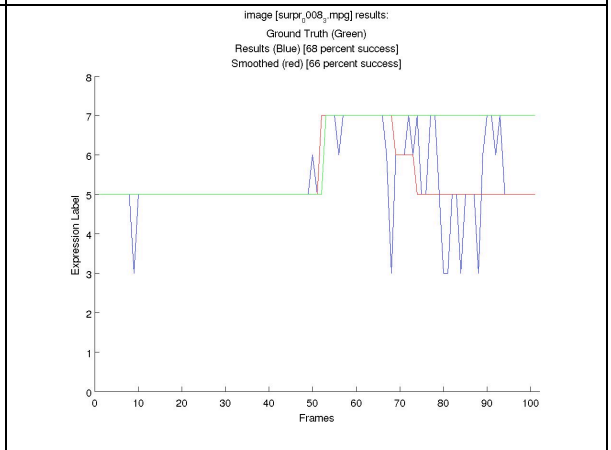
7.5. Neutral

PCA	LGBP + SVM
	
	
	

7.6. Sadness

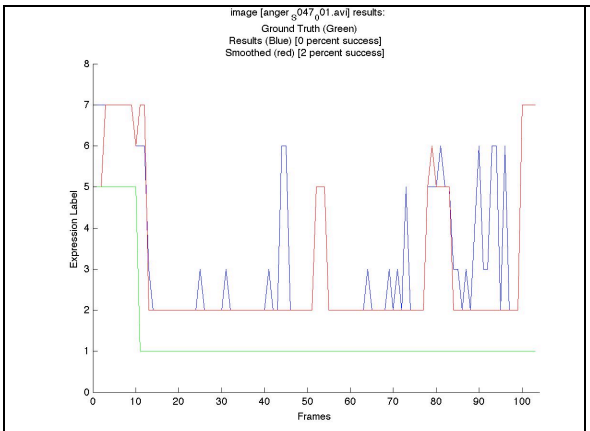
PCA	LGBP + SVM
	
	
	

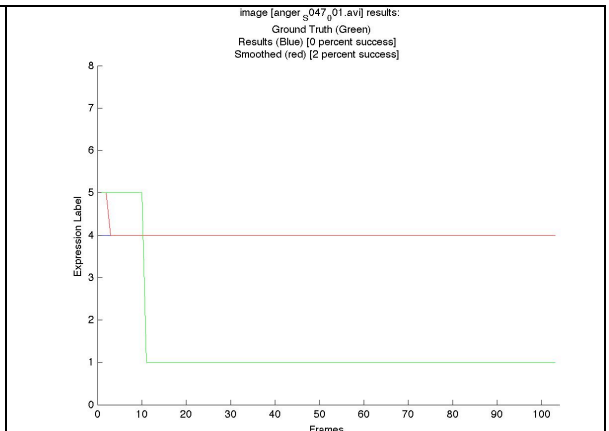
7.7. Surprise

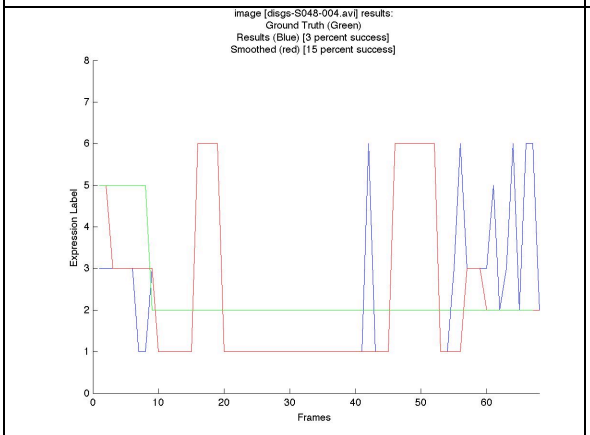
PCA	LGBP + SVM
	
	
	

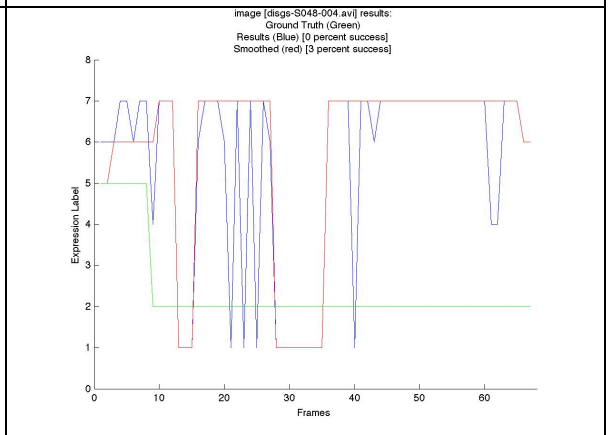
7.8. MMI Database

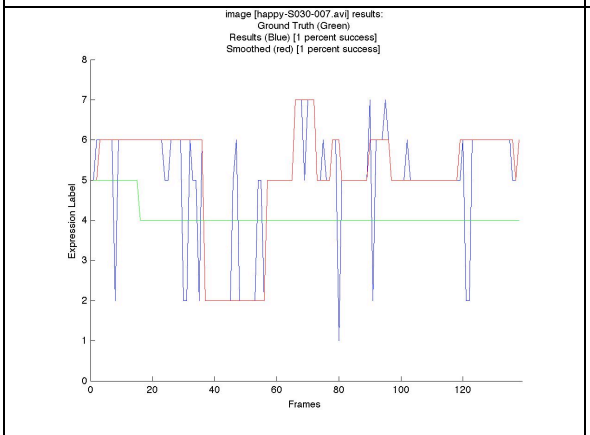
PCA	LGBP + SVM

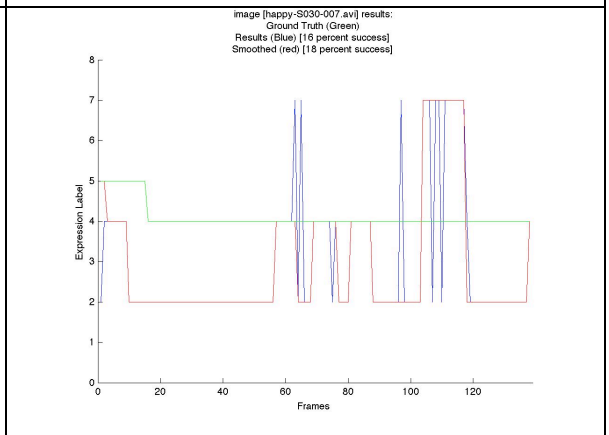


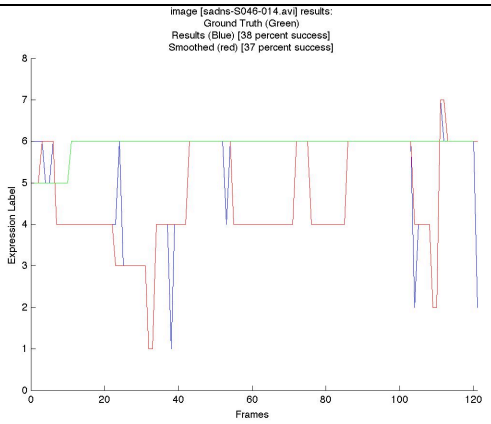
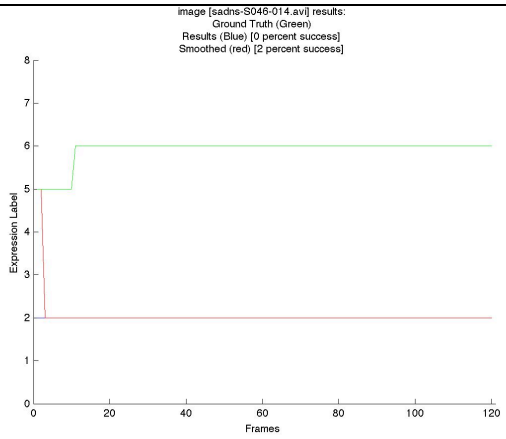
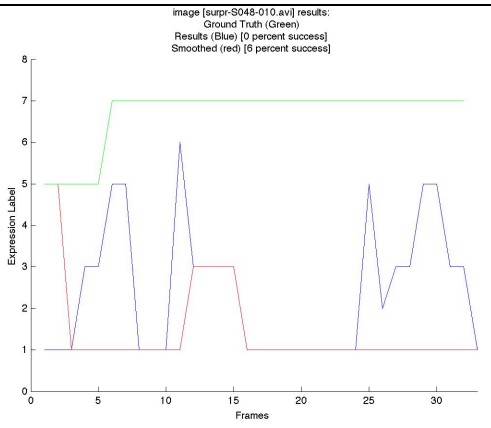
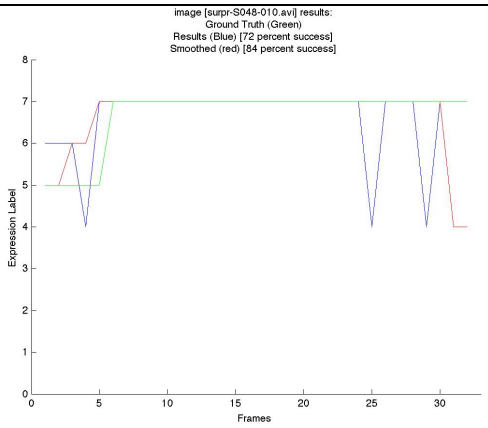










PCA	LGBP + SVM
 image [sadns-S046-014.avi] results: Ground Truth (Green) Results (Blue) [38 percent success] Smoothed (red) [37 percent success] Expression Label Frames	 image [sadns-S046-014.avi] results: Ground Truth (Green) Results (Blue) [0 percent success] Smoothed (red) [2 percent success] Expression Label Frames
 image [surpr-S048-010.avi] results: Ground Truth (Green) Results (Blue) [0 percent success] Smoothed (red) [6 percent success] Expression Label Frames	 image [surpr-S048-010.avi] results: Ground Truth (Green) Results (Blue) [72 percent success] Smoothed (red) [84 percent success] Expression Label Frames