SENTIMENT ANALYSIS

A PROJECT REPORT

Submitted by

SOUDEEP DUTTA
SOUPARNA BISWAS
AISHWARYA ROY
SUBHADEEP NEOGI

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

Department of COMPUTER SCIENCE & ENGINEERING

BRAINWARE UNIVERSITY
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

December,2022

SENTIMENT ANALYSIS

Submitted by

SOUDEEP DUTTA
SOUPARNA BISWAS
AISHWARYA ROY
SUBHADEEP NEOGI

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

BRAINWARE UNIVERSITY
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project report "**SENTIMENT ANALYSIS**" is the bonafide work of "**SOUDEEP DUTTA, SOUPARNA BISWAS, AISHWARYA ROY, SUBHADEEP NEOGI"** who carried out the project work under my supervision.

SIGNATURE
DR. DEBDUTTA PAL

HEAD OF THE DEPARTMENT

COMPUTER SCIENCE & ENGINEERING

SIGNATURE
MR. RITESH PRASAD

SUPERVISOR
ASSISTANT PROFESSOR

COMPUTER SCIENCE & ENGINEERING

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

	Table of Contents	
Chapter	**Title**	**Page No.**

Abstract

Sentiment analysis is a type of natural language processing that is used to determine the attitude or emotion of a speaker, writer, or other subject. It is used to analyze large amounts of text data to identify and extract subjective information such as opinions, attitudes, and emotions. This analysis can be used to better understand customer sentiment, predict customer behavior, and inform marketing decisions. The process involves using various methods to identify sentiment-bearing words and phrases, classify sentiment as either positive or negative, and then measure the sentiment intensity. It can also be used to identify sentiment-related topics and track sentiment trends over time. By understanding the sentiment of customers, businesses can better understand the needs of their customers and develop strategies to meet those needs.

Introduction

Computer systems and computer software are replacing physical labour as the world increasingly moves toward digitalization. Pen and paper are a considerably less accurate and inconsistent option. Numerous uses for the project on face recognition with sentiment analysis might be found in daily life. We can quickly analyse a person's behaviour using sentiment analysis. System for face recognition appeared inevitable. Nowadays, facial recognition technology is used by both governments and businesses worldwide. This article provides you with the knowledge you need to create a facial recognition system for sentiment analysis as well as the mathematical equations for the algorithms you'll be employing. After the theory, the actual implementation of those algorithms is covered in detail, which was the most crucial aspect of the project.

Objective

- **Unlocking Phones**: We can use it on smartphones to unlock phone and apps using face recognition.
- With Emotion Detection employees can make better decisions, increase their concentration and performance at work, manage stress, and adopt healthier and more productive work styles.
- Emotion-sensing smart home gadgets might tailor entertainment (music, films, TV shows, or pictures) to the user's mood.
- **User–user behaviour**: This includes behaviour between two users, for example, sending friend requests or messages from one user to another, playing games, blocking, chatting, etc.
- **User–community behaviour**: This includes behaviour between a user and a community, for example, joining or leaving a community, being a fan, or participating in any community event.
- **User–entity behaviour**: This category includes writing a blog, post, or review or uploading a photo to social media.

Planning

Facial recognition is a method of recognising or verifying a person's identification by looking at their face. People can be identified in pictures, films, or in real time using facial recognition technology. It's a form of Biometric Identification. Although the technology is mostly utilised for security and law enforcement, there is growing interest in other applications. The implementation of face recognition technology includes the following three stages:

• Data Acquisition

Acquisition of Face Data - Acquisition and Processing of Face Data is first step in the face recognition system. In this step face images are collected on real-time from webcam or may be at static time i.e., from website and stored in the database. The collected face images should have the pose, illumination and expression etc. variation in order to check the performance of the face recognition system under

these conditions. Processing of face database requires sometimes otherwise causes serious effect on the performance of face recognition systems due to changes in the illumination condition, background, lighting conditions, camera distance, and thus the size and orientation of the head. Therefore, input image is normalized and some image transformation methods apply on the input image.

- **Input processing**

Extracting Face Feature - This process can be defined as the process of extracting relevant information from a face image. In feature extraction, a mathematical representation of original image called a biometric template or biometric reference is generated, which is stored in the database and will form the basis (vector) of any recognition task. Later these extracted features are used in recognition. After that greyscale pixel is considered as initial feature.

- **Face image classification and decision making**

Recognition of Face - In this process, once the features are extracted and selected, the next step is to classify the image. For that appearance-based face recognition algorithms use a wide variety of classification methods. Such as PCA, LDA, Fisher face etc. In classification, the faces are compared for the similarity between faces from the same individual and different individuals after all the face images in database are represented with relevant features. Sometimes feature extraction & recognition process are done simultaneously.

There are about “80 nodal points” on a human face (distance between the eyes, width of the nose, depth of the eye socket, cheekbones, jaw line, chin). Using these facial features are judged and real time images are identified. The task in hand was to identify the optimal solution to land a working project developed using python.

Requirement Analysis

HARDWARE REQUIREMENTS

SOFTWARE REQUIREMENTS

- HARDWARE REQUIREMENTS:
  - ❖ Hard Disk - Minimum 512MB
  - ❖ Ram- 8GB
  - ❖ Processor- Minimum Intel Core I3
  - ❖ Graphics Drive- Not needed
- SOFTWARE REQUIREMENTS:
  - ❖ OS - Windows 10
  - ❖ VSCODE
  - ❖ Python 3.11
  - ❖ **OpenCV Package/Module**: The OpenCV (Open-Source Computer Vision) library is a computer vision library. Both a trainer and a detector are included in OpenCV. OpenCV was utilized to provide a robust infrastructure for PC vision applications and to accelerate their use in business, machine perception is important. It supports Windows, Linux, Mac OS, iOS, and Android and includes C++, Python, and Java interfaces. OpenCV was created with a strong focus on computational proficiency on programs that are run in real-time. It's written in improved C/C++, and it supports multi-core processing. This library will be used to accomplish the task. It can take advantage of the hardware thanks to OpenCL, the hidden heterogeneous platform. More than 2500 upgraded algorithms are included in the collection, which includes a large number of excellent machine learning algorithms.

These algorithms can be used to distinguish and perceive faces, recognize objects, group human activities in recordings, track camera developments, track moving articles, extricate 3D models of objects, create 3D point clouds from stereo cameras, combine pictures to deliver a high-resolution picture of a whole scene, find similar images from a picture database, expel red eyes from pictures taken with flash, take after eye developments, perceive view, and distinguish and perceive faces. OpenCV has a client network of about 47 thousand people and an estimated number of downloads of over 14 million. The library is widely used. The Haar Cascade classifier is a face detector included with OpenCV. To train a classifier to recognize faces, two sets of photos are used, one of which contains all of the photographs containing faces and the other of which does not. Models for classifiers are created using these two sets. Using negative and positive images, feature extraction is carried out. The object detection framework is used by the Haar classifier.

Advantages of using OpenCV module:

- It's a large library and its accessible free of expense.
- Since OpenCV library is written in C/C++ it is very quick.
- An Image is fundamentally and exhibit for which Python's NumPy - NumPy and SciPy.org - SciPy.org modules can be utilized.
- The exhibit controls in NumPy module are exceedingly upgraded for speed. The face detection can be performed at 15 frames per second speed for 384*288-pixel photos, which are fast compared to other modules.
- OpenCV gives algorithmic efficiency fundamentally to process real-time projects. Also, it has been planned in a way that permits it to take advantage of hardware acceleration.
- Sometimes in substantial scale ventures, not every person is alright with Python and codes in C++, so one need to fit in with the bound together coding style hence OpenCV supports Multilanguage. When it comes to the installation part, we have installed using below command. Installation command: pip install OpenCV v. NumPy: NumPy is the most important Python package for logical computing.

It includes, among other things, the following:

• Intense N-dimensional array object.

• Tools for integrating C/C++ code. NumPy can be used as a capable multi-dimensional container of non-specific information in addition to its indisputable logical uses. This allows NumPy to work seamlessly and fast with a variety of databases. vi. Face Recognition Package/Module: The world's easiest face recognition module can be used to recognise and edit faces from Python or from the command line. The accuracy of the model is around 99.38 percent. A simple face recognition command line programme is also included, which allows you to run face recognition on a folder of photographs from the command line. Installation command: pip install face recognition.

System Flow

All identification or authentication technologies operate using the following four stages: a. Capture: A physical or behavioural sample is captured by the system during Enrolment and also in identification or verification process. b. Extraction: Unique data is extracted from the sample and a template is created. c. Comparison: The template is then compared with a new sample. d. Match/non match: The system decides if the features extracted from the new Samples are a match or a non-match.

COMPONENTS OF FACE RECOGNITION SYSTEM:

a) **Enrolment Module/Face Detection**: An automated mechanism that scans and captures a digital or an analog image of a living personnel characteristics. Face recognition begins with the detection of faces in a picture, which is done using the HOG approach. The histogram of oriented gradients (HOG) is a type of histogram. The process begins with a black-and-white conversion of an image. A set of surrounding pixels is chosen for each pixel in an image in order to determine how dark that pixel is in comparison to the others. This procedure is repeated for each and every pixel in an image. Every pixel is eventually replaced by arrows. Gradients are arrows that are created by combining the magnitude and angle of the image. The following formulas are used to determine the gradients Gx and Gy for each pixel:

$$Gx\,(x, y) = H\,(x + 1, y) - H\,(x - 1, y)$$

$$Gy\,(x, y) = H\,(x, y + 1) - H\,(x, y - 1)$$

After these calculations, the magnitude and the direction of the gradient are obtained as:

$G(x, y) = \sqrt{Gx}\ (x, y)\ 2 + Gy\ (x, y)\ 2$

$\theta(x, y) = argtan(\ Gy\ (x, y)\ /Gx\ (x, y))$

The magnitude and direction are separated into many cells. A 9- point histogram is created for each cell, and each bin produces the gradient intensity. Once the histogram computation for all cells is complete, four cells are merged to make a block. Thus, overlapping combination is carried out. The 9-point histograms of each cell in a block are concatenated to generate a 36-point feature vector for all four cells in the block. Then, to lessen the effect of changes in contrast between photos of the same face, normalising is done. In this way, the faces can be easily found in any image. If the image size is 128x64, then the total HOG feature is: $Tf = 7 * 15 * 36 = 3780$ (31)

Here, 36 is the feature vector, 7 and 15 are the blocks in horizontal and vertical directions, respectively. The HOG method goes through 8 steps to collect the feature vectors. Those feature vectors obtain the HOG feature according to the input image.

b) **Face Encoding**: Another entity which handles compression, processing, storage and compression of the captured data with stored data. FaceNet is used to extract features from a person's face after it has been detected. It's a convolutional neural network developed by Google's Florian Schroff, Dmitry Kalenichenko, and James Philbin and published in 2015. In general, the CNN is taught to recognise images, objects, and numbers. FaceNet, on the other hand, takes a facial image as input, extracts the feature from convolutional and max-pooling layers, and creates a vector of 128 measurements from fully connected layers. The term embedding refers to the 128 measures that make up a generic representation of a human face. FaceNet uses this embedding to train the neural network classifier's accuracy using the triplet loss function. Three vector variables are fed into the triplet loss function: an anchor, a positive, and a negative. An anchor is the first known image of a person, a positive is another image of the same person, and a negative is a different person's image. The embedding of anchor images in neural networks is trained

to be close to positive embedding and far away from negative embedding. The neural network is trained when the embeddings yield near measurements, and it can create 128 measurements for any face.

c) **Identification Module**: The final step is to compare the test image's embedding to the database image's embedding. SVM, a machine learning technique, can be used to classify the test image with the best match in this scenario. The distance between two data points is calculated using the Euclidean distance formula. The similar technique can be used with picture embeddings. The faces are from the same individual if the distance between these embeddings is small, and vice versa.

d) **Single Triplet Training**: Triplet training steps are used to train the facial recognition system. The provided image of the person is then converted into 128 real-valued integers, resulting in 128 measurements for a single input image in the dataset. This is also known as the dataset image quantification technique. The objective is to use numerous datasets to train the system. The dataset is viewed as a portrait of people's faces, with each individual being represented by a different colour. Direct classifiers are used to turn faces into encodings, which are subsequently saved in order to compare real-time frames.

Finding the name of the person from the encodings: To discover the person's name from the encodings, first find the individual from the dataset using the pre-processed measurements from the training phase. This can be done by comparing the closest measurements to the test image from the dataset provided. This is then followed by the use of any classifier technique, rather than fancy deep learning techniques. In this situation, a simple linear SVM classifier was applied. After the training phase, the classifier takes the measures created by the algorithm and compares them to the closest matched measurement from the dataset, which takes milliseconds to complete. As a result, the classifier's output is the name of the person.

e) **Face Analysis:** Face analysis is a field of detecting human faces from a visual input, extracting feature values, and recognizing who the subjects are. It is widely utilized in areas such as ID verification, access control, and AR face filters/stickers. For tracking facial expressions, we are using Deepface module which is currently available in python. Deep Face comes with a strong facial

attribute analysis module for age, gender, emotion, and race/ethnicity prediction. While Deep Face's facial recognition module wraps existing state-of-the-art models, its facial attribute analysis has its own models.

Proposed Design

About the model

DeepFace is a deep learning facial recognition system created by a research group at Facebook. It identifies human faces in digital images. The program employs a nine-layer neural network with over 120 million connection weights and was trained on four million images uploaded by Facebook users.
the DeepFace method reaches an accuracy of 97.35% ± 0.25% on Labeled Faces in the Wild (LFW) data set where human beings have 97.53%.

Architecture

The DeepFace system consists of four modules: 2D alignment, 3D alignment, frontalization, and neural network. An image of a face is passed through them in sequence, resulting in a 4096-dimensional feature vector representing the face. The feature vector can then be further processed for many different tasks. For example, to identify the face, one can compare it against a list of feature vectors of known faces, and identify the face with the most similar feature vector.

DeepFace uses fiducial point detectors based on existing databases to direct the alignment of faces. The facial alignment begins with a 2D alignment, and then continues with 3D alignment and frontalization. That is, DeepFace's process is two steps. First, it corrects the angles of an image so that the face in the photo is looking forward. To accomplish this, it uses a 3-D model of a face.

2D alignment

The 2D alignment module detects 6 fiducial points on the detected face — the center of the eyes, tip of the nose and mouth location. These points are translated onto a warped image to help detect the face. However, 2D transformation fails to compensate for rotations that are out of place.

3D alignment

In order to align faces, DeepFace uses a generic 3D model wherein 2D images are cropped as 3D versions. The 3D image has 67 fiducial points. After the image has been warped, there are 67 anchor points manually placed on the image to match the 67 fiducial points. A 3D-to-2D camera is then fitted that minimizes losses. Because 3D detected points on the contour of the face can be inaccurate, this step is important.

Frontalization

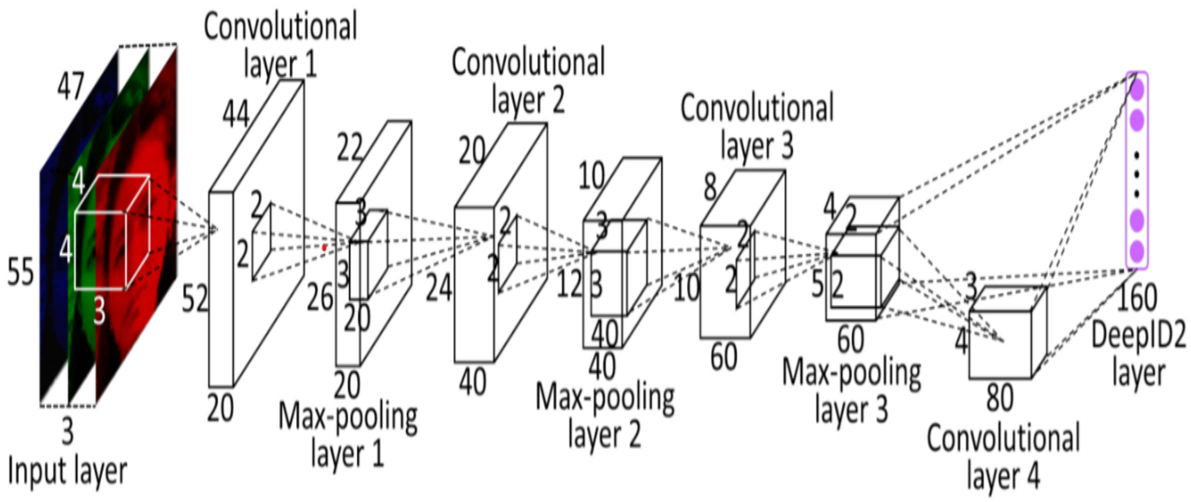
Because full perspective projections are not modeled, the fitted camera is only an approximation of the individual's actual face. To reduce errors, DeepFace aims to warp the 2D images with smaller distortions. Also, the camera P is capable of replacing parts of the image and blending them with their symmetrical counterparts.

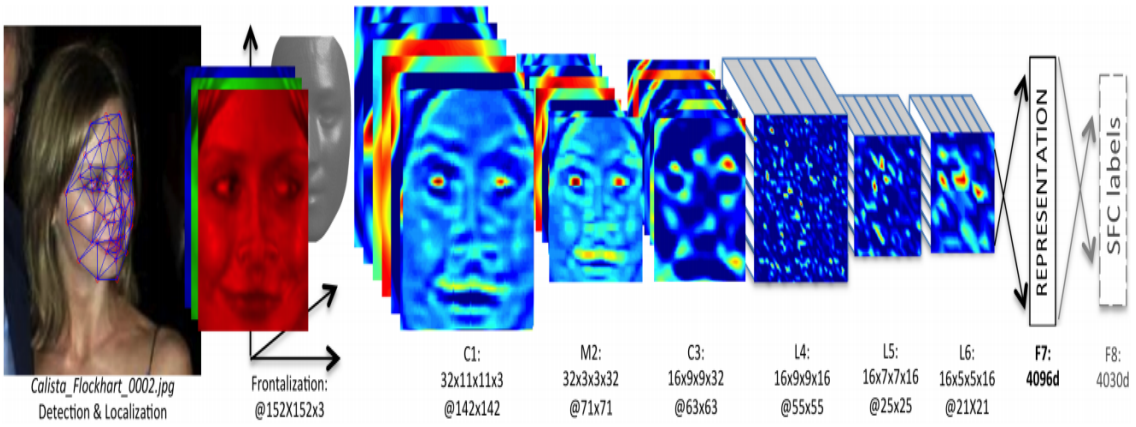
Neural network

The neural network is a sequence of layers, arranged as follows: convolutional layer - max pooling - convolutional layer - 3 locally connected layers - fully connected layer.

The input is an RGB image of the face, scaled to resolution and the output is a real vector of dimension 4096, being the feature vector of the face image.

In the 2014 paper, an additional fully connected layer is added at the end to classify the face image into one of 4030 possible persons that the network had seen during training time.





Development

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime as dt
from deepface import DeepFace

face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')  #This 'XML' file contains a pre-trained model that was created through extensive training and uploaded
#by Rainer Lienhart on behalf of Intel in 2000.
#Rainer's model makes use of the Adaptive Boosting Algorithm (AdaBoost) in order to yield better results and accuracy.

path = './ImagesDB' # path for sample images
```

```
imageList = []
classNames = []
myList = os.listdir(path)   # extracting the list of images for once from the path

date= dt.today()    # date module function
date_str= date.strftime("%d %b, %Y")    # typecasting dateTime type to string format

for cl in myList:
    curframe = cv2.imread(f'{path}/{cl}')   # This library function is from open cv
module , where it reads the pixels of all the images from mylist
    imageList.append(curframe)
    classNames.append(os.path.splitext(cl)[0])

def findEncodings(images):  # returning the encoded values of the list of images by
using face_recognition module
    encodeList = []
    for frame in images:
        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(frame)[0]
        encodeList.append(encode)
    return encodeList   # returning the list of arrays of final encoded values of the
images

encodeListKnown = findEncodings(imageList)
print(len(encodeListKnown))

cap = cv2.VideoCapture(0)   # an opencv module that opens the webcam that
captures an array of incoming frames
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 500)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 500)
cap.set(cv2.CAP_PROP_FPS, 60)
# print(cap.get(cv2.CAP_PROP_FPS))
```

```
# print(cap.get(cv2.CAP_PROP_FRAME_WIDTH))

while True: # event loop
 success, frame = cap.read()
 result = DeepFace.analyze(frame, actions=['emotion'], enforce_detection=False) #
DeepFace.analyze uses the haarcascade_frontalface_default.xml file to map the
current face pixels with its predefined values
 '''Haar cascade is an algorithm that can detect objects in images, irrespective of their
scale in image and location.
 This algorithm is not so complex and can run in real-time. We can train a
haar-cascade detector to detect various
 objects like cars, bikes, buildings, fruits, etc. '''

 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
 faces = face_cascade.detectMultiScale(gray, 1.1, 4)

 for (x, y, w, h) in faces:
  cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
 font = cv2.FONT_HERSHEY_SIMPLEX
 cv2.putText(frame, result['dominant_emotion'], (50, 50),font, 3, (0, 0, 255), 2,
cv2.LINE_4)
 #cv2.imshow('Original Video', frame)
 frameS = cv2.resize(frame, (0, 0), None, 0.25, 0.25)
 frameS = cv2.cvtColor(frameS, cv2.COLOR_BGR2RGB)
 facesCurFrame = face_recognition.face_locations(frameS)
 encodesCurFrame =face_recognition.face_encodings(frameS, facesCurFrame)

 for encodeFace, faceLoc in zip(encodesCurFrame,facesCurFrame):

  matches =face_recognition.compare_faces(encodeListKnown,encodeFace)
  faceDis =face_recognition.face_distance(encodeListKnown,encodeFace)
  matchIndex = np.argmin(faceDis)
```

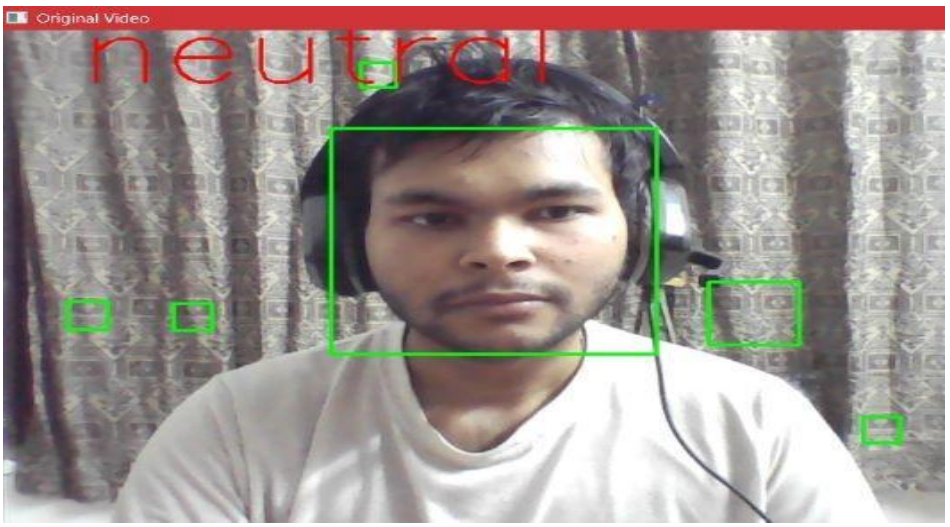
```
 if matches[matchIndex]:

  name = classNames[matchIndex]. upper ()
  y1, x2, y2, x1 = faceLoc
  y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
  cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0),2)   # graphics that outline the part of face
  cv2.rectangle(frame, (x1, y2 - 35), (x2+50, y2), (0, 255, 0), cv2.FILLED)
  cv2.putText(frame, name, (x1 + 5, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
  cv2.rectangle(frame, (x1 -30, y2), (x2 + len(date_str) + 40, y2+40), (255, 0, 0), cv2.FILLED)
  cv2.putText(frame, date_str, (x1 - 30, y2 + 30), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)

 cv2.imshow('Webcam', frame)
 key = cv2.waitKey(1)
 if key == ord('e'):    # Press 'e' to escape the window
  break
```

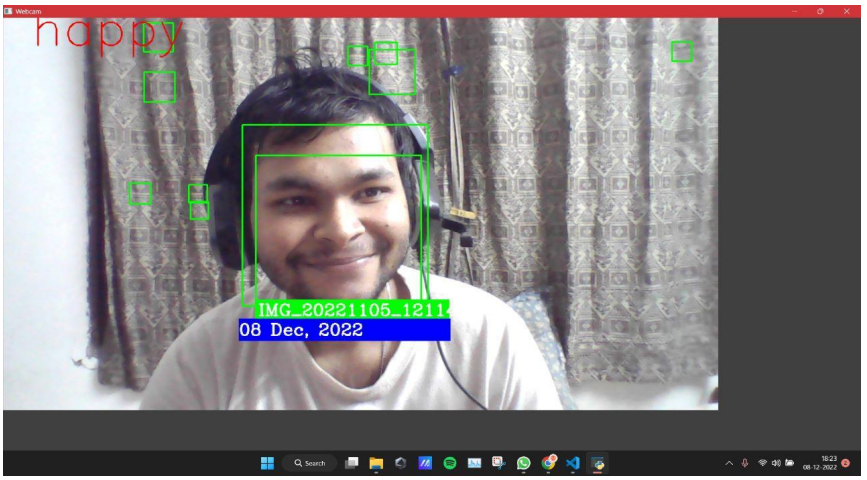
Experimental Result

1) **Neutral**




```
{
  'emotion': {
    'angry':  0.000577753,
    'disgust':  1.48E-10,
    'fear':  0.00388789,
    'happy':  0.003795712,
    'sad':  0.028636202,
    'surprise':  0.000585135,
    'neutral':  99.9625206,
  },
  'dominant_emotion':  'neutral'
}
```

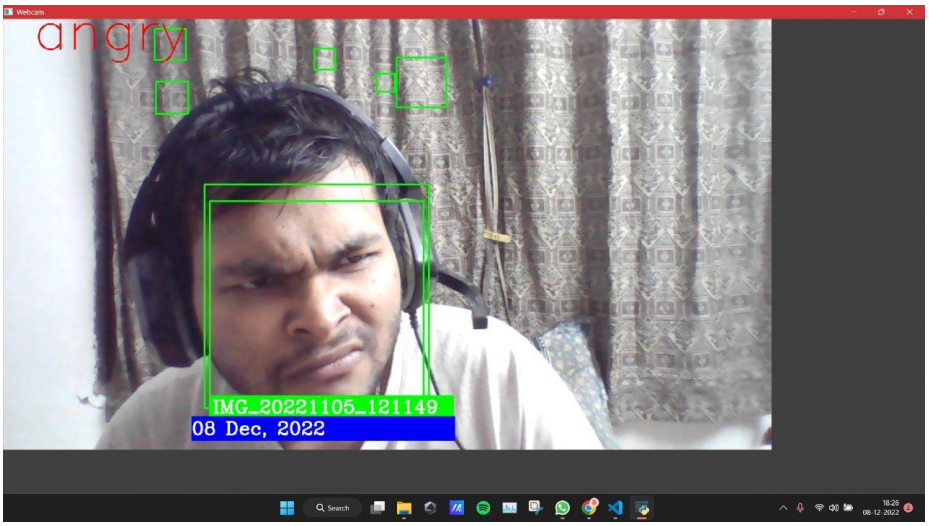

2) **Happy**




```
{
  'emotion':{
```

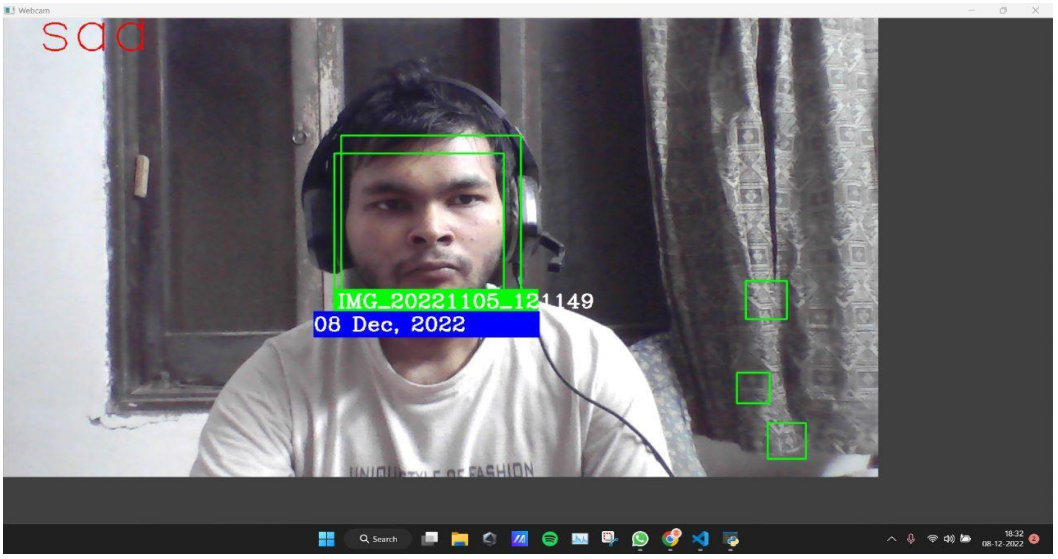
```
    'angry': 0.000000017,
    'disgust': 1.07E-14,
    'fear': 0.000000049,
    'happy': 99.99995232,
    'sad': 0.0000437,
    'surprise': 8.81E-09,
    'neutral': 0.00000723,
  },
  'dominant_emotion': 'happy'
}
```

3) **Angry**



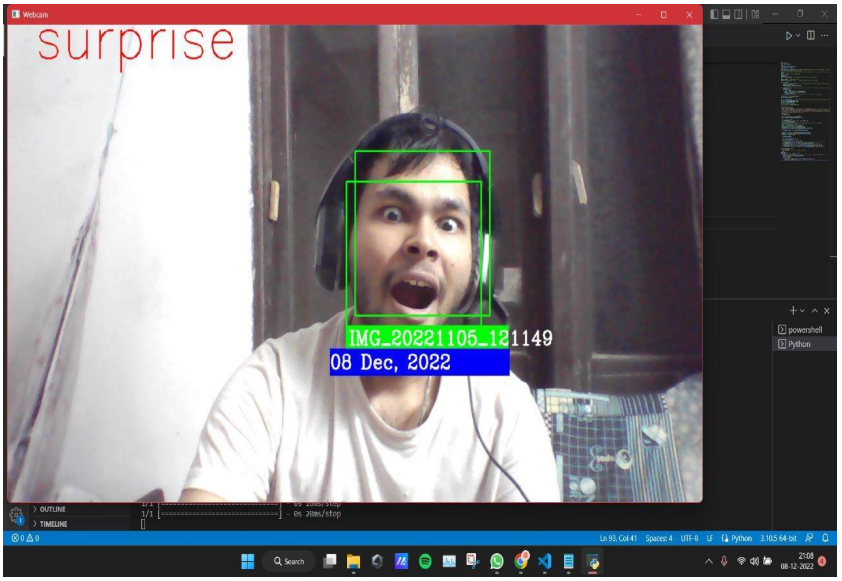

```
{
  'emotion':{
    'angry': 99.59673283,
    'disgust': 0.003035753,
    'fear': 0.333397017,
    'happy': 0.039892905,
    'sad': 0.023209352,
    'surprise': 0.0000908,
    'neutral': 0.003641219,
  },
  'dominant_emotion': 'angry'
}
```

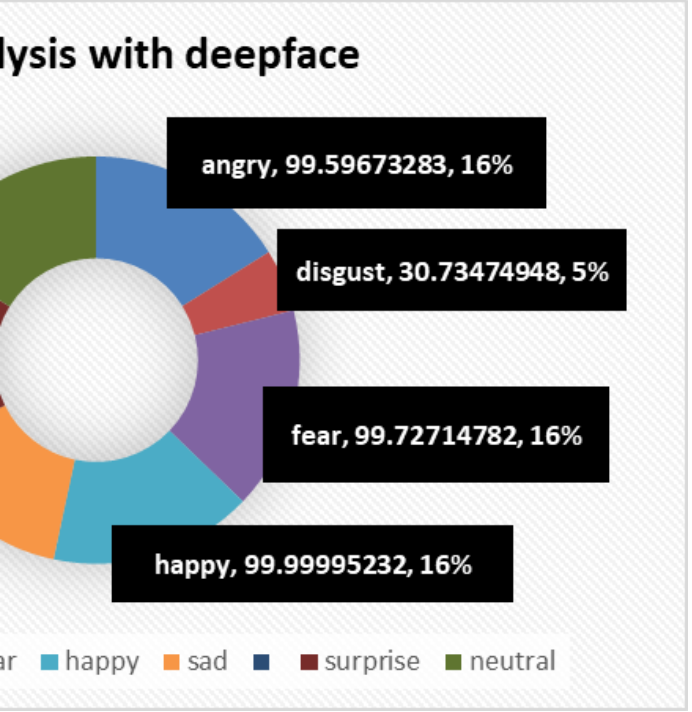
4) **Sad**




```
{
  'emotion':{
    'angry': 0.213653268,
    'disgust': 0.053344853,
    'fear': 9.604147822,
    'happy': 0.066823699,
    'sad': 88.53947528,
    'surprise': 6.26E-08,
    'neutral': 1.52255334,
  },
  'dominant_emotion': 'sad'
}
```


5) **Surprise**






```
{
  'emotion':{
    'angry':  0.000170726,
    'disgust':  1.45E-124,
    'fear':  0.051017484,
    'happy':  0.0000384,
    'sad':  3.67E-08,
    'surprise':  99.94876385,
    'neutral':  0.0000128,
  },
  'dominant_emotion':  'surprise'
}
```


Future Scope

Sentiment analysis using face recognition is a promising area of research that has the potential to revolutionize the way in which we interact with customers and understand their feelings. With further advances in facial recognition technology, sentiment analysis using face recognition can be used to identify emotions such as happiness, sadness, surprise, anger, and fear. This could be particularly useful in the customer service industry where it could be used to analyze customer behavior and provide better customer service. Additionally, this technology could be used to study the impact of marketing campaigns, advertisements, or other experiences on customers' emotions. Finally, it could be used to develop more accurate psychological assessments and to better understand mental health issues. In the future, sentiment analysis using face recognition can help businesses and organizations better understand their customers and build better relationships with them.

Conclusion

Sentiment analysis using face recognition is an innovative way of understanding human

emotions and sentiments. It can be used in a variety of applications such as customer service, marketing, and video surveillance. By combining facial recognition technology with natural language processing, it can accurately detect emotions and sentiments in real-time. With its fast and accurate results, sentiment analysis using face recognition can be an effective tool for gaining insights into customer sentiment and improving customer experience.

REFERENCES

- Andrejevic, M., & Selwyn, N. (2020). Facial recognition technology in schools: Critical questions and concerns. Learning, Media and Technology, 45(2), 115-128.
- Baron, R. J. (1981). Mechanisms of human facial recognition. International Journal of Man-Machine Studies, 15(2), 137-178.
- Benton, A. (1990). Facial recognition 1990. Cortex, 26(4), 491-499.
- Hamann, K., & Smith, R. (2019). Facial recognition technology. CRIM. JUST, 9. Hamann, K., & Smith, R. (2019). Facial recognition technology. CRIM. JUST, 9.

https://pypi.org/

https://pypi.org/project/documentation/

- Kaur, P., Krishan, K., Sharma, S. K., & Kanchan, T. (2020). Facial recognition algorithms: A literature review. Medicine, Science and the Law, 60(2), 131-139.
- Khan, M., Chakraborty, S., Astya, R., & Khepra, S. (2019, October). Face Detection and Recognition Using OpenCV. In 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) (pp. 116-119). IEEE.
- Khan, S., Akram, A., & Usman, N. (2020). Real time automatic attendance system for face recognition using face API and OpenCV. Wireless Personal Communications, 113(1), 469-480.
- Khattak, A., Asghar, M. Z., Ali, M., & Batool, U. (2022). An efficient deep learning technique for facial emotion recognition. Multimedia Tools and Applications, 81(2), 1649-1683.

- Mann, M., & Smith, M. (2017). Automated facial recognition technology: Recent developments and approaches to oversight. University of New South Wales Law Journal, The, 40(1), 121-145.
- Mellouk, W., & Handouzi, W. (2020). Facial emotion recognition using deep learning: review and insights. Procedia Computer Science, 175, 689-694.
- Smith, M., & Miller, S. (2022). The ethical application of biometric facial recognition technology. Ai & Society, 37(1), 167-175.
- Suzon, A. K. (2022). Face mask detection in real time using python.
- Wakurdekar, S., Rikhra, A., Sharma, M., & Singh, S. REAL TIME FACIAL RECOGNITION USING OPENCV PYTHON TECHNIQUES.
- Woodward Jr, J. D., Horn, C., Gatune, J., & Thomas, A. (2003). Biometrics: A look at facial recognition. RAND CORP SANTA MONICA CA.