International Journal of Scientific & Engineering Research, Volume 7, Issue 2, February-2016
ISSN 2229-5518

Facial Expression Detection Using FACS

Vinaya Tinguria , Sampada Borukar[y], Aarti Awathare[z],Prof. K.T.V. Talele[x] Sardar Patel Institute of Technology, Mumbai-400058, India, vinaya.t20@gmail.com
[y]Sardar Patel Institute of Technology, Mumbai-400058, India, sampadaborukar@gmail.com [z]Sardar Patel Institute of Technology, Mumbai-400058, India, awathareaarti@gmail.com [x]Sardar Patel Institute of Technology, Mumbai-400058, India, talelesir@gmail.com

Abstract—Human emotions and intentions are very often communicated by changes in one or a few discrete facial features and cannot be classified into a small set of prototypic expressions. Many a times people communicate emotions that might otherwise be very difficult to communicate through their expressions. Thus understanding these nuances would definitely help in maintaining a peaceful co-existence. In this project, we propose to develop a system for detecting the basic human expressions using a Facial Action Coding System (FACS). We propose to develop an Automatic Facial Action Recognition System to analyse facial expressions based on both permanent facial features (brow, eyes, mouth) and transient facial features (deepening of facial furrows). The system will recognize the fine changes in facial expression into action units (AUs) of the Facial Action Coding System(FACS). We are using more than one techniques namely: Contour Matching, a rule based method, Support Vector Machine (SVM) and Local Binary Pattern (LBP) to increase the overall efficiency of the code.

Index Terms—Contour Matching, Local Binary Pattern, Support Vector Machine, Action Units, Facial Action Coding System

I. INTRODUCTION

Automatic recognition of facial gestures (i.e., facial muscle activity) is rapidly becoming an area of intense interest in the research field of machine vision. FACIAL expressions play a significant role in our social and emotional lives. They are visually observable, conversational, and interactive signals that clarify our current focus of attention and regulate our interactions with the environment and other persons in our vicinity [22]. They are our direct and naturally preeminent means of communicating emotions [12], [22]. Therefore, automated analyzers of facial expressions seem to have a natural place in various vision systems, including automated tools for behavioral research, lip reading, bimodal speech processing, video conferencing, face/visual speech synthesis, affective computing, and perceptual man-machine interfaces. It is this wide range of principle driving applications that has lent a special impetus to the research problem of automatic facial expression analysis and produced a surge of interest in this research topic.

Facial Action Coding System:

From several methods for recognition of facial gestures, the facial action coding system (FACS) is the best known and most commonly used in psychological research. It is a system designed for human observers to describe changes in the facial expression in terms of visually observable activations of facial muscles. Ekman and Friesen first developed the Facial Action Coding System(FACS) for describing and identifying facial expression using Action Units(AUs). 30 AUs out of the 40 that they defined, were related to the contractions of specific facial muscles: 12 for upper face, and 18 for lower face. Action Units can occur individually or in combinations. When AUs occur in combination they may be additive, in which the combination does not change the appearance of the constituent AUs, or nonadditive, in which the appearance of the constituents does change. The number of basic action units might seem small, but more than 7000 different combinations have been observed.FACS provides the descriptive power necessary to describe the details of facial expression. Commonly occurring AUs and their

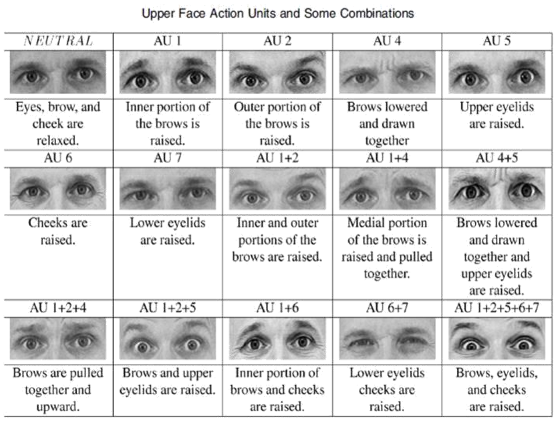


Fig. 1: Lower face action units and few combina-tions

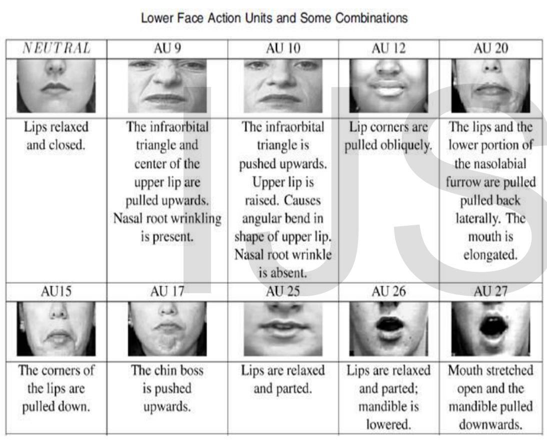


Fig. 2: Upper face action units and few combina-tions

various combinations are shown in Figures 1 and 2. An example of additive AU can be AU 4+5 where AU4 (brows lowered and drawn together) and AU5 (Upper eyelids raised) both can be observed individually. In contrast, an example of the non additive effect appears in AU 1+4. AU 4 appears differently depending on whether it occurs alone or in combinations. When AU4 occurs alone, the brows are drawn together and lowered. In AU 1+4, the brows are drawn together but are raised due to action of AU1.

Methods of Implementation:

1) Contour Matching:
   Contour Matching is performed to identify if the expression in the image obtained is Neu-tral. Contour matching is performed by first calculating the contours of a given image and that of a predefined neutral image. Following this, the contour shapes are matched, and the degree to which they match indicates if the expression is neutral.
2) Support Vector Machine:
   Support Vector Machine is a popular tech-nique for classication . SVM performs an implicit mapping of data into a higher dimen-sional feature space, where linear algebra and geometry can be used to separate data that is only separable with nonlinear rules in the input space.
3) Rule Based technique:
   In this technique, we classify the expressions based on distance vector. Firstly, the image is split into various parts: eyes, nose, cheeks, lips etc. Then, by using various rules for classifying each expression, the most accurate expression is predicted.
4) Local Binary Pattern:
   LBP is used to extract the local texture infor-mation of grayscale image. LBP can extract the local texture feature of facial expression image. LBP helps to extract the nuances of changes among different kinds of facial ex-pression, but ignores the global information of the whole facial expression image.

II. BASIC BLOCK DIAGRAM

Circuit Description:

Preprocessing:

The preprocessing basically consists of resizing the image. All the images which are used as input are converted to a size of 20X20.

Face Detection:

Face detection is a computer technology that determines the locations and sizes of

Fig. 3: Block Diagram

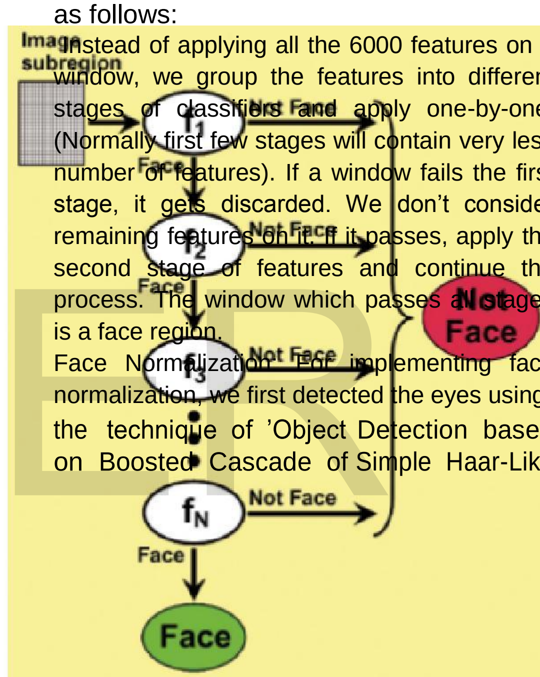
human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies. It is implemented using Object Detection Technique based on Boosted Cascade of Simple Haar-Like features developed by Paul Viola and Michael Jones. The algorithm uses three Haar-like features which are input to basic classifiers. These features are:

1) Edge features

2) Line features

3) Center-surround features

The cascade of classifiers is carried out as follows:



Instead of applying all the 6000 features on a window, we group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, it gets discarded. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

Face Normalization: For implementing face normalization, we first detected the eyes using the technique of 'Object Detection based on Boosted Cascade of Simple Haar-Like

Fig. 4: stages followed for face detection

features' developed by Paul Viola and Michael Jones in a similar fashion as was done for face detection.

Face Normalization is implemented as given in fig. 5:

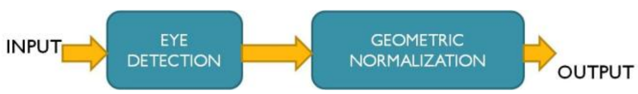


Fig. 5: Face Normalization

Geometric Normalization in turn is carried out as shown in Fig. 6:

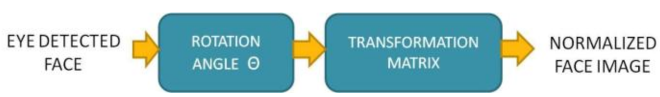


Fig. 6: Geometric Normalization

For geometric normalization, we first calculate the angle through which the face is to be rotated. This is done by finding the angle by which the eyes are rotated. This angle can be found out by using the formula:

$$= \tan^{1} dy=dx \quad (1)$$

where dy and dx are the distances as shown in Fig. 7.

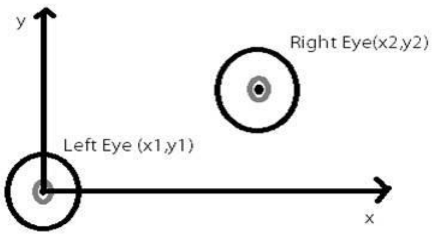


Fig. 7

The transformation matrix that we used to obtain the normalized images is:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_c(1-\cos\theta) + yc\sin\theta \\ \sin\theta & \cos\theta & y_c(1-\cos\theta) - xc\sin\theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where

x_C= Width of the input image

y_C= Width of the input image x;
y= Width of the input image x^0 ;
y^0 = Width of the input image
The input to the Face detection and normaliza-tion module is shown in Fig. 8 and the output is shown in Fig. 9

Feature Extraction:

Contractions of the various facial muscles change the appearance of permanent and tran-sient facial features. Permanent facial features are facial components such as eyebrows, eyes, and mouth. Their shape and location can alter immensely with expressions (e.g., pursed lips versus delighted smile). Transient facial fea-tures are any facial lines that did not become

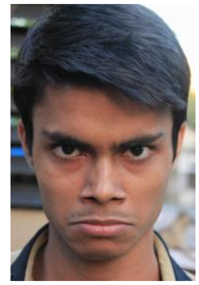


Fig. 8: Input image to face detection/normalization module

Fig. 9: Output image from face detec-tion/normalization module

permanent with age but appear with expres-sions. To reason about shown facial expression that produced it, we first detected facial fea-tures and their current appearance.
We have used two different approaches for facial feature extraction namely, Canny Edge Detection and Local Binary Pattern. Both of these techniques are explained in detail below.

1) Canny Edge Detection:
   The Canny Edge Detector was developed by John F. Canny in 1986. This algorithm aims to satisfy three main criteria:

   a) Low error rate: Which means a good detection of only existent edges

   b) Good localization: The distance between edge pixels detected and real edge pixels need to be minimized.

c) Minimal response: Only one response per edge.

The input and output of canny edge de-tection for a facial feature is shown in Fig. and Fig.

Fig. 10: Input for canny edge detection

Fig. 11: Output of canny edge detection

The algorithm for Canny Edge detection is as follows:

a) Step 1: Filter out any noise. The Gaussian filter is used for this purpose.

b) Step 2: Find the intensity gradient of the image. For this, we follow a procedure analogous to Sobel:

a. Apply a pair of convolution masks (in x and y directions)

$$G_x = \begin{bmatrix} 1 & 0 & +1 \\ 2 & 0 & +2 \\ 1 & 0 & +1 \end{bmatrix} \quad (2)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \quad (3)$$

b. Find the gradient strength and di-rection with:

$$G = \sqrt{G_x^2 + G_y^2} \quad (4)$$

$$= \arctan G_y = G_x \quad (5)$$

The direction is rounded to one of the four possible angles(namely 0, 45, 90 or 135).

c) Non- maximum suppression is applied. This removes pixels that are not considered to be part of an edge. Hence, only thin lines (candidate edges) will remain.

d) Hysteresis: The final step. Canny uses two thresholds (upper and lower):

a. If a pixel gradient is higher than the upper threshold, the pixel is accepted as an edge.

b. If a pixel gradient value is below the lower threshold, then it is rejected.

c. If the pixel gradient is between the two thresholds, then it will be accepted only if it is connected to a pixel that is above the upper threshold.

2) Local Binary Pattern:

supposing $g_c(x_c; y_c)$ is any pixel within a local area of a face image, g_c as the center of a 33 window and the other eight points are $g_0; ::::; g_7$.Define the local area texture as $T_{LBP} = t(g_c; g_0::::g_7)$ and carry on binary processing for the other eight pixels within the window using the threshold, here set the gray value of cen-ter pixel in the window as the threshold.

$$T_{LBP} = t(s(g_0 \quad (g_c); :::::s(g_{(7)} \quad g_c)) \quad (6)$$

where

$$s(x) = 1; x > 0 \quad (7)$$

$$s(x) = 0; x \quad 0 \qquad (8)$$

Read out an 8-bit binary number in the clockwise direction as an eigenvalue of the central pixel. Convert the binary num-ber into a decimal number by the fol-lowing formula for each symbol function. Then, LBP code which is described the spatial structure of local image texture feature is

$$T_{LBP}(x_c; y_c) = \sum_{u=0}^{7} s(g_u \quad g_c)2^u \quad (9)$$

Then the texture feature of image can be described by counting the facial expres-sion image histogram.

Action Unit Detection:

This is the final step for identifying the Action Units. For realizing this step, we first classified the various action units for different expres-sions. A huge range of action units is divided by manual observation into 7 classes, namely, neutral, happy, unhappy, surprised, disgust, anger and fear. Our final aim is to identify the Action Units from various images given as the input and thus identify the expression of the individual. For this, we have used two different approaches. They are: Support Vector Machine (SVM) and a rule based method.

1) Support Vector Machine:

If a training set is given

$$T = (x^i; y^i); i = (1; ::::; l)(10)$$

the new test data x is classified by the following function:

where $_i$ are Lagrange multipliers of a dual optimization problem, and $K(x^i; x)$ is a kernel function given a nonlinear mapping that embeds input data into feature space,kernels have the form of

$$K(x^i; y^i) = ((x_i) \quad (x_j)) \quad (11)$$

SVM finds a linear separating hyperplane with the maximal margin to separate the training data in feature space. b is the pa-rameter of the optimal hyperplane.SVM allows domain-specific selection of the kernel function. Though new kernels are being proposed, the most frequently used kernel function are the linear, polyno-mial, and RBF kernels. SVM makes bi-nary decisions. Multi-class classification here is accomplished by a cascade of binary classifiers together with a voting scheme.

2) Rule Based Method:

In this method, we will be deciding vari-ous rules based on the feature properties that are a characteristic of each and ev-ery expression. These rules will basically be dependent on distance vectors. To identify these characteristics, the output from the canny edge detection must go through a series of preprocessing stages from which the distance vectors can be calculated. From these distance vectors, we can identify the action units present in the expression. On the basis of these action units, we can then in turn identify the expression from their combinations.

III. RESULTS

We have trained the Support Vector Machine by giving 16 positive and 20 negative images from our Indian database and 41 positive and 63 negative from our Taiwanese database. After this, we tested the machine for Indian database by giving 25 nega-tive and 14 positive images, after which we got an accuracy of 69%. Similarly, we tested the machine for the Taiwanese database by giving 41 negative and 17 positive expressions. For this database, we obtained an accuracy of 75.8%.

IV. CONCLUSION

In this paper, we proposed a novel, automated method for detecting facial actions based upon changes in contours of facial components and/or face profile contour detected in a static frontal-view face image. We have approached the problem at hand by two different methods, that is, by a rule based method and by Support Vector Machine to make sure that high efficiency is achieved.

References

[1] www.opencv.com
[2] Y. Tian, T. Kanade, and J.F. Cohn, Facial Expression Analy-sis,"Recognizing Action Units for Facial Expression Analy-sis" in IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, COL. 23, FEBRUARY 2001
[3] B. Fasel and J. Luettin, Automatic facial expression analysis: a survey, Pattern Recognition, vol. 36, pp. 259275, 2003
[4] Caifang Shan,Shaogang Gong,and Peter W.McOwen, "Fa-cial Expression Recognition Using Local Binary Pattern"
[5] LUO yuan,WU Cai-ming,ZHANG Yi,"Facial expression feature extraction using hybrid PCA and LBP"
[6] www.bogotobogo.com
[7] www.stackoverflow.com
[8] Maja Pantic, Member, IEEE, and Leon J. M. Rothkrantz, "Facial Action Recognition for Facial Expression Analysis From Static Face Images" in IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICSPART B: CY-BERNETICS, VOL. 34, NO. 3, JUNE 2004
[9] S.P.Khandait, Dr. R.C.Thool, P.D.Khandait, "Automatic Fa-cial Feature Extraction and Expression Recognition based on Neural Network" in (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 2, No.1, January 2011
[10] James J. Lien, Jeffrey F. Cohn, Takeo Kanade, Ching-Chung Li, "Automated Facial Expression Recognition Based on FACS Action Units" in Proceedings of FG98, April 14-16, 1998 in Nara, Japan.