VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE3088 ARTIFICIAL INTELLIGENCE

PROJECT TITLE:

EMOTION DETECTION

TEAM MEMBER:

Shivani Gokulram Naidu 19MIA1006

Madasu Deepike 19MIA1066

Harinisrig 19MIA1069

FACULTY NAME:

Dr. S.L. Jayalakshmi

SLOT:

B2

INDEX

ACKNOWLEDGMENT

Primarily, we would like to thank the almighty for all the blessings he showered over us to complete this project without any flaws. The success and final outcome of this assignment required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along with the completion of our project. Whatever we have done is only due to such guidance and assistance by our faculty, Dr. S.L. JAYALAKSHMI, to whom we are really thankful for giving us an opportunity to do this project. Last but not the least, we are grateful to all our fellow classmates and our friends for the suggestions and support given to us throughout the completion of our project.

ABSTRACT

Face detection has been around for ages. Taking a step forward, human emotion displayed by face and felt by brain, captured in either video, electric signal (EEG) or image form can be approximated. Human emotion detection is the need of the hour so that modern artificial intelligent systems can emulate and gauge reactions from face. This can be helpful to make informed decisions be it regarding identification of intent, promotion of offers or security related threats. Recognizing emotions from images or video is a trivial task for human eye, but proves to be very challenging for machines and requires many image processing techniques for feature extraction. Several machine learning algorithms are suitable for this job. Any detection or recognition by machine learning requires training algorithm and then testing them on a suitable dataset. This paper explores feature extraction techniques which would help us in accurate identification of the human emotion.

INTRODUCTION

Recognizing human expressions and emotions has drawn the attention of researchers, as the capability of recognizing one's expressions helps in human-computer interaction, to right advertising campaigns, and crowning with an augmented and enhanced human communication ,by amending the emotional intelligence ("EQ") of humans.

There are many ways to inspect the recognition of human expressions, ranging from facial expressions, body posture, voice tone etc. In this paper we have focused on facial expression recognition.

Facial Emotion Recognition(FER) is a thriving research area in which lots of advancements like automatic translation systems, machine to human interaction are happening in industries.

Why AI is needed for this project?

Facial emotion recognition is the process of detecting human emotions from facial expressions. The human brain recognizes emotions automatically, and software has now been developed that can recognize emotions as well. This technology is becoming more accurate all the time, and will eventually be able to read emotions as well as our brains do.

AI can detect emotions by learning what each facial expression means and applying that knowledge to the new information presented to it. Emotional artificial intelligence, or emotion AI, is a technology that is capable of reading, imitating, interpreting, and responding to human facial expressions and emotions.

EXISTING SYSTEM

Facial Expression Recognition Applications:

Monitoring facial expressions of several people over a period of time provides great insights if used carefully, so for this reason we can use this technology in the following applications.

1: Smart Music players that play music according to your mood:

Think about it, you come home after having a really bad day, you lie down on the bed looking really sad & gloomy and then suddenly just the right music plays to lift up your mood.

2: Student Mood Monitoring System:

Now a system that cleverly averages the expressions of multiple students over a period of time can get an estimate of how a particular topic or teacher is impacting students, does the topic being taught stresses out the students, is a particular session from a teacher a joyful experience for students.

3: Smart Advertisement Banners:

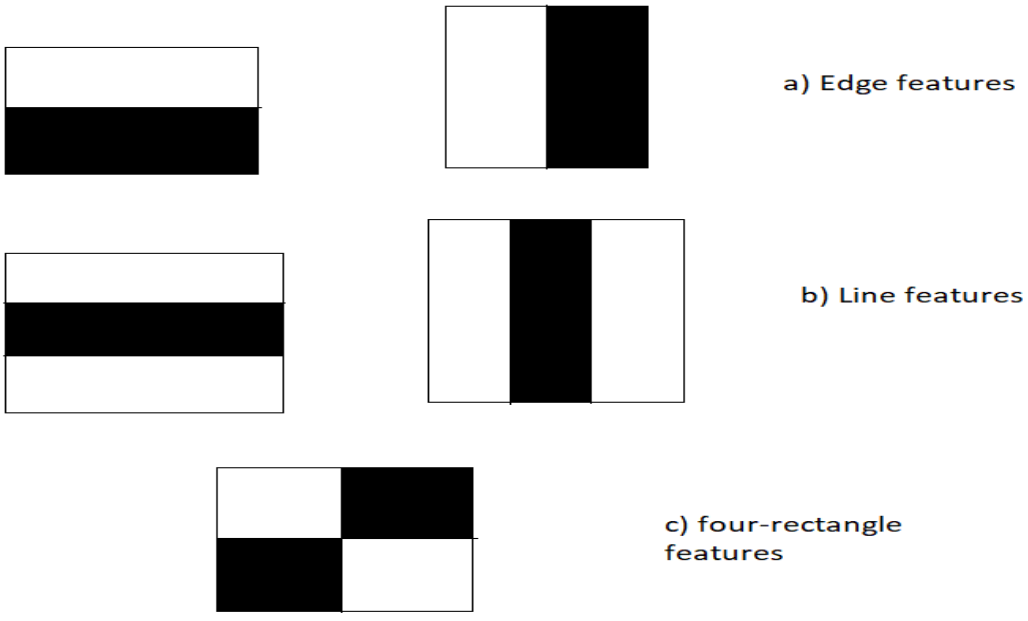
Think about smart advertisement banners that have a camera attached to it, when a commercial airs, it checks real-time facial expressions of people consuming that ad and informing the advertiser if the ad had the desired effect or not. Similarly, companies can get feedback if customers liked their products or not without even asking them.

PROPOSED METHODOLOGY

Here we are going to see how to detect faces using a cascade classifier in OpenCV Python. Face detection has much significance in different fields of today's world. It is a significant step in several applications, face recognition (also used as biometrics), photography (for auto-focus on the face), face analysis (emotion recognition), video surveillance, etc

It works in four stages:

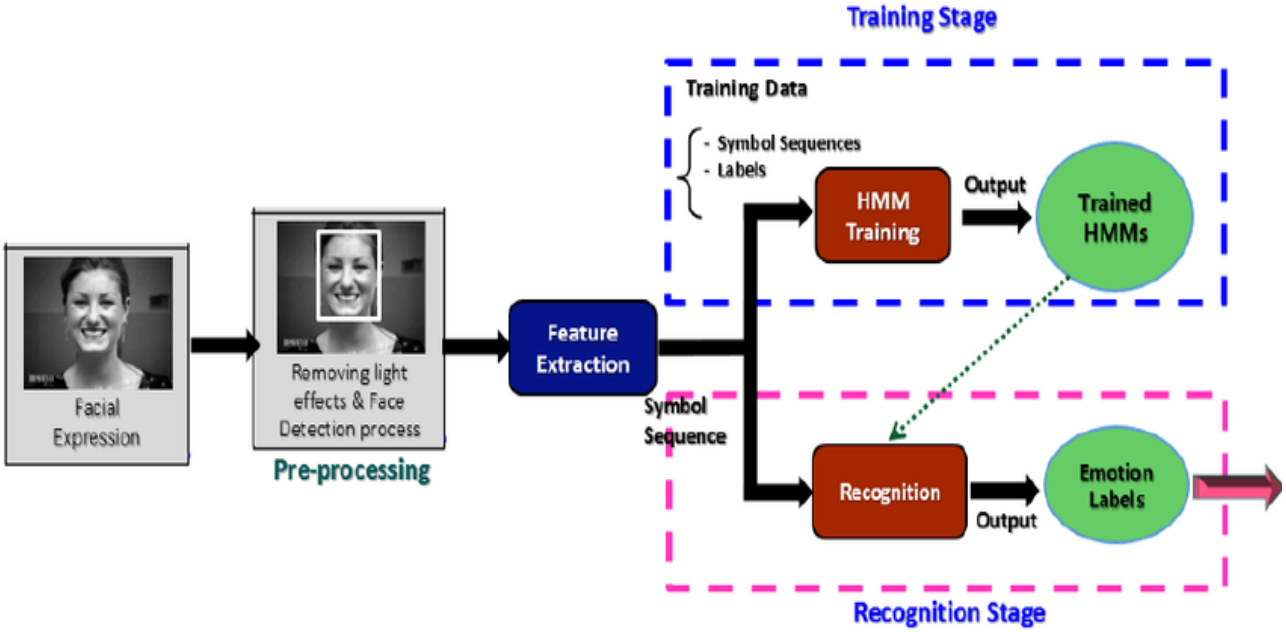
- Haar-feature selection: A Haar-like feature consists of dark regions and light regions. It produces a single value by taking the difference of the sum of the intensities of the dark regions and the sum of the intensities of light regions. It is done to extract useful elements necessary for identifying an object. The features proposed by viola and jones are:

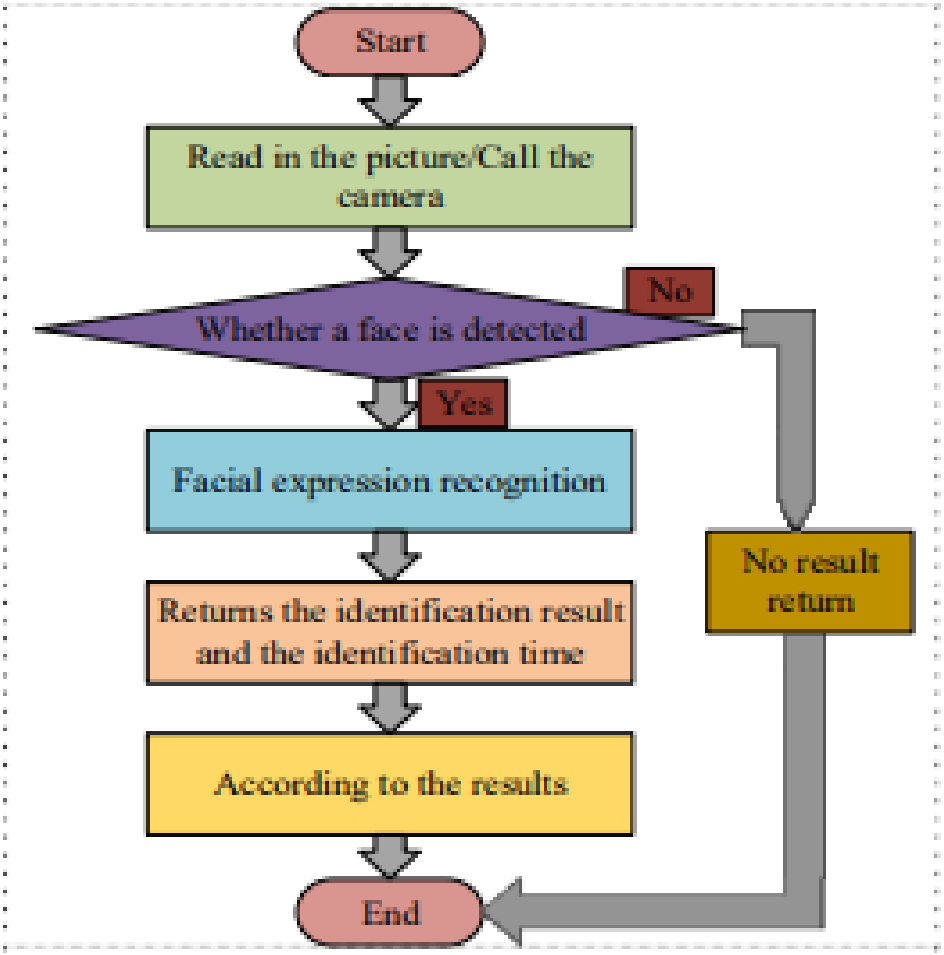


- Creation of Integral Images: A given pixel in the integral image is the sum of all the pixels on the left and all the pixels above it. Since the process of extracting Haar-like features involves calculating the difference of dark and light rectangular regions, the introduction of Integral Images reduces the time needed to complete this task significantly.
- AdaBoost Training: This algorithm selects the best features from all features. It combines multiple “weak classifiers” (best features) into one “strong classifier”. The generated “strong classifier” is basically the linear combination of all “weak classifiers”.
- Cascade Classifier: It is a method for combining increasingly more complex classifiers like AdaBoost in a cascade which allows negative input (non-face) to be quickly discarded while spending more computation on promising or positive face-like regions. It significantly reduces the computation time and makes the process more efficient.

OpenCV comes with lots of pre-trained classifiers. Those XML files can be loaded by cascadeClassifier method of the cv2 module. Here we are going to use haarcascade_frontalface_default.xml for detecting faces.

ARCHITECTURE DIAGRAM





MODULES

The strategy starts with the preparing cascadeClassifier by pleasing an input image (static or dynamic) by accumulating a convolution layer, pooling layer, flatten layers, and dense layers. Convolution coats will be added for better accuracy for large datasets. The dataset is collected from CSV file (in pixels format) and it's converted into images and then classify emotions with respective expressions. Here emotions are classified as happy, sad, angry, surprise, neutral, disgust, and fear with lots of images for the training dataset and huge images for testing.

Each emotion is expressed with different facial features like eyebrows, opening the mouth, Raised cheeks, wrinkles around the nose, wide-open eyelids and many others. Trained the large dataset for better accuracy and result that is the object class for an input image. Based on those features it performs convolution layers and max pooling. In this proposed system, Deep learning is used with the help of Keras, contains several Models. Among those models, the facial emotion of the human is classified.

Here emotions are classified as happy, sad, angry, surprise and neutral with lots of images for the training dataset and huge images for testing. Each emotion is expressed with different facial features like eyebrows, opening the mouth, Raised cheeks, wrinkles around the nose, wide-open eyelids and many others. Trained the large dataset for better accuracy and result that is the object class for an input image.

TECHNOLOGIES AND INTEGRATION

The recognition of facial expressions is the core of the system. When the trained model file is obtained, in addition to using test set data to test the network training results, the more important thing is to use the corresponding code to complete the call to the model, so as to test whether the model can achieve the expected good results in the expression picture data under the real environment.

Facial expression recognition is a process performed by humans or computers, which consist of:-

1. **Locating faces in the scene** (e.g., in an image; this step is also referred to as face detection)
2. **Extracting facial features** from the detected face region (e.g., detecting the shape of facial components or describing the texture of the skin in a facial area; this step is referred to as facial feature extraction),
3. **Analyzing the motion of facial features** and/or the changes in the appearance of facial features

Figure 1. The overall process for facial expression recognition

IMPLEMENTATION DETAILS(SOFTWARES USED)

The implementation of the whole system in software will be completed around the network training and expression recognition using pandas, numpy, matplotlib, sklearn etc to complete the design of the application layer interface. The overall system has been developed using python with support of Jupyter notebook and Visual Studio Code.

But here I want to highlight three important libraries that have been used:

- **OpenCV**: This is one of the library which is widely used in processing images, particularly real time images.
- **Keras**: This is one of the library which is used to code deep learning models. In its back-end it uses Tensorflow.
- **DeepFace:** This is one of the library which is used to face recognition and facial attribute analysis library for Python.

Dataset Description

The dataset used in this project work has been taken from the Kaggle.com available at (https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data) i.e. a FER dataset. The data consists of 48x48 pixel grayscale images of faces.

The task is to categorize each face based on the emotion shown in the facial expression in to one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

train.csv contains two columns, “emotion” and “pixels”. The “emotion” column contains a numeric code ranging from 0 to 6, inclusive, for the emotion that is present in the image. The “pixels” column contains a string surrounded in quotes for each image. The contents of this string a space-separated pixel values in row major order. test.csv contains only the “pixels” column and your task is to predict the emotion column.

The training set consists of 28,709 examples. The public test set used for the leaderboard consists of 3,589 examples. The final test set, which was used to determine the winner of the competition, consists of another 3,589 examples.

RESULT & DISCUSSION:

The research of facial expression recognition has made new development in the era of artificial intelligence, and the related achievements in this field have been greatly transformed into the industrial field, and better technical application direction has also been obtained.

In view of the application requirements, it is necessary to build a human-computer interactive visual facial expression recognition system.

Facial expression recognition system should have the following characteristics: easy to operate and visualization, high degree of intelligence, strong universality, strong real-time.

So here we have derived the output of 7 major emotions namely :

- Angry
- Disgust
- Fear
- Happy
- Neutral
- Sad
- Surprise

These emotions were recognised by inserting the path for the image to see if it predicts the correct class. Lets see the output of all the 7 emotions:

Method-1: Using Keras

ANGRY:

```
path = "/content/train/Angry/download(1).jpg"
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
plt.title("input image")
plt.show()
```

DISGUST:

```
path = "/content/train/Disguist/download(1).jpg"
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
plt.title("input image")
```

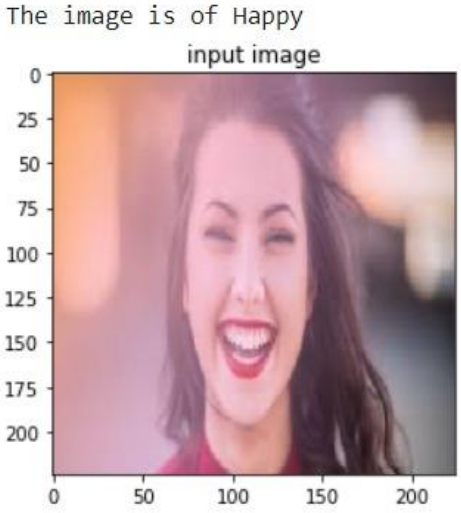
```
plt.show()
```

FEAR:

```
  path = "/content/train/Fear/download(1).jpg"
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
plt.title("input image")
plt.show()
```

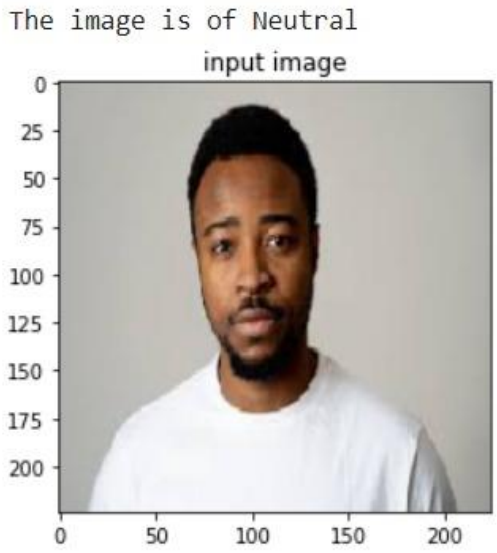
HAPPY:

```
  path = "/content/train/Happy /download(1).jpg"
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
plt.title("input image")
plt.show()
```



NEUTRAL:

```
  path = "/content/train/Neutral/download(1).jpg"
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
plt.title("input image")
plt.show()
```



SAD:

```
  path = "/content/train/Sad/download(1).jpg"
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
plt.title("input image")
plt.show()
```

SURPRISE:

```
  path = "/content/train/Suprise /download(1).jpg"
```

```
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
plt.title("input image")
plt.show()
```

We have also randomly taken a picture from net to see the prediction of emotion that it gives to check the accuracy .

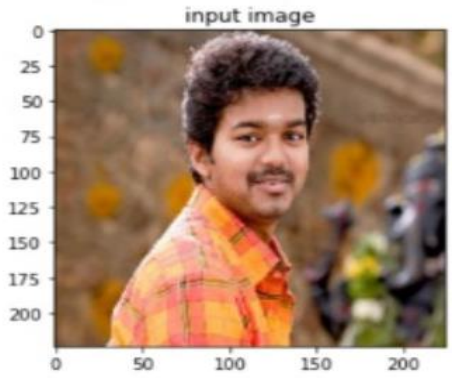
CODE:

```
  path = "/content/Vijay.jpg"
img = load_img(path, target_size=(224,224) )
  i = img_to_array(img)/255
input_arr = np.array([i])
input_arr.shape
  pred = np.argmax(model.predict(input_arr))
  print(f" the image is of {op[pred]}")
  # to display the image
plt.imshow(input_arr[0])
```

```
plt.title("input image")
plt.show()
```

OUTPUT:

```
the image is of Happy
```



Method-2: Using FER

```
from fer import FER
import matplotlib.pyplot as plt
%matplotlib inline
test_image_one = plt.imread("/content/Happy.jpg")
emo_detector = FER(mtcnn=True)
# Capture all the emotions on the image
captured_emotions = emo_detector.detect_emotions(test_image
_one)
# Print all captured emotions with the image
print(captured_emotions)
plt.imshow(test_image_one)
# Use the top Emotion() function to call for the dominant emotio
n in the image
dominant_emotion, emotion_score = emo_detector.top_emotion
(test_image_one)
```

```
[{'box': (161, 0, 425, 425), 'emotions': {'angry': 0.06, 'disgust': 0.0, 'fear': 0.0, 'happy': 0.93, 'sad': 0.01, 'surprise': 0.0, 'neutral': 0.0}}]
happy 0.93
```


```
print(dominant_emotion, emotion_score)
```

```
happy 0.93
```

Method-3: Using Deepface

In [24]:
```
from deepface import DeepFace
predictions= DeepFace.analyze(img)
```

```
Action: race: 100%|████████████████████████████████████████████████████| 4/4 [00:17<00:00,  4.39s/it]
```

In [25]:
```
predictions
```

Out[25]:
```
{'emotion': {'angry': 2.5760504840377507e-06,
  'disgust': 1.1235212413196471e-09,
  'fear': 9.375357157281883e-08,
  'happy': 98.59557151794434,
  'sad': 2.8454857670112688e-05,
  'surprise': 0.01309231884079054,
  'neutral': 1.391306333243847},
 'dominant_emotion': 'happy',
 'region': {'x': 250, 'y': 80, 'w': 260, 'h': 260},
 'age': 26,
 'gender': 'Man',
 'race': {'asian': 5.875943228602409,
  'indian': 36.31830811500549,
  'black': 28.7826806306839,
  'white': 3.9336882531642914,
  'middle eastern': 3.466768190264702,
  'latino hispanic': 21.622616052627563},
 'dominant_race': 'indian'}
```

29]:
```
from deepface import DeepFace
img = cv2.imread('Dhoni.jpg')
imgplot = plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
obj = DeepFace.analyze(img_path = "Dhoni.jpg", actions = ['age', 'gender', 'race', 'emotion'])
print(obj["age"]," years old ",obj["dominant_race"]," ",obj["dominant_emotion"]," ", obj["gender"])
```

```
Action: emotion: 100%|████████████████████████████████████████████████████| 4/4 [00:01<00:00,  2.35it/s]
```

```
26  years old  indian   happy   Man
```

Real-time Classification:

We have also prepared a real time emotional detection that will used to detect the emotion of a person in a live manner. That is in this project, OpenCV's Haar Cascade is used to for the detection and extraction of the region containing the face from the video feed of webcam.
The video is converted to grayscale and the detected face is contoured or enclosed within a region defined to surround the face. This region of interest is resized to the size of the image and compared. The emotion is predicted and on top of face above contour rectangle.

Code:

```
import os
import cv2
import numpy as np
from keras.preprocessing import image
import warnings
warnings.filterwarnings("ignore")
from keras.preprocessing.image import load_img, img_to_array
from keras.models import  load_model
import matplotlib.pyplot as plt
import numpy as np

# load model
model = load_model("best_model.h5")


face_haar_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')


cap = cv2.VideoCapture(0)

while True:
    ret, test_img = cap.read()  # captures frame and returns boolean value and
captured image
    if not ret:
        continue
    gray_img = cv2.cvtColor(test_img, cv2.COLOR_BGR2RGB)

    faces_detected = face_haar_cascade.detectMultiScale(gray_img, 1.32, 5)
```

```
    for (x, y, w, h) in faces_detected:
        cv2.rectangle(test_img, (x, y), (x + w, y + h), (255, 0, 0),
thickness=7)
        roi_gray = gray_img[y:y + w, x:x + h]  # cropping region of interest
i.e. face area from  image
        roi_gray = cv2.resize(roi_gray, (224, 224))
        img_pixels = image.img_to_array(roi_gray)
        img_pixels = np.expand_dims(img_pixels, axis=0)
        img_pixels /= 255

        predictions = model.predict(img_pixels)

        # find max indexed array
        max_index = np.argmax(predictions[0])

        emotions = ('angry','disgust', 'fear', 'happy', 'sad', 'surprise',
'neutral')
        predicted_emotion = emotions[max_index]

        cv2.putText(test_img, predicted_emotion, (int(x), int(y)),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

    resized_img = cv2.resize(test_img, (1000, 700))
    cv2.imshow('Facial emotion analysis ', resized_img)

    if cv2.waitKey(10) == ord('q'):  # wait until 'q' key is pressed
        break

cap.release()
cv2.destroyAllWindows
```

Output:

ACCURACIES

SURPRISE

[{'box': (34, 18, 69, 69), 'emotions': {'angry': 0.13, 'disgust': 0.0, 'fear': 0.22, 'happy': 0.26, 'sad': 0.04, 'surprise': 0.3, 'neutral': 0.05}}]

Surprise Accuracy: 0.3

ANGRY

[{'box': (140, 13, 82, 82), 'emotions': {'angry': 1.0, 'disgust': 0.0, 'fear': 0.0, 'happy': 0.0, 'sad': 0.0, 'surprise': 0.0, 'neutral': 0.0}}]

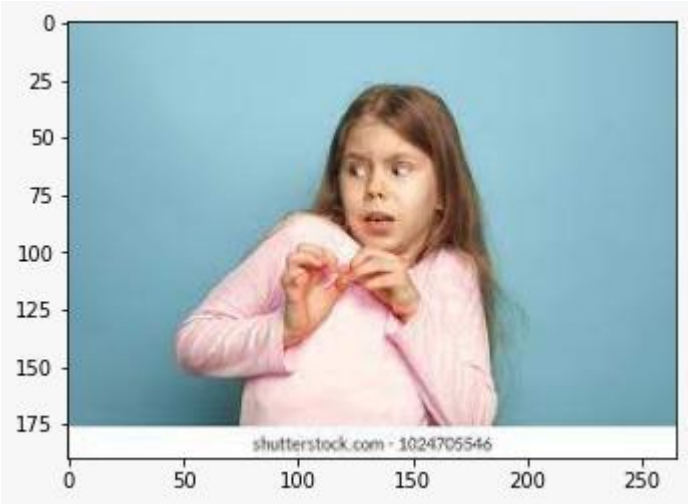
Accuracy of anger: 1.0

DISGUST

[{'box': (107, 42, 65, 65), 'emotions': {'angry': 0.9, 'disgust': 0.08, 'fear': 0.01, 'happy': 0.01, 'sad': 0.0, 'surprise': 0.0, 'neutral': 0.0}}]

Disgust Accuracy : 0.9

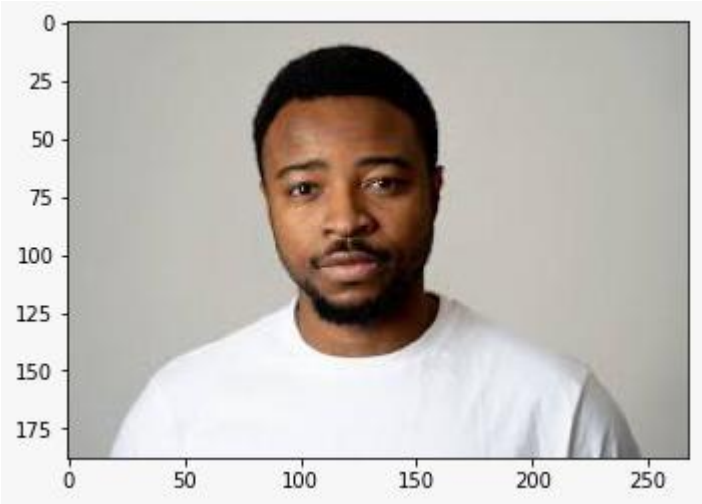
FEAR



[{'box': (111, 43, 54, 54), 'emotions': {'angry': 0.03, 'disgust': 0.0, 'fear': 0.12, 'happy': 0.01, 'sad': 0.01, 'surprise': 0.81, 'neutral': 0.02}}]

Fear Accuracy: 0.81

NEUTRAL



[{'box': (76, 33, 91, 91), 'emotions': {'angry': 0.01, 'disgust': 0.0, 'fear': 0.0, 'happy': 0.01, 'sad': 0.13, 'surprise': 0.0, 'neutral': 0.85}}]

Neutral Accuracy: 0.85

SAD

[{'box': (153, 32, 51, 51), 'emotions': {'angry': 0.09, 'disgust': 0.0, 'fear': 0.04, 'happy': 0.02, 'sad': 0.71, 'surprise': 0.01, 'neutral': 0.14}}]

Sad Accuracy: 0.71

AVERAGE ACCURACY

```
[68] avg_accuracy= (emotion_score1+emotion_score2+emotion_score3+emotion_score4+emotion_score5+emotion_score6+emotion_score7)/7
     avg_accuracy

     0.7742857142857142
```

ACCURACY OF THIS PROJECT

```
[71] print("Accuracy of this project:", avg_accuracy*100)

     Accuracy of this project: 77.42857142857143
```

CONCLUSION:

Emotions play an important role in human sphere of life. This project was comprehended to assess and gather the significant and efficient emotion recognition techniques developed. Today, we have a wide range of methods and considerable amount of work has been done in the fields of facial emotion recognition.

The 7 elementary emotions which humans display are sadness, surprise, disgust, happiness, fear ,neutral and anger. Facial Emotion Recognition (FER) is largely done through two categories of methods, namely, feature and model based techniques.Here we are doing the feature technique.

In the carried out experiments, for 7 emotional states, we achieved a very good classification accuracy of emotions - 96% for the images. Experiments were carried out using the FER2013 dataset uploading it in google colaboratory and also using visual code for real time recognition . Certainly, the classification accuracy was influenced by the way users play specific facial expressions. In real conditions the classification accuracy can be affected by many additional factors. When you feel real emotions, facial expressions can vary greatly - may be exposed to a greater or lesser extent.

FUTURE SCOPE:

There are numerous limitations and a wide scope of improvements in this field. Till now, only seven basic emotions have been successfully identified.
Research should be carried out for identifying more than seven emotions.
It can be applied to real distance education. The application of emotion recognition in virtual learning environments is a much-researched topic. In addition to the change of uncertainty factors makes teachers and students face pattern is more complex, so the emotion recognition in the online learning network application mode is a very challenging topic.
Accordingly, it is proposed that the subject requires further research from the following aspects:

1. Due to the fact that this research does not involve the illumination and pose of the image, it is uncertain how much these factors influence facial expressions and thus the final emotion recognition. All of these issues need to be explored in future research and validated by experiments.
2. To make the theory and technology of emotion recognition fully meet the practical requirement, there are suggesting that the comprehensive application of image processing, pattern recognition, computer vision and neural networks, psychology, cognitive science, and integrates with other biometric authentication methods and methods of human-computer interaction perception based on the in-depth and meticulous research work.

REFERENCES

[1] Marek Kowalski, Jacek Naruniec, Tomasz Trzcinski, Deep Alignment Network: A convolutional neural network for robust face alignment"

[2] Recognizing Facial Expressions Using Deep Learning by Alexandru Savoiu Stanford University and James Wong Stanford University [http://cs231n.stanford.edu/reports/2017/pdfs/224.pdf]

[3] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, "Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition"

[4] Ivona Tautke, Tomasz Trzcinski, Adam Bielski, "I Know how You Feel: Emotion Recognition with Facial Landmarks"

[5] B. Hasani and M. H. Mahoor, "Facial expression recognition using enhanced deep 3d convolutional neural networks," in Proceedings of CVPRW. IEEE, 2017.

[6] Daniel Llatas Spiers, "Facial Emotion Detection Using Deep Learning" .

[7] Sivo Prasad Raju, Saumya A and Dr. Romi Murthy, "Facial Expression Detection using Different CNN Architecture Hybrid Vehicle Driving", Centre for Communications, International Institute of Information Technology.

[8] Deepesh Lekhak, "Facial Expression Recognition System using Convolutional Neural Network", Tribhuwan University Institute of Engineering.

[9] Jie Hu, Li Shen, and Gang Sun "Squeeze-and-excitation networks", in Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141, 2018.

[10] A literature survey on Facial Expression Recognition using Global Featuresby Vaibhavkumar J. Mistry and Mahesh M. Goyani,International Journal of Engineering and Advanced Technology (IJEAT),April,2013 [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.645.5162&rep=rep1&typ e=pdf]