* 1. **Machine Learning**

# CHAPTER 1 INTRODUCTION

Machine learning is a type of artificial intelligence that allows computer systems to automatically improve their performance on a task through experience. It involves the use of algorithms and statistical models to analyze and identify patterns in data, and then use those patterns to make predictions or decisions without being explicitly programmed. Machine learning can be categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning involves training a machine learning model on labeled data, where the model learns to predict an output variable based on a set of input variables. Unsupervised learning, on the other hand, involves training a model on unlabeled data, where the model must identify patterns or structure within the data without any specific guidance. Reinforcement learning involves training a model through a trial-and-error process, where the model receives feedback in the form of rewards or penalties for each decision it makes.

Machine learning has numerous applications across many fields, including computer vision, natural language processing, speech recognition, fraud detection, recommendation systems, and autonomous vehicles, among others.

## Deep Learning

Deep learning is a subset of machine learning that involves the use of neural networks with multiple layers. These networks are designed to simulate the way the human brain works, by using layers of interconnected nodes or "neurons" to process and analyze data.

In a deep neural network, each layer extracts features or patterns from the

1

data and passes that information to the next layer, which then extracts even more complex features. The output of the final layer can be used to make predictions or decisions based on the input data.

Deep learning has been highly successful in a variety of applications, including image and speech recognition, natural language processing, and computer vision. One of the key advantages of deep learning is that it can automatically learn and extract relevant features from raw data, without the need for manual feature engineering, making it highly effective for processing unstructured data such as images and text.

## Convolutional Neural Network

A CNN or Convolutional Neural Network, is a type of deep neural network commonly used in image and video processing applications. It is designed to automatically learn and extract features from images through a process called convolution.

The key feature of a CNN is its use of convolutional layers, which consist of a set of filters or "kernels" that slide across an input image, performing mathematical operations at each position to produce a feature map. The resulting feature map highlights patterns or features of the image that are relevant to the given task, such as edges, shapes, or textures.

In addition to convolutional layers, a typical CNN also includes pooling layers, which reduce the spatial dimensions of the feature map by down sampling or subsampling the data, and fully connected layers, which perform the final classification or regression based on the extracted features.

CNNs have achieved state-of-the-art performance in a variety of computer vision tasks, including image classification, object detection, and segmentation, and

are widely used in industry and academia for a wide range of applications such as

autonomous vehicles, facial recognition, and medical imaging.

## Object Detection

Object detection is an important aspect of face recognition technology as it helps in identifying and locating objects of interest within an image or video frame. In the context of face recognition, object detection is used to identify and locate faces within an image or video stream.

Objection detection for face recognition involves the use of various algorithms and techniques to detect and locate faces in an image or video stream. These algorithms use machine learning and deep learning techniques to analyse the image or video frame and identify the location of the face. The most commonly used algorithms for object detection in face recognition include Haar cascades, Histogram of Oriented Gradients (HOG), and Convolutional Neural Networks (CNN).

Haar cascades use a series of simple classifiers to detect the presence of a face within an image or video frame. HOG algorithms use gradient orientation information to detect the edges of objects within an image, including faces. CNNs use a deep neural network to learn and classify the features of faces, allowing for more accurate and reliable detection.

Overall, objection detection is a crucial component of face recognition technology as it allows for accurate and reliable identification and authentication of individuals in a variety of settings, including security and surveillance, identity verification, and biometric authentication systems.

## Face Recognition

Face recognition is a computer vision technology that involves identifying individuals from digital images or videos. Face recognition technology has numerous applications, including security systems, surveillance, and attendance tracking. In Python, face recognition can be implemented using the face recognition library, which is built on top of deep learning models.

The face recognition library in Python utilizes convolutional neural networks (CNNs) to extract facial features from images and videos. The CNNs are trained on large datasets of faces to learn to identify unique features that differentiate one person from another. These unique features are then used to recognize individuals in new images and videos.

The face recognition library in Python supports different types of face recognition algorithms, including the Eigenfaces algorithm, Fisherfaces algorithm, and Local Binary Patterns Histograms (LBPH) algorithm is used in attendance system. These algorithms differ in their approach to feature extraction and matching, but they all rely on deep learning models to identify individuals accurately.

To use the face recognition library in Python, one needs to first install the library and its dependencies, including OpenCV, Dlib library and NumPy. Once installed, the library can be used to load pre-trained models, detect faces in images and videos, and recognize individuals. The library also supports training custom models on custom datasets, which can be used for specific applications. The dlib library in python used to provide tools for face detection, landmark detection, face recognition, face shape prediction and it can able to detect faces under low light and partial occlusion with combination of ML algorithms like HOG algorithm. Attendance system will identify the faces of person who works in Office or Laboratory will announce the workspace name and name of the employee.

## Temperature Detection

DS18B20 temperature sensor are used to measure the body temperature of individuals as they enter a building or room, allowing for the detection of fever, which is one of the common symptoms of COVID-19.

In a facial recognition attendance system that includes temperature sensors, the system first detects the individual's face using facial recognition technology, and then uses the temperature sensor to measure their body temperature. If the temperature reading is within the acceptable range, the individual is granted access to the building or room, and their attendance is marked.

The temperature sensor used in these systems is typically a non-contact infrared sensor that measures the infrared radiation emitted by the body. This type of sensor is ideal for use in attendance systems because it does not require physical contact with the individual, reducing the risk of disease transmission.

## Result Analysis

The Results of attendance system are announced via voice engine and store data about Registered candidate or employee details. The Three results are provided as output are

* **Authentication of Employee**:
  + By Identifying Employee identity by using Facial Recognition and dataset and broadcast by means of voice welcome message.
* **Thermal Analysis:**
  + After Authentication of Employee, Temperature are inspected by sensor and stated by voice and if the temperature is higher, then it will notify to System admin.
* **Database:**
  + At last, The Records are stored in format of Comma-Separated Values or CSV file for future purpose.

# CHAPTER 2 SYSTEM ANALYSIS

## Literature Survey

**[1] Khushbu Gupta, Aakanksha S, A Review on Face Detection based Attendance System with Temperature Monitoring, 20 December 2020.**

The face is the identity of a person. The method to exploit this physical feature have seen a great change since the advent of image processing techniques. The attendance is taken in every school, colleges and library. Traditional approach for attendance is professor calls student name & record attendance. Now a days, Machine Learning has been highly explored for computer vision applications. So, we use the concept of machine learning in Face – recognition for automatic attendance systems. In this project, we perform the face recognition and face detection algorithms, to provide the computer systems the ability of finding and recognizing human faces fast and precisely in images or videos so that the systems can used in giving attendance. Along with the face detection temperature measurement is also done using mlx90614 infrared temperature sensor.

**[2]** **Busra Kocacinar, Bilal Tas, Fatma Patlar Akbulut, (Member, Ieee), Cagatay Catal , And Deepti Mishra, (Senior Member, Ieee) ,** **A Real-Time CNN-Based Lightweight Mobile Masked Face Recognition System, June 13,2022.**

Due to the global spread of the Covid-19 virus and its variants, new needs and problems have emerged during the pandemic that deeply affects our lives. Wearing masks as the most effective measure to prevent the spread and transmission of the virus has brought various security vulnerabilities. Today we are going through times when wearing a mask is part of our lives, thus, it is very important to identify individuals who violate this rule. Besides, this pandemic makes the traditional biometric authentication systems less effective in many cases such as facial security checks, gated community access control, and facial attendance. So far, in the area of masked face recognition, a small number of contributions have been accomplished. It is definitely imperative to enhance the recognition performance of the traditional face recognition methods on masked faces. Existing masked face recognition approaches are mostly performed based on deep learning models that require plenty of samples. Nevertheless, there are not enough image datasets containing a masked face. As such, the main objective of this study is to identify individuals who do not use masks or use them incorrectly and to verify their identity by building a masked face dataset. On this basis, a novel real-time masked detection service and face recognition mobile application was developed based on an ensemble of fine-tuned lightweight deep Convolutional Neural Networks (CNN). The proposed model achieves 90.40% validation accuracy using 12 individuals’ 1849 face samples. Experiments on the five datasets built in this research demonstrate that the proposed system notably enhances the performance of masked face recognition compared to the other state-of-the-art approaches.

**[3]** **Chaoyou Fu, Xiang Wu, Yibo Hu, Huaibo Huang, and Ran He\*, Senior Member, IEEE, DVG-Face: Dual Variational Generation for Heterogeneous Face Recognition, July, 2020.**

Heterogeneous Face Recognition (HFR) refers to matching cross-domain faces and plays a crucial role in public security. Nevertheless, HFR is confronted with challenges from large domain discrepancy and insufficient heterogeneous data. In this paper, we formulate HFR as a dual generation problem, and tackle it via a novel Dual Variational Generation (DVG-Face) framework. Specifically, a dual variational generator is elaborately designed to learn the joint distribution of paired heterogeneous images. However, the small-scale paired heterogeneous training data may limit the identity diversity of sampling. In order to break through the limitation, we propose to integrate abundant identity information of large-scale visible data into the joint distribution. Furthermore, a pairwise identity preserving loss is imposed on the generated paired heterogeneous images to ensure their identity consistency. As a consequence, massive new diverse paired heterogeneous images with the same identity can be generated from noises. The identity consistency and identity diversity properties allow us to employ these generated images to train the HFR network via a contrastive learning mechanism, yielding both domain-invariant and discriminative embedding features. Concretely, the generated paired heterogeneous images are regarded as positive pairs, and the images obtained from different samplings are considered as negative pairs. Our method achieves superior performances over state-of-the-art methods on seven challenging databases belonging to five HFR tasks, including NIR-VIS, Sketch-Photo, Profile-Frontal Photo, Thermal-VIS, and ID-Camera.

**[4]** [**A Arjun Raj**](https://ieeexplore.ieee.org/author/37088461211)**, [Mahammed Shoheb](https://ieeexplore.ieee.org/author/37088461414),** [**K Arvind**](https://ieeexplore.ieee.org/author/37088459498)**,**[**K S Chethan**](https://ieeexplore.ieee.org/author/37086534950)**, A Face Recognition Based Smart Attendance System, June 19, 2020.**

Education institutes today are concerned about the consistency of students ' performance. One cause of this decrease in student performance is the inadequate attendance. There are several ways to mark your attendance, the most common ways to sign or call the students. It took longer and was problematic. From now on, a computer-based student attendance checking system is required that supports the faculty to keep records of attendance. We have used an intelligent attendance system based on face recognition in this project. We have proposed to implement a "Smart Attendance System for Face Recognition" through this large applications are incorporated. The present implementation includes facial identification that is time saving and eradicates the possibilities of proxy attendance due to the facial authorization. This system can now be used in an area in which participation plays an important role. Raspberry Pi, Open CV and Dlib using python are the basic requirements for this system. The system implemented uses LBPH face recognizer to identify the face of the person in real time. Eigen faces and Fisher faces are affected both by light and we cannot ensure perfect light conditions in real life. An improvement in the LBPH faces recognizer to overcome this problem. This system compares the image of the test and the training image and determines who is and is not present. The attendance data is stored in an excel sheet that is automatically updated in the system. If a student is absent a message will be automatically sent to their parent's phone number using GSM. Student's can check their attendance using an Android application that we have developed using MIT app Inventor.

**[5] Dr. V Suresh, Srinivasa Chakravarthi Dumpa, Chiranjeevi Deepak Vankayala, HaneeshaAduri, Jayasree Rapa, Facial Recognition Attendance System Using Python and OpenCv, March 5, 2020.**

The main purpose of this project is to build a face recognition-based attendance monitoring system for educational institution to enhance and upgrade the current attendance system into more efficient and effective as compared to before. The current old system has a lot of ambiguity that caused inaccurate and inefficient of attendance taking. Many problems arise when the authority is unable to enforce the regulation that exist in the old system. The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals are mailed to the respective faculty.

## Existing System

The existing system is based on Face Recognition Attendance Management System which integrates images of person’s face authentication into the process of attendance management for both staff and student. It is made up of two processes namely; enrolment and authentication. Attendance management is the act of managing attendance or presence in a work setting to minimize loss due to employee downtime. Attendance control has traditionally been approached using time clocks and timesheets, but attendance management goes beyond this to provide a working environment which maximizes and motivates employee attendance.

## Disadvantages

* Traditional projects did not use Real time liveness of people’s image.
* People must submit the picture in order to detect face so it will take more time.
* Accuracy will be low.

## Proposed System

The Goal of the proposed system is to realize an attendance system that can be managed through face recognition. To be a good attendance system, it must have several characteristics, namely a high recognition rate, fast recognition, and the ability to cope with many different faces. This work uses two-stage networks to reach the goal of face detection and recognition. First, the system will continuously detect the input screen. When a face is detected on the screen, the system will immediately crop the face. Then, the cropped face image will be sent to the face recognition model to generate a facial feature vector, which will be compared with the facial features previously built into the database to confirm the identity of the person. Finally, the recognition results will be announced by voiceover and body temperature will be measured by using temperature sensor. After this process will be done, it will store a user information on CSV file format. This method can be used to record the temperature of people while assisting them to punch in and out. In today’s rampant COVID-19 environment, it is important to measure and record the temperature of people entering and exiting.

## Advantages

* For security purpose the face recognition will validate the liveness of people.
* High reliability and accuracy.
* Time consumption is less.
* Real – time streaming makes the system more convenient to detect the face recognition.

# CHAPTER 3 SYSTEM REQUIREMENTS

## Hardware Specification

* CPU type : Intel I5
* RAM size : 4.00 GB.
* Monitor type : 1-inch color monitor.
* Keyboard type : Qwerty keyboard.
* Webcam

## Hardware Description

* + 1. **webcam**

The system uses inbuilt USB 2.0 webcam for capturing and performing social distance operations. A webcam is a video camera that feeds or streams an image or video in real time to or through a computer to a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video. For example, Apple's iSight camera, which is built into Apple laptops, iMacs and a number of iPhones, can be used for video chat sessions, using the Messages instant messaging program. Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires much bandwidth, such streams usually use compressed formats. The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions.

The term "webcam" (a clipped compound) may also be used in its original sense of a video camera connected to the Web continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its web page over the Internet. Some of them, for example, those used as online traffic cameras, are expensive, rugged professional video cameras.

## Software Specification

* OS : window 7
* Software tool: python 3.9

## Software Description 3.4.1Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is a high-level programming language designed to be easy to read and simple to implement. It is open source, which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to Java and .NET virtual machines.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

## 3.4.2 Command Prompt

Command Prompt is a command line interpreter application available in most Windows operating systems. It's used to execute entered commands. Most of those commands automate tasks via scripts and batch files, perform advanced administrative functions, and troubleshoot or solve certain kinds of Windows issues.

Command Prompt is officially called Windows Command Processor, but it's also sometimes referred to as the command shell or cmd prompt, or even by its filename, cmd.exe. Command Prompt is sometimes incorrectly referred to as "the DOS prompt" or as MS-DOS itself. Command Prompt is a Windows program that emulates many of the command line abilities available in MS-DOS, but it's not MS- DOS. Command Prompt is available on every Windows NT-based operating system which includes Windows 10, Windows 8, Windows 7, Windows Vista, Windows XP, and Windows 2000, as well as Windows Server 2012, 2008, and 2003.

Windows PowerShell, an advanced command line interpreter available in recent Windows versions, supplements the command executing abilities available in Command Prompt. Windows PowerShell may eventually replace the Command Prompt in a future version of Windows.

# CHAPTER 4 SYSTEM DESIGN

## Architecture Diagram

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. Architecture diagram is shown in Fig 4.1



Training Dataset

CNN

Algorithm

Model

Testing dataset for prediction

Classifier Module

Stress Relief

Person under Stress

Stress detection using CNN result

CNN Algorithm

Image processing

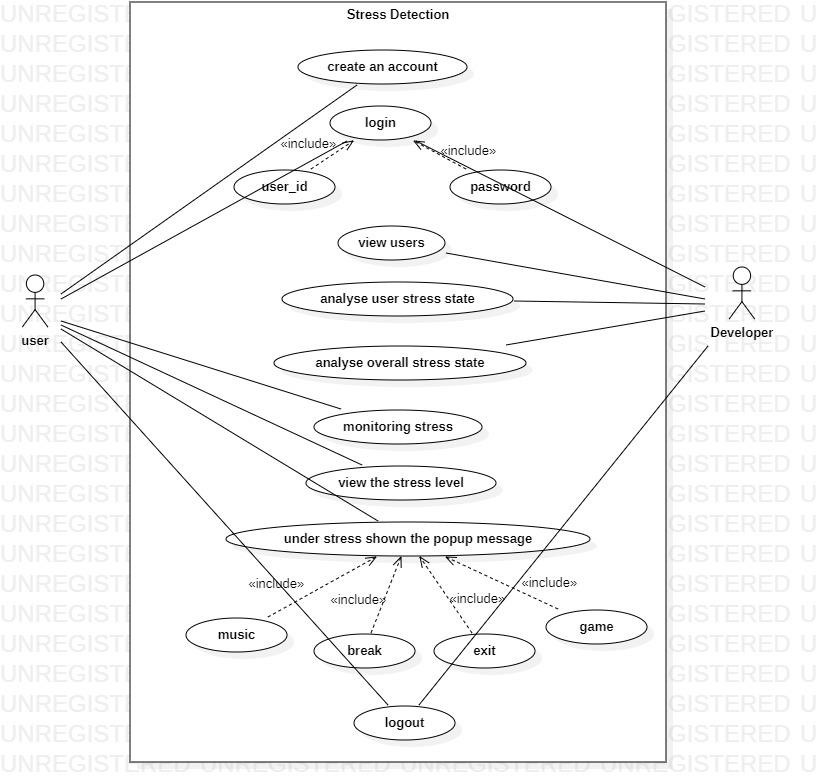
Live video capturing

Face Acquisition and Pre-processing module

**Fig 4.1 Architecture diagram**

## Usecase Diagram

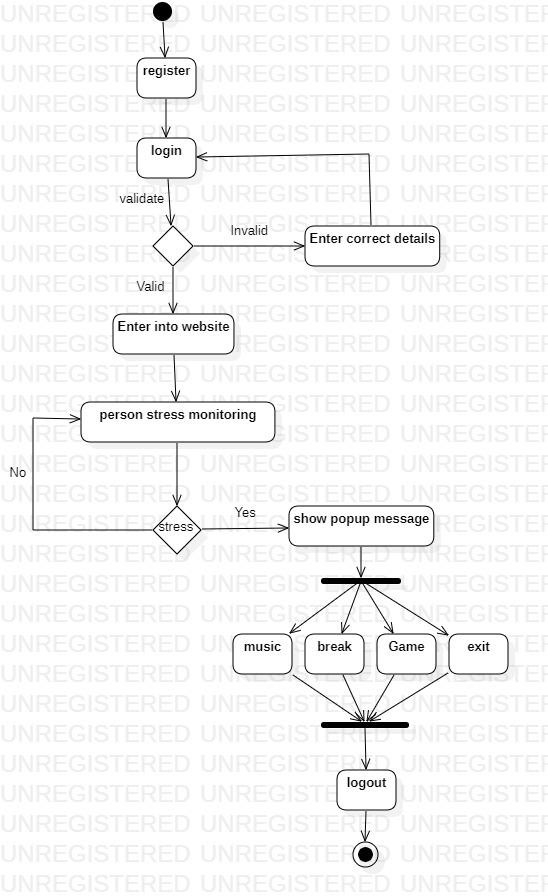
A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. Use case diagram is shown in the Fig 4.2



**Fig 4.2 use case diagram**

## Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. Activity diagram is shown in the Fig 4.3



**Fig 4.3 activity diagram**

## List of Modules

* + - Data CollectionCHAPTER 5 SYSTEM IMPLEMENTATION
    - Stress detection Using CNN Algorithm
    - Auto Relaxing Alert
    - Performance Analysis

## Module Description

* + 1. **Data Collection**

Web design is a process of planning, conceptualizing and arranging content online. By using website builder, the webpage gets created. Then HTML and CSS language going to design the unique webpage.

**Create a registration form:** The first step is to create a registration form where users can enter their personal information, including their desired username and password.

**Store user information:** Once a user submits the registration form, their information is stored in a database. This typically includes their username and password.

**Implement password security measures:** It is important to implement password security measures to protect users' accounts from unauthorized access. This may include requiring users to create strong passwords that meet specific criteria (e.g., minimum length, use of special characters).

**Implement login functionality:** Once a user has registered, they can log in using their username and password. This typically involves creating a login form where users can enter their credentials.

**Authenticate users:** When a user logs in, their username and password are compared against the information stored in the database to authenticate them. If the credentials match, the user is granted access to their account.

## Stress Detection Using CNN Algorithm

First preprocess the data so that input data needs to be preprocessed to ensure that it is in a suitable format for the CNN model. This may involve resizing images, normalizing values, and converting data to a suitable format for the CNN model. Next split the data of the preprocessed data needs to be split into training, validation, and testing sets. The training set is used to train the CNN model, the validation set is used to optimize the hyper parameters of the model, and the testing set is used to evaluate the performance of the model. Then define the CNN architecture based on the input data and the task at hand. The architecture may include convolutional layers and fully connected layers. The number and size of the layers may vary depending on the complexity of the input data and the desired performance of the model. The CNN model needs to be trained using the training data. During training, the weights of the model are adjusted to minimize the error between the predicted outputs and the actual outputs. The hyper parameters of the model need to be optimized using the validation set. This may involve adjusting parameters such as learning rate, batch size, and number of epochs to improve the performance of the model. The performance of the CNN model needs to be evaluated using the testing set. This involves measuring metrics such as accuracy, precision, and recall to determine how well the model can detect stress. Once the CNN model is trained and evaluated, it can be used to make predictions about the presence or absence of stress in new input data. The facial landmarks idxs dictionary dataset is used to detect stress in indices of left and right eyebrows using CNN

Algorithm. For each frame, the stress levels are determined using the normalize values approach. To returns a stress value and stress label using eyebrow distance, current frame distance and calculated normalized value.

## Auto Relaxing Alert

To relief a stress using auto relaxing alert message will be displayed. To overcome stress will set some activities like music, games, break, etc. Reduce stress by engaging in these activities.

**Music:** Listening to calming or uplifting music can help reduce stress and improve mood. It can also help distract from negative thoughts or emotions. Some people find that playing music or singing can also be a helpful way to cope with stress.

**Games:** Engaging in a fun activity like playing games can help take the mind off of stressors and provide a sense of relaxation. This could include board games, or card games. Some people also find physical activities like sports or exercise to be a helpful form of stress relief.

**Break:** Taking a break from work or responsibilities can be an important way to manage stress. This could involve taking a walk, practicing meditation or deep breathing, or simply sitting in a quiet space for a few minutes. After the end of 15 minutes you can continue your work.

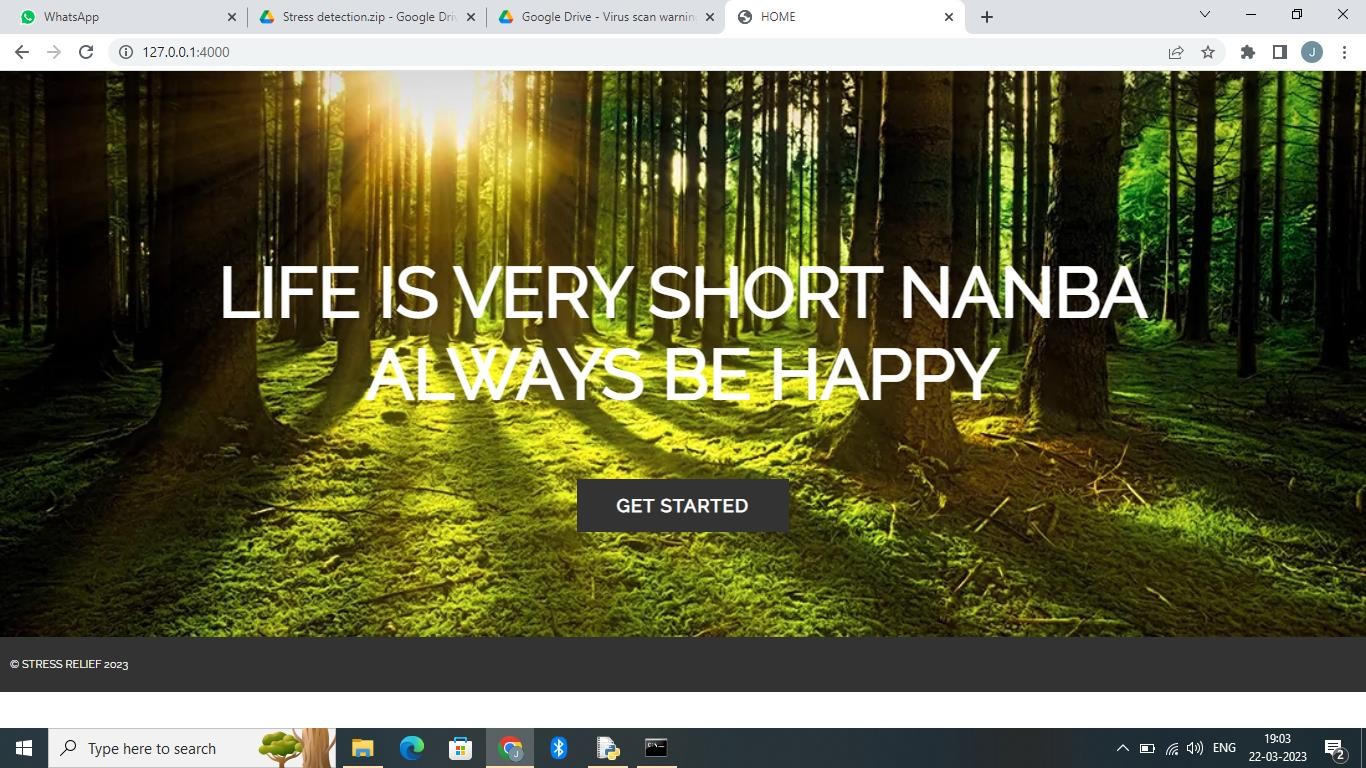
**Exit:** By selecting this option the employer leave from the pop-up message and it continue monitoring process.

## Performance Analysis

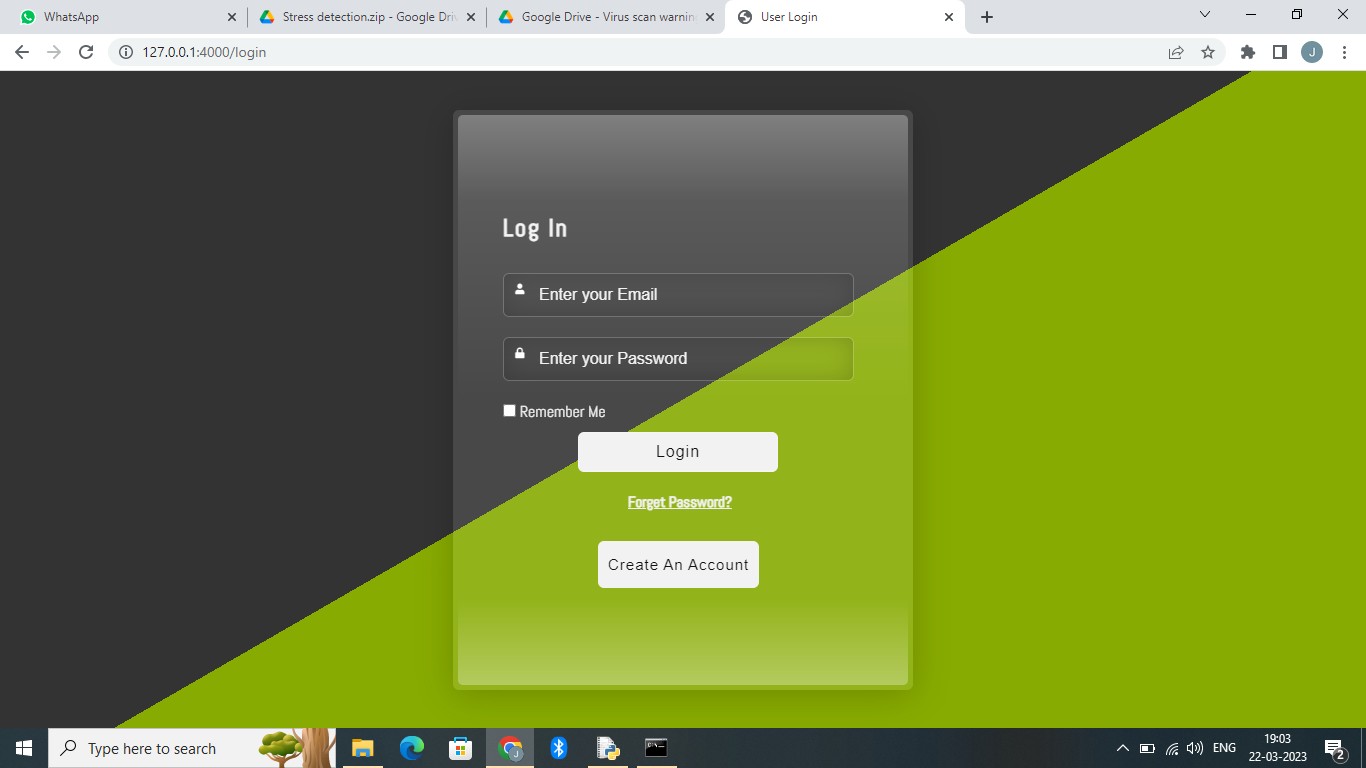
The choice of data used to train and test a deep learning model is crucial for its performance. The dataset used should be large and diverse enough to cover various stress scenarios and should also be balanced across different classes. The

dataset should also be annotated with stress labels to enable supervised learning. To compare the performance and accuracy of stress using various Machine Learning (ML) models. Accuracy, precision, recall, and F1-Score are used to assess the data's performance. Then to display stress level and stress label on the frame using OpenCV’s put Text method and the frame is shown on window. The stress value is also appended to a list of stress values for each frame and the normalized value is plotted using matplotlib.

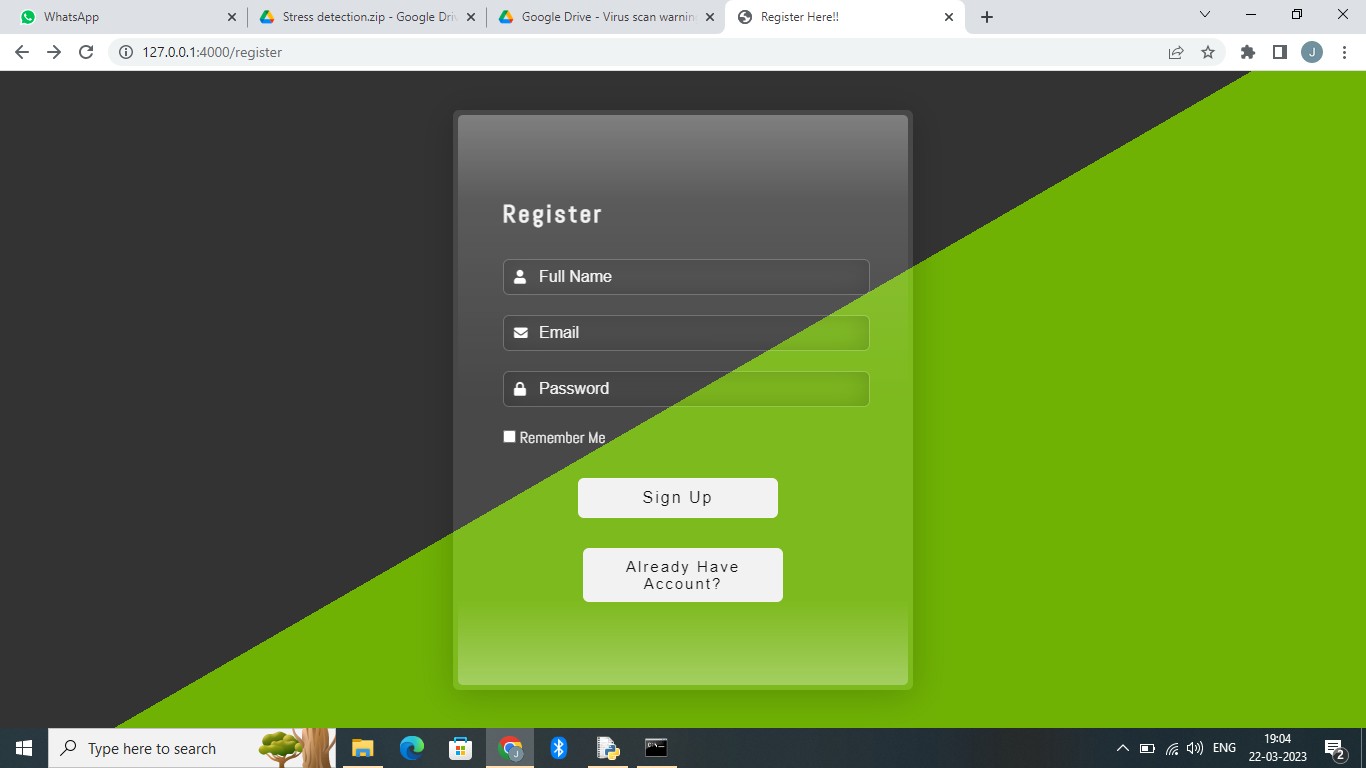
# CHAPTER 6 RESULTS AND DISCUSSION



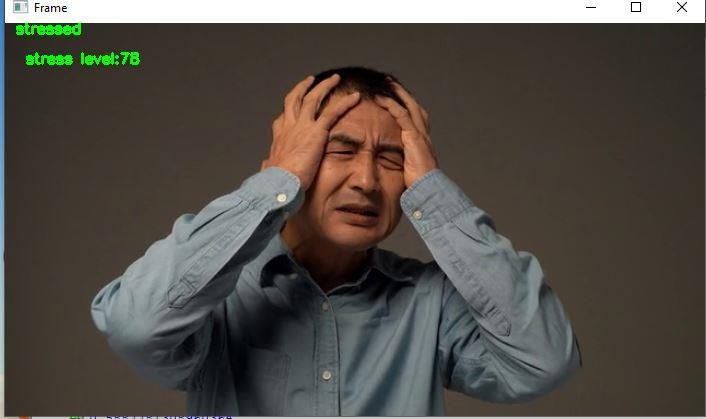
**Fig 6.1 Stress Detection Webpage**



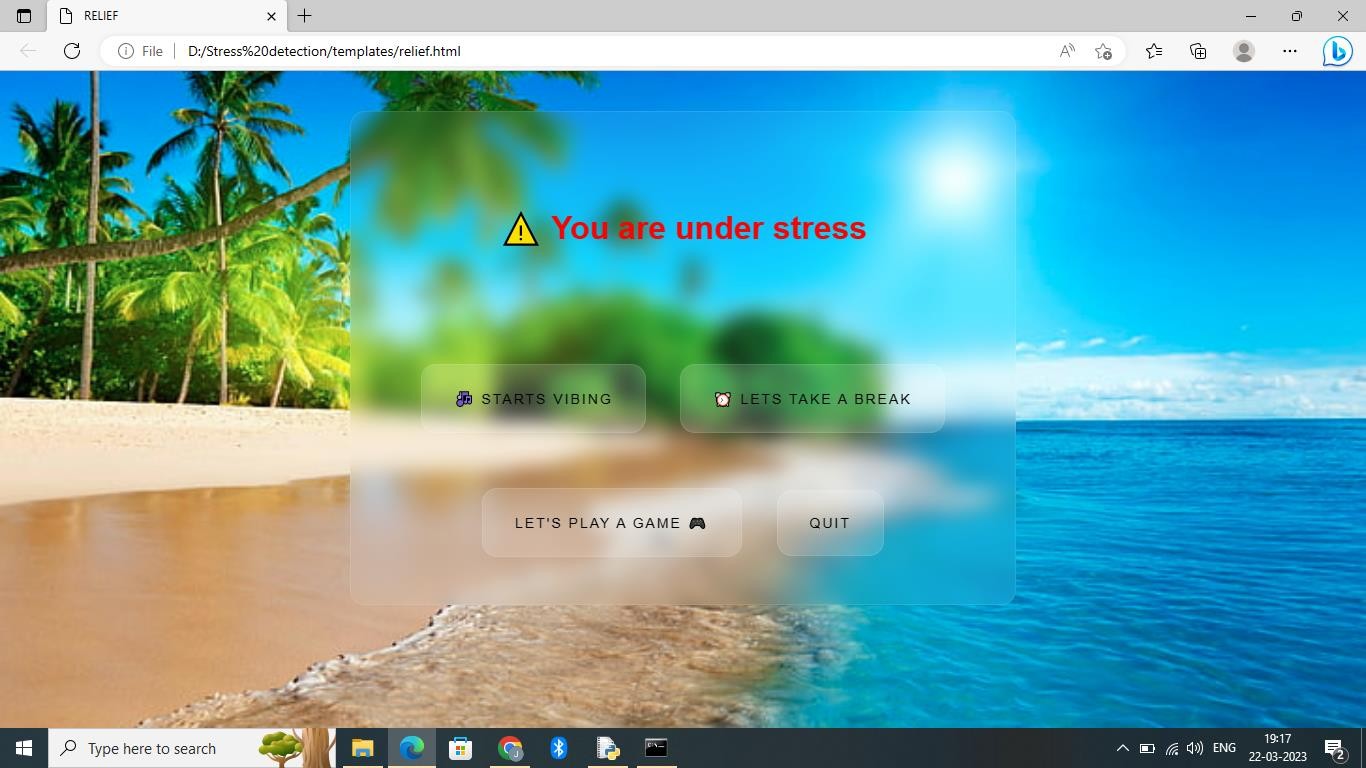
**Fig 6.2 Login Page**



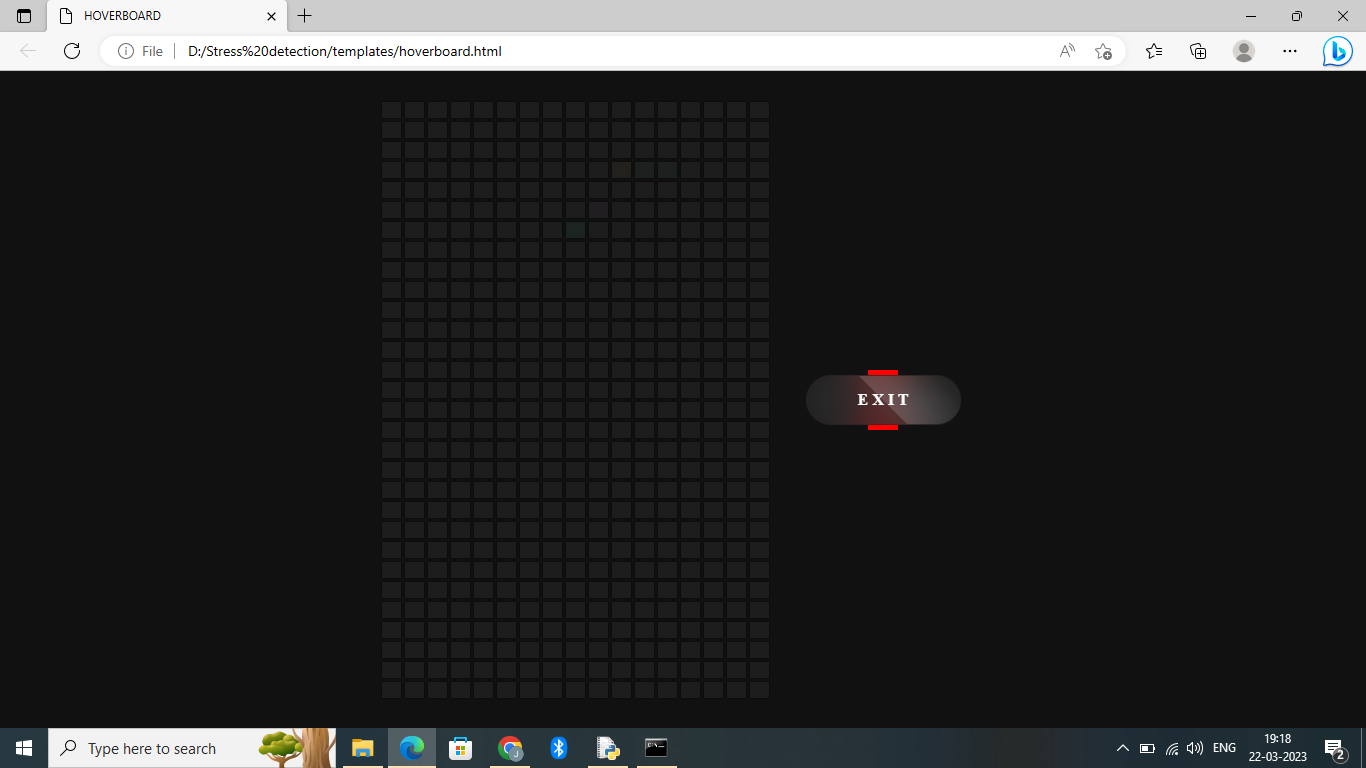
**Fig 6.3 Sign-up page**



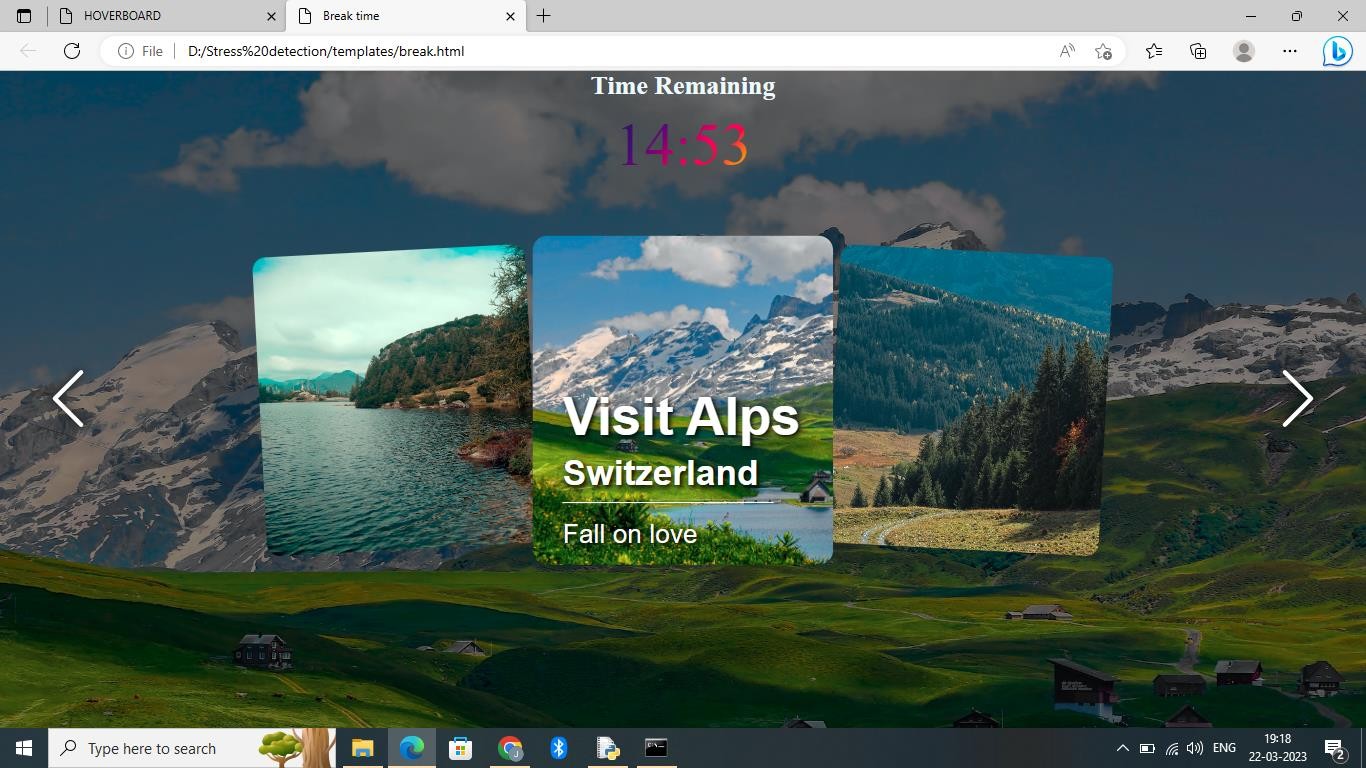
**Fig 6.4 Stress Monitoring**



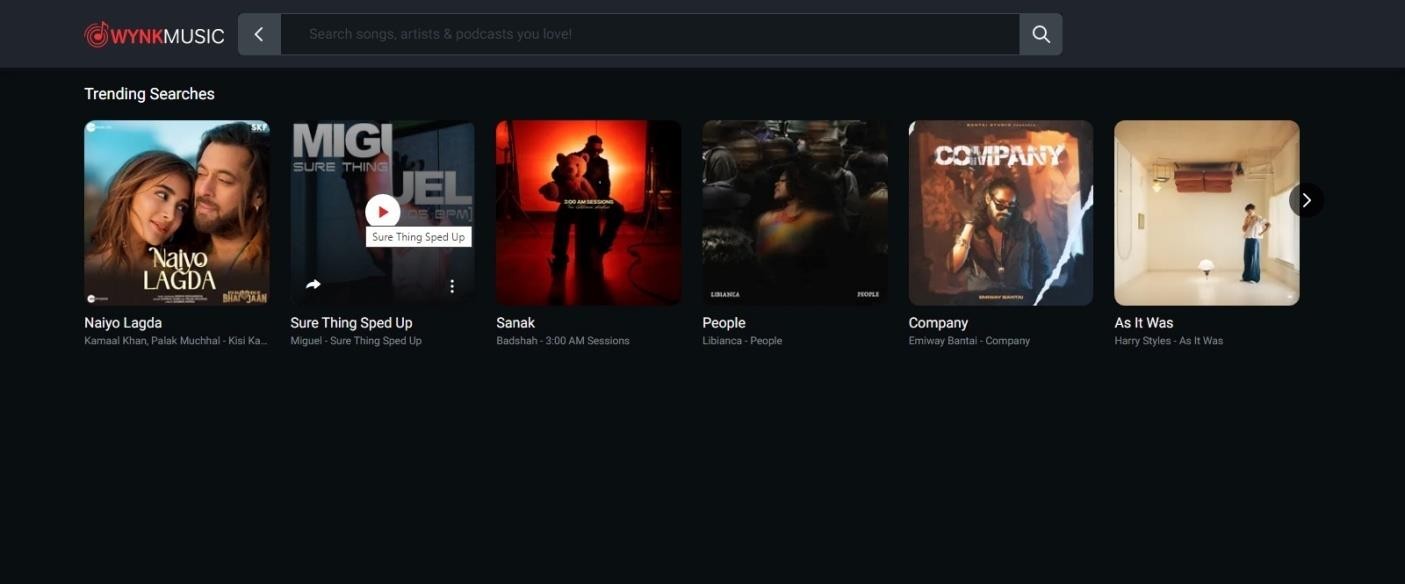
**Fig 6.5 pop-up message page**



**Fig 6.6 Game Page**



**Fig 6.7 Break Option Page**



**Fig 6.8 Music Page**

**CHAPTER 7**

# SAMPLE CODING

## Attendance System Code:

import cv2

import numpy as np

import face\_recognition

import os

from datetime import datetime

import pyttsx3

import serial.tools.list\_ports

path = 'ImagesAttendance'

images = []

personNames = []

myList = os.listdir(path)

print(myList)

for cl in myList:

curImg = cv2.imread(f'{path}/{cl}')

images.append(curImg)

personNames.append(os.path.splitext(cl)[0])

print(personNames)

def findEncodings(images):

encodeList = []

for img in images:

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

encode = face\_recognition.face\_encodings(img)[0]

encodeList.append(encode)

return encodeList

def markAttendance(name):

with open('Attendance.csv','r+') as f:

myDataList = f.readlines()

nameList = []

for line in myDataList:

entry = line.split(',')

nameList.append(entry[0])

if name not in nameList:

now = datetime.now()

dtString = now.strftime('%H:%M:%S')

f.writelines(f'\n{name},{dtString}')

encodeListKnown = findEncodings(images)

print('Encoding Complete')

cap = cv2.VideoCapture(0)

while True:

success, img = cap.read()

imgS = cv2.resize(img,(0,0),None,0.25,0.25)

imgS = cv2.cvtColor(imgS, cv2.COLOR\_BGR2RGB)

facesCurFrame = face\_recognition.face\_locations(imgS)

encodesCurFrame = face\_recognition.face\_encodings(imgS,facesCurFrame)

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

matches = face\_recognition.compare\_faces(encodeListKnown,encodeFace)

faceDis = face\_recognition.face\_distance(encodeListKnown,encodeFace)

print(faceDis)

matchIndex = np.argmin(faceDis)

if matches[matchIndex]:

name = personNames[matchIndex].upper()

print(name)

y1,x2,y2,x1 = faceLoc

y1, x2, y2, x1 = y1\*4,x2\*4,y2\*4,x1\*4

cv2.rectangle(img,(x1,y1),(x2,y2),(0,255,0),2)

cv2.rectangle(img,(x1,y2-35),(x2,y2),(0,255,0),cv2.FILLED)

cv2.putText(img,name,(x1+6,y2-6),cv2.FONT\_HERSHEY\_COMPLEX,1,(255,255,255),2)

markAttendance(name)

engine = pyttsx3.init()

engine.setProperty("rate", 120)

engine.say(name)

engine.say("Welcome To K L N college of Engineering")

engine.runAndWait()

ports = serial.tools.list\_ports.comports()

serialInst = serial.Serial()

portsList = []

for onePort in ports:

portsList.append(str(onePort))

print(str(onePort))

val = input("Select Port: COM")

for x in range(0, len(portsList)):

if portsList[x].startswith("COM" + str(val)):

portVar = "COM" + str(val)

print(portVar)

serialInst.baudrate = 9600

serialInst.port = portVar

serialInst.open()

while True:

if serialInst.in\_waiting:

packet = serialInst.readline()

print(packet.decode('utf').rstrip('\n'))

cv2.imshow('Webcam',img)

cv2.waitKey(1)

# CHAPTER 8 SYSTEM TESTING

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that “al gears mesh”, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

## Unit Testing

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as ‘module testing’. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some

validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

## Test Strategy And Approach

Field testing will be performed manually and functional tests will be written in detail.

## Test Objectives

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

## FEATURES TO BE TESTED

* Verify that the entries are of the correct format.
* No duplicate entries should be allowed.
* All links should take the user to the correct page.

## Integration Testing

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

1. Top-down integration testing.
2. Bottom-up integration testing.

## Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* Valid Input - identified classes of valid input must be accepted.
* Invalid Input - identified classes of invalid input must be rejected.
* Functions - identified functions must be exercised.
* Output - identified classes of application outputs must be exercised.
* Systems/Procedures - interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

## Testing Techniques/Strategies 8.4.1White Box Testing

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

## 8.4.2 Black Box Testing

* Black box testing is done to find incorrect or missing function
* Interface error
* Errors in external database access
* Performance errors.
* Initialization and termination errors

In ‘functional testing’, is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called ‘black box testing’. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

## Software Testing Strategies

* + 1. **Validation Testing**

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

## User Acceptance Testing

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

## Output Testing

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

# CHAPTER 9

**CONCLUSION AND FUTURE ENHANCEMENT**

## Conclusion

* It is designed to 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.
* It is used to detect the stress of the user based on some standard conversion and image processing mechanisms.
* Analyze the stress levels by using Machine Learning algorithms which generates the results that are more efficient.

## Future Enhancement

* Developing a chatbot will lead to greater satisfaction and trust among peoples.
* Voice over will be developed for future preferences on people.
* Software application will be implemented to access easily.
* Videos will be added for relaxation of people.

# CHAPTER 10 REFERENCES

1. NishaVishnupantKimmatkar and B. VijayaBabu, (2021)"Novel Approach for Emotion Detection and Stabilizing Mental State by Using Machine Learning Techniques", Computers, vol. 10, no. 37, pp. 1-20.
2. Muhammad Zubair and Changwoo Yoon, (2020) "Multilevel mental stress detection using ultra-short pulse rate variability series", vol. 57, pp. 101736.
3. Alice D Souza and Dm Rio D Souza,(2019) "Affective Interaction based Hybrid Approach for Emotion Detection using Machine Learning", Second International Conference on Smart Systems and Inventive Technology.
4. MdFahimRizwan, Rayed Farhad and FarhanMashuk,(2019) "Design of a biosig- nal based Stress Detection System using machine learning techniques", International Conference on Robotics Electrical and Signal Processing Techniques (ICREST).
5. B. Padmaja, V. V. Rama Prasad and K. V. N. Sunitha,(2018) "A Machine Learning approach for stress detection using a wireless physical activity tracker", International Journal of Machine Learning and Computing.
6. KoldijkSaskia, Mark A. Neerincx and Wessel Kraaij,(2018) "Detecting Work Stress in offices by combining unobtrusive sensors", IEEE Transactions on affective computing.
7. Dharan,(2018) "Machine Learning Techniques for Stress Prediction in Working Employees," IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).
8. NishaRaichur, NidhiLonakadi, and Priyanka Mural, (2017)"Detection of Stress Using Image Processing and Machine Learning Techniques," vol.9, no. 3S.
9. G. Giannakakis, D. Manousos, F. Chiarugi. Zenonos, A., Khan, A., Kalogridis, G., Vatsikas, S., Lewis, T., Sooriyabandara, M.(2017)"Stress and anxiety detection using facial cues from videos," Biomedical Signal Processing and Control, vol. 31, pp. 89-101.
10. Alberto Greco, Gaetano Valenza, Antonio Lanata, Enzo Pasquale Scilingo and Luca Citi,(2016)"IEEE cvxEDA: A Convex Optimization Approach to Electrodermal Activity Processing", IEEE transactions on biomedical engineering.
11. A. Mestanikova, I. Ondrejka, M. Mestanik, I. Hrtanek, E. Snircova and I. Tonhajzerova,(2016)"Electrodermal activity in adolescent depression" in Pulmonary Infection and Inflammation., Springer.
12. Ghaderi, A., Frounchi, J., Farnam, A.(2015) “Machine learning based signal processing using physiological signals for stress detection”.22nd Iranian Conference on Biomedical Engineering (ICBME).
13. V. Sandulescu, S. Andrews, D. Ellis, N. Bellotto and O.M. Mozos,(2015) "Stress Detection Using Wearable Physiological Sensors" in Artificial Computation in Biology and Medicine. IWINAC 2015 Lecture Notes in Computer Science, Springer.