**EMOTION DETECTION**

**Aim- To make a Facial Emotion Recognition program to detect live emotion through web cam of a person.**

**Literature-**

The primary criteria of any emotion detection approach in real world conditions are  
emotion detection accuracy and robustness. For instance, people communication in  
the real world conditions are not in the form of only deliberate facial expressions of  
emotions. It is a mixture of facial expressions with speech process. Also, people in the  
social communication uses different intensity of emotions such as low, medium and  
high to express their intensions towards their goal. Facial expression in speech with  
different intensity of emotions (such as low, medium and high) can affect emotion  
detection’s accuracy and robustness.

**Facial Expression Recognition using Transfer Learning-**

Transfer learning is a new machine learning method based on existing knowledge to solve different but related problems. The goal of transfer learning is to apply the knowledge learned from an environment into a new environment. Compared with the traditional machine learning,

we consider the concept of Transfer Learning whereby features learnt from generic images of large-scale datasets can be used to train models of smaller databases without losing the generalization ability.

Facial Expression Recognition (FER) has gained importance and popularity among the Vision Community since the series of emotion recognition competitions such as FER2013 and EmotiW. made it possible to acquire sufficient training data from real-world. In particular, Deep Learning (DL) techniques have shown to cope well with emotion recognition in the wild. Training data is key to all DL techniques. However, the variations in the training sets pose a problem of insufficient samples leading to the common ‘overfitting’ or lack of generalization issues as well as large intra-class variability.

Standard pre-processing tasks such as face alignment and image normalization is often needed.

Convolutional Networks (ConvNet) typically consist of three layers namely CONV, POOL and FC (fully connected). These layers are stacked to form a full ConvNet architecture. The Convolutional Layer (CONV) has a set of learnable filters that are convolved with the original images providing as output specific activation feature maps.

It is well known that direct application of DL architectures on relatively small databases suffer from overfitting. Two approaches exist that attempt to overcome this issue by either using additional task-oriented data to pre-train the networks or use well known pre-trained models such AlexNet, VGG, VGG-face, GoogleNet.

Because of the intra-variations in expressions influenced by factors such as age, gender, culture, etc., for FER systems to perform well, it is required to have abundant samples during training.

Our intention here is to use the FER2013 database a training set and transfer learning.

**Algorithm A: With Data Augmentation on Kaggle FER2013 Database**

Stage 1. Pre-Processing

1) Load data: database: FER2013, image size: 48x48 = 2304 vector.

#classes=7 = [ 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, and 6=Neutral]

2) Split data:

(Training: test) = (28273,7067) 3) Augment data: rotation, scaling, shift along X and Y axes

Stage 2. Creating the Network Add layers sequentially

Stage 3. Training the Network: Num epochs=100 Fit model on batches with real-time augmentation

Stage 4. Learning decision: determine loss on training and test sets over the training epochs

Stage 5. Making Predictions: • Test on individual images. Evaluate trained model on test set.

This shows accuracy of 78% which is significant.

In the traditional classification and learning, in order to ensure the accuracy and reliability of the classification model trained, need to meet two basic assumptions: the first is the training samples and test samples meet with independent distribution; the second is that it need enough training data.

Transfer learning does not have to be like the traditional machine learning as the training samples and test samples need to be independent and identically distributed. At the same time, compared with the traditional network with random initialization, the learning speed of transfer learning is much faster.

because the image in the dataset is the facial expression image captured by the digital camera, and some of the color image, and some gray image, so the image must be converted to grayscale image. Finally, in order to remove the interference of the background and hair improve the accuracy of image classification, we should cut out the face region in the image, and use the clipping image for training, verification and testing.

**TensorFlow-**

TensorFlow is the second generation of artificial intelligence research and development system developed by Google, which supports convolutional neural network (CNN), recurrent neural network (RNN) and other depth neural network model.

**Objectives-**

***-first task is to locate face for taking expression.***

***-to draw a rectangle around it.***

***-text appearance box/area.***

***-checking each frame for change in emotion so that it can change text automatically.***

***-train our image/data.***

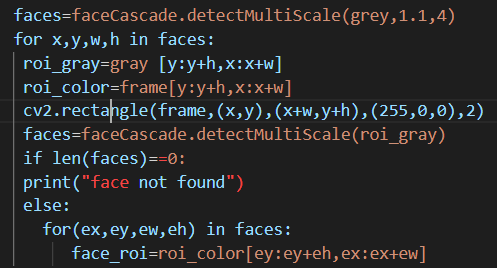
***-emotion detection using different images***

**How to attain our objectives-**

**For face detection we will be using OpenCV**

OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel.

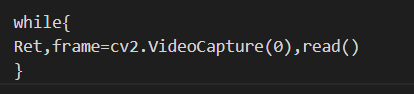
**To draw a rectangle over it-**



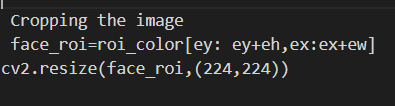
**To draw a text-(example) cv2 also help us in it**



**To check for frame- read frame to frame**

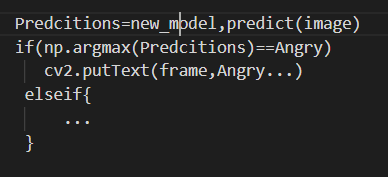


**For train our image –we have to resize-grayscale-crop our images**



**For detecting emotion-**

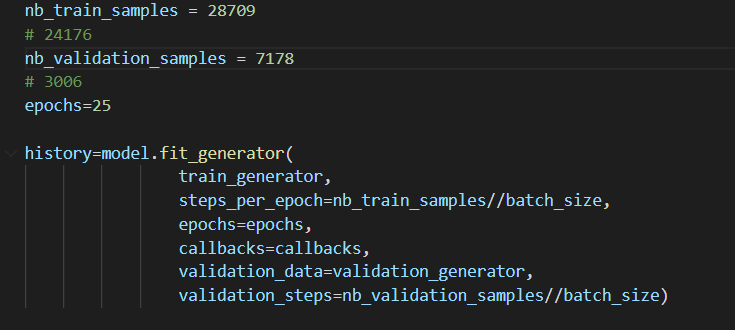
**We compare every frame to our dataset and TensorFlow (transfer learning) will match it with every folder and tell us (Predict) how much it is matching based on number. The we just have to select that folder name and display it, that is the emotion on screen.**



**Advantages-**

**1- In this method (transfer learning) we have to train our model with predefined datasets.**

**Where more the dataset more the accuracy.**

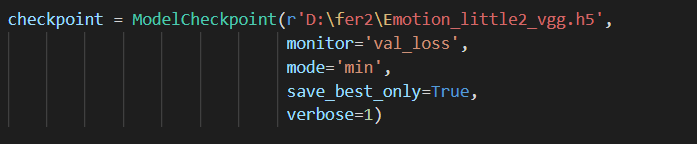


**While in other methods either we have to use Deep face or predefined libraries .in which does not have enhancement option.**



**2- Once we have created trained data file then we can use it in any environment/program just like a normal file.**

**We create our own trained model by-**



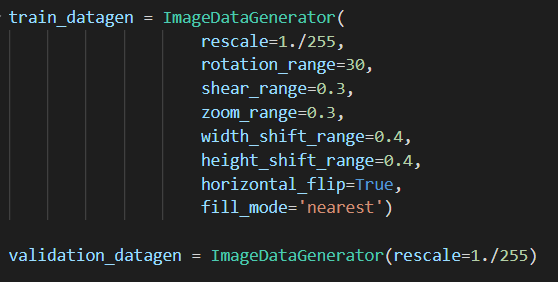
**While in others we have to import or set environment for another program.**

**3- Once its trained it become super-fast, just like to run a normal C program to execute.**

**While does not put any load on GPU/CPU.**

**4- Actually for improving our accuracy we have trained our data for 4- 5 images which we make form one image.**

**By different angle, rescale, rotation etc.**



**5-We can have more control over everything. We can set how many images it uses to train we can actually change one emotion name to other.**



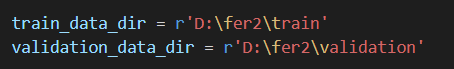
**6-Most important it is not difficult to understand. The way is easy we just have to know about keras little bit training and OpenCV.**

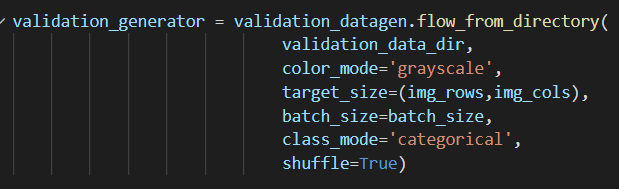
**We need only 4 files at max.**

**While in Py torch it requires too much files and code to do this task.**

**7-we can also improve accuracy by defining that if 2nd epoch has high accuracy, then overwrite first.**

**8- We have validation data also in which we have to validate our result first while training our data.**





**9-**

**If we use vs code then it will be very helpful it will tell us what library I have to import or what error I am getting. It's not that big deal but this maybe a factor for someone.**

**10-**

**We can also reduce classes or increase classes/number of emotions.**

**By simply download that class image dataset and train and now you have new emotion in your program.**

