School of Computer Science and Engineering

VIT University

Vellore – 632014, India

Emotion-recognition

a project report submitted by

Sarthak Mehta

16BCE0961

Pratik Hotchandani

16BCE2097

Ronhit Neema

16BCE2099

under the supervision of

SELVAKUMAR K

Associate Professor (SCOPE)

Certificate

This is to certify that the project work titled '**Emotion-Recognition**' is a bona fide work, submitted by **Sarthak Mehta** (16BCE0961), **Pratik Hotchandani** (16BCE2097) and **Ronhit Neema** (16BCE2099), under the supervision of **Mr. Selvakumar K** The contents of this project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

____

Sarthak Mehta

____

Pratik Hotchandani

____

Ronhit Neema

Contents

Acknowledgement

We would like to express our gratitude to our professor, Mr. Selvakumar who gave us the golden opportunity to work on the topic ‘**Emotion Recognition**’, and also for his guidance and encouragement in carrying out this project work. This dissertation would not have been possible without the support of our professor. We would also like to thank and appreciate one and all who directly or indirectly have lent their helping hand in this venture.

Abstract

Human computer interaction has many significant fields of work and expression/ emotion detection is one of them. In order to detect a facial expression the system should analyse various variability of human faces like colour, posture, expression, orientation, lighting etc. Detecting facial features is a pre-requisite to emotion recognition. This is achieved by observing the parts if the face, like eyes, lips movement etc.

Emotion recognition is a technique used in software that allows a program to "read" the emotions on a human face using advanced image processing. Companies have been experimenting with combining sophisticated algorithms with image processing techniques that have emerged in the past ten years to understand more about what an image or a video of a person's face tells us about how he/she is feeling and not just that but also showing the probabilities of mixed emotions a face could has.

Keywords

Facial expression recognition; Fisher faces; Principal Component Analysis; Eigenfaces; Euclidean Distance

Introduction

Facial expressions are important cues for non-verbal communication among human beings. This is only possible because humans are able to recognize emotions quite accurately and efficiently. An automatic facial emotion recognition system is an important component in human machine interaction. Recognizing human emotion can have numerous applications in various contexts. While the most promising one's probably the man-machine interaction, patient monitoring, studying a suspect for anti-social motives etc. might be other useful areas for emotion recognition. With emotion recognition system the centre can analyse customer's reaction on seeing certain product or advertisement or upon receiving a particular piece of information or message. Based on the response whether they are happy or sad or disgusted, etc. the service centre can modify their approached. In a generalized form of a facial expression recognition system, an input sensing device such as a webcam obtained the input image from a subject and then it communicates with the computer. After detection of the facial area, representative feature from the emotionally expressive face image are extracted, it is then pre-processed and a classifier is used to classify them into one of the emotion classes such as anger, disgust, surprise, happy, neutral etc. There are several detection method as well as classifier algorithms that can be used in the detection and classification. A dynamic model of emotions is presented in this research based on a comprehensive eigen space based approach. Eigen space is a feature space that best encodes the variation in the eigen faces. The eigen faces maybe thought of as a set of feature space which characterise the over all variations among face images.

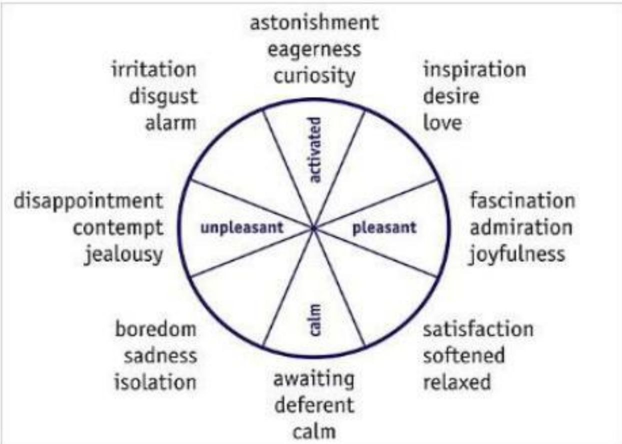
Now with the easy availability, it is possible for a person to implement neural networks or machine learning provided he/she has a prior knowledge of the topic. Human emotion recognition plays an important role in the interpersonal relationship. The automatic recognition of emotions has been an active research topic from early eras. Therefore, there are several advances made in this field. Emotions are reflected from speech, hand and gestures of the body and through facial expressions. Hence extracting and understanding of emotion has a high importance of the interaction between human and machine communication. With facial emotion detection, algorithms detect faces within a photo or video, and sense micro expressions by analysing the relationship between points on the face, based on curated databases compiled in ac ademic environments. Today we all use emoticons or as we call the emojis in our chats or texts, and they have been used to express emotion we can't express through texts. Incorporating that idea into emotion analysis techniques, we can make it more interesting.

Emotion recognition and analysis is being used by many technological giants like Apple Inc, Google, Amazon, Microsoft, Facebook etc in understanding their customer needs through the emotions and also the users can create their own emoticons using just their face expressions. The recent developments in this field have been in the field of marketing. Customer sentiments and emotions are targeted in order to make the brand reach the market. Many of the most successful marketing campaigns and initiatives are focused on emotions.

Related Works

1 **Taxonomy:**

Before starting to recognize emotions, we focus on all the emotions that can be recognized by humans all over the globe. Emotion theorists and psychologists have defined several models for emotion classification ranging from universally displayed basic emotions to culturally specific complex ones. Out of the various models in emotion research, there are two that have dominated facial expression research: Ekman's basic set of emotions and Russell's circumflex model of affect.



Ekman and Freisen in 1971 proposed six prototypical (basic) emotions - anger, disgust, fear, joy, sadness, and surprise - which are displayed universally among human beings and are recognized from human facial expressions

2. **Existing Work**:

Even though the emergence of this knowledge has been over a few decades, it still remains an area of research and welcomes new techniques and algorithms each time. Naturally, identification of one’s correct emotional state from the measurements of the physiological conditions is also difficult. More subjects excited with stimulus responsible for arousal of a specific emotion, have a manifestation for mixed emotions. Emotion recognition becomes more complex, when subjects arouse mixed emotions. The work by Ekman and Friesen in 1971 on facial expression recognition was based on extracting features from different regions of face, e.g. cheek, chin, and wrinkles. It reports a direct correlation of facial expression with the eyes, the eye-brows, and the mouth.

The facial features were extracted using different feature filters like gabor wavelet filter and/or discrete cosine transformation. Another idea which uses a completely different technique is the work of Hamit Soyel and Hasan Demiral also implemented the techniques of facial expression detection using 3D facial feature distances. The face was scanned and transformed to its 3D space, similar transform was done for expression training and prediction. Muid Mufti and Assia Khanam developed a fuzzy rule based emotion recognition technique using facial expression recognition. This approach was very different from the conventional techniques and was the first of its kind.

Proposed Method

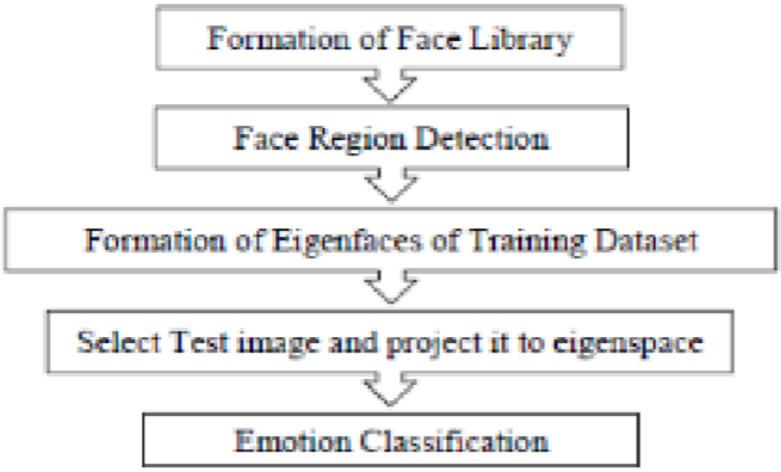


Fig 1: Proposed approach

1. Formation of face library:

It involves the preparation of dataset upon which the learning algorithm will work. We have used different datasets of faces available on internet example CK+ dataset. Data will be trained using fisher face classifiers to recognize different emotions.

2. Feature Selection and Extraction:

Various edge detectors like wavelet and wavelet packets, discrete cosine transformation and Gabor filters including shape/colour features. The face space is usually computed by a principal components analysis or linear discriminant analysis (Fisher's Linear Discriminant) of the face database. Both these analyses are classical methods for multivariate analysis. They can form eigenfaces or fisher faces.

Gabor Wavelet-based Features: Gabor wavelet filters are multi-scale and selective to specific directional changes in the image. They can therefore be used to obtain invariance to scale change and to investigate the effect of locally oriented image features. In addition, they achieve a certain amount of localized normalization for illumination.

3. Face Detection:

Commonly Used Techniques involve finding faces in images with controlled background which uses images with a plain mono-colour background, or use them with a predefined static background and then removes the background always giving the face boundaries. Finding faces by colour is another method that uses the typical skin colour to find face segments. But it doesn't work with all kinds of skin colours, and is not very robust under varying lighting conditions. Colour provides a computationally efficient yet effective method which is robust under rotations in depth and partial occlusions. It can be combined with other methods such as motion and appearance-based face detection. Human skin forms a relatively tight cluster in colour space even when different races are considered. Face colour distributions are normally modelled as Gaussian mixtures.

4. Training algorithm for emotion recognition:

Various algorithms are available for recognising faces:

1. Fisher Face Classifier

This algorithm is based on Principal Components Analysis(PCA). These Eigenfaces are the eigenvectors associated to the largest eigenvalues of the covariance matrix of the training data. This is indeed a powerful way to represent the data because it ensures the data variance is maintained while eliminating unnecessary existing correlations among the original features (dimensions) in the sample vectors.

After the eigenface of the selected test image is obtained, its Euclidean distance is calculated with the mean of the eigenfaces of the training dataset. Then the Euclidean distance is compared with the eigenvalues of the eigenvectors i.e. the distances between the eigenfaces of the training dataset and their mean image. The training images corresponding to various distances from the mean image are labelled with expressions like happy, sorrow, fear, surprise and anger and when the Euclidean distance between the test image eigenface and mean image matches the distances of the mean image and training dataset's eigenfaces, the emotion is classified and named as per the labelled train images.

2. Convolutional Neural Network:

In machine learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal pre-processing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. Convolutional networks were inspired by biological processes in which the connectivity pattern between neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. They have applications in image and video recognition, recommender systems and natural language processing.

3. HAAR Classifier:

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it.

Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle. They are based on the property of contrasts between different parts of the face. A feature window is used to classify these contrasts as face features.

For each feature, it finds the best threshold which will classify the faces to positive and negative Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates and new weights are calculated. The process is continued until required accuracy or error rate is achieved or a required number of features are found. There are thousands of these feature classifiers, grouping them into different stages and applying these stages one by one gives better results.

HAAR was chosen out of all three methods for several reasons .The first step in facial feature detection is detecting the face. This requires analysing the entire image. The second step is using the isolated face(s) to detect each feature. Since each the portion of the image used to detect a feature is much smaller than that of the whole image, detection of all three facial features takes less time on average than detecting the face itself. Regionalization also greatly increases the accuracy of the detection.

HAAR cascade classifier has shown excellent performance for the images which contain the simple background.

To handle the large databases HAAR cascade classifier is the best detector in terms of speed and reliability. Even the image is affected by illumination, face detection results are more accurate using HAAR cascade classifier.

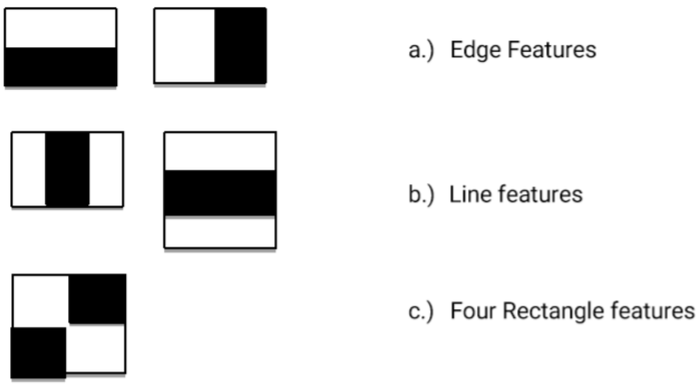


Fig 2. Haar Feature

5. **Packages used:**

1. keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Use Keras if you need a deep learning library that:

1. Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).

2. Supports both convolutional networks and recurrent networks, as well as combinations of the two.

3. Runs seamlessly on CPU and GPU.

2. **imutils**

This package includes a series of OpenCV + convenience functions that perform basics tasks such as translation, rotation, resizing, and skeletonization.

3. **cv2**

Just because they are not official packages doesn't mean you should feel uncomfortable using them, but it's important for you to understand that they are not endorsed and supported directly by the official OpenCV.org team.

All that said — there are four OpenCV packages that are pip-installable on the PyPI repository:

1. **opencv-python:** This repository contains just the main modules of the OpenCV library. If you're a PyImageSearch reader you do not want to install this package.

2. **opencv-contrib-python:** The opencv-contrib-python repository contains both the main modules along with the contrib modules — this is the library I recommend you install as it includes all OpenCV functionality.

3. **opencv-python-headless:** Same as opencv-python but no GUI functionality. Useful for headless systems.

4. **opencv-contrib-python-headless:** Same as opencv-contrib-python but no GUI functionality. Useful for headless systems.

4. **numpy:**

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object

- sophisticated (broadcasting) functions

- tools for integrating C/C++ and Fortran code

- useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Result and Discussions:

1. Real Time Video code

```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np

# parameters for loading data and images
detection_model_path =
'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/_mini_XCEPTION.102-0.66.hdf5'

# hyper-parameters for bounding boxes shape
# loading models
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry" ,"disgust","scared", "happy", "sad",
"surprised",
 "neutral"]


#feelings_faces = []
#for index, emotion in enumerate(EMOTIONS):
   # feelings_faces.append(cv2.imread('emojis/' + emotion + '.png',
-1))

# starting video streaming
cv2.namedWindow('your_face')
camera = cv2.VideoCapture(0)
while True:
    frame = camera.read()[1]
    #reading the frame
    frame = imutils.resize(frame,width=300)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces =
face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,
minSize=(30,30),flags=cv2.CASCADE_SCALE_IMAGE)

    canvas = np.zeros((250, 300, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
        faces = sorted(faces, reverse=True,
        key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
                    # Extract the ROI of the face from the grayscale
image, resize it to a fixed 28x28 pixels, and then prepare
            # the ROI for classification via the CNN
        roi = gray[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (64, 64))
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)


        preds = emotion_classifier.predict(roi)[0]
        emotion_probability = np.max(preds)
        label = EMOTIONS[preds.argmax()]


    for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
                # construct the label text
                text = "{}: {:.2f}%".format(emotion, prob * 100)

                # draw the label + probability bar on the canvas
               # emoji_face = feelings_faces[np.argmax(preds)]


                w = int(prob * 300)
                cv2.rectangle(canvas, (7, (i * 35) + 5),
```

```
                (w, (i * 35) + 35), (0, 0, 255), -1)
                cv2.putText(canvas, text, (10, (i * 35) + 23),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45,
                (255, 255, 255), 2)
                cv2.putText(frameClone, label, (fX, fY - 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
                cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY +
fH),
                              (0, 0, 255), 2)
#    for c in range(0, 3):
#        frame[200:320, 10:130, c] = emoji_face[:, :, c] * \
#        (emoji_face[:, :, 3] / 255.0) + frame[200:320,
#        10:130, c] * (1.0 - emoji_face[:, :, 3] / 255.0)


    cv2.imshow('your_face', frameClone)
    cv2.imshow("Probabilities", canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

camera.release()
cv2.destroyAllWindows()
```

2. Emotion Classifier code

```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np

# parameters for loading data and images
detection_model_path =
'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/_mini_XCEPTION.102-0.66.hdf5'

# hyper-parameters for bounding boxes shape
# loading models
face_detection = cv2.CascadeClassifier(detection_model_path)
emotion_classifier = load_model(emotion_model_path, compile=False)
EMOTIONS = ["angry" ,"disgust","scared", "happy", "sad",
"surprised",
 "neutral"]


#feelings_faces = []
#for index, emotion in enumerate(EMOTIONS):
   # feelings_faces.append(cv2.imread('emojis/' + emotion + '.png',
-1))

# starting video streaming
cv2.namedWindow('your_face')
camera = cv2.VideoCapture(0)
while True:
    frame = camera.read()[1]
    #reading the frame
    frame = imutils.resize(frame,width=300)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces =
face_detection.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5,
minSize=(30,30),flags=cv2.CASCADE_SCALE_IMAGE)

    canvas = np.zeros((250, 300, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
        faces = sorted(faces, reverse=True,
        key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
                    # Extract the ROI of the face from the grayscale
image, resize it to a fixed 28x28 pixels, and then prepare
            # the ROI for classification via the CNN
        roi = gray[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (64, 64))
```

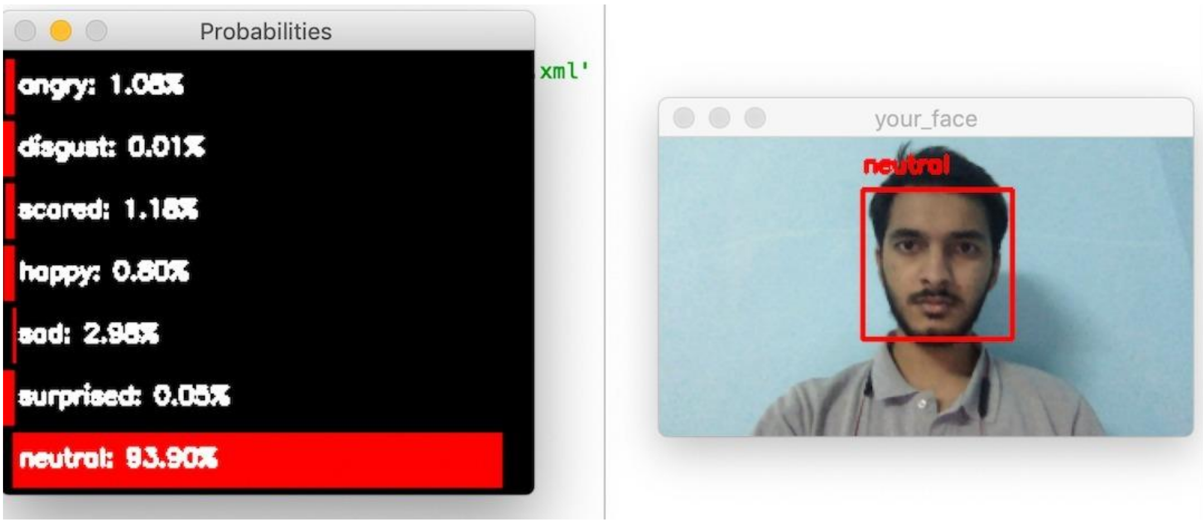
```
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)


        preds = emotion_classifier.predict(roi)[0]
        emotion_probability = np.max(preds)
        label = EMOTIONS[preds.argmax()]


    for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
                # construct the label text
                text = "{}: {:.2f}%".format(emotion, prob * 100)

                # draw the label + probability bar on the canvas
               # emoji_face = feelings_faces[np.argmax(preds)]


                w = int(prob * 300)
                cv2.rectangle(canvas, (7, (i * 35) + 5),
                (w, (i * 35) + 35), (0, 0, 255), -1)
                cv2.putText(canvas, text, (10, (i * 35) + 23),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45,
                (255, 255, 255), 2)
                cv2.putText(frameClone, label, (fX, fY - 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
                cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY +
fH),
                              (0, 0, 255), 2)
#    for c in range(0, 3):
#        frame[200:320, 10:130, c] = emoji_face[:, :, c] * \
#        (emoji_face[:, :, 3] / 255.0) + frame[200:320,
#        10:130, c] * (1.0 - emoji_face[:, :, 3] / 255.0)


    cv2.imshow('your_face', frameClone)
    cv2.imshow("Probabilities", canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

camera.release()
cv2.destroyAllWindows()
```

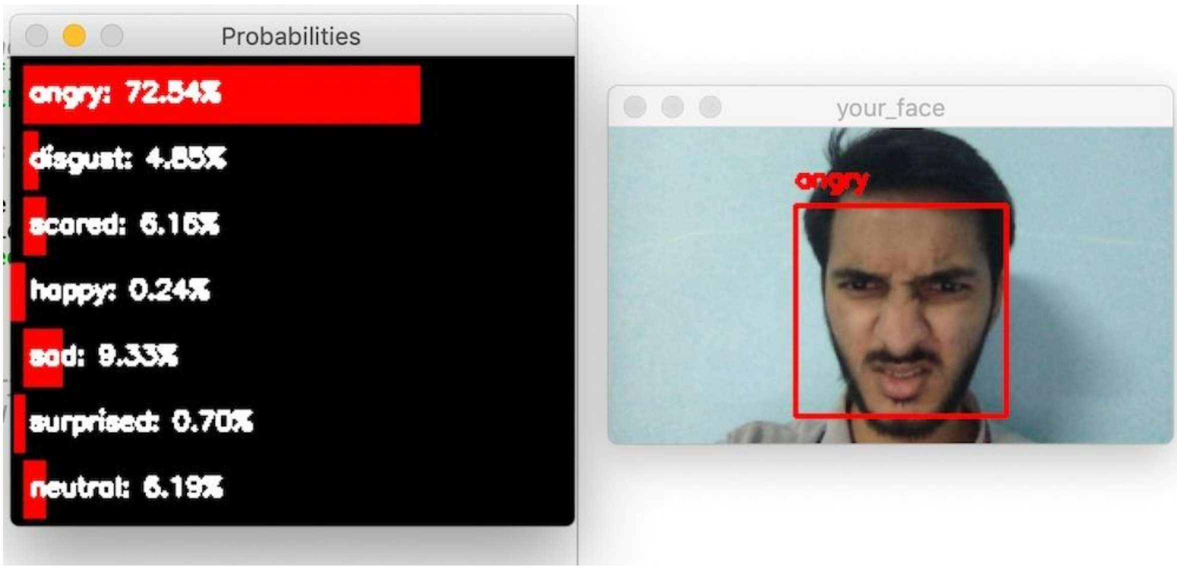
3. Load and Process code

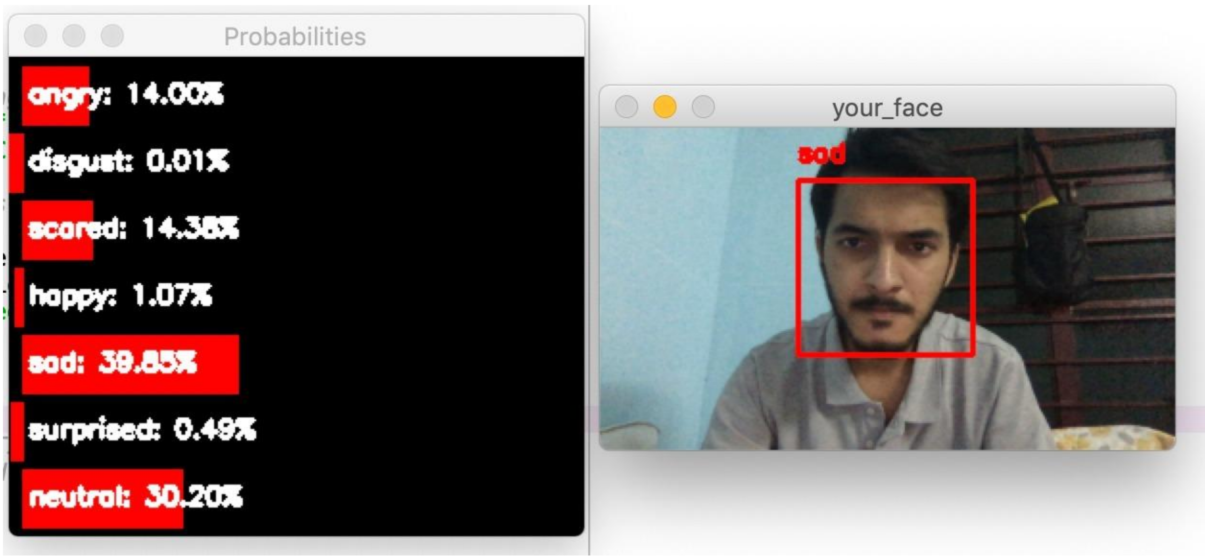
```
import pandas as pd
import cv2
import numpy as np


dataset_path = 'fer2013/fer2013/fer2013.csv'
image_size=(48,48)

def load_fer2013():
        data = pd.read_csv(dataset_path)
        pixels = data['pixels'].tolist()
        width, height = 48, 48
        faces = []
        for pixel_sequence in pixels:
            face = [int(pixel) for pixel in pixel_sequence.split('
')]
            face = np.asarray(face).reshape(width, height)
            face = cv2.resize(face.astype('uint8'),image_size)
            faces.append(face.astype('float32'))
        faces = np.asarray(faces)
        faces = np.expand_dims(faces, -1)
        emotions = pd.get_dummies(data['emotion']).as_matrix()
        return faces, emotions

def preprocess_input(x, v2=True):
    x = x.astype('float32')
    x = x / 255.0
    if v2:
```

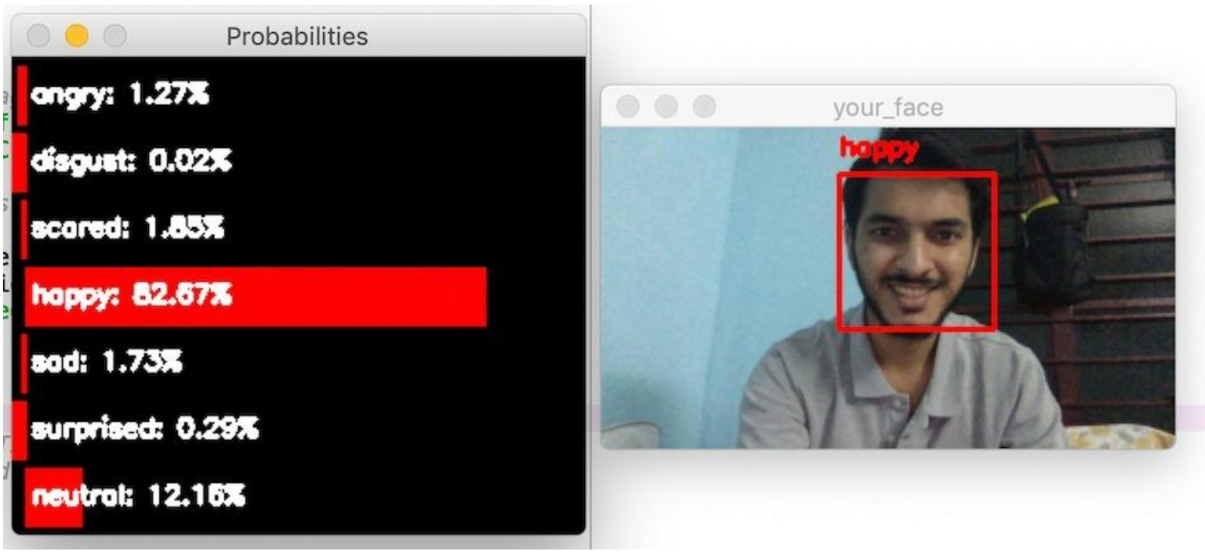
```
        x = x - 0.5
        x = x * 2.0
    return x
```

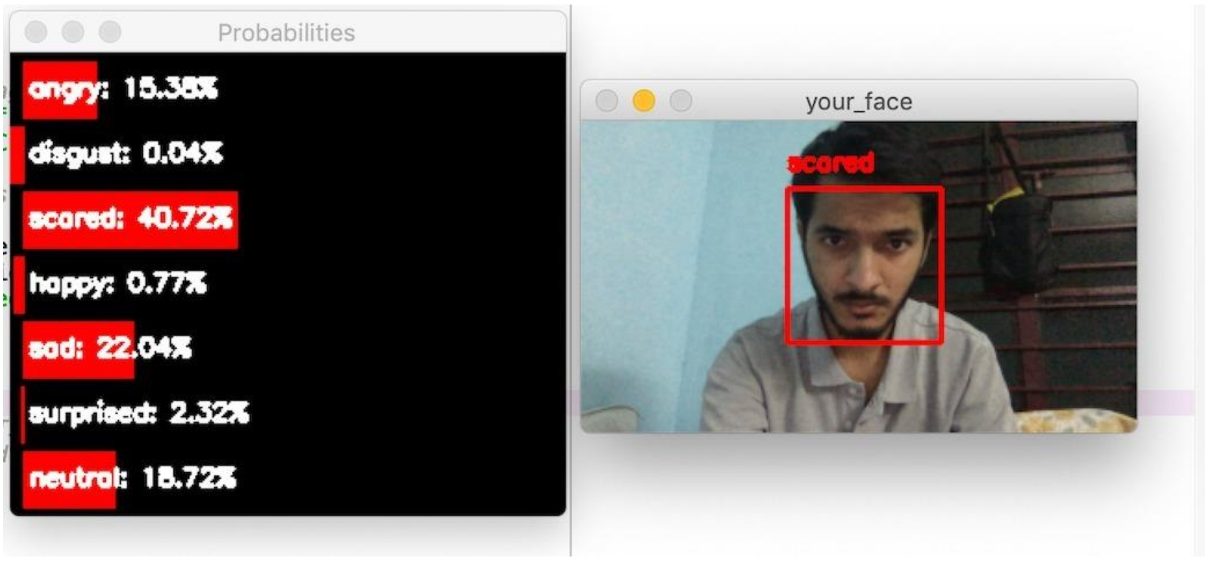
Output:-











Conclusion:

Extensive efforts have been made over the past two decades in academia, industry, and government to discover more robust methods of assessing truthfulness, deception, and credibility during human interactions. Efforts have been made to catch human expressions of anyone. Emotions are due to any activity in brain and it is known through face, as face has maximum sense organs. Hence human facial activity is considered. The objective of this project is to give brief introduction towards techniques, application and challenges of automatic emotion recognition system using tensorflow and opencv libraries in machine learning.

References:-

1. S. L. Happy, A. George and A. Routray, "A real time facial expression classification system using Local Binary Patterns," in IEEE 4th International Conference on Intelligent Human Computer Interaction, 2012.

2. C. Juanjuan, Z. Zheng, S. Han and Z. Gang, "Facial expression recognition based on PCA reconstruction,“ in International Conference on Computer Science and Education, 2010.

3. Liu, Chengjun, and Harry Wechsler. "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition." IEEE Transactions on Image processing 11.4 (2002): 467-476.

4.Faraji, Mohammad Reza, and Xiaojun Qi. "Face recognition under varying illumination with logarithmic fractal analysis." IEEE signal processing letters 21.12 (2014): 1457-1461.

5. De, Anurag, Ashim Saha, and M. C. Pal. "A human facial expression recognition model based on Eigen face approach." Procedia Computer Science 45 (2015): 282-289.

6. Zhao, Wenyi, et al. "Discriminant analysis of principal components for face recognition." Face Recognition. Springer, Berlin, Heidelberg, 1998. 73-85.

7. Cevikalp, Hakan, et al. "Discriminative common vectors for face recognition." IEEE Transactions on pattern analysis and machine intelligence 27.1 (2005): 4-13.

8. Martinez, Aleix. "Fisherfaces." Scholarpedia 6.2 (2011): 4282.

9. Turk, Matthew A., and Alex P. Pentland. "Face recognition using eigenfaces." Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on. IEEE, 1991.

10. Belhumeur, Peter N., João P. Hespanha, and David J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." IEEE Transactions on pattern analysis and machine intelligence 19.7 (1997): 711-720.

11. Wilson, Phillip Ian, and John Fernandez. "Facial feature detection using Haar classifiers." Journal of Computing Sciences in Colleges 21.4 (2006): 127-133.

12. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

13.. Pushpaja V. Saudagare, D.S. Chaudhari, Facial Expression Recognition using Neural Network –An Overview, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012.

14. P. Ekman and Friesen. Constants across cultures in the face and emotion. Journal of Personality and Social Psychology, 17(2):124-129, 1971

15. Viola, P., & Jones, M. (2001, July). Robust real-time face detection. In null (p. 747). IEEE.

16. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features.
In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-I). IEEE.