

# Mood Detection: Implementing a facial expression recognition system

---

Neeraj Agrawal, Rob Cosgriff and Ritvik Mudur

## 1. Introduction

Facial expressions play a significant role in human dialogue. As a result, there has been considerable work done on the recognition of emotional expressions and the application of this research will be beneficial in improving human-machine dialogue. One can imagine the improvements to computer interfaces, automated clinical (psychological) research or even interactions between humans and autonomous robots.

Unfortunately, a lot of the literature does not focus on trying to achieve high recognition rates across multiple databases. In this project we develop our own mood detection system that addresses this challenge. The system involves pre-processing image data by normalizing and applying a simple mask, extracting certain (facial) features using PCA and Gabor filters and then using SVMs for classification and recognition of expressions. Eigenfaces for each class are used to determine class-specific masks which are then applied to the image data and used to train multiple, one against the rest, SVMs. We find that simply using normalized pixel intensities works well with such an approach.

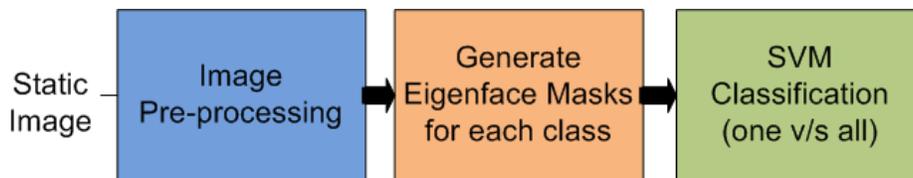


Figure 1 – Overview of our system design

## 2. Image pre-processing

We performed pre-processing on the images used to train and test our algorithms as follows:

1. The location of the eyes is first selected manually
2. Images are scaled and cropped to a fixed size (170 x 130) keeping the eyes in all images aligned
3. The image is histogram equalized using the mean histogram of all the training images to make it invariant to lighting, skin color etc.
4. A fixed oval mask is applied to the image to extract face region. This serves to eliminate the background, hair, ears and other extraneous features in the image which provide no information about facial expression.

This approach works reasonably well in capturing expression-relevant facial information across all databases. Examples of pre-processed images from the various datasets are shown in Figure-2a below.



Figure 2a – *Top*: Original images, *Bottom*: Processed images with mask

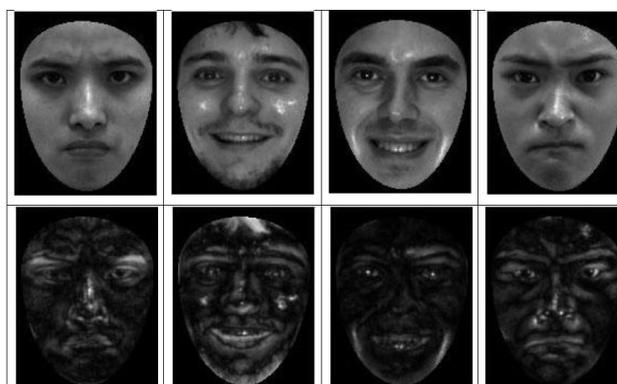


Figure 2b – *Top*: Pre-processed images, *Bottom*: L1 norm of the Gabor bank features

### 3. Feature extraction

#### 3.1. Normalized pixel intensities

Every image in our training set is normalized by subtracting the mean of all training set images. The masked region is then converted to a column vector which forms the feature vector. This is a common (albeit naïve) approach [1] and produces a feature vector of length 15,111 elements.

#### 3.2. Gabor filter representations

Gabor filters are often used in image processing and are based on physiological studies of the human visual cortex [2]. The use of Gabor filtered facial images has been shown to result in improved accuracy for facial expression recognition [1][3][4]. One approach to using these filters is to generate a bank of filters across multiple spatial frequencies and orientations. The filtered outputs are then concatenated, and down-sampling or PCA is often used to reduce dimensionality. We use an approach similar to [3] that provides competitive results, and use the L1 norm of each of the Gabor bank features for a given image. Our Gabor bank contains filters at 5 spatially varying frequencies and 8 orientations. Figure-2b shows examples of our Gabor features.

### 4. Eigenface masks

The feature vectors discussed above suffer from high dimensionality, which can cause overfitting during classification. One approach to reducing the dimension of the feature vectors is to apply principal component analysis. In [5], eigenfaces are used to generate a mask that eliminates pixels that vary little across training samples in different labels. In our system, we modify the approach to generate a separate mask for each expression class. The procedure is as follows:

1. PCA is applied separately to images in each class and the first 10 principal components are stored to represent the class subspace
2. Images of a given class are projected onto all other subspaces and then reconstructed
3. The average reconstruction error is determined for all training samples within a class.
4. Pixels above the 90<sup>th</sup> percentile rank (i.e. high reconstruction error) are used in the mask for the corresponding class. This gives a feature vector length of 2240.

This approach stresses those facial regions (and pixels) that are most significant in defining a particular expression. A few samples of the generated masks are shown in Figure-3 below.

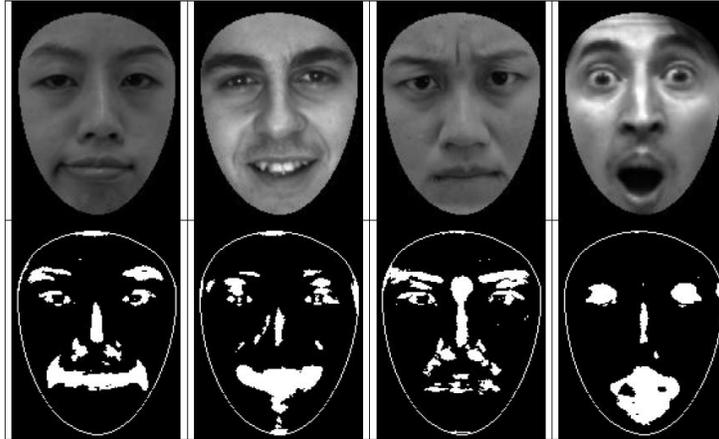


Figure 3 – *Top*: Training images of class contempt, happy, angry and surprised (from left to right); *Bottom*: Masks generated by our method

## 5. Classification

In order to use the eigenface masks described above, we train five different one-against-rest SVMs (one for each expression class except neutral). The algorithm is as follows:

<b>Classification Algorithm</b>	
<u>Training</u>	<p><i>For each class <math>i</math> except Neutral {</i></p> <p style="padding-left: 2em;"><i>Separate data into two groups of</i></p> <p style="padding-left: 4em;"><i>➤ label <math>i</math> v/s not label <math>i</math></i></p> <p style="padding-left: 2em;"><i>Construct feature vectors using the class mask for class <math>i</math></i></p> <p style="padding-left: 2em;"><i>Train one-vs-rest SVM</i></p> <p style="padding-left: 2em;"><i>}</i></p>
<u>Testing</u>	<p><i>For each class <math>i</math> except Neutral{</i></p> <p style="padding-left: 2em;"><i>Construct feature vector for the test image using the class mask for class <math>i</math></i></p> <p style="padding-left: 2em;"><i>Calculate probability of the test image being of label <math>i</math> using the corresponding SVM model</i></p> <p style="padding-left: 2em;"><i>}</i></p> <p><i>If (max(Prob) &gt; threshold)</i></p> <p style="padding-left: 2em;"><i>Label = class with the max probability</i></p> <p><i>Else</i></p> <p style="padding-left: 2em;"><i>Label = Neutral</i></p>

Table 1 – Classification algorithm using eigenface masks

## 6. Experimentation and Result

In order to focus on recognition across databases, we combined three publicly available image datasets for training and testing our system. An appropriate subset of CMU PIE[6], TFEID[7], and AR[8] databases were chosen. Only frontal images were chosen with uniform illumination throughout the image, with multiple expressions of each subject, and subjects were limited to those without glasses. Our final dataset comprises of 639 images and six facial expressions.

For the evaluation of our system, we used the publicly available libsvm [9] library. The SVM was trained using a Gaussian radial basis function kernel. We also used parameter fitting to get the best values for the parameters  $\gamma$  ( $\gamma = 0.04$ ) and  $C$  ( $C = 2$ , coefficient in the regularization term) to the SVM kernel in the following equation:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2), \gamma > 0.$$

Since, our experimentation involves dividing the dataset randomly to get training and testing images, each experiment was repeated for 1000 iterations and average results obtained for overall accuracy. Figure 4 shows the ROC curve for neutral expression with varying threshold parameter. We used the ROC to determine a threshold which gave a small false positive rate while also correctly labeling most of the true positive (true neutral) expressions. We obtained a threshold of 0.40

In order to evaluate the overall performance for each expression, we used 30% of images of each class for testing and a varying fraction of the remaining images for training. Figure 4 shows the training set error as we vary the training set size. The test error generally decreases with increase in training set size for all classes.

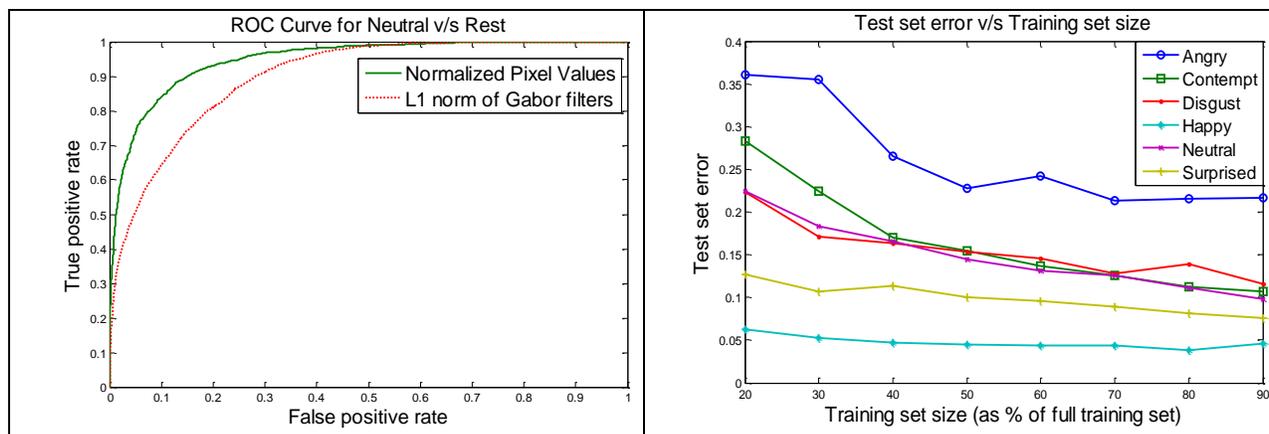


Figure 4 – Left: ROC curve for Neutral v/s Rest of the labels; Right: Test set error v/s training set size

Table 2 shows the accuracy obtained for various expressions when using 30% images for testing and remaining images for training. The accuracy for surprised and happy expressions was observed to be relatively higher than other expressions. (since these expressions did not vary much across datasets).

	Angry	Contempt	Disgust	Happy	Neutral	Surprised
Gabor	87.60	71.40	88.50	89.25	70.96	94.31
Pixel	78.30	89.30	88.42	95.43	90.16	92.46

Table 2 – Accuracy (%) of recognition for each class, using both Gabor and normalized pixels as features

## 7. Conclusion

In our final system, an approach using eigenface masks was developed and implemented to classify facial expressions across publicly available databases. These learned masks along with feature vectors of the image were used to train the SVM. This method produced competitive results across significantly varying databases.

The results were similar for both the normalized pixel value and the gabor filtered feature vectors, with neither representation being clearly superior to the other. The performance across databases is indicative of the method's robustness to variations in facial structure and skin tone when recognizing the expression.

However, both representations of the feature vectors end up being very high dimensional. This hurts the run time of the algorithm hampering the ability to use it in real time, as well as leading to possible over fitting of the data during training. Increasing the number of available training images would help to compensate for over fitting. Additional improvements to the method may be obtained by experimenting with other techniques. For example using ICA to further reduce dimensionality [10], the use of AdaBoost in conjunction with gabor representations of the image [4], and the use of facial action units [11].

## References

- [1] B. Fasel, Juergen Luetttin, Automatic facial expression analysis: a survey, *Pattern Recognition*, vol. 36, no. 1, pp. 259-275, January 2003
- [2] Daugman, J.G., "Two-dimensional spectral analysis of cortical receptive field profiles", *Vision Res.*, vol 20 (10), pp. 847-56
- [3] Seyed Mehdi Lajevardi, Margaret Lech, "Averaged Gabor Filter Features for Facial Expression Recognition," *Digital Image Computing: Techniques and Applications*, pp. 71-76, 2008
- [4] Bartlett, M.S.; Littlewort, G.; Frank, M.; Lainscsek, C.; Fasel, I.; Movellan, J., "Recognizing facial expression: machine learning and application to spontaneous behavior," *Computer Vision and Pattern Recognition*, 2005. IEEE Computer Society Conference on , vol.2, no., pp. 568-573 June 2005
- [5] Carmen Frank and Elmar Noth, "Optimizing Eigenfaces by Face Masks for Facial Expression Recognition," *Computer Analysis of Images and Patterns* vol. 2756/2003, pp. 646-654
- [6] Sim, T., Baker, S., and Bsat, M. 2002. The CMU Pose, Illumination, and Expression (PIE) Database. In *Proceedings of the Fifth IEEE international Conference on Automatic Face and Gesture Recognition* (May 20 - 21, 2002). FGR. IEEE Computer Society, Washington, DC, 53.
- [7] Li-Fen Chen and Yu-Shiuan Yen. (2007). Taiwanese Facial Expression Image Database [<http://bml.ym.edu.tw/download/html>]. Brain Mapping Laboratory, Institute of Brain Science, National Yang-Ming University, Taipei, Taiwan.
- [8] A.M. Martinez and R. Benavente, "The AR face database," CVC Tech. Report #24, 1998
- [9] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [10] Bruce A. Draper, Kyungim Baek, Marian Stewart Bartlett, J. Ross Beveridge, Recognizing faces with PCA and ICA, *Computer Vision and Image Understanding*, Volume 91, Issues 1-2, Special Issue on Face Recognition, July-August 2003, Pages 115-137
- [11] Ying-li Tian, Takeo Kanade, Jeffrey F. Cohn, "Recognizing Action Units for Facial Expression Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97-115, Feb. 2001