EMOTION DETECTION USING FACIAL RECOGNITION

Project report in partial fulfilment of the requirement for the award of the degree of

Bachelor of Technology

In

CSE (IOT, BCT, CYS)

Submitted By

Deeptangshu Saha	University Roll No. 12021002029109
Bhaskar Das	University Roll No. 12021002029110
Nilay Mukherjee	University Roll No. 12021002029112
Himadri Dutta	University Roll No. 12021002029104
Ankit Kumar	University Roll No. 12021002029111
Shubhayu Mandal	University Roll No. 12021002029154
Rupam Ghosh	University Roll No. 12021002029113
Naimur Hossaion Sk	University Roll No. 12021002029080
Sampad Narayan Bhattacharjee	University Roll No. 12021002029106
Shantanu Bera	University Roll No. 12021002029141

Under the guidance of

Prof. Sweta Saha

Department of CSE (IOT, BCT, CYS)

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160.

CERTIFICATE

This is to certify that the project titled **Emotion Detection using Facial Recognition** submitted by Deeptangshu Saha (University Roll No. 12021002029109), Bhaskar Das (University Roll No. 12021002029110), Nilay Mukherjee (University Roll No. 12021002029112), Himadri Dutta (University Roll No. 12021002029104), Ankit Kumar (University Roll No. 12021002029111), Shubhayu Mandal (University Roll No. 12021002029154), Rupam Ghosh (University Roll No. 12021002029113), Naimur Hossaion Sk (University Roll No. 12021002029080), Sampad Narayan Bhattacharjee (University Roll No. 12021002029106) and Shantanu Bera (University Roll No. 12021002029141) students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfilment of requirement for the degree of Bachelor of Computer Science (IOT, BCT, CYS), is a Bonafede work carried out by them under the supervision and guidance of Prof. Sweta Saha during 7th Semester of academic session of 2024 - 2025. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

Signature of Guide

Signature of Head of Department

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide Prof. Sweta Saha of the Department of CSE (IOT, BCT, CYS), UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Deeptangshu Saha

Bhaskar Das

Nilay Mukherjee

Rupam Ghosh

Himadri Dutta

Ankit Kumar

Shubhayu Mandal

Naimur Hossaion Sk

Sampad Narayan Bhattacharjee

Shantanu Bera

TABLE OF CONTENTS

ABSTRACT

This project develops a real-time face and emotion detection system using Python's DeepFace library, integrating deep learning and computer vision techniques to analyse facial expressions and classify emotions such as happiness, sadness, anger, fear, surprise, disgust, and neutrality. The system uses pre-trained models like VGG-Face and Facenet for face detection and emotion classification, while OpenCV handles real-time processing. The application is designed to work effectively in diverse environments, adapting to variations in lighting, pose, and background.

The system has practical applications in healthcare, education, and customer service, helping monitor emotional states, assess engagement, and enhance user interactions. However, challenges like dataset bias, overlapping emotions, and privacy concerns need addressing for more equitable and ethical use. Future improvements include expanding emotion categories, enhancing robustness, and incorporating privacy-preserving measures. This project lays the foundation for more intelligent human-computer interactions through real-time emotion recognition.

INTRODUCTION

Emotions are central to human interactions, influencing decision-making, communication, and behaviour. Among the various ways to interpret emotions, facial expressions stand out as a universal, non-intrusive, and reliable method. These expressions reveal insights into emotional states, making them a valuable resource for both understanding and responding to human behaviour. The field of emotion detection using facial recognition harnesses advancements in computer vision and artificial intelligence to analyse facial expressions and infer emotions, offering a powerful tool for real-time emotional analysis.

At the foundation of this field lies the Facial Action Coding System (FACS), which identifies facial muscle movements linked to emotions like happiness, sadness, anger, fear, surprise, disgust, and neutrality. Modern systems build on this framework, leveraging machine learning and deep learning algorithms to process facial data and classify emotions with remarkable precision. By analysing subtle facial cues and patterns, these systems deliver insights that are both accurate and actionable.

Emotion detection using facial recognition has garnered significant attention due to its potential to revolutionize industries. In healthcare, it aids in mental health monitoring and early diagnosis of emotional disorders. In education, it enhances learning by adapting content to students' emotional states. Customer service applications include analysing customer satisfaction, while in security, it helps identify suspicious or aggressive behaviour in real time.

Despite its promise, the field faces challenges such as cultural variability in emotional expressions, overlapping emotional states, and ethical concerns related to privacy and consent. Addressing these challenges is essential for wider adoption and ethical deployment.

This paper explores the methodologies, applications, and challenges of emotion detection using facial recognition, highlighting its transformative potential while emphasizing the need for further research to address its limitations and ensure ethical implementation.

LITERATURE SURVEY

Some of the facial recognition systems are:

- Eigen faces algorithm with PCA
- Haar-Cascade Classifiers Viola Jones
- Support Vector Machine (SVM)
- DeepFace for emotion detection

2.1 Eigen faces algorithm with PCA

2.1.1 Theory

Eigenfaces are a set of eigenvectors used in the computer vision problem of human face recognition. Eigenfaces assume ghastly appearance. They refer to an appearance-based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner. Specifically, the eigen faces are the principal components of a distribution of faces, or equivalently, the eigenvectors of the covariance matrix of the set of face images, where an image with N x N pixels is considered a point (or vector) in N2-dimensional space. The idea of using principal components to represent human faces was developed by Sirovich and Kirby[3].

2.1.2 Drawbacks

The tests conducted on various subjects in different environments shows that this approach has limitations over the variations in light, size and in the head orientation, nevertheless this method showed very good classifications of faces(85% success rate). A good recognition system should have the ability to adapt over time. Reasoning about images in face space provides a means to learn and subsequently recognize new faces in an unsupervised manner. When an image is sufficiently close to face-space (i.e., it is face-like) but is not classified as one of the familiar faces, it is initially labelled as ”unknown” . The computer stores the pattern vector and the corresponding unknown image. If a collection of ”unknown” pattern vectors cluster in the pattern space, the presence of a new but unidentified face is postulated. A noisy image or partially occluded face would cause recognition performance to degrade. The eigenface approach does provide a practical solution that is well fitted to the problem of face recognition. It is fast, relatively simple, and has been shown to work well in constrained environment[4].

2.2 Haar-Cascade Classifiers Viola Jones

2.2.1 Theory

Motivated by the challenge of face detection, P. Viola and M. Jones proposed an object detector framework using Haar-like features, which has been widely used by other works not only for face detection, but also for object locations. Thanks to the Open Computer Vision Library implementation

[17], the general object detector framework has become popular and motivated the community to generate their own object classifiers. These classifiers use Haar-like features that are applied over the image. Only those image regions, called sub-windows, that pass through all the stages of the detector are considered to contain the target object. Fig. 3 shows the detection cascade schematic with N stages. The detection cascade is designed to eliminate a large number of negative examples with a little processing.[5]

2.2.2 Drawbacks

Based on the experiment before the system can classify and detect the face in many cases and conditions. With six types of condition and five times of trial, obtain the accuracy by 75,33% by using V-J and 80,22% by using HOG. V-J algorithm can detect frontal face very well in images, regarding of their scale, pose, makeup, expression, and illumination, but rather difficult to detect the face who have occlusions like using helm, eyeglass, and mask. The V-J algorithm can perform in real-time on many applications and hardware, the main problem with Haar cascades is in the parameter called detect multiscale and scale factor. If the scale factor is too low, many pyramid layers will be evaluated, this will help to detect more than one faces in images, but the detection process will be slower and increases the false-positive detection rate. On the other hand, if scale factor is too large, it cannot detect the face in small pixel. The recommended size for datasets at least above 250*250 pixels. The HOGs more accurate than V-J for face detection, it can represent local appearance very well[6].

2.3 Support Vector Machine (SVM)

2.3.1 Theory

Support vector machine is a supervised learning model with associated learning algorithms, it analyses data for classification and regression analysis. SVM training algorithm differentiates categories making it non probabilistic binary linear classifier. SVM separates categories by a clear gap as wide as possible also known as margin. SVM also performs non-linear classification using Kernel trick. SVM is capable of delivering higher classification accuracy. SVM can be used to detect text, digit, image classification and object detection. SVM constructs hyper-plane between 2 or more clusters; hyper-plane can be used to detect outliers among data. To achieve optimal parameter setting SVM requires extensive cross validation commonly known as model selection. The choice of a kernel function, the standard deviation of the Gaussian kernel, training data, relative weights of slack variable impact the overall results. SVM minimizes the empirical classification error and maximizes the geometric margin. SVM is based on Structural Risk Minimization (SRM). SVM maps input vector to a higher dimensional space with maximal separating hyper plane.[2]

2.3.2 Drawbacks

In the previous discussion, we learned that the superiority of SVM made it a significant development in the fields of modern machining, predicting protein and face detection. We also learned that the model of SVM is only determined only by the support vectors, so it is very convenient for us to train. But SVM also has some drawbacks[6], for instance, the training efficiency of the existing SVM for large-scale sample data sets of practical problems cannot reach the ideal training efficiency. Therefore, the future development of SVM may concentrate more on how to further improve the SVM algorithm. In addition, despite the fact that SVM has outstanding advantages in theory, compared with theoretical research, applied research on application still lags behind. Therefore, how to apply SVM more in people’s daily life and explore new application areas of SVM will be the emphasis of future research.[7]

2.3.3 DeepFace

DeepFace, originally developed for facial recognition, has been adapted for emotion detection by leveraging its deep learning capabilities and pre-trained models such as VGG-Face and Facenet. It analyses facial features to classify emotions like happiness, sadness, anger, fear, surprise, disgust, and neutrality. This technology is widely applied in areas such as healthcare, where it aids in mental health monitoring, education for evaluating student engagement, and customer service to enhance interactions based on emotional feedback. Its real-time processing capability and ease of integration into existing systems make it a valuable tool for applications requiring dynamic emotional insights.

2.3.4 Drawbacks

However, emotion detection using DeepFace is not without its challenges. Bias in training datasets can lead to reduced accuracy for certain demographic groups, raising concerns about fairness and inclusivity. The interpretation of complex or overlapping emotions remains a technical hurdle, particularly in real-world scenarios with diverse facial expressions. Additionally, ethical concerns such as privacy, consent, and the potential misuse of emotional data pose significant barriers to adoption. Addressing these drawbacks is essential for ensuring that the technology is applied responsibly and equitably across industries.

PROBLEM STATEMENT

In a world where effective human-computer interaction is increasingly essential, understanding human emotions has emerged as a critical area in enhancing user experience across various applications. Traditional approaches often rely on direct user input, which can be intrusive or unreliable. However, these methods lack real-time adaptability and the intuitive recognition of emotions that would lead to more personalized and meaningful user interactions.

Facial expressions are one of the most powerful indicators of human emotions and can be analysed to interpret emotional states such as happiness, sadness, anger, surprise, fear, and neutrality. The challenge lies in developing a system capable of accurately detecting and classifying these emotions from live facial data, as expressions can vary widely among individuals due to cultural, demographic, and personal differences. Furthermore, factors like lighting, angle, and background noise make the detection process technically complex.

The goal of this project is to build an emotion detection system using facial recognition that can classify basic emotions based on real-time facial expressions. Using machine learning and computer vision algorithms, the system will process live or recorded facial data, analyse distinct facial landmarks, and categorize emotions. The solution has applications in fields such as customer service, mental health assessment, e-learning, and marketing, providing valuable insights to adapt experiences according to users' emotional feedback.

This project will leverage Python libraries like OpenCV, TensorFlow, Keras and DeepFace to design and train the model for emotion detection, offering a real-time, non-intrusive solution to enhance the depth and adaptability of human-computer interaction.

PROPOSED SOLUTION

I developed a program that detects faces and recognizes emotions in real time using Python's DeepFace library. This application harnesses the capabilities of pre-trained deep learning models, such as VGG-Face and Facenet, provided by DeepFace to achieve accurate facial detection and emotion classification. By capturing video feed through a webcam, the program identifies faces in each frame and processes them to classify emotions such as happiness, sadness, anger, fear, surprise, disgust, and neutrality. The integration of computer vision libraries like OpenCV ensures efficient face detection and seamless frame-by-frame analysis, enabling the program to operate in real time. Once a face is detected, DeepFace analyses its features, maps them to trained datasets like FER2013, and predicts the associated emotional state. The results are dynamically displayed on the live video feed, making the interface interactive and user-friendly. The program employs preprocessing techniques to optimize performance. Detected facial regions are resized and normalized to ensure compatibility with the DeepFace models, improving the accuracy of emotion recognition. It is robust against variations in lighting, pose, and background, allowing it to function effectively across diverse environments. For instance, the use of OpenCV’s Haar cascades or DNN-based face detection enhances the program’s ability to localize faces accurately, even in challenging scenarios. The design emphasizes modularity and scalability, allowing for future integration with additional facial analysis tasks, such as age or gender detection. This application has significant potential in real-world scenarios. In healthcare, it could be used for mental health monitoring, helping therapists track patients’ emotional states during sessions or over time. In education, the program could evaluate student engagement during online classes and adapt teaching methods to suit their emotional states. Customer service systems could integrate this tool to analyse client emotions and tailor interactions accordingly, improving customer satisfaction. Additionally, the program could enhance user experiences in gaming or human-computer interaction by enabling systems to respond to emotional cues dynamically. Despite its strengths, there are challenges that must be addressed. The accuracy of emotion detection depends heavily on the quality and diversity of training datasets. Biases in these datasets could lead to lower accuracy for certain demographic groups, raising fairness concerns. Moreover, the interpretation of complex or overlapping emotions remains a technical hurdle. For example, distinguishing between similar emotional states, such as fear and surprise, can be challenging in real-time scenarios. Ethical considerations are equally important, particularly regarding privacy and consent. Since facial data is sensitive, measures must be implemented to ensure secure handling and storage of data, as well as transparency about how the data is used.

Future improvements to the program could include incorporating larger and more diverse datasets to reduce bias, expanding the range of recognized emotions, and enhancing the system’s ability to handle dynamic, multi-emotional scenarios. Privacy-preserving mechanisms, such as anonymization or on-device processing, could further strengthen the program’s ethical framework. By addressing these challenges, the program could be refined into a powerful tool for emotion detection, with applications in fields ranging from healthcare to entertainment and beyond.

EXPERIMENTAL SETUP AND RESULT ANALYSIS

Requirements:

- OpenCV
- Face Recognition
- DeepFace
- Keras
- Tensorflow
- Pillow
- NumPy
- Pandas
- DLib

Code[1]:

```
import cv2
import face_recognition
import numpy as np
from simple_facerec import SimpleFacerec
from attendance import mark_attendance
from deepface import DeepFace

# Encode faces from a folder
sfr = SimpleFacerec()
sfr.load_encoding_images("images/")

frame_resizing = 0.25
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')


def detect_known_faces(frame_in):
    small_frame = cv2.resize(frame_in, (0, 0), fx=frame_resizing,
fy=frame_resizing)
    # Find all the faces and face encodings in the current frame of
video
    # Convert the image from BGR color (which OpenCV uses) to RGB
color (which face_recognition uses)
    rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
```

```
    face_locations_in =
face_recognition.face_locations(rgb_small_frame)
    face_encodings =
face_recognition.face_encodings(rgb_small_frame, face_locations_in)
    known_face_encodings = np.load('ImageEncoding.npy')
    image_names = open("ImageNames.txt", "r")
    known_face_names = []
    for known_face_names_var in image_names:
        known_face_names.append(known_face_names_var)
    face_names_in = []

    for face_encoding in face_encodings:
        # See if the face is a match for the known face(s)
        matches =
face_recognition.compare_faces(known_face_encodings, face_encoding)
        name_in = "Unknown "

        # # If a match was found in known_face_encodings, just use
the first one.
        # if True in matches:
        #     first_match_index = matches.index(True)
        #     name = known_face_names[first_match_index]
        # Or instead, use the known face with the smallest distance
to the new face
        face_distances =
face_recognition.face_distance(known_face_encodings, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name_in = known_face_names[best_match_index]
            mark_attendance(name_in[:-1])
        face_names_in.append(name_in[:-1])

    # Convert to numpy array to adjust coordinates with frame
resizing quickly
    face_locations_in = np.array(face_locations_in)
    face_locations_in = face_locations_in / frame_resizing
    return face_locations_in.astype(int), face_names_in


# def mark_attendance(name_in):


# Load Camera
cap = cv2.VideoCapture(0)
```

```
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1080)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 1920)
# cap = cv2.resize(cap, (1920, 1080))

while True:
    ret, frame = cap.read()
    # emotion_detect(ret, frame)
    # Detect Faces
    # faces = face_cascade.detectMultiScale(gray_frame,
scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))
    face_locations, face_names = detect_known_faces(frame)
    for face_loc, name in zip(face_locations, face_names):
        y1, x2, y2, x1 = face_loc[0], face_loc[1], face_loc[2],
face_loc[3]
        face_roi = frame[y1:y2, x1:x2]
        result = DeepFace.analyze(face_roi, actions=['emotion'],
enforce_detection=False)
        emot = result[0]['dominant_emotion']
        cv2.putText(frame, name, (x1, y1 - 10),
cv2.FONT_HERSHEY_DUPLEX, 1, (0, 0, 200), 2)
        cv2.putText(frame, emot, (x1, x2 - 10),
cv2.FONT_HERSHEY_DUPLEX, 1, (0, 0, 200), 2)
        cv2.rectangle(frame, (x1, y1), (x2, y2), (200, 0, 0), 4)

        # cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 0, 255), 4)

    cv2.imshow("FaceDetect", frame)

    key = cv2.waitKey(1)
    if key == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

Code[2]:

```
import os
from datetime import datetime


def mark_attendance(name_in):
    now = datetime.now()
    filename = now.strftime("%Y_%m_%d") + ".csv"
```

```
        name_list = []
        if os.path.exists(filename):
            with open(filename, 'r+') as f:
                my_data_list = f.readlines()
                for line in my_data_list:
                    entry = line.split(',')
                    name_list.append(entry[0])
                if name_in not in name_list:
                    f.writelines("\n" + name_in + "," +
    now.strftime('%H:%M:%S'))
        else:
            with open(filename, 'a+') as f:
                f.writelines("Name,Time")
                my_data_list = f.readlines()
                for line in my_data_list:
                    entry = line.split(',')
                    name_list.append(entry[0])
                if name_in not in name_list:
                    f.writelines("\n" + name_in + "," +
    now.strftime('%H:%M:%S'))
```

Result Analysis:

The above is the driver code and works as expected. Though the code works as expected it also identifies the faces which are in the database with an accuracy of more than 80%. The only drawback being that the face size needs to be of a certain size or it won't detect the face at all. However, the current distance is not so ideal and thus it needs to be improved further.

CONCLUSION & FUTURE SCOPE

The real-time face and emotion detection program built using Python's DeepFace library offers significant potential in various fields, including mental health, education, customer service, and entertainment. By utilizing deep learning models like VGG-Face and Facenet, along with computer vision techniques from OpenCV, the program accurately detects faces and classifies emotions such as happiness, sadness, anger, fear, and surprise. Its ability to process video feeds in real time makes it a versatile tool, enhancing human-computer interaction and providing valuable emotional insights. However, challenges such as dataset bias, complex overlapping emotions, and privacy concerns remain, which must be addressed to ensure the system's fairness and responsible use.

To improve the program, future developments should focus on expanding and diversifying the training datasets to reduce biases and ensure accuracy across all demographics. Additionally, increasing the range of emotions detected and enhancing the system’s robustness to handle varying real-world conditions—like lighting changes, extreme angles, and occlusions—would make the system more reliable. Privacy concerns could be addressed by incorporating privacy-preserving measures, such as on-device processing and data encryption, ensuring user data is protected and compliant with regulations. Further advancements could include integrating emotion detection with other modalities, such as voice or gesture recognition, to improve accuracy and context understanding. This could lead to more adaptive applications, such as personalized learning environments or emotion-aware customer service systems. Ultimately, expanding the capabilities of the system while maintaining ethical standards will be key to maximizing its potential in transforming how machines interact with human emotions.

BIBLIOGRAPHY

[1] Manop Phankokkruad and Phichaya Jaturawat. Influence of facial expression and viewpoint variations on face recognition accuracy by different face recognition algorithms. In 2017 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pages 231–237. IEEE, 2017.

[2] Sanmoy Paul and Sameer Kumar Acharya. A comparative study on facial recognition algorithms. In e-journal-First Pan IIT International Management Conference–2018, 2020.

[3] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. Josa a, 4(3):519–524, 1987.

[4] Y Vijaya Lata, Chandra Kiran Bharadwaj Tungathurthi, H Ram Mohan Rao, A Govardhan, and LP Reddy. Facial recognition using eigenfaces by pca. International Journal of Recent Trends in Engineering, 1(1):587, 2009.

[5] Rafael Padilla, CFF Costa Filho, and MGF Costa. Evaluation of haarcascade classifiers designed for face detection. World Academy of Science, Engineering and Technology, 64:362–365, 2012.

[6] C Rahmad, R An Asmara, DRH Putra, I Dharma, H Darmono, and I Muhiqqin. Comparison of viola-jones haarcascade classifier and histogram of oriented gradients (hog) for face detection. In IOP conference series: materials science and engineering, page 012038. IOP Publishing, 2020.

[7] Zhao Jun. The development and application of support vector machine. In Journal of Physics: Conference Series, page 052006. IOP Publishing, 2021.