

# Facial Expression Analysis by Support Vector Regression

Chao Fan<sup>1</sup>, Hossein Sarrafzadeh<sup>1</sup>, Farhad Dadgostar<sup>1</sup>, and Hamid Gholamhosseini<sup>2</sup>

<sup>1</sup>Institute of Information and Mathematical Sciences  
Massey University  
Private Bag 102 904 NSMC, Auckland, New Zealand  
{c.fan, h.a.sarrafzadeh, f.dadgostar}@massey.ac.nz

<sup>2</sup>Department of Electrical & Electronic Engineering  
Auckland University of Technology  
Private Bag 92006, Auckland 1020  
hamid.gholamhosseini@aut.ac.nz

## Abstract

Real-time Facial Expression analysis is one the important topics in the development of the next generation affect-sensitive user interfaces. However current algorithms and techniques are computationally expensive, and therefore are not suitable for real-time applications. In this paper we present a real-time system for analyzing the six basic facial expressions of the human user, using regression analysis. In this research we applied linear lighting correction algorithm and contrast-limited adaptive histogram equalization to enhance facial features. Polygonized vector image are applied instead of pixel represented image.

**Keywords:** Support vector regression, Face detection.

## 1 Introduction

Implementing real-time facial expression analysis system is one of the current research focuses in human-computer interaction (HCI) and affect-sensitive user interfaces. This is a challenge task because a real-world application should be robust against different facial features which are dependent to race, sex, age, and lighting condition. On the other hand algorithms which are more accurate require more computation power, which makes them less suitable for using with current technology of the home or office computers.

So far, several methods were used for this purpose, such as skin color detection, motion detection, and pattern matching. Some of the like skin color detection and motion detection are fast method but don't improve machine intelligent method and were not successful. Some others like ANNs require a huge amount of calculation.

Formulating the facial expression recognition problem as a classification problem is not a new approach to the pattern recognition community. One of the classical approaches to this problem is vectorizing facial features and using the vectorized data as the input of a classifier. Although selecting the vectors which are more useful and choosing the classification method are still two open questions. Choosing a large number of feature vectors is not preferred, because producing an accurate classifier

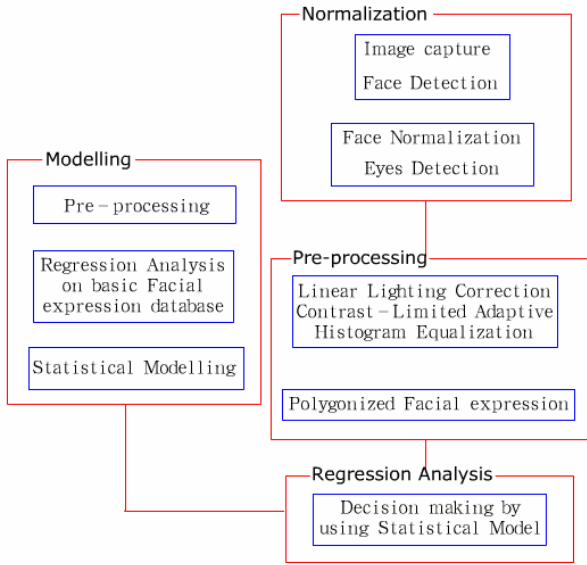
requires a larger amount of training data and slower classification speed. Therefore choosing lesser feature vectors together with studying approaches which require fewer amounts of training data and faster classification can improve this research branch.

In recent years, Support Vector Regression (SVR) is being grown up quickly. SVR has been successful in many fields. This technique is based on analyzing the high dimensional feature space for classification. The SVR classifiers also called Support Vector Machin (SVM).

In this research we present the application of SVR for facial expression analysis. Our method has three basic steps. The first step is face and eye detection in the image. We used a modified version of the Viola and Jones [1] HAAR-like feature analysis algorithm as a pre-processor to speed up detecting the face and eye area . The second step is vectorizing the facial features. We used polygon approximation technique to reduce the amount of data which should be transferred to the classifier. Final step is classifying the facial expression features using the SVR technique.

## 2 System Overview

There are four major blocks in this system as indicated in Figure 1 and the purpose of each subsystem briefly described in the following paragraphs.



**Figure 1. Block diagram of the system**

**Face image normalization:** Captures the image from web camera. Then, applies SVR and Viola-Jones algorithm to detection face and eye location.

**Pre-processing:** Smooth the input image using the lighting correction algorithm and contrast-limited adaptive histogram equalization. In addition, produces polygonized vector image using polygonal approximation algorithm.

**Statistical Modeling block (training):** Selects the six basic emotions from the facial expression. Facial expressions are selected from different race, sex, lighting condition from face database. Each group includes about 2000 image. There are a total of 12000 images representing different facial expressions. These images were processed by pre-processing block. Regression analysis is done on the polygonized data to find a stable model.

**Regression analysis:** Uses statistical models to predict the class of the captured image.

### 3 Methodology

#### 3.1 Support Vector Machine Overview

Support vector machine is a very effective method for general purpose pattern recognition. SVM is particularly a good tool to classify a set of points which belong to two or more classes. SVMs are based on statistical learning theory and try to find the biggest margin to separate different classes. SVM embed data into a high

dimensional feature space. This method uses the hyper-plane that separates the largest possible fraction of points of the same class on the same side, while it maximizes the distance of either class from the hyper-plane. Hence there is only the inner product involved in SVM, learning and predicting is much faster than a multilayer neural network.

Vapnik and et al. [2, 3] introduced the SVM as a supervised learning algorithm. This algorithm operates by mapping training set into a high-dimensional feature space, and separate positive and negative samples. In statistical learning theory, for some classes of well-behaved data, the choice of the maximum margin hyper-plane will lead to maximal generalization when predicting the classification of previously unseen examples. The basic concept of SVM is as follows:

Assume we want to fit a function  $g(x)$  through a set of samples  $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$ .  $x \in \mathbb{R}^n$  and  $y \in \{-1, +1\}$ .  $y$  is the label of two different classes [4].

$$y(x) = \text{sign}[w^T \varphi(x) + b]$$

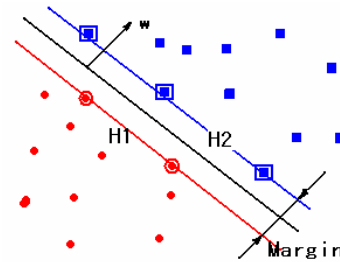
$\Phi(x)$  maps input data to a high dimensional feature space as follows.

$$\begin{aligned} w^T \varphi(x_i) + b &\geq +1, & \text{if } y_i = +1 \\ w^T \varphi(x_i) + b &\leq -1, & \text{if } y_i = -1 \end{aligned}$$

The above equation can be combined into one equation:

$$y_i (\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i$$

These points lie on the hyper-plane  $H_1 : w^T \Phi(x) + b = 1$  and  $H_2 : w^T \Phi(x) + b = -1$ , The margin between these two hyper-plane is:  $d = 2/\|w\|$ , shows in Figure 2. The points lie one the  $H_1$  and  $H_2$  are support vectors.



**Figure 2. Support Vectors and Margin**

The maximum the margin between  $H_1$  and  $H_2$  is equitant minimizing  $\|w\|^2$ , subject to constraints. The Lagrange multipliers  $\alpha$  is introduced such that:

$$L_P \equiv \frac{1}{2} \|w\|^2 - \sum_{i=1}^l \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^l \alpha_i$$

To solve the above equation, we take derivative to  $w$  and  $b$ , and then we will have:

$$w = \sum_i \alpha_i y_i x_i$$

and

$$\sum_i \alpha_i y_i = 0.$$

Then we substitute them to equation then we have:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

This problem called Quadratic Programming problem [5], in the solution  $\alpha > 0$  are called "Support Vectors". Support Vectors lie on the hyper-planes  $H_1$  or  $H_2$ . For finding the Support Vectors a package like LIBSVM [6] can be used.

In this equation,  $\langle X_i \cdot X_j \rangle$  is called Kernel which maps to the feature space. Some possible kernels are:

$$K(x, x_i) = x_i^T x \text{ (linear SVM)}$$

$$K(x, x_i) = (x_i^T x + \tau)^d \text{ (polynomial SVM of degree } d)$$

$$K(x, x_i) = \exp\{-\|x - x_i\|_2^2 / \sigma^2\} \text{ (RBF kernel)}$$

$$K(x, x_i) = \tanh(\kappa x_i^T x + \theta) \text{ (MLP kernel)}$$

Thus, by using Support Vectors, decision function can be written as follows, which only requires the inner product.

$$y(x) = \text{sign}[\sum_{i=1}^{\#sv} \alpha_i y_i K(x, x_i) + b]$$

### 3.2 Images capture and face detection

We have used a simple web camera as the image grabbing device, mounted on top of the monitor. The image grabber provides the frontal view of the face of the user. The lighting condition is normal and constant (Figure 3). The distance between the face and the image grabber assumed was the range of 0.5 to 1 meter. The size of image is 320 pixels in width and 240 pixels in height.



Figure 3. A sample of the input image.

### 3.3 Finding the face location

For locating the face area in the image, we have used a real-time face detection algorithm developed by Viola and Jones [1]. This algorithm computes rectangle

features by using an intermediate representation of the image which is called the integral image. The integral image at location  $(x, y)$  is sum of the pixels above and to the left of  $(x, y)$ .

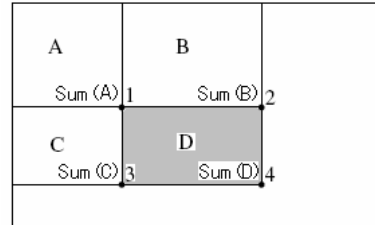


Figure 4. Integral image

The sum all grey pixels value (Figure 4) of D is Sum (D) + Sum (B) + Sum(C) – Sum (A).

For face detection problem, eyes region much darker than skin region, so we may simply apply a filter to remove those region without such property. We use an 18 \* 18 mask filter to estimate the face and eyes region in the image (Figure 5), with the following rules:

$$\text{Sum (B)} > \text{Sum (A) and}$$

$$\text{Sum (B)} > \text{Sum(C).}$$

$$\text{Sum (D+E+F)} > \text{Sum (A+B+C)}$$

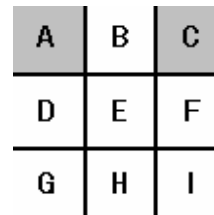


Figure 5. Filter for face detection

Using this method we can simply remove about 97% of the non-face regions in the image. In Figure 6, the green rectangle shows possible face locations.

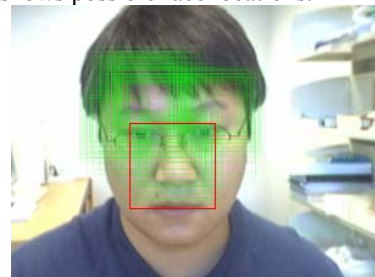


Figure 6. Possible face locations

### 3.4 Support Vector Machine classification

The training database for face detection contains 4000 jpeg images, including 2000 face and 2000 non-face images. Each image is 18x18 pixels. Figure 6 show two samples of database.

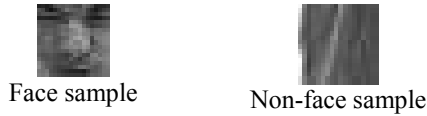


Figure 7. Training samples for face detection

Training database uses LIBSVM package to find support vectors, kernel function is using 'radial basis function':

$$\Phi = \exp(-\gamma|x_i - x_j|^2)$$

Decision function is as follows:

$$f_j(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^j \beta_i k(\mathbf{x}, \mathbf{z}_i) + b_j \right).$$

Figure 8 presents the best location of the detected face.



Figure 8. Face detection result SVM.

### 3.5 Eye Detection using SVM

For finding the eye location in the image, we applied a similar procedure which we used for face detection, within the detected face region. The eye detector uses a dataset of 2000 images for training, including 1000 eye images and 1000 non-eye images of size 15x15 pixels (Figure 9).

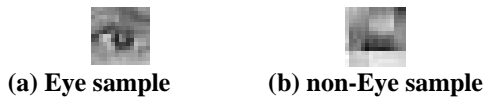


Figure 9. Eye sample

The iris in the eye region is normally darker than the rest of the image. Therefore we used the filter in Figure 10 for approximating the position of the eye pupil.

A	B	C
D	E	F
G	H	I

Figure 10. Filter for eye detection

Eye region has following property:

$$\begin{aligned} \text{Sum}(A + B + C) &> \text{Sum}(D + E + F) \\ \text{Sum}(G + H + I) &> \text{Sum}(D + E + F) \end{aligned}$$

In Figure 11, we showed the result of applying these two methods for locating the face and eyes.

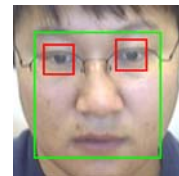


Figure 11. Result of eye detection and face detection

### 3.6 Normalizing the face image

Large-size images are less preferable for representing facial features, because consequently require a huge computer memory and computation time for processing. On the other hand, face region which is captured by the input device may be varying in size. Therefore it is necessary to normalize the image to a certain and preferably small size. We scale the face region to 200 pixels in width and 200 pixels in height. The approximate positions of the eyes are (50,150) and (150,150) (Figure 12).

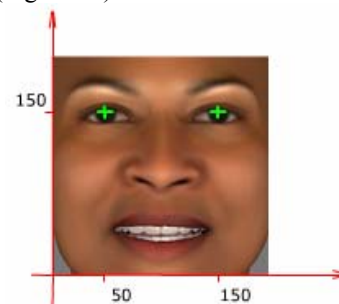


Figure 12. Positions of the eyes in the normalized image

### 3.7 Pre-processing

The purpose of the pre-processing is facial feature extraction. For training the Support Vector Machine, we concentrated on facial features which provide better results. On the other hand, the light intensity and direction may be different in the final application (Figure 13).



**Figure 13. Before and after lighting correction**

Therefore we applied the linear lighting correction using the Gaussian–Elimination with parameters a, b, c as follows.

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix} = I(x,y)$$

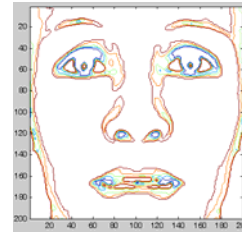
In addition, histogram equalization can improve image contrast. Instead of mapping the histogram from the whole image, we applied Contrast-Limited Adaptive Histogram Equalization (CLAHE) for contrast enhancement. Original image is divided into small regions (tiles), and then CLAHE applies on each region. The boundaries for each tile are combined using bilinear interpolation.

Our observation from the empirical results shows that a block of size 16x16 for CLAHE and a real scalar equal to 0.02 for contrast enhancement limit, are the best parameters for a face image of 200x200 pixels (Figure 14).



**Figure 14. Pre-processing**

In the next step, the eight-level polygon approximation algorithm is used on the image together with morphological operators to smooth the facial curves (Figure 15).



**Figure 15. The final result of pre-processing**

### 3.8 Statistical Modeling for Facial Expression

Support Vector Machine is a good tool to find the maximum margin between the different objects. We modeled the facial expression in statistical regression with the least squares regression algorithm.

Lets matrix X denotes our facial expression database which generated by the pre-processing. Each row of X represents a training sample, and Y is a column vector denotes the target six emotions. For example 1 denotes happiness face and etc. g(X) denotes the model which we are looking for. We can measure squared difference between g(X) and Y using:

$$\mathbb{F}(X,Y) = (g(X) - Y)^2, \text{ and } g(X) = X'w$$

Where w is called the weight vector.

$$\|\zeta\| = \langle y - Xw, y - Xw \rangle$$

Therefore to calculate w, we set the derivative of the above equation equal to 0, and we will have:

$$X'Xw + \lambda w = (X'X + \lambda I)w = X'y$$

Primal solution:

$$w = (X'X + \lambda I)^{-1}X'y$$

The regression function is:

$$g(x) = x'w = x'(X'X + \lambda I)^{-1}X'y$$

Dual solution:

$$X'Xw + \lambda w = X'y, \text{ Implies}$$

$$w = \frac{1}{\lambda} (X'y - X'Xw) = X' \frac{1}{\lambda} (y - Xw) = X'\alpha$$

and

$$\alpha = (XX' + \lambda I)^{-1}y$$

Regression function

$$g(x) = \sum_{i=1}^m \alpha_i \langle x, x_i \rangle$$

X is our training sample in equation, and Y is output. By finding  $\alpha$ , we can model the emotion. The calculation only involves inner product. Therefore the whole process can be done in real time. We applied principle component Analysis to reduce the dimensions of this matrix.

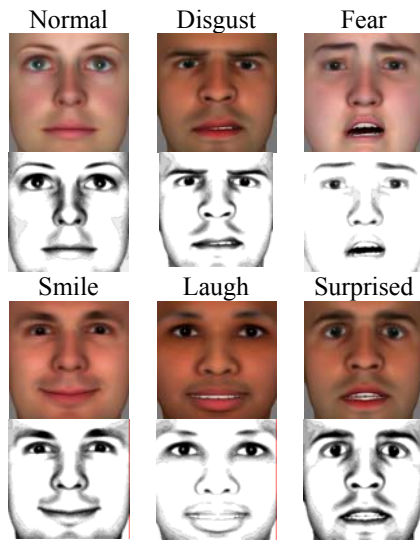


Figure 16. Feature extraction from the training images

### 3.8 Regression analysis result

We tested different kernel model on the image database, and the results presented in Table 1.

Table 1. The result of applying different kernel models

	Linear Kernel Model	Polynomial Kernel Model	RBF Kernel Model
Normal	75%	89%	89%
Disgust	65%	75%	85%
Fear	68%	83%	86%
Smile	73%	87%	93%
Laugh	82%	92%	96%
Surprised	87%	93%	94%

## 4 Conclusion and future work

In this paper we presented a real-time system for face detection and facial expression analysis. This system includes a pre-processor which locates face and eyes within the image and does enhancements. Facial expression analysis is done using a Support Vector Machine. For training the SVM we used a database of images representing different facial expressions of the six basic emotions. This approach requires less computation power in comparison to alternative solutions, and can be applied for real-time applications. Current prototype of the system, analyses a single frame as input. To increase the accuracy, in recognizing emotions from facial expressions, we are exploring a higher dimension which is the sequence of detections. We believe the extra dimension, provides more information for locating the maximized variation for different expressions.

## References

- [1] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, pp. 137-154, 2004.
- [2] V. N. Vapnik, "The nature of statistical learning theory," New York, NY, 1995.
- [3] V. N. Vapnik, "Statistical learning theory," New York: NY, 1998.
- [4] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge, UK: Cambridge University Press, 2000.
- [5] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," in *Data Mining and Knowledge Discovery*, vol. 2. Boston: Kluwer Academic Publisher, 1998, pp. 121-167.
- [6] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," 2001.