IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 9, Issue 5, Ver. V (Sep - Oct. 2014), PP 40-45
www.iosrjournals.org

Person identification through emotions using violas Jones algorithm

Dolly Reney[1], Dr.Neeta Tripathi[2]
[1](*Electronics and telecommunication, C. V. Raman* College of Engineering */ CVRU,India)*
[2](*Electronics and telecommunication, Shri Shankaracharya Group of Institution/ CSVTU,India)*

***Abstract**: This paper presents an algorithm to identify a person through his emotional state using neural network which are completed in two phase, first is database creation in which we take user image and an audio file as an input, and add the features to the database. This is done by the code named createDatabase.m in MATLAB. Second is evaluation of Database where the system evaluates the parameters and compares the same with the database and the best matching person and emotion are fetched and displayed at the output using KNN classifier.*

***Keyword:** MATLAB, KNN CLASSIFIER,CREATEDATABASE.M*

I. Introduction

Over last few years attempt has been made to recognize person through speech recognition, similarly attempt has also been made through face recognition. Very little work has been done to recognize the person through his emotional status using these two biometrics parameter. A problem of personal verification and identification is an actively growing area of research. Face, voice, lip movements, hand geometry, odor, gait, iris, retina, fingerprint are the most commonly used authentication methods. All of these psychological and behavioral characteristics of a person are called biometrics. The driving force of the progress in this field is due to the growing role of the internet and electronic transfers in modern society.

This research is mainly interested in face detection issue and speaker's emotional state. The task of detecting a face in an image is not an easy problem because many difficulties arise and must be taken into account. For starters, faces will generally occupy very little area in most images and they are usually located arbitrarily. This means that face detection algorithms must search over all areas of any given image to be successful. Some methods do a preliminary scan over the image in an attempt to find the areas of interest early on. Furthermore, algorithms must take into account the fact that faces vary greatly in many aspects such as size, complexion and how they are accessorized. Further, faces in images can look very different depending on orientation and pose. For example, a face seen from a profile perspective will have a completely different set of defining characteristics than a face seen head on. There is currently no simple solution to address these issues, but one technique that is having some success is to train a detector using a set of images that spans many of these variations. Finally, another aspect that introduces a lot of difficulties is occlusions. Occlusions usually happen very unpredictably and are thus very hard to address.One of the method for face detection is viola Jones algorithm, this describe how one can use machine-learning techniques to construct sets of meaningful features that will detect faces. The Viola-Jones method is quite fast already, but if we're able to make assumptions about the images that are being processed, optimizations can be made to further speed up detection. Similarly for emotion recognition KNN classifier is used. Emotions arise suddenly in response to a particular event and remain for seconds or minutes, while moods are uncertain in nature and remain for hours or days. It has been suggested that there are four most common basic emotions these are "happiness, sorrow, anger and fear". Instance-based classifiers such as the kNN classifier operate on the premises that classification of unknown instances can be done by relating the unknown to the known according to some distance/similarity function. The intuition is that two instances far apart in the instance space defined by the appropriate distance function are less likely than two closely situated instances to belong to the same class.

The rest of the paper is organized as follows. Section II gives algorithm of the reported approaches. Section III describes implementation. In Section IV, gives result and discussion. Section V, gives conclusions and future work

II. Algorithm

2.2.1 Features

The problem of face detection can be viewed as a problem of binary classification of image frame as either containing or not containing a face. In order to be able to learn such a classification model, we first need to describe an image in terms of features, which would be good indicators of face presence or absence on a given image.

For their face detection framework Viola and Jones [3] decided to use simple features based on pixel intensities rather than to use pixels directly. They motivated this choice by two main factors:
1. Features can encode ad-hoc domain knowledge, which otherwise would be difficult to learn using limited training data.
2. Features-based system operates must faster than a pixel-based system.
The author's defined three kinds of Haar-like rectangle features (see Figure1):
1. Two-rectangle feature was defined as a difference between the sums of the pixels within two adjacent regions (vertical or horizontal),
2. Three-rectangle feature was defined as a difference between two outside rectangles and an inner rectangle between then,
3. Four-rectangle feature was defined as a difference between diagonal pairs of rectangles.

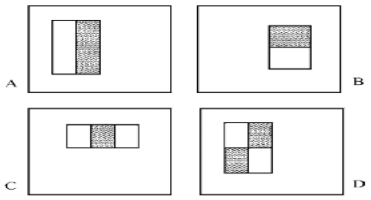

Figure 1: Rectangle features example:
(A) and (B) show two-rectangle
features,(C) shows three-rectangle
feature, and (D) shows four-rectangle

To enable an efficient computation of rectangle features, the authors presented an intermediate representation of an image, called an integral image.The value of integral image at location x; y is defined by the sum of the pixels from the original image above and to the left of location x; y inclusively (see Figure 2).
Formally,

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y'), \qquad ($$

where ii(x; y) is an integral image and i(x; y) is an original image.

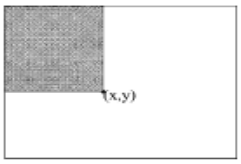

Figure 2: Integral image

Maintaining a cumulative row sum at each location x; y, the integral image can be computed in a single pass over the original image. Once it is computed, rectangle features can be calculated using only a few accesses to
it (see Figure 3):

- Two-rectangle features require 6 array references,
- Three-rectangle features require 8 array references, and
- Four-rectangle features require 9 array references.

The authors defined the base resolution of the detector to be 24x24. In other words, every image frame should be divided into 24x24 sub-windows, and features are extracted at all possible locations and scales for each such sub-

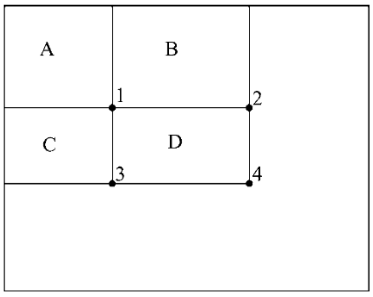

Figure 3: Calculation example. The sum of the pixels within rectangle D can be computed as 4 + 1 - (2 + 3), where 1-4 are values of the integral image.

Window. This results in an exhaustive set of rectangle features which counts more than 160,000 features for a single sub-window.

2.2.2 Learning Classification Functions

The complete set of features is quite large - 160,000 features per a single 24x24 sub-window. Though computing a single feature can be done with only a few simple operations, evaluating the entire set of features is still extremely expensive, and cannot be performed by a real-time application.

Viola and Jones assumed [3] that a very small number of the extracted features can be used to form an effective classifier for face detection. Thus, the main challenge was to find these distinctive features. They decided to use AdaBoost learning algorithm as a feature selection mechanism. In its original form, AdaBoost is used to improve classification results of a learning algorithm by combining a collection of weak classifiers to form a strong classi_er. The algorithm starts with equal weights for all examples.In each round, the weight are updated so that the misclassified examples receive more weight.By drawing an analogy between weak classifiers and features, Viola and Jones decided to use AdaBoost algorithm for aggressive selection of a small number of good features, which nevertheless have significant variety.

Practically, the weak learning algorithm was restricted to the set of classification functions, which of each was dependent on a single feature. A weak classifier h(x; f; p; θ) was then defined for a sample x (i.e. 24x24 sub-window) by a feature f, a threshold θ, and a polarity p indicating the direction of the inequality:

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta, \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

The key advantage of the AdaBoost over its competitors is the speed of learning. For each feature, the examples are sorted based on a feature value. The optimal threshold for that feature can be then computed in a single pass over this sorted list. To achieve this, we need to maintain four sums for each element in the sorted list:

- T+ - the total sum of positive example weights,
- T- - the total sum of negative example weights,
- S+ - the sum of positive weights below the current example, and
- S- - the sum of negative weights below the current example.

The error for a threshold which splits the range between current and previous example in the sorted list is the minimum of the error of labeling all the examples below the current example as negative (and all other examples as positive) versus the error of labeling all the examples below the current example as positive (and all other examples as negative). Formally,

$$e = \min\left(S^{+} + \left(T^{-} - S^{-}\right), S^{-} + \left(T^{+} - S^{+}\right)\right)$$

In their paper [8], Viola and Jones show that a strong classifier constructed from 200 features yields reasonable results - given a detection rate of 95%, false positive rate of 1 to 14,084 was achieved on a testing dataset. These results are promising. However, authors realized that for a face detector to be practical for real applications, the false positive rate must be closer to 1 in 1,000,000. The straight forward technique to improve detection performance would be to add features to the classifier. This, unfortunately, would lead to increasing computation time and thus would turn the classifier into inappropriate for real-time applications.

2.2.3 Detectors Cascade

There is a natural trade-of between classifier performance in terms of detection rates and its complexity, i.e. an amount of time required to compute the classification result. Viola and Jones [3], however, were looking for a method to speed up performance without compromising quality. As a result, they came up with an idea of detectors cascade (see Figure 4). Each sub-window is processed by a series of detectors, called cascade, in the following way. Classifiers are combined sequentially in the order of their complexity, from the simplest to the most complex. The processing of a sub-window starts, then, from a simple classifier, which was trained to reject most of negative (non-face) frames, while keeping almost all positive (face) frames. A sub-window proceeds to the following, more complex, and classifier only if it was classified as positive at the preceding stage. If any one of classifiers in a cascade rejects a frame, it is thrown away, and a system proceeds to the next sub-window. If a sub-window is classified as positive by all the classifiers in the cascade, it is declared as containing a face.

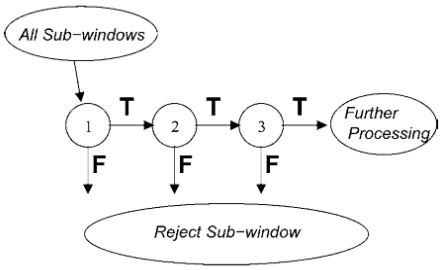

Figure 4: Detectors cascade

Since most of sub-windows do not contain faces, only a small number of them are processed by all the classifiers in a cascade. The majority of sub-windows are rejected during first few stages, involving simple classifiers solely, which results can be evaluated very rapidly. Thereby, the overall detection performance meets the requirements of real-time processing, and yet it yields high detection rates, comparable with alternative, much slower, techniques.

III. Implementation

This section describes an implementation of the system for face detection and emotion recognition algorithm which is carried sequentially as:-

A. **Database creation**:- It proceeds as follows

Step 1. Pick the image using the command

[file path] = uigetfile('*.*','Pick the Face Image');

Step 2. Image is read using command given below

img = imread([path file]);

Step 3. Convert

img = rgb2gray(img);

RGB2GRAY - Convert RGB image or color map to grayscale. RGB2GRAY converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

Step 4. img(double img)-

Converts the intensity image to double precision, rescaling the data if necessary. If the input image is of class double, the output image is identical.

Step 5. Then face detection algorithm is applied using FaceDetect ('haarcascade_frontalface_alt2.xml',img)

The code implements Viola-Jones adaboosted algorithm for face detection by providing a mex implementation of OpenCV's face detector. The function returns Nx4 matrix. In case no faces were detected, N=1 and all four entries are -1. Otherwise, N=number of faces in the image and the vector contains the x, y, width and height information of the face.

Step 6. We get the face and then we apply edge detection on the image.

Step 7. Finally we apply the properties of image using the command region props which Measure properties of image regions (blob analysis). REGIONPROPS does not accept a binary image as its first input. There are two common ways to convert a binary image to a label matrix:

1. L = bwlabel(BW);

2. L = double(BW)

Suppose that BW were a logical matrix containing these values:

1 1 0 0 0 0
1 1 0 0 0 0
0 0 0 0 0 0
0 0 0 0 1 1
0 0 0 0 1 1

The first method of forming a label matrix, L = bwlabel(BW), results in a label matrix containing two contiguous regions labeled by the integer values 1 and 2. The second method of forming a label matrix, L = double(BW), results in a label matrix containing one discontiguous region labeled by the integer value 1. Since each result is legitimately desirable in certain situations, REGIONPROPS does not accept binary images and convert them using either method. We should convert a binary image to a label matrix using one of these methods (or another method if appropriate) before calling REGIONPROPS.here we are using regionprops(bwlabel(face_edge) command

Step 8. After face detection we call for emotion menu which includes happy, Sad, Anger, Surprise and Normal.

Step 9. Now pick the person's voice sample

9.1 We find the Mel Matrix

9.2 And Get the Pitch, Frequency Components and Harmonic Components.

Finally image as well as sound is saved. and file is run to get output as original image and extracted image along with emotion.

B. Evaluation of database:

Once you have saved your database with enough entries (say 50 entries with varying emotions), then we must run this file to recognize emotions. This uses knnclassify to classify the input face, and find its emotions.

Process moves as:

1. Select the saved database
2. Input sound to the system
3. The system evaluates the parameters and compares the same with the database.
4. Now the best matching person and emotion are fetched and displayed at the output.

Programme of evaluate database is same as in create database, with the difference that here we proceed with the saved image of create database and uses knn classify to classify the input face, and find its emotions . The knn classifier is a nearest-neighbor classification object, where both distance metric ("nearest") and number of neighbors can be altered. The object classifies new observations using the predict method. The object contains the data used for training, so can compute resubstitution predictions. In our project Knn is used to detect 5 features, which are edge counts, areas, major axis, minor axis and eccentricities and mfcc to analyse sound and recognize it.

IV. Result:-

As our project is carried in two phase, our result has two phases, result of create database and evaluate database respectively

1. Result of create database programme

A. capturing image in different emotion as shown:-

Fig.5. person in different emotions

B. Input emotions and name of person then color map is converted to grayscale and we get extracted image as follows:-

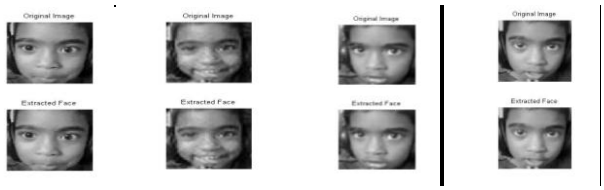

Fig.6. original and extracted images in different emotions

C. Result of Evaluate database programme:- Once we have saved our above images in extracted form is saved in folder and we proceed to evaluate database programme,and when these saved database is fetched with sound with the help of KNN classifier a menu is displayed showing the person name and his emotion as shown in fig.

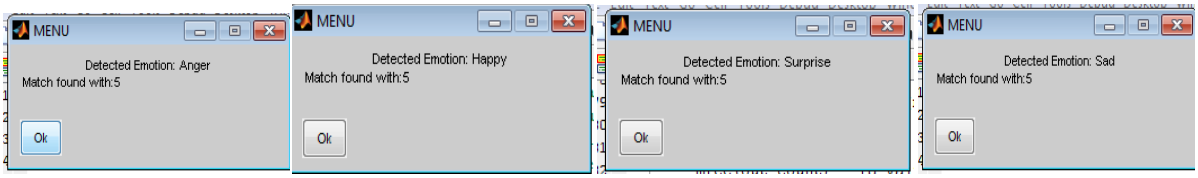

Fig.7 Menu displaying emotion with perfect match

V. Conclusion:-

In this paper image of various person along with emotion database is created. Initially all the images are captured in JPEG.format and sound are cut in .mp3 format and then converted into(.wav) file. From experimentation and result it is proved that person can be identified through his emotion which will be beneficial in medical and security applications. It is observed that we can add more emotion for more perfection and also our programming can be used for any language.

References

Journal Papers:

[1] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi, Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques, JOURNAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010.

[2] Freund, Yoav and Schapire, Robert E. "A decision-theoretic generalization of online learning and an application to boosting", Journal of Computer and System Sciences, No 55, 1997

[3] Paul Viola and Michael Jones. Robust real-time face detection. International Journal of Computer Vision, 57:137{154, 2004.

Proceedings Papers:

[4] Crow, F, "Summed-area tables for texture mapping", in Proceedings of SIGGRAPH,18(3):207-212, 1984

[5] Lien hart, R. and Maydt, J., "An extended set of Haar-like features for rapid object Detection", ICIP02, pp. I: 900-903, 2002

[6] Papageorgiou et al, "A General Framework for Object Detection", International Conference on Computer Vision, 1998

[7] Rowley et al. "Neural network-based face detection" IEEE Patt. Anal. Mach. Intell.**20:22-38**

[8] Theo Ephraim, Tristan Himmelman, and Kaleem Siddiqi. Real-time viola-jones face detection in a web browser. In Proceedings of the 2009 Canadian Conference on Computer and Robot Vision, CRV '09, pages 321{328, Washington, DC, USA, 2009. IEEE Computer Society.

[9] M Jones and P Viola. Fast multi-view face detection. Mitsubishi Electric Research Lab TR2000396, (July), 2003.

[10] J. Kovac, P. Peer, and F. Solina. Human skin color clustering for face detection. In EUROCON 2003, volume 2, pages 144{148, 2003.