A Project Report On

Multiple Face Emotion Detection

Submitted by

Divija Nagaraju – 14IT112

Mukta Kulkarni – 14IT220

Pooja Soundalgekar – 14IT230

Yogitha A N – 14IT251

IV SEM B.Tech (IT)

in partial fulfillment for the award of the degree of

Bachelor of Technology
In

Information Technology
At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal.

April 2016

1. Problem Statement

To provide an application that can detect multiple faces of people, detect facial emotion and classify them as happy or sad. This application can be put to use in real life situations as meetings, seminars or conferences and predict feedback based on the emotion detected.

2. Objective

The project aims to present multiple face emotion detection and it’s implementation in real life situations like meetings and conferences. For this, the application captures a video in real time, detects all the human faces in the frame and classifies the face emotions as happy or not-happy (neutral/sad).The time for which the video captures the faces can be specified by the user. After the video is captured for the specified amount of time, the satisfaction of the meet can be analyzed by displaying the percentage of people who are happy taking both time and number into consideration.

3. Implementation

The project presents implementation of a program that uses a webcam to capture the video in real time. The openCV computer vision library has been used to preprocess the webcam image. Python packages ‘numpy’ and ‘csv’ are used to read the data set values. The dataset is trained on by a logistic regression algorithm to evaluate for the newly captured image through real time.

The application starts by allowing the user to set maximum capture time for the video. The openCV library is used to detect multiple faces and localize their mouth area.

The image is then vectorized, i.e. the image is resized such that the output is 28 X 10 pixels image containing person’s mouth and surrounding areas. The images are converted to grayscale and then the pixel data is stored as a vector of length 280.

A logistic regression program has been used that will take the image vector and determine whether that person was happy or not. The model is trained on the dataset using gradient descent. A set of neutral or sad images and a set of happy images are used to train the model. The input is the 280 length vector provided through the real time captured frame. A set of weights is applied to it to yield a scalar output. The proximity of the scalar to 0 or 1 determines what result the model gives.

The application captures frame for the specified time period and increments the ‘happycounter’ if the model yields result positive result else increments the ‘sadcounter’ for each detected face. The result is then determined by calculating the total happiness percentage.

4. Algorithmic Approach

4.1 Detection and cropping of face-

```
crop(self,area):
    self.crop = self.img[area[0][1]:area[0][1] + area[0][3], area[0][0]:area[0][0]+area[0][2]]
    return self.crop
cropface(self,x,y,w,h):
    self.cropface=cf[y: y + h, x: x + w]
```

4.2 Detection and cropping of mouth-

```
mouth_in_lower_face(mouth,face):
  if (mouth[0][1] > face[0][1] + face[0][3] * 3 / float(5)
    and mouth[0][1] + mouth[0][3] < face[0][1] + face[0][3]
    and abs((mouth[0][0] + mouth[0][2] / float(2))
      - (face[0][0] + face[0][2] / float(2))) < face[0][2] / float(10)):
    return True
  else:
    return False

findmouth
      Load the classifiers
      Run the Classifiers
      if detectedMouth:
            for mouth in detectedMouth:
                  if mouth_in_lower_face(mouth,maxFace):
                        filteredMouth.append(mouth)

      maxMouthSize = 0
      for mouth in filteredMouth:
            if mouth[0][3]* mouth[0][2] > maxMouthSize:
            maxMouthSize = mouth[0][3]* mouth[0][2]
            maxMouth = mouth
     return maxMouth
```

4.3 Analysis of each frame-

```
working
        Load training data
        Train the data with logistic regression
        input time
        faces = self.face.detectMultiScale()
        for (x, y, w, h) in faces
                cv.SaveImage("webcam.jpg", cv.fromarray(frame))

           print x,y,w,h
           self.img = cv.LoadImage("webcam.jpg")
           cropping=self.img[y-30: y + h+50, x-30: x + w+50]
           if(i==1):
                cv.SaveImage("crop1.jpg", cropping)
           else:
                cv.SaveImage("crop2.jpg", cropping)

              if(i==1):
                 self.img = cv.LoadImage("crop1.jpg")
              else:
                 self.img= cv.LoadImage("crop2.jpg")
              mouth = m.findmouth(self.img)
              if mouth != 2:
                 mouthimg = self.crop(mouth)
                 if(i==1):
                    cv.SaveImage("webcam-m1.jpg", mouthimg)
                else:
                    cv.SaveImage("webcam-m2.jpg", mouthimg)

                if(i==1):
                    result = lr.predict(self.vectorize('webcam-m1.jpg'))
                else:
                    result = lr.predict(self.vectorize('webcam-m2.jpg'))
                if result == 1:
                    print "face",i, ": You are smiling! :-) "
                    smile=smile+1
                else:
                    print "face",i, ":You are not smiling :-/ "
                    nosmile=nosmile+1
            else:
                print "face",i, ":Failed to detect mouth. our face is tilted."
            i=i+1
        cv2.imshow('preview', frame)
    if cv2.waitKey(5) != -1:
        break
```

5. Result

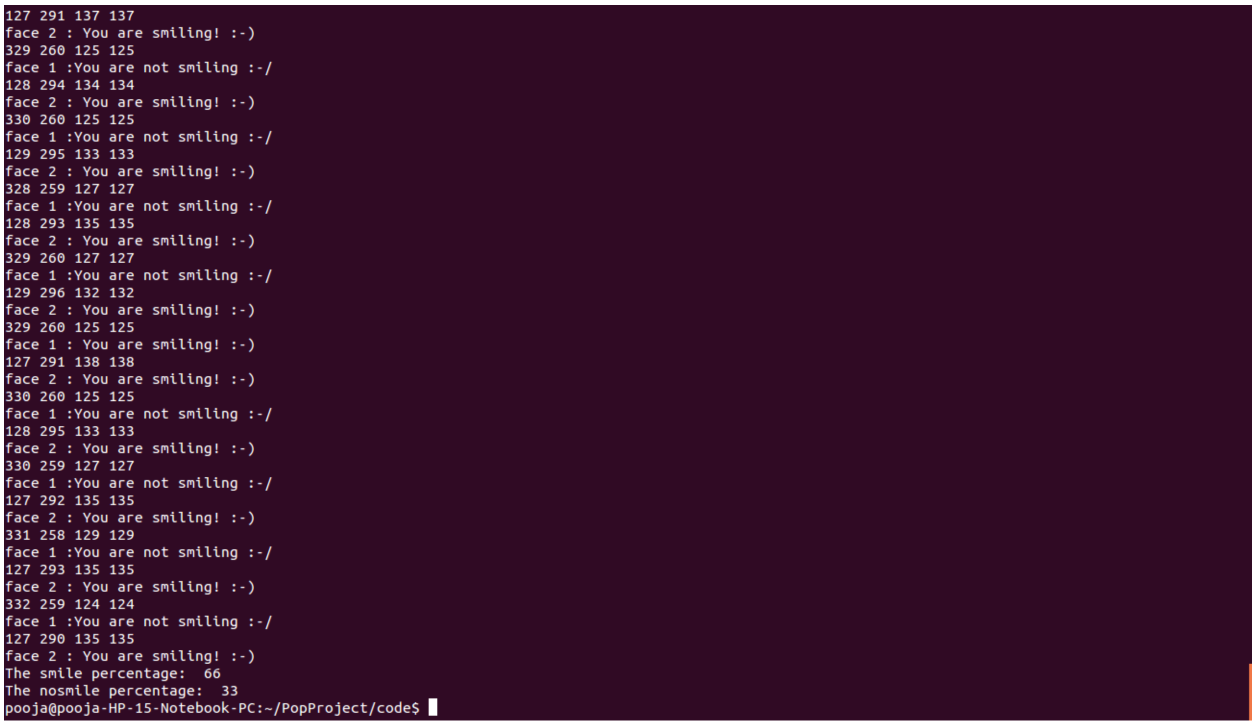
5.1 ScreenShots

5.1.1 Program Under Execution

Requirements:A linux system with cv/ cv2 and numpy installed.

Commands executed:

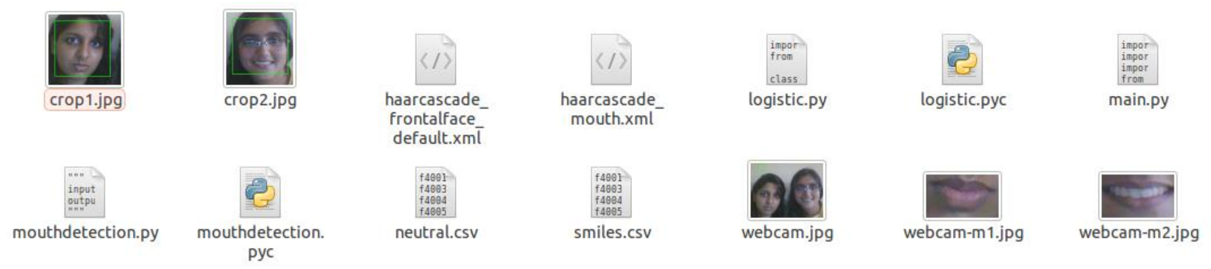
cd Pop_Project-->cd code-->python main1.py



```
127 291 137 137
face 2 : You are smiling! :-)
329 260 125 125
face 1 :You are not smiling :-/
128 294 134 134
face 2 : You are smiling! :-)
330 260 125 125
face 1 :You are not smiling :-/
129 295 133 133
face 2 : You are smiling! :-)
328 259 127 127
face 1 :You are not smiling :-/
128 293 135 135
face 2 : You are smiling! :-)
329 260 127 127
face 1 :You are not smiling :-/
129 296 132 132
face 2 : You are smiling! :-)
329 260 125 125
face 1 : You are smiling! :-)
127 291 138 138
face 2 : You are smiling! :-)
330 260 125 125
face 1 :You are not smiling :-/
128 295 133 133
face 2 : You are smiling! :-)
330 259 127 127
face 1 :You are not smiling :-/
127 292 135 135
face 2 : You are smiling! :-)
331 258 129 129
face 1 :You are not smiling :-/
127 293 135 135
face 2 : You are smiling! :-)
332 259 124 124
face 1 :You are not smiling :-/
127 290 135 135
face 2 : You are smiling! :-)
The smile percentage:  66
The nosmile percentage:  33
pooja@pooja-HP-15-Notebook-PC:~/PopProject/code$
```

5.1.2 Data Storage

The data being stored i.e the cropped images of the people attending the conference can be used for forming an attendance system.This will require combination with an appropriate face recognition system.



The results are as seen in the screenshots. The output consists of the number of faces which are detected in the real time video capture. The co-ordinates of each of the faces are also shown. The Emotion of each face is detected and classified as happy (smile) or neutral/sad (no smile).The final result is calculated based on this result for each face and the happiness percentage is calculated and displayed.

6. Conclusion

The application presented through the project can be used as a method of obtaining honest feedback. However the challenges posed in this process include low accuracy due to presence of multiple structural components, face orientation, imaging conditions and so on. The accuracy can be improved by using higher level learning algorithms to detect more facial points which can be used for multiple facial emotion recognition. Through emotion recognition a new are called Affect computing is coming into picture.It is a domain that focuses on user emotions when interacting with a computer.As the emotional state of a person may affect concentration, task solving decision making skills,the vision of this is to make systems able to recognize human emotions and influence them in order to enhance productivity and effectiveness of working with computers.