International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 5, Issue 04, April-2019

FACE EXPRESSION DETECTION USING OPENCV

A. Sravya[1], M. Bhavana[2], Dr. V. Sireesha[3]

[1]*Dept of CSE, Vasavi College Of Engineering, Hyderabad, Telangana, India*
[2]*Dept of CSE, Vasavi College Of Engineering, Hyderabad, Telangana, India,*
[1]*Dept of CSE, Vasavi College Of Engineering, Hyderabad, Telangana, India,*

Abstract*— Computer vision is a field which teaches computers to gather information they see. It is a way computers learn from their environment and meet our demands. In the past few years face recognition has become popular and is being extensively used in various computer vision fields. This has motivated researches and neuroscientists to dig more on expression detection since it has many applications in automatic access control system. In this work, we detect user's emotions using his facial expressions. The expressions are predicted from the pre-existing image available in the memory and also real time. In order to build a face expression detection system, we scan the training dataset and compare it with the test data which helps the classifier to predict the expression. This implementation is done using Haar-cascade algorithm in order to detect the face. Due to its efficiency, Haar-like rectangle features play an important role in face detection. Also, an algorithm known as fisher face is used to detect the expression from the given dataset loaded by the user. These algorithms are provided by OpenCV which is an open source computer vision library with various kinds of algorithms built in. It is applicable in various computer vision projects. This work has been implemented using Python Idle 2.7, Open Source Computer Vision Library (OpenCV) and NumPy.

Keywords*— Haar cascade, Fisherface, OpenCV, Numpy, Adaboost.

I. INTRODUCTION

Image processing is extensively applied in order to get an enhanced image and to extract some useful information out of it. An efficient way of converting an image into a digital form subsequently performing various operations on it is image processing. It is similar to digital signal processing where a 2D image is taken as an input, in which the pixel values range from 0 to 255.

II. ALGORITHM SPECIFIC TO FACE EXPRESSION

A. Haar Cascade Algorithm

Haar cascade is based on Haar Wavelet[8] technique to analyze pixels in the image and compute them into squares by function. A training dataset is used which is a machine learning method of training the classifier. This process involves integral images which tells us the faces detected. Haar Cascade[1] is based on **Adaboost** learning algorithm[8]. It selects small number of features from a large number and gives us an efficient classifier.

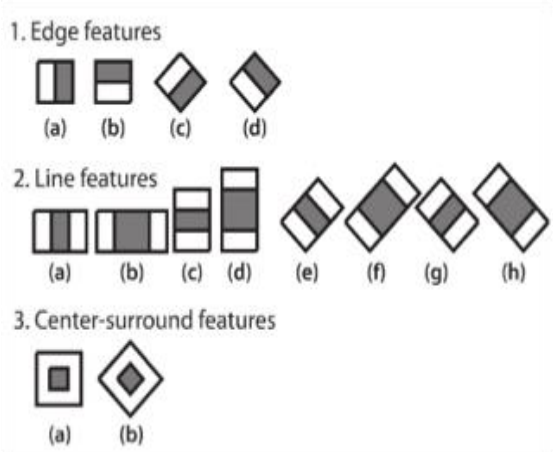


Fig.1 Feature Extraction

Fisher Face Algorithm

One way to represent the input data is by finding a subspace which represents most of the data variance. This can be obtained with the use of Principal Components Analysis (PCA). When applied to face images, PCA yields a set of eigenfaces[2]. Subspace representation is a set of face images, that results in basis vectors defining the space known as Fisherfaces.

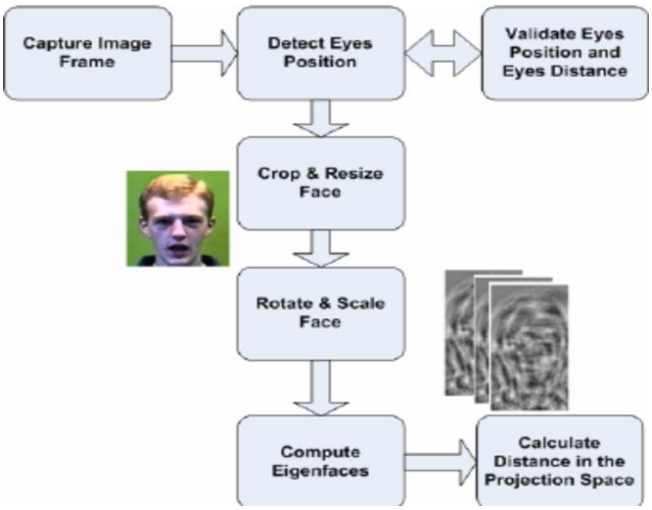


Fig.2 Workflow diagram of face expression

2.1 Introduction to OpenCV & Numpy

OpenCV: OpenCV is an Open Computer Vision Library[3]. It is portable on platforms like digital signal processors. It is a tool which reduces number of lines, errors, and also avoid memory leakage. OpenCV is a trainer and detector. It contains pre defined classifier for face, eyes, smile, etc., Various features like GPU acceleration[4] have been added to the existing libraries which enhances the tool's features.

Numpy: NumPy is a library for Python programming language. It provides large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It is an open-source software that has an array object of N-dimension. It is used to integrate C/C++ and Fortran code for solving linear algebraic expressions, and Fourier transforms[7].

Example: >>> import numpy as np
>>> a = np.arange(15).reshape(3, 5)
>>> aarray([[0, 1, 2, 3, 4], [5, 6, 7, 8, 9], [10, 11, 12, 13, 14]])
>>> a.shape
(3, 5)
>>> a.ndim
2
>>> a.dtype.name
'int64'

The types of expressions detected are:

A. Neutral
B. Happy
C. Anger
D. Disgust
E. Surprise
F. Fear
G. Sad
H. Contempt

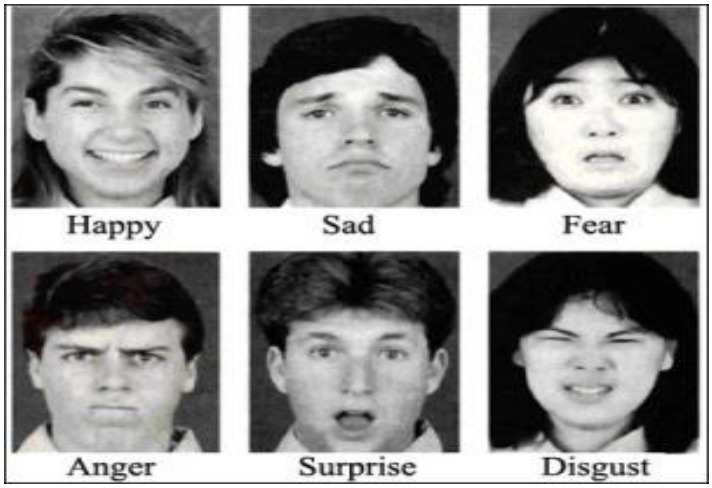


Fig.3 Cohn Kanade dataset

III. STEPS IMPLEMENTED TO PERFORM EXPRESSION DETECTION USING OPENCV

1. After installing Open CV and Python Idle 2.7, we must create the dataset. Here, we create a dataset by analysing group of images so that our result is accurate and there is enough data to extract sufficient information.
2. The database is then classified into two different directories. First directory contains the images and the second directory contains information about different types of expressions.
3. Different types of classes can be used in OpenCV for emotion recognition, but we will be mainly using Fisher Face. FisherfaceRecognizer= createFisherFaceRecognizer().
4. Now, we extract faces using OenCV which provides classifiers that are responsible for detecting faces. face_detector= cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
5. The dataset is split into Training set and Classification set. The training set is used to teach the type of emotions by extracting information from a number of images and the classification set is used to estimate the classifier performance. The images should be of same size for better results.
6. Each image's subject is analyzed, converted to grayscale, cropped and saved to a directory. Conversion of Color Image to Gray Scale:

There are basically two methods to convert a color image to a gray scale image [5]:

A. Average Method

In Average method, mean of RGB(Red, Blue, Green)colors in a present color image is calculated . We get,
Grayscale= (R+G+B)/3;
Sometimes instead of grayscale image we get black image. This is due to the converted image we get ie.,33% each of Red, Blue & Green. Therefore, to solve this problem we use the second method called Weighted Method or Luminosity Method.

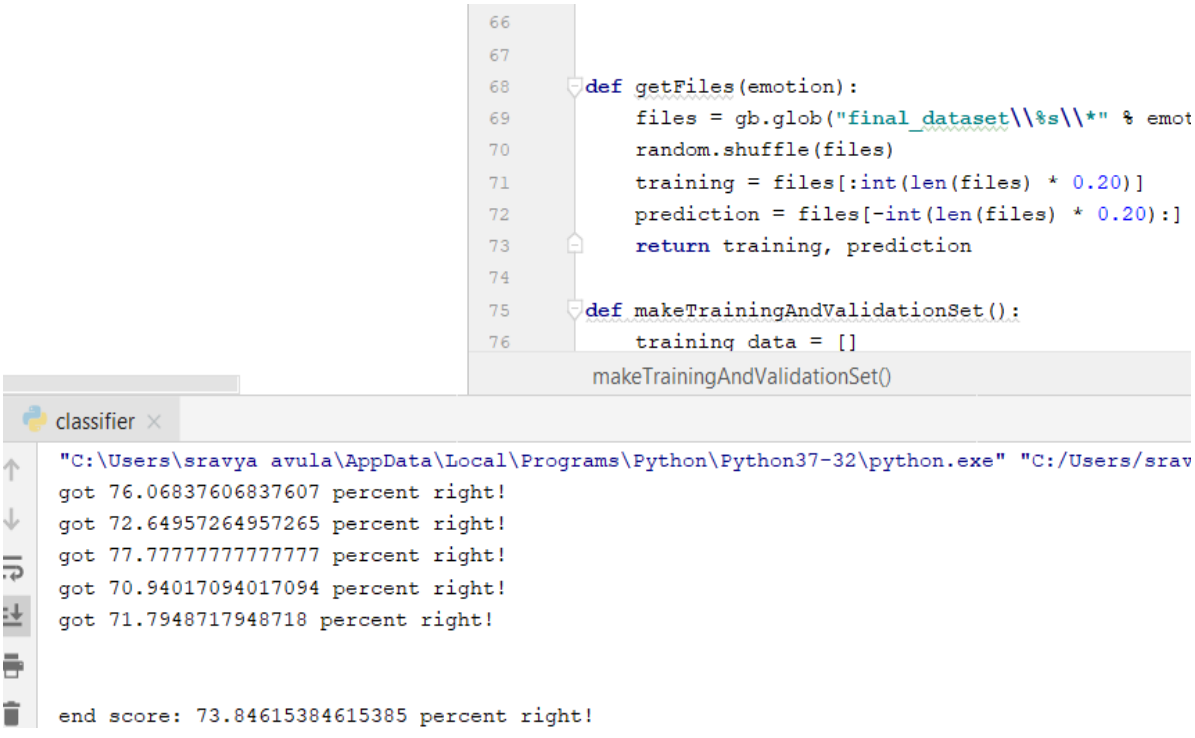
B. Weighted or Luminosity Method

To limit the problem in Average Method, we use Luminosity method. In this method, we decrement the presence of Red Color and increment the color of Green Color and the blue color's percentage is in between these two colors.
Thus, by the equation we get, Grayscale= ((0.3 * R) + (0.59 * G) + (0.11 * B)).
We use this because of the wavelength patterns of these colors. Blue has the least wavelength while Red has the maximum wavelength.

7. Finally, we compile training set as 80% of the test data and classify the remaining 20% on the classification set.

IV. RESULTS

CASE 1: WHEN THE SIZE OF TRAINING DATA SET IS 20 % OF THE FINAL DATA SET WE GET:



CASE 2 : WHEN THE SIZE OF TRAINING DATA SET IS 40 % OF THE FINAL DATA SET WE GET:

```
    random.shuffle(files)
    training = files[:int(len(files) * 0.40)]
    prediction = files[-int(len(files) * 0.40):]
    return training, prediction

def makeTrainingAndValidationSet():
    training_data = []
```

getFiles()

classifier

```
"C:\Users\sravya avula\AppData\Local\Programs\Python\Python37-32\python.exe" "C:/Users/srav
got 71.84873949579831 percent right!
got 81.09243697478992 percent right!
got 81.5126050420168 percent right!
got 81.5126050420168 percent right!
got 83.19327731092437 percent right!


end score: 79.83193277310923 percent right!

Process finished with exit code 0
```

CASE 3 : WHEN THE SIZE OF TRAINING DATA SET IS 60 % OF THE FINAL DATA SET WE GET:

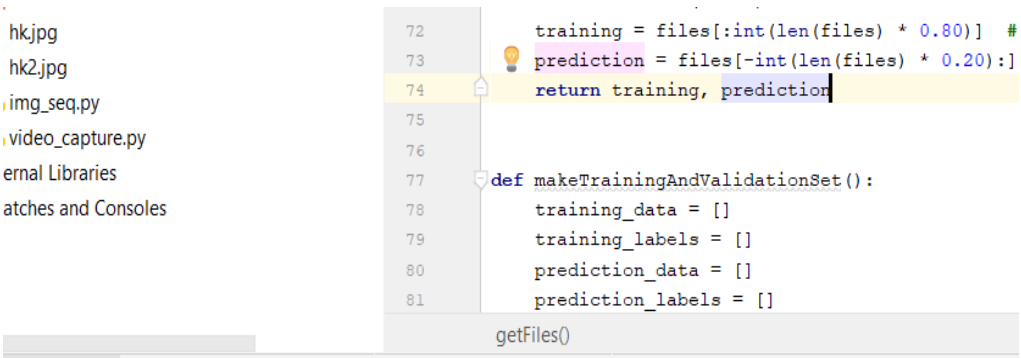
hk.jpg
hk2.jpg
img_seq.py
video_capture.py
ernal Libraries
atches and Consoles

```
    training = files[:int(len(files) * 0.60)]  #
    prediction = files[-int(len(files) * 0.40):]
    return training, prediction


def makeTrainingAndValidationSet():
    training_data = []
    training_labels = []
    prediction_data = []
    prediction_labels = []
```

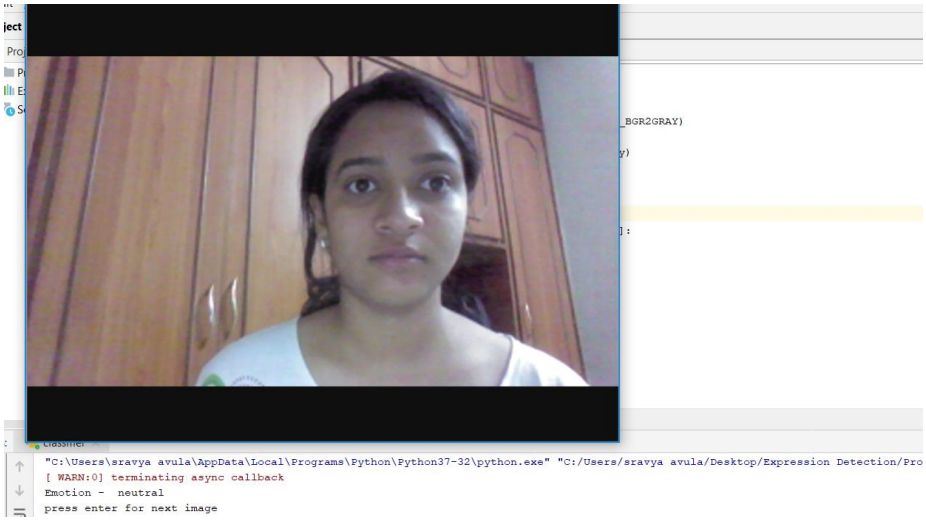
classifier

```
D:\forPro\MajorProject-1\ExpressionRecogniser\venv\Scripts\python.exe D:/forPro/MajorProjec
got 79.83193277310924 percent right!
got 85.71428571428571 percent right!
got 78.57142857142857 percent right!
got 81.9327731092437 percent right!
got 78.57142857142857 percent right!


end score: 80.92436974789916 percent right!
```

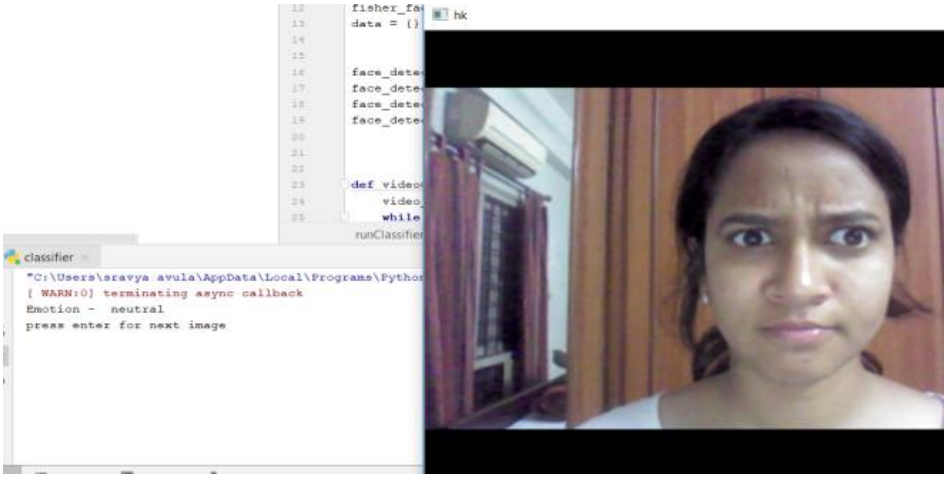
CASE 4 : WHEN THE SIZE OF TRAINING DATA SET IS 80 % OF THE FINAL DATA SET WE GET:



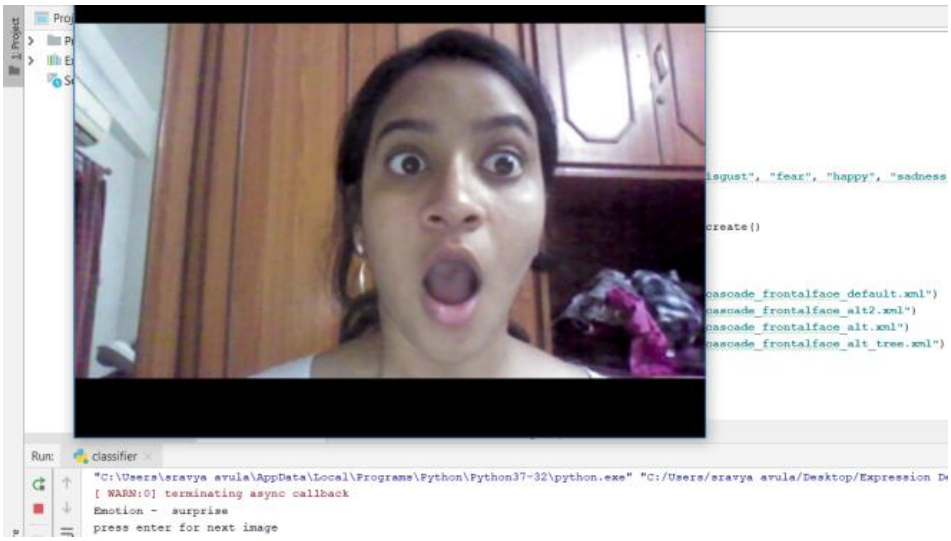
REAL TIME EXPRESION 1: NEUTRAL



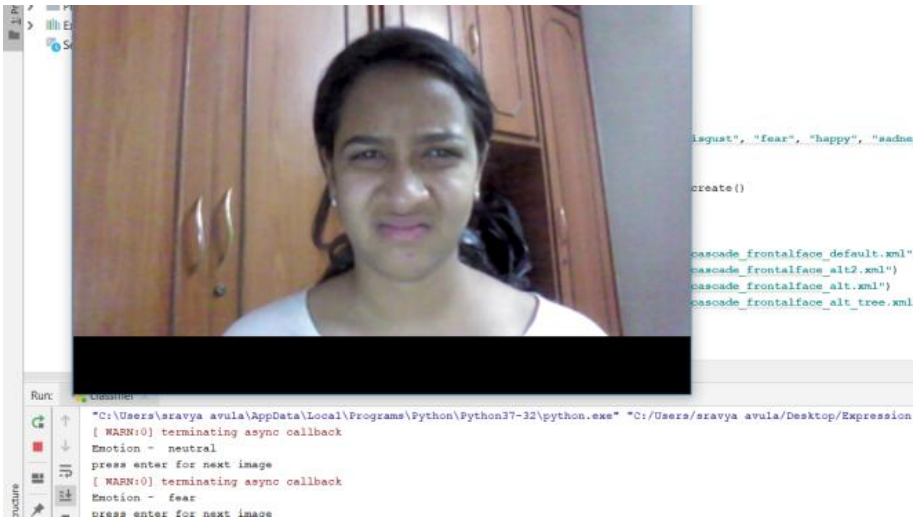
REAL TIME EXPRESSION 2: ANGER



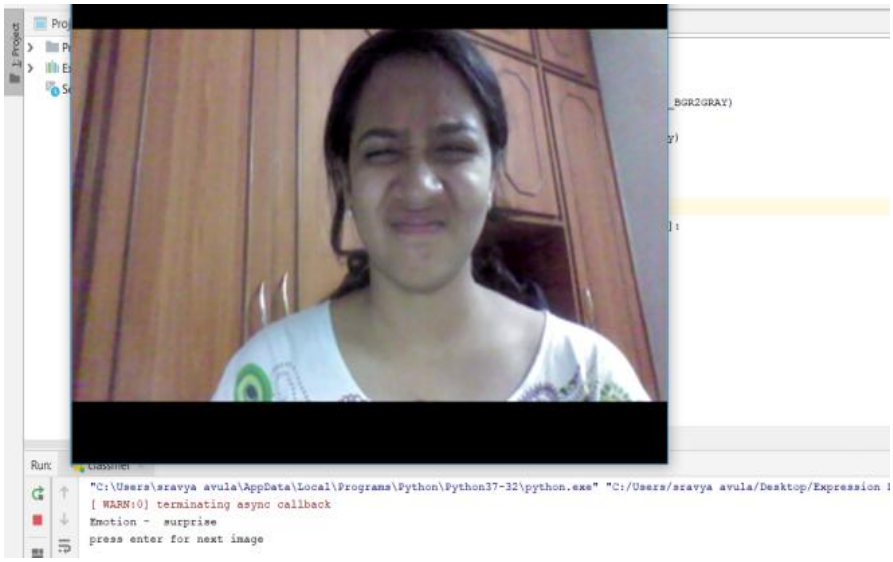
REAL TIME EXPRESSION 3: SURPRISE



REAL TIME EXPRESSION 4: CONTEMPT

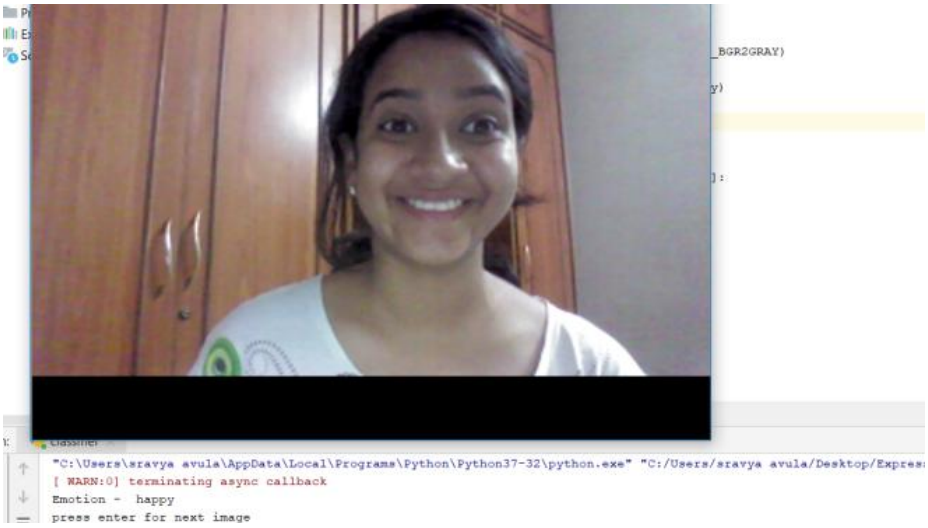


REAL TIME EXPRESSION 5: DISGUST

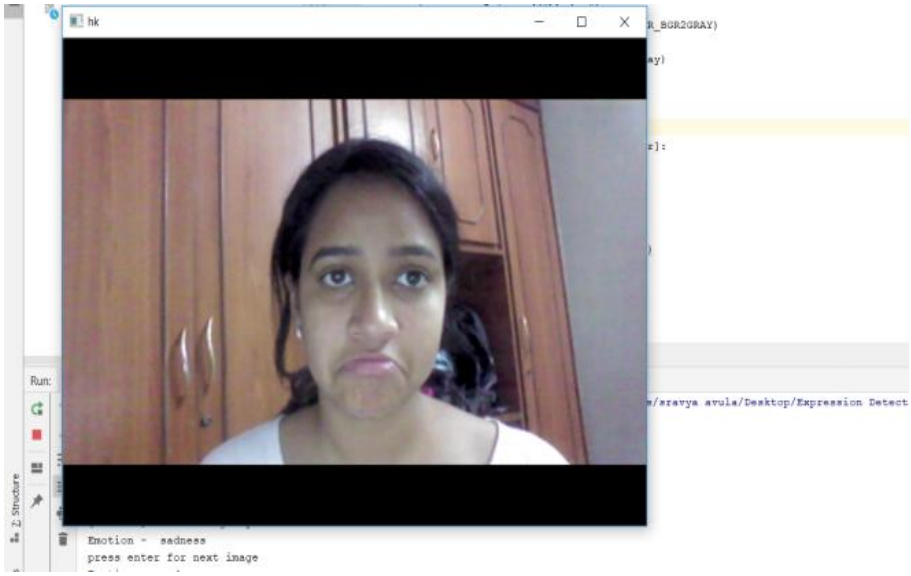


REAL TIME EXPRESSION 6: FEAR

REAL TIME EXPRESSION 7: HAPPY



REAL TIME EXPRESSION 8: SADNESS



4.1 OBSERVATIONS

Table 1: Accuracy prediction for existing subjects

Training data set (as % of final dataset)	Prediction data set (as % of final dataset)	Accuracy of emotion detection
20	20	73.85
40	40	79.83
60	40	80.92
80	20	83.08

Table 2: Accuracy prediction for real time face subject

V. APPLICATIONS

Loads of tasks and processes can be performed if one can become aware about the intricacies and endless possibilities offered under the field of emotion detection.
Some of the widely used applications are:
1. Improvised web development
Internet is expanding in a large scale and the service providers are interested in collecting tons of data from the users. Correspondingly, all the content and advertisements are played based on the users' profile. Subsequently, adding intricate details about the different human emotions can provide much more precise behavioral models of different types of users.
2. App and product development
In engineering processes, testing of a product can be done with ease by understanding the emotions. It's a long-established fact that level of comfort with different software products depends hugely upon human emotions. A products overall look and feel can also alter human feelings which in turn makes the person buy the product or not. Thus, researching about different emotional states of a human body and how it is influenced by usage of different products is a matter of prime importance for related industry professionals.

Actual expression	Predicted Expression
Neutral	Neutral
Anger	Neutral
surprise	Surprise
Contempt	Fear
Disgust	Surprise
Fear	Surprise
Happy	Happy
Sadness	Sadness
Total Accuracy	**57.14**

VI. CONCLUSION AND FUTURE WORK

Artificial Intelligence can be used to solve intriguing tasks such as emotion detection, although this task was quite convolute even more when using a great number of images.
We have developed a computer vision system that automatically recognizes facial expressions with subtle differences and also estimates expression intensity. Feature point tracking together with the principal component analysis, and high gradient component analysis, and is then used to discriminate subtle differences in facial expressions.
In Present Work, we recognized different types of human emotions using Python 2.7, Open CV & (CK and CK+) Database [6] and got some interesting insight about it.
For existing subjects of eight different expressions, the result was approximately 70-80% accurate.

The classifier has tried to learn the dataset and has given us some satisfying results. The results show that our system has achieved high accuracy in facial expression recognition.
We have implemented the same methodology for a real time subject with the same eight expressions and we have achieved an accuracy of 57.14%. We are going to push further to increase the efficiency and produce more accurate results. Emotion detection is an inseparable part of computer vision. Loads of tasks and processes can be performed if one can become aware about the intricacies and endless possibilities offered under the field of emotion detection.

VII. REFERENCES

[1] https://docs.opencv.org/3.4.2/d7/d8b/tutorial_py_face_detection.html
http://www.rroij.com/open-access/comparison-between-face-recognition-algorithmeigenfaces-fisherfaces-and-elastic-bnch-graph-matching-187-193.php?aid=37590J
[2] France,46-53- Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z.,& Matthews, I. (2010). The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression. Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101. Vision with OpenCV. *Queue* 10, 4, Pages \ Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, and Victor Eruhimov.2012. Realtime Computer
[3] https://opencv.org/
[4] https://www.tutorialspoint.com/dip/
[5] https://pshychmnemonics.wordpress.com/2015/07/03/primary-emtions
[6] https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/
[7] http://dsp-book.narod.ru/PWSA/8276_01.pdf