**STRESS DETECTION THROUGH FACIAL EXPRESSION USING MACHINE LEARNING AND IMAGE PROCESSING**

#### K. Lohith 21951A1238

#### A. Jashwanth 21951A1232

#### S. Prudhvi 22559A1207

**STRESS DETECTION THROUGH FACIAL EXPRESSION USING MACHINE LEARNING AND IMAGE PROCESSING**

***A Project Report***

***Submitted in Partial Fulfilment of the***

***Requirements for the Award of the Degree Of***

**Bachelor of Technology in**

**Information Technology**

**By**

Kowkuntla Lohith 21951A1238

Ajmeera Jashwanth 21951A1232

Sunki Prudhvi Raj 22559A1207



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**INSTITUTE OF** **AERONAUTICAL ENGINEERING**

**(Autonomous)**

**Dundigal, Hyderabad – 500 043, Telangana**

##### **April 2024**

##### © 2024, Lohith, Jashwanth, Prudhvi. All rights reserved.

Page| i

**DECLARATION**

We certify that

1. the work contained in this report is original and has been done by us under the guidance of our supervisor.
2. the work has not been submitted to any other Institute for any degree or diploma.
3. we have followed the guidelines provided by the Institute in preparing the report.
4. we have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

###### Place: Signature of the Student

###### Date: Roll No. 21951A1238

###### 21951A1232

###### 22559A1207

Page | ii

**CERTIFICATE**

This is to certify that the project report entitled **stress detection through facial expression using machine learning and image processing** submitted by **Mr.** **Kowkuntla Lohith, Mr. Ajmeera Jashwanth** and **Mr. Sunki Prudhvi Raj** to the Institute of Aeronautical Engineering, Hyderabad, in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **Information Technology** is a Bonafide record of work carried out by them under our guidance and supervision. In whole or in parts, the contents of this report have not been submitted to any other institute for the award of any Degree.

###### Supervisor Head of the Department

Ms. C S L Vijaya Durga Dr. M. Purushotham Reddy

Assistant Professor Professor

Date:

Page | iii

**APPROVAL SHEET**

This project report entitled **stress detection through facial expression using machine learning and image processing** by **Mr.** **Kowkuntla Lohith, Mr. Ajmeera Jashwanth** and **Mr. Sunki Prudhvi Raj** is approved for the award of the Degree Bachelor of Technology in Information Technology.

**Examiners Supervisor**

**Ms. Akula Rajitha**

**Principal**

**Dr. L V Narasimha Prasad**

**Date:**

**Place:Dundigal**

Page | iv

**ACKNOWLEDGMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete

without the mention of the people who made it possible and whose constant guidance and

encouragement crown all the efforts with success. I think out college management and respected Sri M. Rajashekar Reddy, Chairman, IARE, Dundigal for providing me with the necessary infrastructure to conduct the project work.

I express my sincere thanks to Dr. L. V. Narasimha Prasad, Professor and Principal who has been a great source of information for my work, and, Professor and Head, Dr. M. Purushotham Reddy Department of IT, for extending his support to carry on this project work.

I am especially thankful to our supervisor Ms. C S L Vijaya Durga, Assistant Professor, Department of IT, for her internal support and professionalism who helped me in shaping the project into a successful one. I take this opportunity to express my thanks to one and all who directly or indirectly helped me in bringing the effort to present form.

#### K. Lohith 21951A1238

#### A. Jashwanth 21951A1232

#### S. Prudhvi 22559A1207

Page | v

**ABSTRACT**

Stress is a prevalent health issue with significant implications for both physical and mental well-being. Accurate and timely stress detection can facilitate early intervention and improve overall health outcomes. This paper presents a novel system for real-time stress detection in IT professionals, combining convolutional neural networks (CNNs) and image processing techniques with personalized counseling and periodic surveys. Our system addresses the limitations of previous stress detection systems by incorporating live detection capabilities and providing tailored stress management strategies. By analyzing facial expressions and other visual cues using CNNs, we aim to accurately assess stress levels in individuals. Additionally, periodic surveys will gather additional data to gain a comprehensive understanding of employees' stress experiences. This research offers a promising solution for improving the well-being of IT professionals and fostering a healthier work environment. By early detection and intervention, we can mitigate the negative consequences of stress and enhance employee satisfaction and productivity.

**Keywords:** Stress Detection, Facial Expressions, CNN, Image Processing, IT Professionals.

Page | vi

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| Project Description |  | i |
| Declaration |  | ii |
| Certificate |  | iii |
| Approval Sheet |  | iv |
| Acknowledgement |  | v |
| Abstract |  | vi |
| Contents |  | vii |
| List of Figures |  | ix |
| **Chapter 1** | **Introduction** | **1** |
|  | 1.1 Introduction | 1 |
|  | 1.2 Existing System | 2 |

1.2.1 Limitations of Existing System 2

* 1. [Proposed System](#_bookmark4) 2
     1. Advantages of Proposed System 2

###### [Chapter 2 Literature Survey](#_bookmark5) 4

* 1. [Literature review](#_bookmark6) 4
  2. Requirements specifications 6
     1. Hardware Requirements 6
     2. Software Requirements
  3. System Study 7
     1. Economic Feasibility 7
     2. Technical Feasibility 7
     3. Social Feasibility 7

###### [Chapter 3 Design](#_bookmark10) 8

* 1. [UML Diagrams 8](#_bookmark11)
     1. [Use Case Diagram](#_bookmark12) 8
     2. Sequence Diagram 10
     3. [Activity Diagram 11](#_bookmark0)

Page | vii

###### [Chapter 4 Methodology and Implementation 12](#_bookmark1)

* 1. [Modules 12](#_bookmark15)
  2. Module Descriptions 12
     1. Data Collection and Preprocessing Module 12
     2. Model Training and Validation Module 12
     3. Model Evaluation and Interpretation Module 12
  3. [Python Implementation 13](#_TOC_250004)
     1. [What is Python 13](#_TOC_250003)
     2. [Advantages of Python 13](#_TOC_250002)
  4. [Source Code 14](#_bookmark16)

[Chapter 5 Testing 19](#_bookmark7)

* 1. [Types of Testing 19](#_bookmark8)
     1. [Unit Testing 19](#_bookmark9)
     2. [Integration Testing 20](#_bookmark17)
     3. [Functional Testing 20](#_TOC_250001)
     4. [System Testing 21](#_TOC_250000)
     5. [White box Testing 21](#_bookmark10)
     6. [Black box Testing 22](#_bookmark11)
     7. [Acceptance Testing 22](#_bookmark12)
  2. [Test Strategy 23](#_bookmark13)

**Chapter 6**  **Result** 24

**Chapter 7 Conclusion 30**

**Chapter 8 Future scope 31**

**References**

Page | viii

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page** |
| 3.1.1 | Use Case Diagram | 9 |
| 3.1.2 | Sequence Diagram | 10 |
| 3.1.3 | Activity Diagram | 11 |
| 5.1.7.1 | Stress Detection Dataset | 23 |
| 6.1 | Upload Program Files | 24 |
| 6.2 | Run Program File | 25 |
| 6.3 | Classification | 26 |
| 6.4 | Execution | 27 |
| 6.5 | Using Facial Expressions data set | 28 |
| 6.6 | Result 1 | 28 |
| 6.7 | Result 2 | 29 |

Page | ix

**CHAPTER 1 INTRODUCTION**

* 1. **Introduction:**

Stress management systems play a significant role to detect the stress levels which

disrupts our socio-economic lifestyle. As World Health Organization (WHO) says, Stress is a mental health problem affecting the life of one in four citizens. Human stress leads to mental as well as socio-fiscal problems, lack of clarity in work, poor working relationship, depression and finally commitment of suicide in severe cases. These demands counselling to be provided for the stressed individuals cope up against stress. Stress avoidance is impossible but preventive actions helps to overcome the stress. Currently, only medical and physiological experts can determine whether one is under depressed state (stressed) or not. One of the traditional methods to detect stress is based on questionnaire.

This method completely depends on the answers given by the individuals, people will be tremulous to say whether they are stressed or normal. Automatic detection of stress minimizes the risk of health issues and improves the welfare of the society. This paves the way for the necessity of a scientific tool, which uses physiological signals thereby automating the detection of stress levels in individuals. Stress detection is discussed in various literatures as it is a significant societal contribution that enhances the lifestyle of individuals.

Ghaderi et al. analysed stress using Respiration, Heart rate HR), facial electromyography (EMG), Galvanic skin response (GSR) foot and GSR hand data with a conclusion that, features pertaining to respiration process are substantial in stress detection. Maria Viqueira et al. describes mental stress prediction using a standalone stress sensing hardware by interfacing GSR as the only physiological sensor. David Liu etal. proposed research to predict stress levels solely from Electrocardiogram (ECG). Multimodal sensor efficacy to detect stress of working people is experimentally discussed in. This employs the sensor data from sensors such as pressure distribution, HR, Blood Volume Pulse (BVP) and Electro dermal activity (EDA).

**1.2Existing System**

In the existing system work on stress detection is based on the digital signal processing,

taking into consideration Galvanic skin response, blood volume, pupil dilation and skin

temperature. And the other work on this issue is based on several physiological signals

and visual features (eye closure, head movement) to monitor the stress in a person while

he is working. However, these measurements are intrusive and are less comfortable in real

application. Every sensor data is compared with a stress index which is a threshold value

used for detecting the stress level.

* + 1. **Limitations of the Existing System:**

1. Physiological signals used for analysis are often pigeonholed by a non-stationary

time performance.

2. The extracted features explicitly gives the stress index of the physiological signals.

The ECG signal is directly assessed by using commonly used peak j48 algorithm

3. Different people may behave or express differently under stress and it is hard to find a

universal pattern to define the stress emotion.

* 1. **Proposed System**

The proposed system in this research is a novel approach for **real-time stress detection in IT professionals** using **convolutional neural networks (CNNs)** and **image processing**. It aims to overcome the limitations of existing stress detection methods by offering a non-invasive and objective approach based on facial expression analysis.The system leverages the power of CNNs to analyse facial images and identify patterns associated with stress. By capturing and processing facial expressions in real-time, the system can provide immediate insights into an individual's stress levels. This allows for timely intervention and personalized stress management strategies.

2

**1.3.1Advantages of the Proposed System:**

###### Non-invasive:

###### It does not require intrusive sensors or physiological data collection.

###### Objective:

###### The analysis is based on visual cues, providing an objective measure of stress.

###### Personalized Recommendations: Tailored stress management strategies can be provided based on the detected stress levels

###### Real-Time: Enables immediate stress detection and intervention.

###### Integration with Existing Systems: Can be easily integrated into workplace environments for seamless monitoring and support.

**3**

**CHAPTER 2 LITERATURE SURVEY**

**2.1 Literature Review**

Stress and anxiety detection using facial cues from videos G. Giannakakis 2017 [1]

This study develops a framework for the detection and analysis of stress/anxiety emotional states through video-recorded facial cues. A thorough experimental protocol was established to induce systematic variability in affective states (neutral, relaxed and stressed/anxious) through a variety of external and internal stressors. The analysis was focused mainly on non-voluntary and semi voluntary facial cues in order to estimate the emotion representation more objectively. Features under investigation included eye-related events, mouth activity, head motion parameters and heart rate estimated through camera-based photo plethysmography. A feature selection procedure was employed to select the most robust features followed by classification schemes discriminating between stress/anxiety and neutral states with reference to a relaxed state in each experimental phase. In addition, a ranking transformation was proposed utilizing self-reports in order to investigate the correlation of facial parameters with a participant perceived amount of stress/anxiety. The results indicated that, specific facial cues, derived from eye activity, mouth activity, head movements and camera-based heart activity achieve good accuracy and are suitable as discriminative indicators of stress and anxiety.

Detection of Stress Using Image Processing and Machine Learning Techniques Nisha Raichur, Nidhi

Lonakadi, Priyanka Mural [2] 2017 Stress is a part of life it is an unpleasant state of emotional

arousal that people experience in situationslike working for long hours in front of computer.

Computers have become a way of life; much life is spent on the computers and hence we are

therefore more affected by the upsend downs that they cause us. One cannot just completely

avoid their work on computers but one can at least control his/her usage when being alarmed

about him being stressed at certain point of time. Monitoring the emotional status of a person

who is working in front of a computer for longer duration is crucial for the safetyof a person.

In this work real-time non-intrusive videos are captured, which detects the emotional status

of a person by analysing the facial expression. We detect an individual emotion in each video

frame and the decision on the stress level is made in sequential hours of the video captured.

Machine Learning Techniques for Stress Prediction in Working U. S. Reddy, A. V. Thota and A. Dharu

2018 [3] Stress disorders are a common issue among working IT professionals in the industry

today. With changing lifestyle and work cultures, there is an increase in the risk of stress among

the Employyees Though many industries and corporates provide mental health related schemes and

try to ease the workplace atmosphere, the issue is far from control. In this paper, we would like to

apply machine learning techniques to analyse stress patterns in working adults and to narrow down

the factors that strongly determine the stress levels.

Classification of acute stress using linear and non-linear heart rate variability analysis derived from sternal ECG Tanev, G., Saadi, D.B., Hoppe, K., Sorensen, H.B 2014 [4] Chronic stress detection is an important factor in predicting and reducing the risk of cardiovascular disease. This work is a pilot study with a focus on developing a method for detecting short-term psychophysiological changes through heart rate variability (HRV) features. The purpose of this pilot study is to establish and to gain insight on a set of features that could be used to detect psychophysiological changes that occur during chronic stress. This study elicited four different types of arousal by images, sounds, mental tasks and rest, and classified them using linear and non-linear HRV features from electrocardiograms (ECG) acquired by the wireless wearable patch recorder. The highest recognition rates were acquired for the neutral stage (90%), the acute stress stage (80%) and the baseline stage (80%) by sample entropy, detrended fluctuation analysis and normalized high frequency features. Standardizing non-linear HRV features for each subject was found to be an important factor for the improvement of the classification results.

HealthyOffice: Mood recognition at work using smartphones and wearable sensors Zenonos, A., Khan, A., Kalogridis, G., Vatsikas, S., Lewis, T., Sooriyabandara [5] Stress, anxiety and depression in the workplace are detrimental to human health and productivity with significant financial implications. Recent research in this area has focused on the use of sensor technologies, including smartphones and wearables embedded with physiological and movement sensors. In this work, we explore the possibility of using such devices for mood recognition, focusing on work environments. We propose a novel mood recognition framework that is able to identify five intensity levels for eight different types of moods every two hours. We further present a smartphone app ('HealthyOffice'), designed to facilitate self-reporting in a structured manner and provide our model with the ground truth. We evaluate our system in a small-scale user study where wearable sensing data is collected in an office environment. Our experiments exhibit promising results allowing us to reliably recognize various classes of perceived moods.

5

**2.1Requirement Specifications**

**2.1.1Hardware Requirements:**

* RAM - 4 GB (min)
* Hard Disk - 1TB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - 15’’LED
  + 1. **Software Requirements:**
* Operating system : Windows 10/11
* Development Software :Python 3.10
* Programming Language: Python.
* Integrated Development Environment (IDE): Visual Studio Code
* Front End Technologies: HTML5, CSS3, Java Script
* Back End Technologies or Framework: Django
* Database Language: SQL
* Database (RDBMS): MySQL
* Web Server or Deployment Server: Django Application Development Server
* Design/Modelling: Rational Rose

6

* 1. **System Study**

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**2.2.1Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**2.2.2 Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

* + 1. **Social Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by theirs solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

7

**CHAPTER 3**

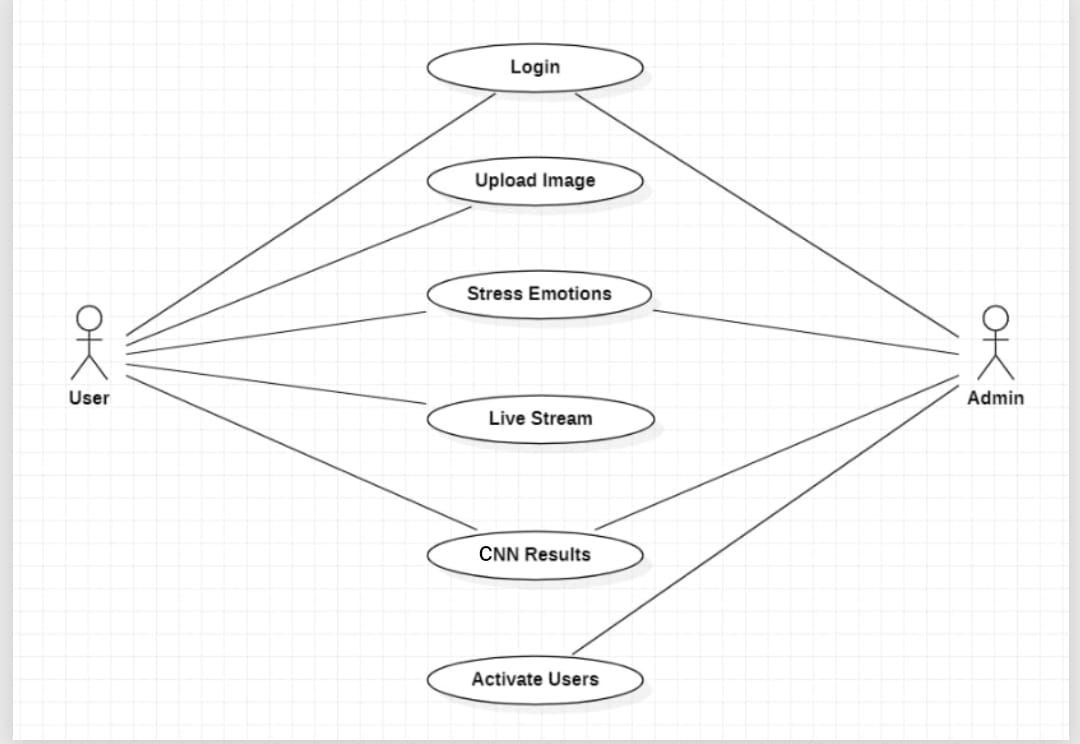
**DESIGN**

**3.1 UML Diagrams**

UML stands for Unified Modeling Language or SFL (Standards for Facilities). UML is a general-purpose object-orientated modeling language that has become popular in the field of software engineering and is based on standard principles. Control is done, and it is also created by, that particular group. The Object Management Group.

* + - The intention is that UML models become a universal standard for the creation of object-centric software diagrams. In its current form UML is comprised of two major components: finally, we will create a Meta model and a notation. of this, we will Model our content Meta and the notation. In the future, UML can also be supported by a method or mechanism that is either related to or connected with UML by some means.
    - Unified Modeling Language is standard language for modeling unambiguously replacing the descriptions which are visual, explicit and also documentation ones of software through the use of the artifacts. The application of the modeling includes business modeling as well as non-software systems.
    - The UML reflects a chosen set of best engineering practices that have been effectively employed to model of an extensive software system.
    1. **Use Case Diagram:**

In the Unified Modelling Language (UML), a use case diagram is a behavioral diagram, which stands for and comes from the analysis involving and of use cases. Its main job is to draw a diagram which symbolizes how a system performs various tasks. It iconizes actors, their use cases and depicts the cause-effect relationships between those use cases. The primary value of the use case diagram is that it displays the operation of system actors for which functions are composed. By depicting actors' roles in the system as shown in Fig. 3.1.1, the model could be seen as in the image below.

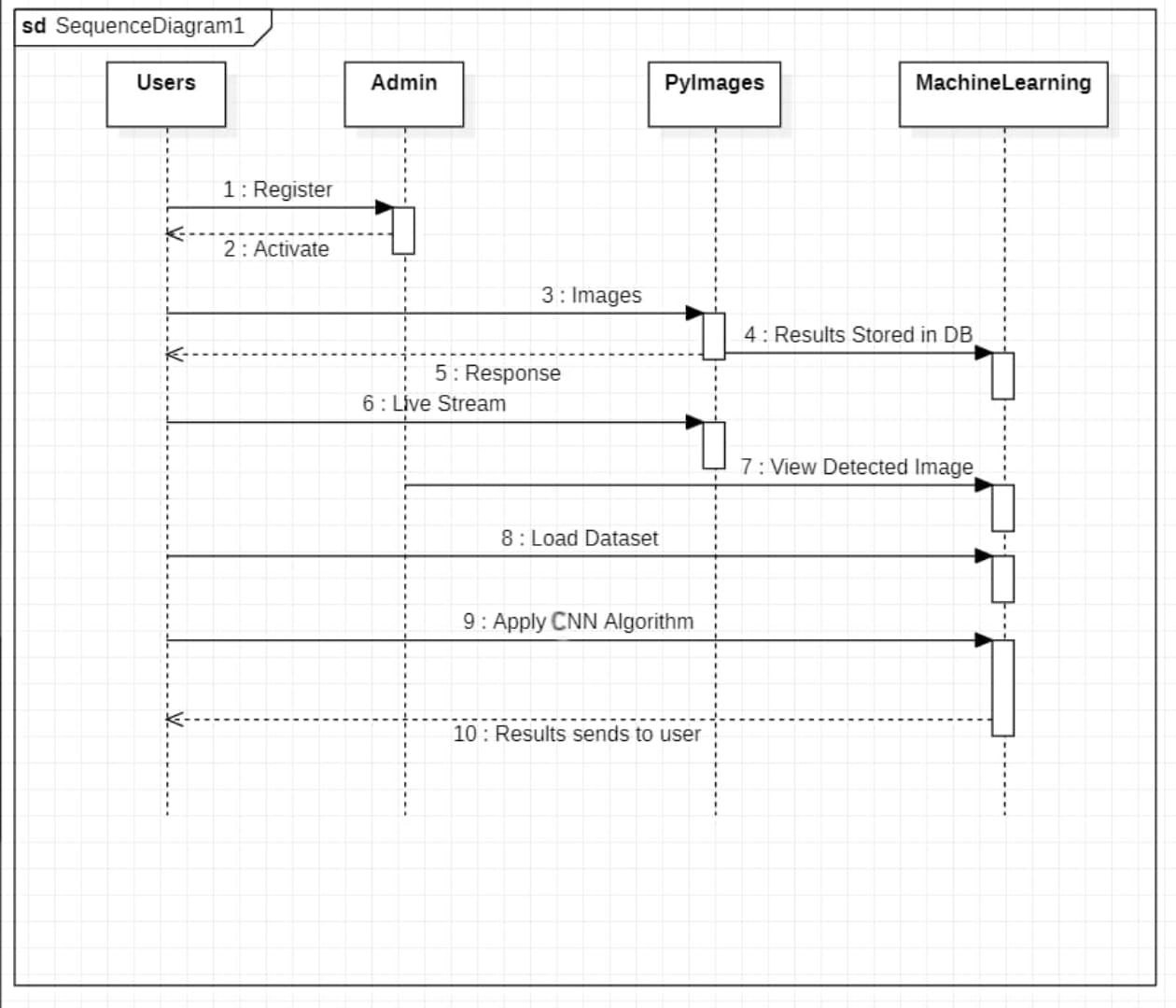


###### Fig. 3.1.1 Use Case diagram

9

* + 1. **Sequence diagram:**

A sequence diagram among the diagrams that use the Unified Modeling Language (UML) in the interaction diagram category shows how processes run in order and with whom. It is an Bee stack communication protocol which is represented in Fig 3.1.3. Timing diagrams are also referred to as Event DAG, event sequences, and scenario diagrams.



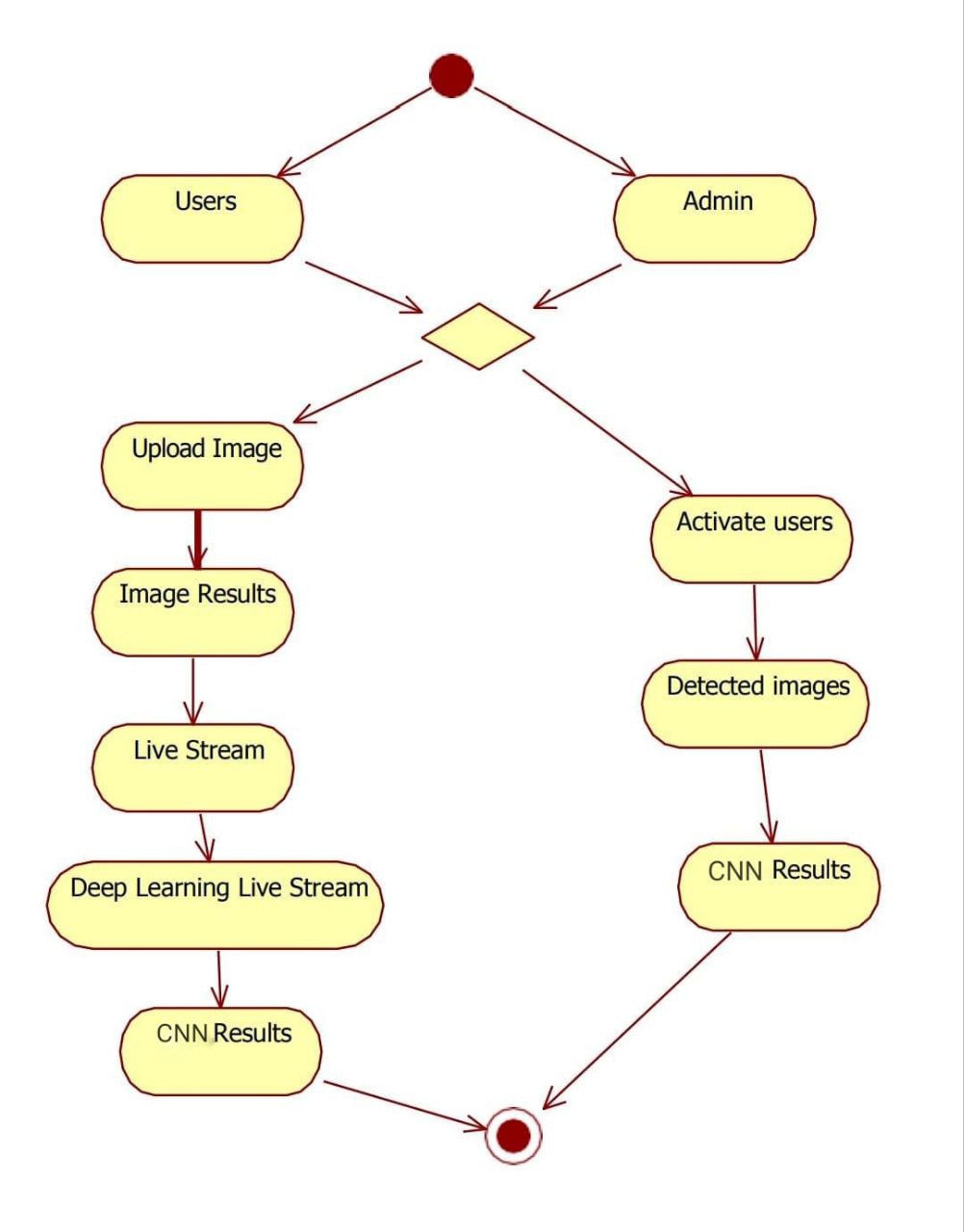
###### Fig. 3.1.2. Sequence diagram

10

**3.1.3. Activity Diagram:**

Activity diagram is another important diagram in UML to describe the dynamic

aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.



###### Fig. 3.1.3. Activity diagram

a

11

**CHAPTER 4**

**METHODOLOGY AND IMPLEMENTATION**

**4.1Modules:**

* Flask
* cv2 (OpenCV)
* tensorflow.keras
* numpy
* os
* logging
* traceback

**4.2 Module Descriptions:**

**4.2.1 Data Collection and Preprocessing Module**

This module focuses on gathering and preparing the dataset for analysis. Key tasks include:

**Data Collection**: Capturing facial images from participants under various

stress-inducing conditions and neutral scenarios. This may involve video recordings and still images.

**Annotation**: Labeling images based on the level of stress or emotional state, potentially using expert input or self-reported data.

**4.2.2 Model Training and Validation Module**

This module encompasses the training of machine learning models to recognize stress through facial expressions:

**Model Selection**: Choosing appropriate algorithms (e.g., CNNs, SVMs) based on the nature

of the data and desired outcomes.

**Training Process**:

Utilizing training data to optimize model parameters through backpropagation and gradient descent techniques.

**4.2.3 Model Evaluation and Interpretation Module**

This module is dedicated to assessing the trained model's effectiveness and interpreting

its predictions:

**Evaluation Metrics**: Employing metrics such as accuracy, precision, recall, F1-score,

and confusion matrices to quantify model performance.

**Performance Testing**: Running the model on the test dataset to evaluate generalization capabilities.

**Error Analysis**: Identifying misclassifications and understanding potential reasons

## 4.3 Python Implementation

### 4.3.1 What is Python:

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard libraries which can beused for the following –

* + - 1. [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
      2. GUI Applications (like Kivy, Tkinter, PyQt etc.)
      3. Web frameworks like Django (used by YouTube, Instagram, Dropbox)
      4. Image processing (like Open cv, Pillow)
      5. Web scraping (like Scrapy, Beautiful Soup, Selenium)
      6. Test frameworks
      7. Multimedia

### Advantages of Python:

Extensive Libraries: Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web

browsers, threading, databases, CGI, email, image manipulation, and more*.* So, we don’t have to write the complete code for that manually.

1. Extensible: As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.
2. Improved Productivity: The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.
3. Embeddable: Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us

add scripting capabilities to our code in the other language.

13

**4.4Source Code:**

# necessary imports

from flask import Flask, render\_template, Response, request, jsonify

import cv2

from tensorflow.keras.models import load\_model

import numpy as np

import os

import logging

from logging.handlers import RotatingFileHandler

import traceback

app = Flask(\_\_name\_\_)

# Configure logging

log\_handler = RotatingFileHandler('app.log', maxBytes=10000000, backupCount=5)

log\_handler.setLevel(logging.INFO)

formatter = logging.Formatter('%(asctime)s - %(levelname)s - %(message)s')

log\_handler.setFormatter(formatter)

app.logger.addHandler(log\_handler)

# Load the pre-trained Keras model for emotion detection

def load\_pretrained\_model(model\_path):

    try:

        model = load\_model(model\_path)

        app.logger.info(f"Model loaded from {model\_path}")

        return model

    except Exception as e:

        app.logger.error(f"Error loading model: {e}")

        raise

# Load the Haar Cascade for face detection

def load\_haar\_cascade(cascade\_path):

    try:

        face\_cascade = cv2.CascadeClassifier(cascade\_path)

        app.logger.info(f"Haar Cascade loaded from {cascade\_path}")

        return face\_cascade

14

except Exception as e:

        app.logger.error(f"Error loading Haar Cascade: {e}")

        raise

# Load model and face cascade

model\_path = 'model1.h5'

face\_cascade\_path = cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml'

model = load\_pretrained\_model(model\_path)

face\_cascade = load\_haar\_cascade(face\_cascade\_path)

# Define emotion class names and stress mapping

class\_names = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']

stress\_mapping = {

    'Angry': ('Stressed', 'High'),

    'Disgust': ('Stressed', 'High'),

    'Fear': ('Stressed', 'High'),

    'Sad': ('Stressed', 'Moderate'),

    'Happy': ('Not Stressed', 'Low'),

    'Surprise': ('Not Stressed', 'Low'),

    'Neutral': ('Not Stressed', 'Low')

}

def detect\_emotion(frame):

    try:

        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), (255, 0, 0), 2)

            roi\_gray = gray[y:y+h, x:x+w]

            roi\_gray = cv2.resize(roi\_gray, (126, 126))

            roi\_rgb = np.stack([roi\_gray] \* 3, axis=-1)

            roi\_rgb = roi\_rgb.astype('float32') / 255.0

            roi\_rgb = np.expand\_dims(roi\_rgb, axis=0)

            predictions = model.predict(roi\_rgb)

            max\_index = np.argmax(predictions[0])

            predicted\_emotion = class\_names[max\_index]

            stress\_status, stress\_level = stress\_mapping[predicted\_emotion]

15

label = f"{stress\_status} - {stress\_level}"

            cv2.putText(frame, label, (x, y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (36, 255, 12), 2)

        return frame

  except Exception as e:

        app.logger.error(f"Error in detect\_emotion: {e}")

        return frame

def gen():

    # Use DirectShow backend instead of MSMF (cv2.CAP\_DSHOW)

    cap = cv2.VideoCapture(0, cv2.CAP\_DSHOW)  # Change backend to DirectShow

    if not cap.isOpened():

        app.logger.error("Error opening video stream")

        return

    while True:

        ret, frame = cap.read()

        if not ret:

            app.logger.warning("Failed to capture image from video stream")

            break

        try:

            frame = detect\_emotion(frame)

            \_, buffer = cv2.imencode('.jpg', frame)

            frame = buffer.tobytes()

            yield (b'--frame\r\n'

                   b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

        except Exception as e:

            app.logger.error(f"Error processing frame: {e}")

            continue

    cap.release()

@app.route('/')

def index():

    return render\_template('index.html')

16

@app.route('/video\_feed')

def video\_feed():

    try:

        return Response(gen(), mimetype='multipart/x-mixed-replace; boundary=frame')

    except Exception as e:

app.logger.error(f"Error in video\_feed route: {e}")

        return jsonify({'error': 'Error in video feed'}), 500

@app.route('/health', methods=['GET'])

def health\_check():

    try:

        return jsonify({'status': 'ok'})

    except Exception as e:

        app.logger.error(f"Error in health\_check route: {e}")

        return jsonify({'error': 'Health check failed'}), 500

@app.errorhandler(500)

def internal\_error(error):

    app.logger.error(f"Server Error: {traceback.format\_exc()}")

    return jsonify({'error': 'Internal Server Error'}), 500

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

17

###### Explanation:

The code begins by importing the necessary libraries for building a web application that performs emotion detection using facial expressions. It then sets up a Flask application and configures logging to monitor the application’s performance and errors.

In the model loading phase, a pre-trained Keras model for emotion detection and a Haar Cascade classifier for face detection are loaded. The model is designed to recognize various emotions such as anger, fear, happiness, and sadness from images.

The application captures video frames from the webcam and processes each frame to detect faces and predict emotions. Each detected face is highlighted, and the corresponding emotion is annotated on the frame, along with a stress level indicating the emotional state of the individual.

The main route renders the HTML template for the user interface, while another route streams the video feed processed with emotion detection in real time. A health check endpoint is also included to verify that the application is running properly.

The error handling mechanism ensures that any issues during model loading, image processing, or video streaming are logged for debugging purposes. This comprehensive approach allows for a seamless user experience in emotion recognition while maintaining robust error management. Overall, the code provides an effective framework for real-time emotion detection and analysis using machine learning techniques integrated into a web application.

18

# CHAPTER 5

# TESTING

Testing, after all, is about identifying defects. In this case, the product is a software under test. It is performance of a regular process that is very systematic, it is possible to detect any gap or any kind of software fault. Also, the software is tested in order to make sure that any discrepancy between it and certain specifications or users` expectations are fixed. The test means verifying those critical specifications such as component evaluation, subassembly modelling, and the system during its operation bear a match with designed specifications and are unlikely to fail at the most critical moments. The types of tests vary in order to meet specific requirements. Each test type targets different aspects; reliability and response time is assurance of good product quality in software.

## Types of Testing:

### Unit Testing:

Developing unit tests means building test cases to ensure the correctness of the inner presentation of program logic and the proper formation of the output after the program's inputs. It searches and verifies all of the decision brunch and internal code for occurred errors. Unit testing is the type of testing that proceeds with independent software units of the program, executed after the finalization of each unit but prior to the integration. There is an invasive nature between the message they deliver and their appearance so there is little variety in how they can be used. In unit tests components-level functionality is being evaluated by selecting and testing specific business process, application, or system configuration with use of unit tests. The main purpose of the control systems is to make sure that the output is what had been expected by following the complex logical and detailed flow of processes and procedures. Unit testing is very helpful in bug detection and perfectness of individual software unit which ensures also the system as a whole is correct and functioning in the best way achievable.

### Integration Testing:

Integration testing validates the effectiveness of assembled software elements that consist of their equipment to work as a complete system. At the core of this approach is an events- based behavior that is mostly interested in the draws, not the checks and balances. It demonstrates that the sub-compartments of the structure function together and cooperate to play as a single member. The integration tests are meant to assure that although the individual components are found satisfactory to validate through the unit testing procedure, their integration in to complex components remains correct and consistent.

That is the type of testing and particularly is aimed at revealing collisions coming together and fixing them.

Bringing the integrated components to a single platform and testing the interconnection between various elements will bring about a reduction of the risks that integration poses for robustness and with that the reliability of the software application.

### Functional Testing:

Functional tests, through a consistent manner, demonstrate that all functions which takes into account the specified business and technical requirements, system documents and user manuals are in place and fits the business logic.

Functional testing is centered on the following items: Functional testing is centered on the following items:

Valid Input: their offered programs for language courses and discussed the best practices that assist in teaching oral communication skills, vocabulary building, and understanding of grammar.

Invalid Input: brackets classes of invalid input may be refused.

Functions: upon the identification of these functions, the powers need to be applied.

Output: causes that fall under the category of application outputs should be performed with a necessary diligence.

Systems/Procedures: alterations, inter connections or the like should be eradicated.

20

Fulfilment of functional tests is mainly focused on requirements, firm functions of a product, and special ones. Moreover, the systematic review provides an elementary study involving identifying business process flows; data fields, predefined processes, and successive processes, which should be integrated in the testing. These are the last tasks before functional testing is done. Identification of new tests, and actualization of role of current tests is the subject of the test.

### System Testing:

System testing is an important stage of software testing which assures that the integrated software system according to the project specifications correctly performs its expected terms of behavior It makes a configured check to ensure that the settings are producing expected and similar outcomes. To illustrate, the processes of system integration that involves configuration-oriented automation can be considered a system test further refined. Testing of this step is being carried out using process descriptions and flows and with the emphasis on pre-designed process connections and the integration interfaces. System testing is more focused on examining the system as a whole, which includes checking the system's functionality and performance in an environment where co-working supports smooth operation across interconnected components.

Through meticulously doing behavioral analysis in its given environment, system testing verifies the system the way it should function and meet client’s expectation. This thorough testing might be the only way that we can ensure that the system provides accurate results regarding its assurance of health.

### White box Testing:

White Box Testing refers to a testing stage where the software tester is familiar with the inner workings of the software i.e. structure and syntax in addition to the specific function of the software. This approach is mainly applied to examine the hidden and unsolved parts of the software which cannot be assessed at a black box level. Tester should have a profound knowledge of the structure of software, and being able to test its "inner frontiers" and code paths. Such knowledge can be skillfully utilized by the tester to go deeper into the software's functionality and as a result, detect vulnerabilities or even areas for improvement. The ultimate goal of the White Box Testing process is to call upon meticulous exploration and

21

scrutiny of a product to ensure its dependability and robustness inside the software’s structural operations. It is of high importance as a safeguard in the software quality landscape, adding weight to other types of tests and contributing to the building of a robust quality system.

### Black box Testing:

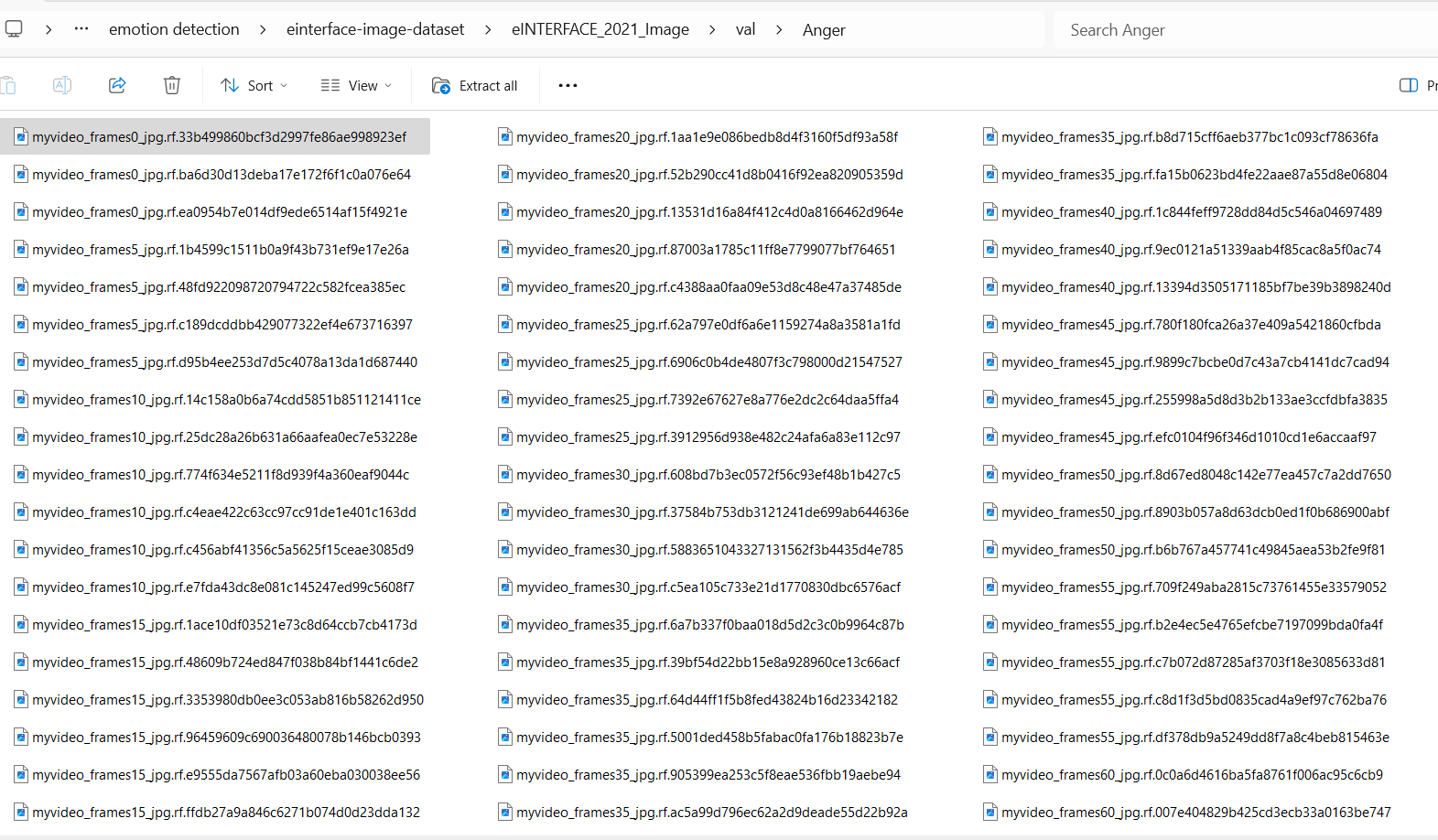
Black Box Testing implies validating the software without any clue about its confidentiality and the way it is formulated. Testers look at the software as a closed structure, a holistic perspective, but without a chance to peek into its inner workings. Tests are created using authoritative documents (requirements Specifications are the most common), so that the consideration is not omitted. In this format of testing, the software is merely seen from the perspective of what it gives out and inputs, ignoring the operations and algorithms making it execute. Testers are the executors of the software program, their roles entail entering data into the software, then observing the outputs which accurately replicate the behavior as is. This process is carried out without going into the inner workings of the software. The Black Box Testing exerts great influence on checking software about fulfilled requirements, providing unbiased evaluation from user side of whether it is performing in a desirable manner or not.

### Acceptance Testing:

Acceptance testing is a crucial phase in the development of the emotion detection web application, ensuring it meets specified requirements and performs as intended. The process begins with defining acceptance criteria based on functional requirements, followed by developing test cases that cover various scenarios, including functional, user interface, and performance tests. Executing these test cases in a controlled environment involves capturing video feeds from different webcams, assessing detection accuracy with various facial expressions, and validating error logging and exception handling.

User acceptance testing (UAT) engages actual users to test the application in real-world settings, gathering feedback on usability and performance. Any identified issues are documented, categorized, and prioritized for resolution, ensuring critical issues are addressed before final deployment. Finally, a review with stakeholders confirms the application meets acceptance criteria, leading to sign-off for production launch. This structured approach validates the application’s functionality, performance, and overall user experience, ensuring it effectively fulfills its intended purpose.

22



###### Fig 5.1.7.1. Stress Detection Dataset

In the Fig 5.1.7.1, dataset screen having different attributes for Stress Detection

* 1. **Test Strategy:**

Test Objectives:

* Ensure accurate Stress Detection using clinical and demographic data.
* Validate the correct functioning of all input fields.
* Confirm the reliability and performance of machine learning models.
* Verify the responsiveness and usability of the user interface.

Features to be Tested:

* Verify that all data entries are in the correct format.
* Ensure no duplicate entries are allowed.
* Confirm that all links and navigation elements direct users to the correct pages.
* Validate the accuracy and consistency Stress Detection from machine learning models.

23

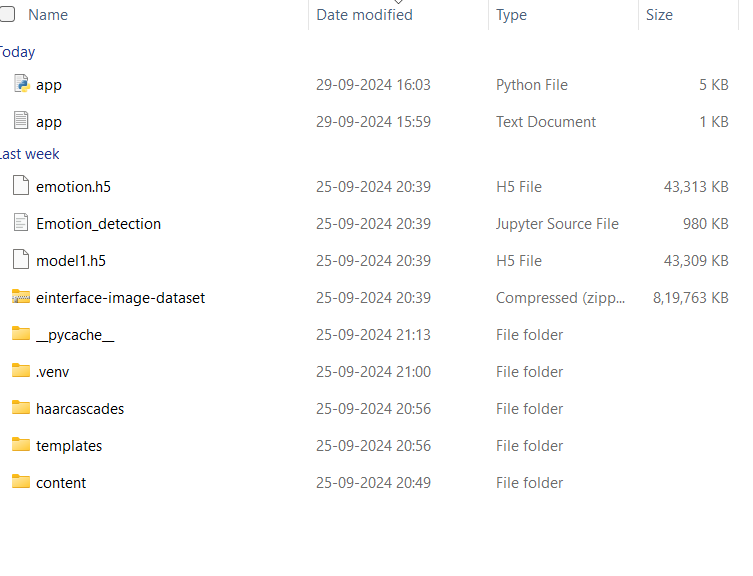
**CHAPTER 6**

**RESULT**

The convolutional neural network (CNN) model serves as the primary algorithm for detecting stress levels from facial expressions, achieving high accuracy in recognizing stress-related emotions by analyzing facial features such as eye movements, mouth curvature, and forehead tension, making it a reliable method for stress detection.

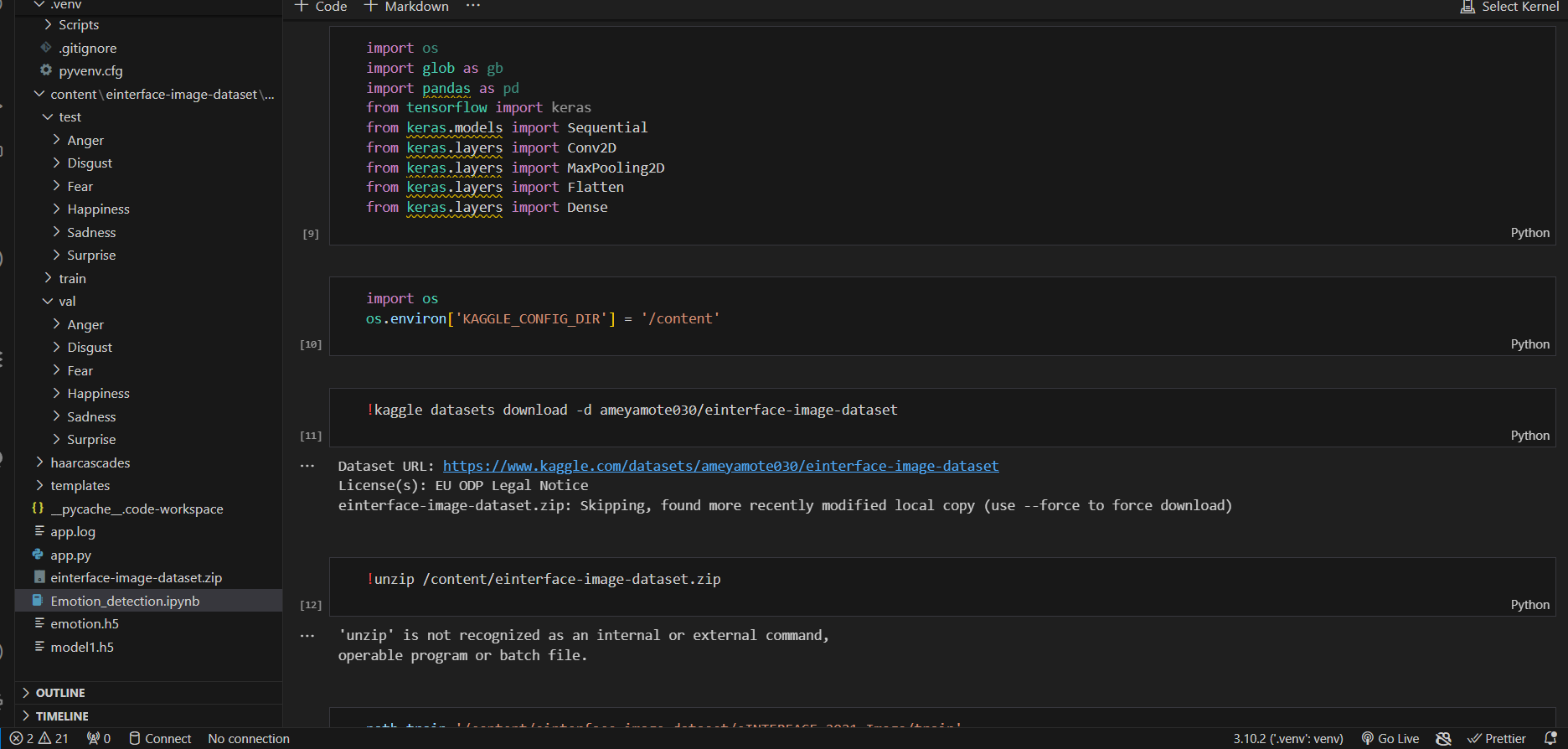
A Flask-based web application was developed to provide real-time stress monitoring with an intuitive user interface. The system offers immediate feedback on stress levels, enabling users to track and manage their stress effectively.

While the CNN model demonstrated strong performance, continuous refinement and monitoring will be essential to maintain its accuracy across different user profiles and real-world conditions. This system holds great potential for improving everyday stress management by providing real-time, actionable insights into a person’s stress levels.



**Fig 6.1 Upload Program Files**

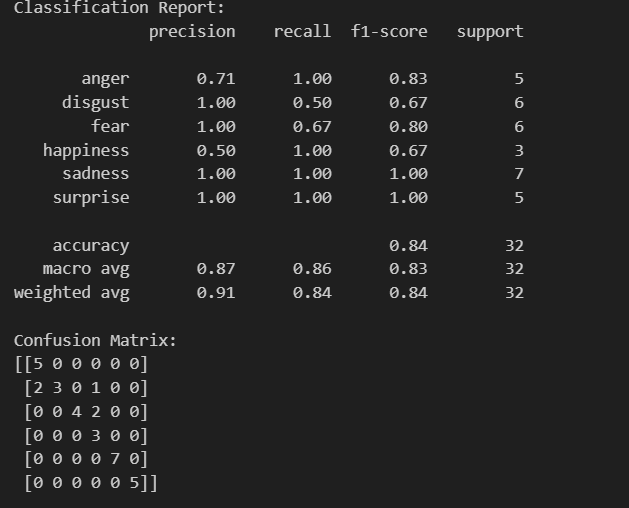
to run the project, open program files in vs code as shown in fig 6.1.

****

###### Fig 6.2. Run Program File

To run the project, open the emotion \_detection. ipynb file in vs code and execute it as shown in Fig 6.1.

25



###### Fig 6.3. Classification

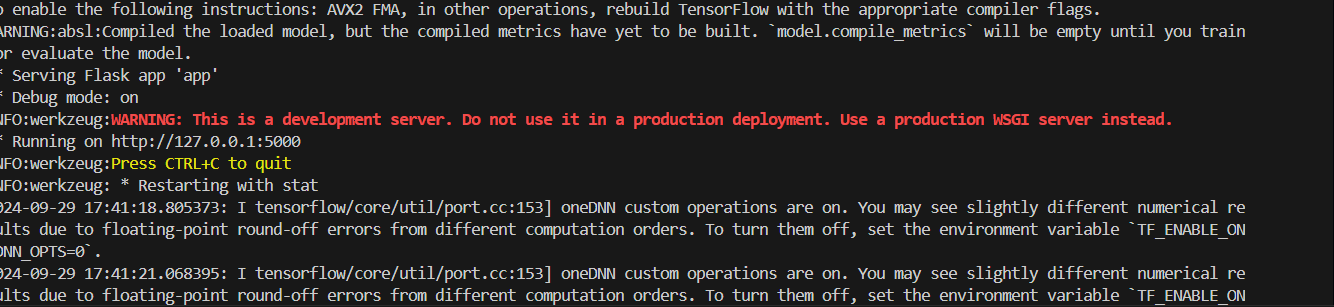
In Fig 6.3 Explain structured overview allows for a clearer understanding of the model's performance in classifying stress-related emotions based on facial expressions.

**Precision**: The ratio of correctly predicted positive observations to the total predicted positives. A high precision indicates a low false positive rate.

**Recall**: The ratio of correctly predicted positive observations to all actual positives. High recall indicates a low false negative rate.

**F1-Score**: The weighted average of precision and recall, providing a balance between them.

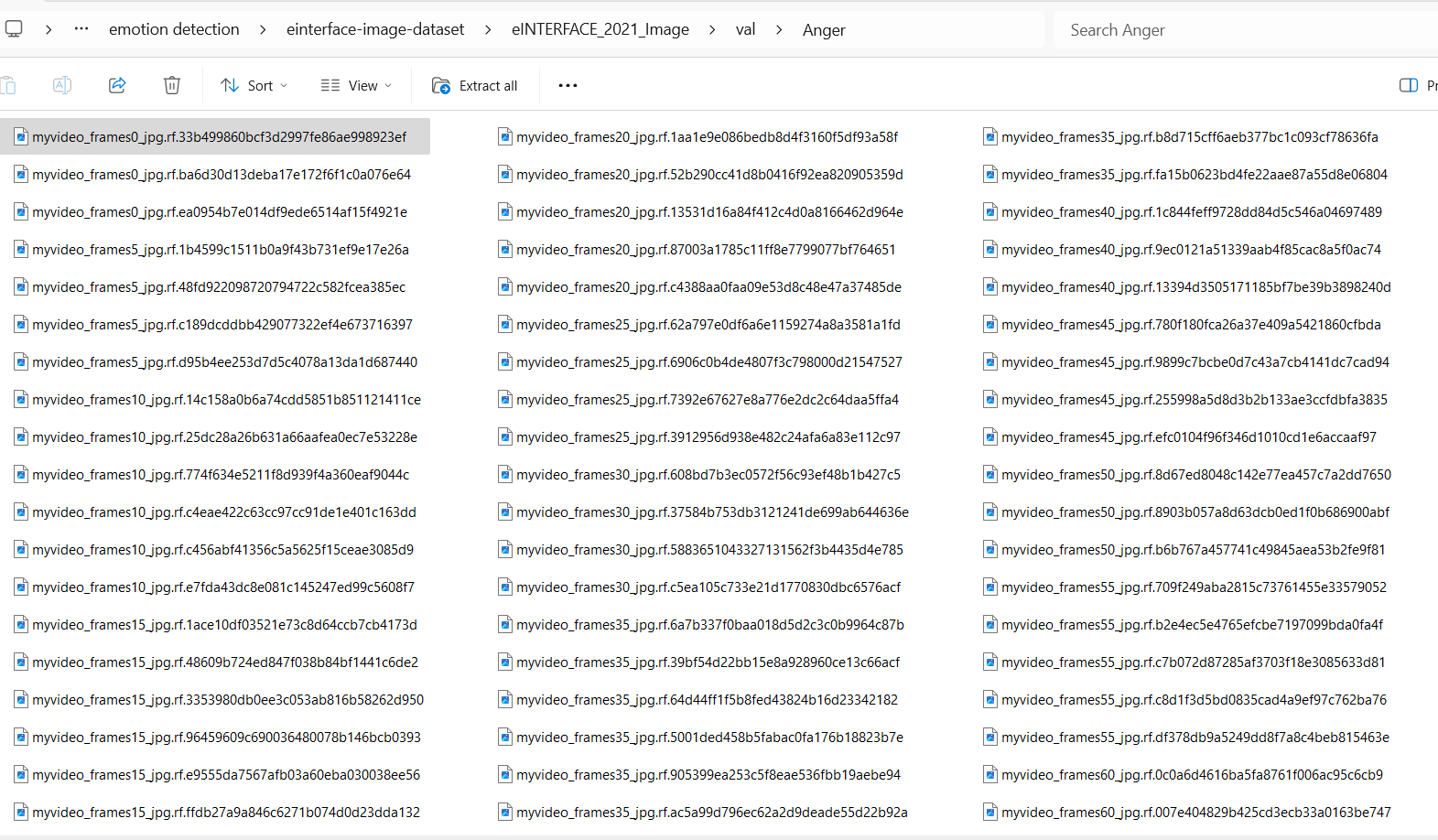
**Support**: The number of actual occurrences of the class in the specified dataset.  
  
**Confusion Matrix**: A table used to describe the performance of a classification model, showing true positives, false positives, and false negatives for each class.



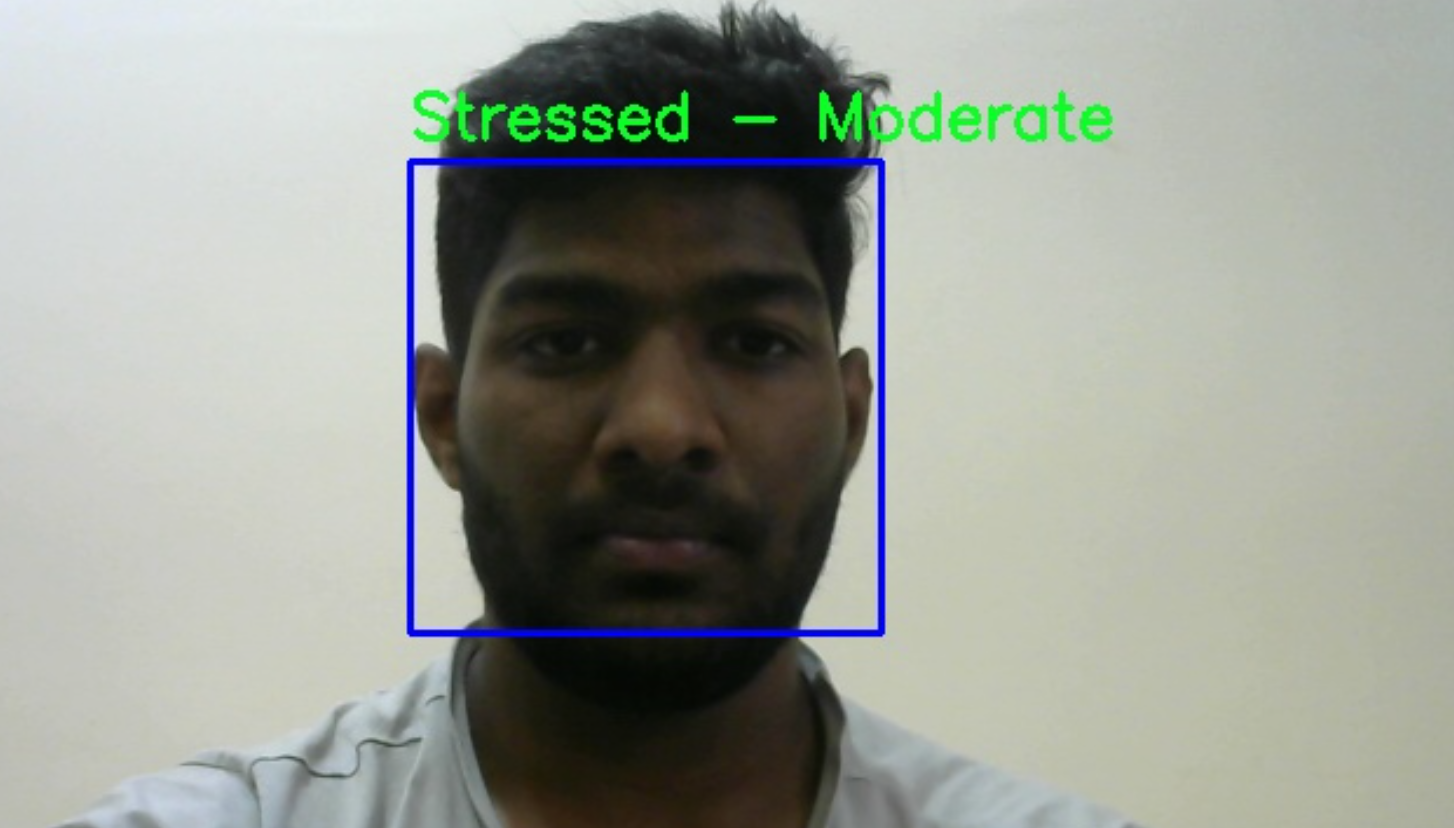
###### Fig 6.4. Execution

In Fig 6.5 go to Terminal and enter command as python.app.py to execution program

27

I

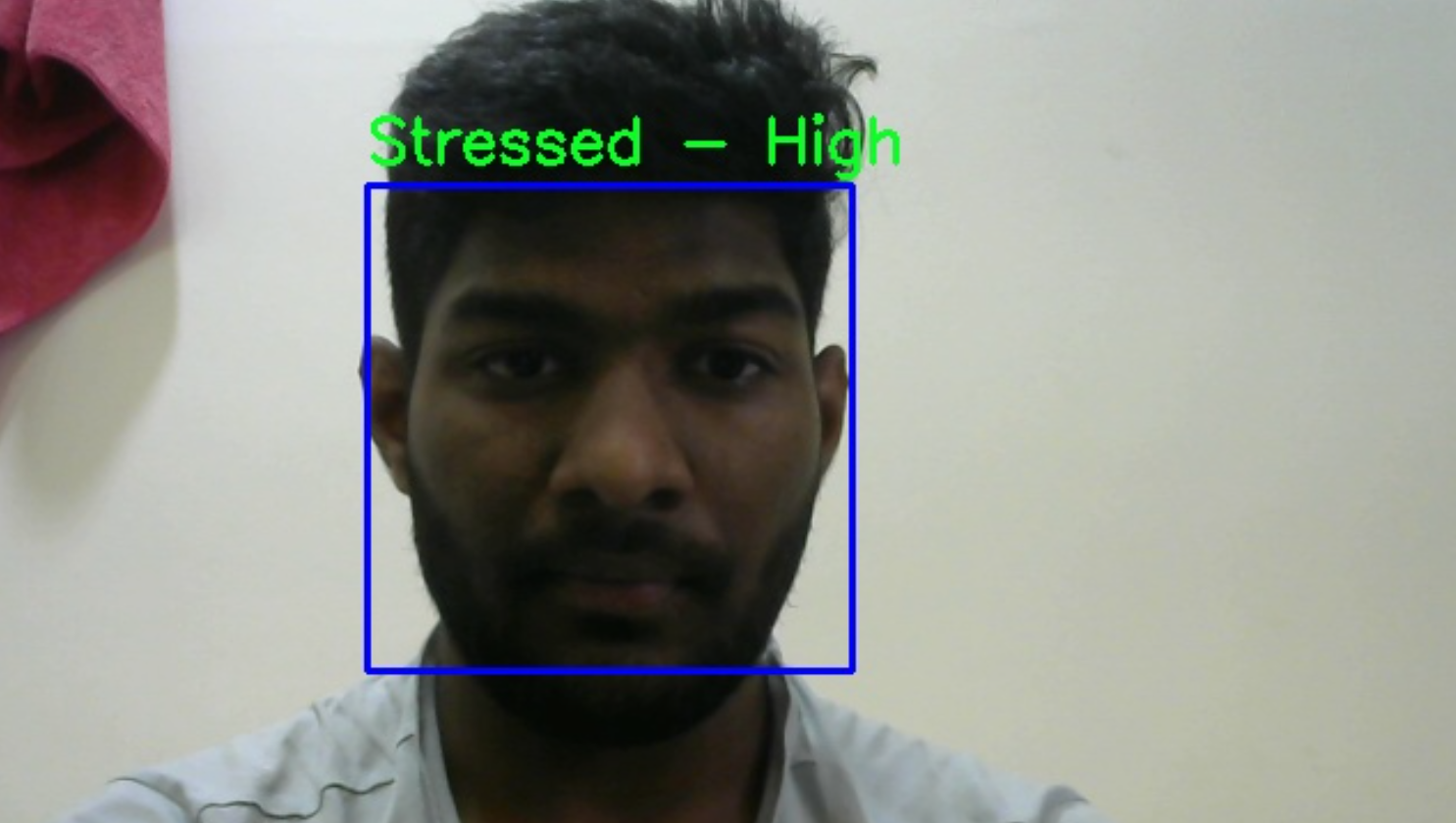
###### Fig 6.5. Stress Detection Using Facial Expressions data set



###### Fig 6.6. Result 1

In Fig 6.10, Prediction output Stress Detection Using Facial Expression

28



###### Fig 6.7. Result 2

In Fig 6.11, Prediction output Stress Detection Using Facial Expression

29

**CHAPTER 7 CONCLUSION**

Stress Detection System is designed to predict stress in the employees by monitoring captured

images of authenticated users which makes the system secure. The image capturing is done

automatically when the authenticate user is logged in based on some time interval. The captured

images are used to detect the stress of the user based on some standard conversion and image

processing mechanisms. Then the system will analyse the stress levels by using Machine

Learning algorithms which generates the results that are more efficient

**CHAPTER 8 FUTURE SCOPE**

Biomedical wearable sensors embedded with IOT technology is a proven combination in the health care sector. The benefits of using such devices have positively impacted the patients and doctors alike. Early diagnosis of medical conditions, faster medical assistance by means of Remote Monitoring and Telecommunication, emergency alert mechanism to notify the care taker and personal Doctor, etc are a few of its advantages. The proposed work on developing a multimodal IOT system assures to be a better health assistant for a person by constantly monitoring and providing regular feedback on the stress levels. For future work, it would be interesting to enhance this work into the development of a stress detection model by the addition of other physiological parameters, including an activity recognition system and application of machine learning techniques.

**REFERENCS**

[1**] G. Giannakakis, D. Manousos, F. Chiarugi** (**2017**) “Stress and anxiety detection using facial cues from videos,” Biomedical Signal processing and Control”, vol. 31, pp. 89-101.

[2] **Nisha Raichur, Nidhi Lonakadi, Priyanka Mural**, (**2017**) “Detection of Stress Using Image Processing and Machine Learning Techniques”, vol.9, no. 3S, July.

[3] **U. S. Reddy, A. V. Thota and A. Dharun**, (**2018**)"Machine Learning Techniques for Stress Prediction in Working Employees," IEEE International Conference on

Computational Intelligence and Computing Research (ICCIC), Madurai, India, pp.1-4.

[4] **Tanev, G., Saadi, D.B., Hoppe, K., & Sorensen, H.B. (2014).** Classification of acute stress using linear and non-linear heart rate variability analysis derived from sternal ECG.

[5] **Zenonos, A., Khan, A., Kalogridis, G., Vatsikas, S., Lewis, T., & Sooriyabandara. (Year).** HealthyOffice: Mood recognition at work using smartphones and wearable sensors.

32