A MINI PROJECT REPORT

ON

STRESS DETECTION IN IT PROFESSIONALS BY USING IMAGE PROCESSING AND MACHINE LEARNING

Submitted in partial fulfilment of the requirement for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

N.CH.S.K. AKASH (20N71A6627)

M. SARITHA (20N71A6616)

M. JEEVAN (20N71A6618)

G. PRALAJ KUMAR (20N71A6604)

Under the guidance of

Dr. K. VENKATA SUBBAIAH

(ASSOCIATE PROFESSOR)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DRK INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Bowrampet(v), via Air force academy, Hyderabad-500043

2023-2024

DRK INSTITUTE OF SCIENCE AND TECHNOLOGY

(Affiliated to JNTU, Hyderabad)

Bowrampet(v), via Air force academy, Hyderabad-500043

CERTIFICATE

This is to certify that the project report entitled **"STRESS DETECTION IN IT PROFESSIONALS BY USING IMAGE PROCESSING AND MACHINE LEARNING"** being submitted by Mr./Mrs **N.CH.S.K. AKASH (20N71A6627), M.SARITHA (20N71A6616),M. JEEVAN (20N71A6618), G.PRALAJ KUMAR (20N71A6604)** in partial fulfillment for the award of the Degree of Bachelor of Technology in computer science and engineering to the Jawaharlal Nehru Technological University is a record of bonafide work carried out by him under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award for any Degree or Diploma.

INTERNAL GUIDE **HEAD OF DEPARTMENT**

Dr. K. VENKATA SUBBAIAH **Mr. K. PRAVEEN**

(ASSOCIATE PROFESSOR) **(ASSOCIATE PROFESSOR)**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are greatly indebted to **Mr. K. PRAVEEN,** Head of the epartment, Computer Science and Engineering, **DRK INSTITUTE OF SCIENCE AND TECHNOLOGY,** JNTU, Hyderabad, for his motivation and timely support in completion of our project work.

We express our deep sense of gratitude to our Internal Guide **Dr.K.VENKATA SUBBAIAH,** Dept of Computer Science and Technology**, DRK INSTITUTE OF SCIENCE AND TECHNOLOGY,** JNTU, Hyderabad, for his/her insightful guidance, kind concern and inspiration without which this work might not have been completed. We learnt many moral and ethical standards in academics and non-academic words from him/his.

We are thankful to all teaching and non-teaching members of Department of Computer Science and Engineering, **DRK INSTITUTE OF SCIENCE AND TECHNOLOGY, JNTU,** Hyderabad, for their cooperation and moral support.

With Sincere Regards,

N.CH.S.K. AKASH	**(20N71A6627)**
M. SARITHA	**(20N71A6616)**
M. JEEVAN	**(20N71A6618)**
G. PRALAJ KUMAR	**(20N71A6604)**

ABSTRACT

The main motive of our project is to detect stress in the IT professionals using vivid Machine learning and Image processing techniques. Our system is an upgraded version of the old stress detection systems which excluded the live detection and periodic analysis of employees and detecting physical as well as mental stress levels in his/her by providing them with proper remedies for managing stress by providing survey form periodically. Our system mainly focuses on managing stress and making the working environment healthy and spontaneous for the employees and to get the detection and the personal counseling but this system comprises of live best out of them during working hours.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF OUTPUT SCREENS

CHAPTER – 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. This 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 method 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 et al. proposed a 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 Electrodermal activity (EDA). An eye tracker sensor is also used which systematically analyses the eye movements with the stressors like Stroop word test and information related to pickup tasks. The authors of performed perceived stress detection by a set of non-invasive sensors which collects the physiological signals such as ECG, GSR,Electroencephalography (EEG), EMG, and Saturation of peripheral oxygen

(SpO2). Continuous stress levels are estimated using the physiological sensor data such EMG, HR, Respiration in. The stress detection is carried out effectively using Skin conductance level (SCL), HR, Facial EMG sensors by creating ICT related Stressors. Automated stress detection is made possible by several pattern recognition algorithms. Every sensor data is compared with a stress index which is a threshold value used for stressor conditions which were tested with Bayesian Network, J48 algorithm and detecting the stress level. The authors of collected data from 16 individuals under four Sequential Minimal Optimization (SMO) algorithm for predicting stress. Statistical features of heart rate, GSR, frequency domain features of heart rate and its variability (HRV), and the power spectral components of ECG were used to govern the stress levels. Various features are extracted from the commonly used physiological signals such as ECG, EMG, GSR, BVP etc., measured using appropriate sensors and selected features are grouped into clusters for further detection of anxiety levels. In, it is concluded that smaller clusters result in better balance in stress detection using the selected General Regression Neural Network (GRNN) model. This results in the fact that different combinations of the extracted features from the sensor signals provide better solutions to predict the continuous anxiety level. Frequency domain features like LF power (low frequency power from 0.04 Hz to0.15Hz), HF power (High frequency power from 0.15Hz to 0.4 Hz), LF/HF (ratio of LF to the HF). and time domain features like Mean, Median, standard deviation of heart signal are considered for continuous real time stress detection in. Classification using decision tree such as PLDA is performed using two stressors namely pickup task and stroop based word test wherein the authors concluded that the stressor based classification proves unsatisfactory. In 2016, Gjoreski et al. created laboratory based stress detection classifiers from ECG signal and HRV features. Features of ECG are analysed using GRNN model to measure the stress level. Heart rate variability (HRV) features and RR (cycle length variability interval length between two successive Rs) interval features are used to classify the stress level. It is noticed that Support Vector Machine (SVM) was used as the classification algorithm predominantly due to its generalization ability and sound mathematical background Various kernels were used to develop models using SVM and it is concluded in that a linear SVM on both ECG frequency features and HRV,

Nowadays as IT industries are setting a new peek in the market by bringing new technologies and products in the market. In this study, the stress levels in employees are also noticed to raise the bar high. Though there are many organizations who provide mental health related schemes for their employees but the issue is far from control. In this paper we try to go in the depth of this problem by trying to detect the stress patterns in the working employee in the companies we would like to apply image processing and machine learning techniques to analyze stress patterns and to narrow down the factors that strongly determine the stress levels. Machine Learning algorithms like KNN classifiers are applied to classify stress. Image Processing is used at the initial stage for detection, the employee's image is clicked by the camera which serves as input. In order to get an enhanced image or to extract some useful information from it image processing is used by converting image into digital form and performing some operations on it. By taking input as an image from video frames and output may be image or characteristics associated with that image.

Image processing basically includes the following three steps

- Importing the image via image acquisition tools.
- Analyzing and manipulating the image.
- Output in which result is altered image or report that is based on image analysis.

System gets the ability to automatically learn and improve from self-experiences without being explicitly programmed using Machine learning which is an application of artificial intelligence (AI). Computer programs are developed by Machine Learning that can access data and use it to learn for themselves. Explicit programming to perform the task based on predictions or decisions builds a mathematical model based on "training data" by using Machine Learning. The extraction of hidden data, association of image data and additional pattern which are unclearly visible in image is done using Image Mining. It's an interrelated field that involves, Image Processing, Data Mining, Machine Learning and Datasets. According to conservative estimates in medical books, 50- 80% machine learning techniques to analyze stress patterns and to narrow down the factors

of all physical diseases are caused by stress. Stress is believed to be the principal cause in cardiovascular diseases. Stress can place one at higher risk for diabetes, ulcers, asthma, migraine headaches, skin disorders, epilepsy, and sexual dysfunction. Each of these diseases, and host of others, is psychosomatic (i.e., either caused or exaggerated by mental conditions such as stress) in nature. Stress has three prong effects.

Subjective effects of stress include feelings of guilt, shame, anxiety, aggression or frustration. Individuals also feel tired, tense, nervous, irritable, moody, or lonely.

- Visible changes in a person's behavior are represented by Behavioral effects of stress. Effects of behavioral stress are seen such as increased accidents, use of drugs or alcohol, laughter out of context, outlandish or argumentative behavior, very excitable moods, and/or eating or drinking to excess.
- Diminishing mental ability, impaired judgment, rash decisions, forgetfulness and/or hypersensitivity to criticism are some of the effects of Cognitive stress.

CHAPTER – 2

SOFTWARE SPECIFICATIONS

The project involved analyzing the design of few applications so as to make the application more user friendly. To do so, it was really important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do.In order to make the application more accessible,the browser version had to be chosen so that it is compatible with most of the Browsers. The requirements required by the proposed project are as follows

→ Software Requirements

→ Hardware Requirements

2.1 HARDWARE REQUIREMENTS

- **System** : Intel i3
- **Hard Disk** : 1 TB.
- **Monitor** : 14’ Colour Monitor.
- **Mouse** : Optical Mouse.
- **RAM** : 4GB.

2.2 SOFTWARE REQUIREMENTS

- **Operating system** : Windows 10.
- **Coding Language** : Python.
- **Front – End** : HTML. CSS
- **Designing** : Html, CSS,JavaScript.
- **Data Base** : SQLite.

CHAPTER – 3

LITERATURE SURVEY

The literature survey study is to carried out to check the impact that the economic that will have on the organization. The amount that will spend on the research and development of the system by the organization will be limited. So, the expenditures must to be justified. Thus, to developed a system with in a budget can be achieved because now a days most of the technologies are used are freely available. But some of the customized products has to be purchase. To maintain a consistency in the organization literature survey, place an integral part. It is placing a crucial role in the development process.

1. Stress and anxiety detection using facial cues from videos

AUTHORS: G. Giannakakis, D. Manousos, F. Chiarugi

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 photoplethysmography. 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.

2. Detection of Stress Using Image Processing and Machine Learning Techniques

AUTHORS: Nisha Raichur, Nidhi Lonakadi, Priyanka Mural

Stress is a part of life it is an unpleasant state of emotional arousal that people experience in situations like 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 ups and 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 safety of a person. In this work a 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. We employ a technique that allows us to train a model and analyze differences in predicting the features. Theano is a python framework which aims at improving both the execution time and development time of the linear regression model which is used here as a deep learning algorithm. The experimental results show that the developed system is well on data with the generic model of all ages.learning algorithm. The experimental results show that the developed system is well on data with the generic model of all ages.

3. Machine Learning Techniques for Stress Prediction in Working Employees

AUTHORS: U. S. Reddy, A. V. Thota and A. Dharun

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 employees. 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 analyze stress patterns in working adults and to narrow down the factors that strongly determine the stress levels. Towards this, data from the OSMI mental health survey 2017 responses of working professionals within the tech-industry was considered. Various

Machine Learning techniques were applied to train our model after due data cleaning and preprocessing. The accuracy of the above models was obtained and studied comparatively. Boosting had the highest accuracy among the models implemented. By using Decision Trees, prominent features that influence stress were identified as gender, family history and availability of health benefits in the workplace. With these results, industries can now narrow down their approach to reduce stress and create a much comfortable workplace for their employees.

4. Classification of acute stress using linear and non-linear heart rate variability analysis derived from sternal ECG

AUTHORS: Tanev, G., Saadi, D.B., Hoppe, K., Sorensen, H.B

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 ePatch 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.

CHAPTER – 4

SOFTWARE REQUIREMENT ANALYSIS

Software requirement analysis simply means complete study,analysing,describing software requirements so that are genuine and needed can be fulfilled to solve problem. There are several activities involved in analysing software requirements.

4.1 EXISTING 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.

DISADVANTAGES OF EXISTING SYSTEM

- Physiological signals used for analysis are often pigeonholed by a Non-stationary time performance.
- 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
- Different people may behave or express differently under stress and it is hard to find a universal pattern to define the stress emotion.

  Algorithm: Bayesian Network, J48

4.2 PROPOSED SYSTEM

The proposed System Machine Learning algorithms like KNN classifiers are applied to classify stress. Image Processing is used at the initial stage for detection, the employee's image is given by the browser which serves as input. In order to get an

enhanced image or to extract some useful information from it image processing is used by converting image into digital form and performing some operations on it. By taking input as an image and output may be image or characteristics associated with that images. The emotion are displayed on the rounder box. The stress level indicating by Angry, Disgusted, Fearful, Sad.

ADVANTAGES OF PROPOSED SYSTEM

- Output in which result is altered image or report that is based on image analysis.
- Stress Detection System enables employees with coping up with their issues leading to stress by preventative stress management solutions.
- We will capture images of the employee based on the regular intervals and then the tradition survey forms will be given to the employees
  Algorithm: K-Nearest Neighbor (KNN)

4.3 MODULES AND THEIR FUNCTIONALITIES

4.3.1 MODULES

- User
- Admin
- Data Preprocess
- Machine Learning

4.3.2 MODULE FUNCTIONALITIES

1. **User**

The User can register the first. While registering he required a valid user email and mobile for further communications. Once the user register then admin can activate the customer. Once admin activated the customer then user can login into our system. First user has to give the input as image to the system. The python library will extract the features and appropriate emotion of the image. If given image contain more than one

faces also possible to detect. The stress level we are going to indicate by facial expression like sad, angry etc.. The image processing completed the we are going to start the live stream. In the live stream also we can get the facial expression more that one persons also. Compare to tensorlflow live stream the tesnorflow live stream will fast and better results. Once done the we are loading the dataset to perform the knn classification accuracy precession scores.

2. Admin

Admin can login with his credentials. Once he login he can activate the users. The activated user only login in our applications. The admin can set the training and testing

data for the project dynamically to the code. The admin can view all users detected results in hid frame. By clicking an hyperlink in the screen he can detect the emotions of the images. The admin can also view the knn classification detected results. The dataset in the excel format. By authorized persons we can increase the dataset size according the imaginary values.

3. Data Preprocess

Dataset contains grid view of already stored dataset consisting numerous properties, by Property Extraction newly designed dataset appears which contains only numerical input variables as a result of Principal Component Analysis feature selection to 6 principal components which are Condition (No stress, Time pressure, Interruption), Stress, Physical Demand, Performance and Frustration.

4. Machine Learning:

K-Nearest Neighbor (KNN) is used for classification as well as regression analysis. It is a supervised learning algorithm which is used for predicting if a person needs treatment or not. KNN classifies the dependent variable based on how similar it is; independent variables are to a similar instance from the already known data. the Knn Classification can be called as a statistical model that uses a binary dependent variable. In classification analysis, KNN is estimating the parameters of a KNN model.

Mathematically, a binary KNN model has a dependent variable with two possible value, which is represented by an indicator variable, where the two values are labeled "0" and "1".

4.4 REQUIREMENT ANALYSIS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

4.4.1 REQUIREMENT SPECIFICATION

Functional Requirements

- Graphical User interface with the User.

Software Requirements

For developing the application the following are the Software Requirements:

1. Python
2. Django

Operating Systems supported

1. Windows 10 64 bit OS

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application the following are the Hardware Requirements:

- Processor : Intel i3
- RAM : 4 GB
- Space on Hard Disk : Minimum 1 TB

4.5 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL 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.

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.

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 the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

CHAPTER – 5

SOFTWARE DESIGN

5.1 SOFTWARE DESIGN

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementation the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

Architecture Diagram

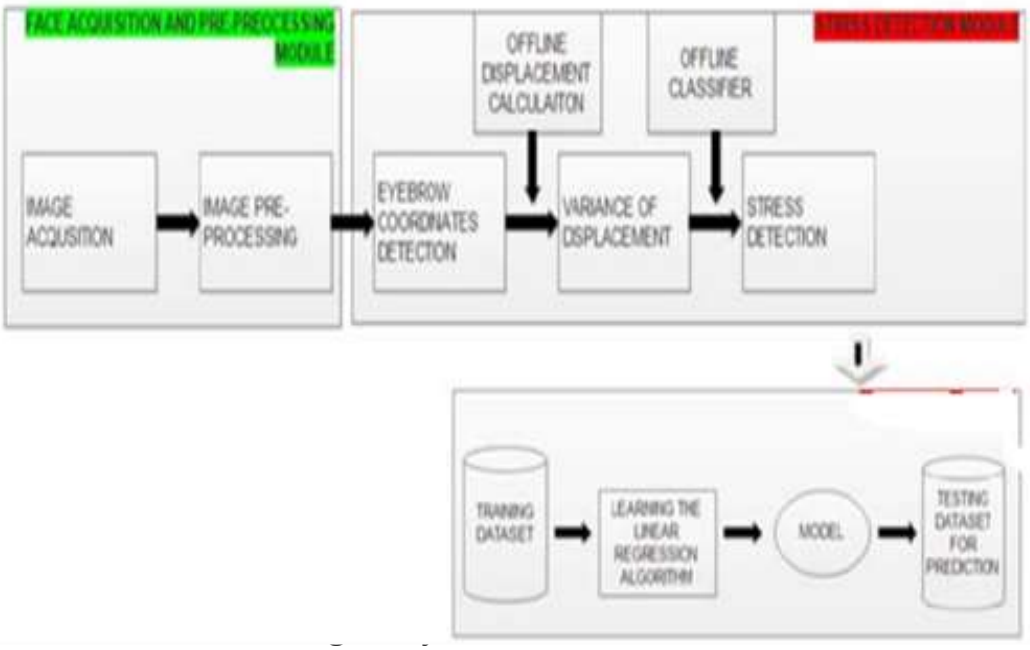


Fig:5.1.1 SYSTEM ARCHITECTURE

5.2 INPUT AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things**:**

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that

the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

5.3 DATAFLOW DIAGRAM

A dataflow diagram illustrates the flow of data between different modules and external entities within the software system. It highlights the interactions and information exchange during encryption and decryption process. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and output data is generated by this system.

1. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

2. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

3. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

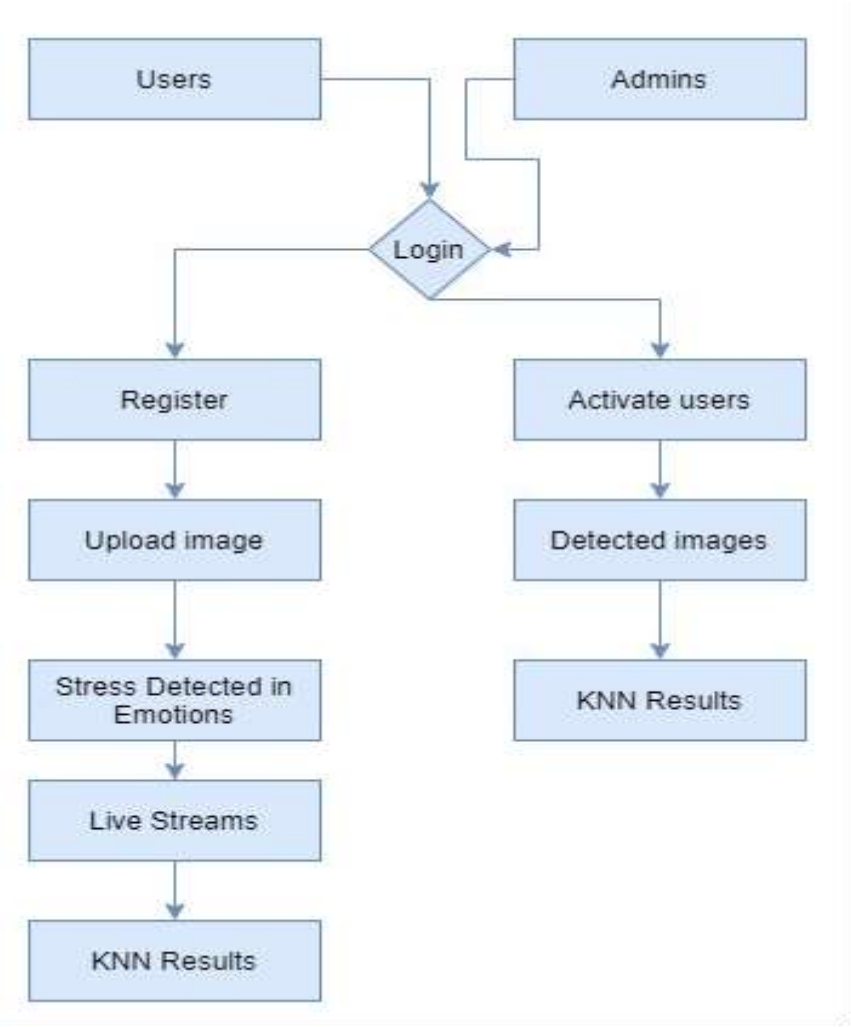


Fig:5.3.1 DATAFLOW DIAGRAM

5.4 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized generalpurpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS

The Primary goals in the design of the UML are as follows

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understandingthe modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations frameworks, patterns and components.
7. Integrate best practices.

5.4.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling 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. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

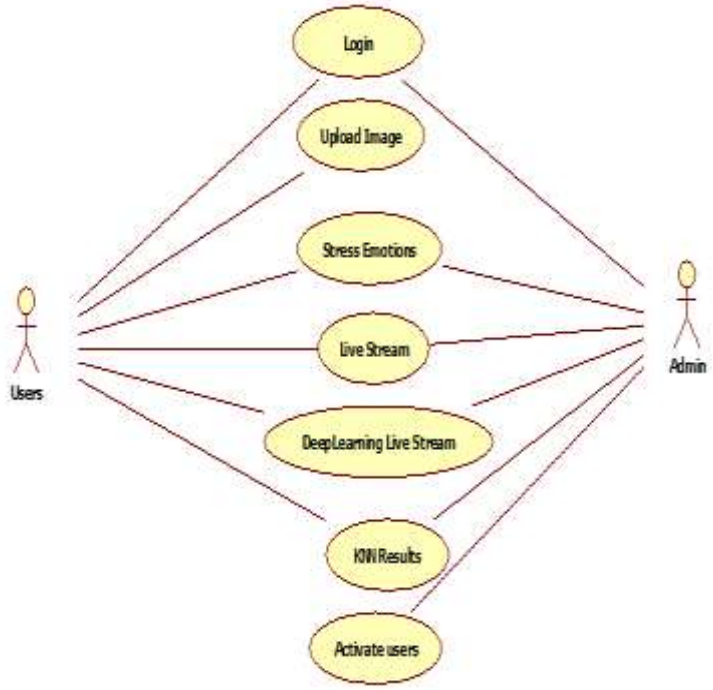


Fig:5.4.1.1 USE CASE DIAGRAM

5.4.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

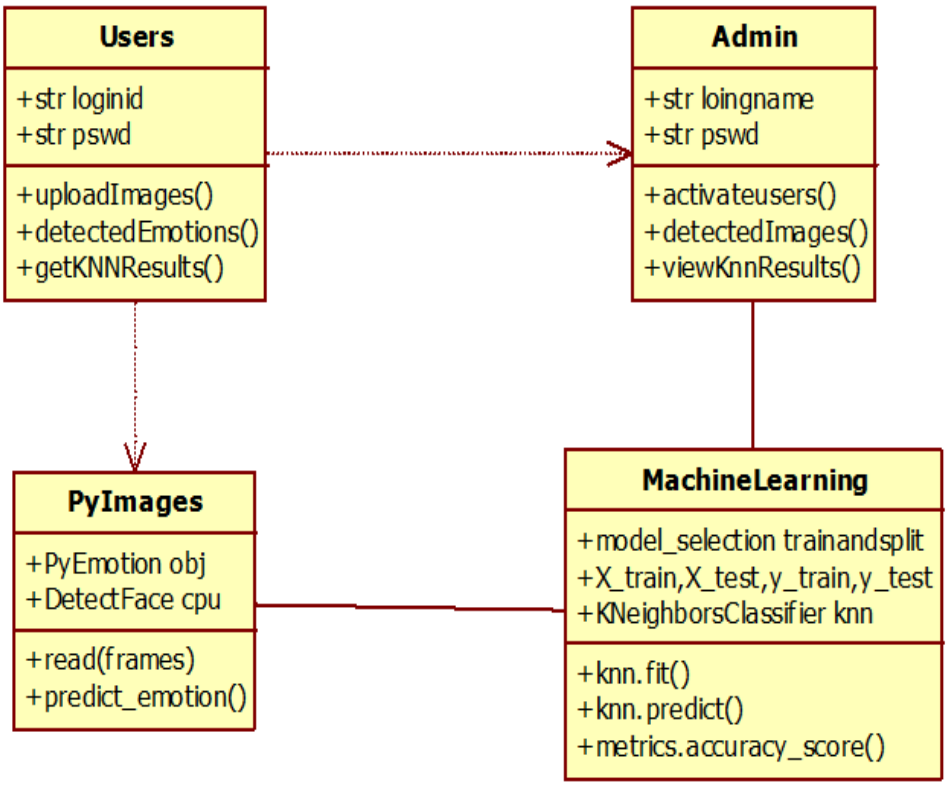


Fig:5.4.2.1 CLASS DIAGRAM

5.4.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a

construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

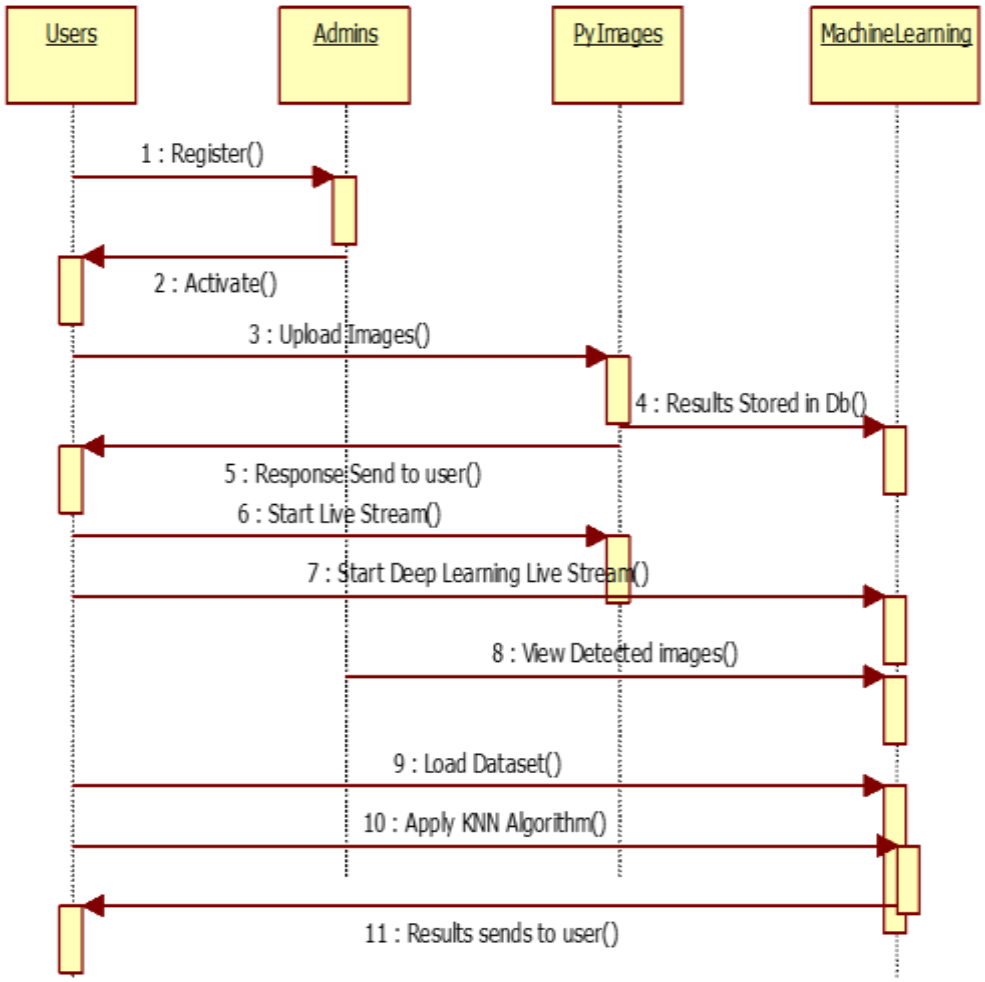


Fig:5.4.3.1 SEQUENCE DIAGRAM

5.4.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling 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.

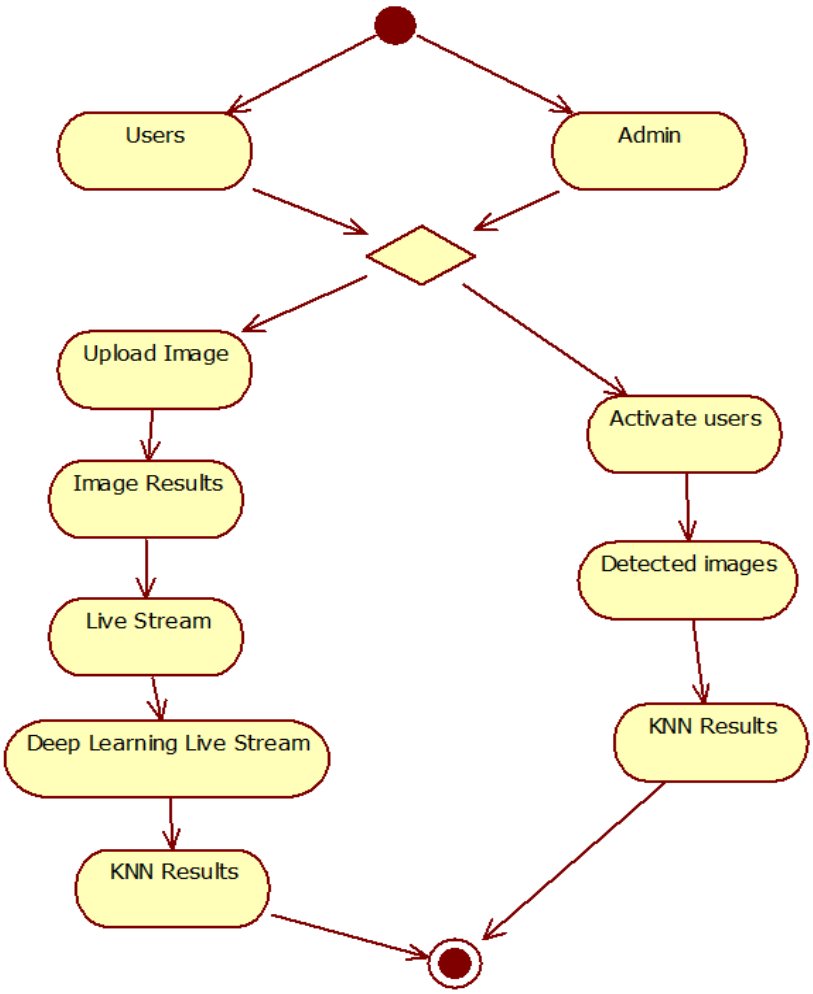


Fig:5.4.4.1 ACTIVITY DIAGRAM

CHAPTER – 6

CODING

User Side views.py

```
from django.shortcuts import render, HttpResponse
from .forms import UserRegistrationForm
from .models import UserRegistrationModel,UserImagePredictinModel
from django.contrib import messages
from django.core.files.storage import FileSystemStorage
from .utility.GetImageStressDetection import ImageExpressionDetect
from .utility.MyClassifier import KNNclassifier
from subprocess import Popen, PIPE
import subprocess

def UserRegisterActions(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            print('Data is Valid')
            form.save()
            messages.success(request, 'You have been successfully registered')
            form = UserRegistrationForm()
            return render(request, 'UserRegistrations.html', {'form': form})
        else:
            messages.success(request, 'Email or Mobile Already Existed')
            print("Invalid form")
    else:
        form = UserRegistrationForm()
    return render(request, 'UserRegistrations.html', {'form': form})

def UserLoginCheck(request):
    if request.method == "POST":
```

```
  loginid = request.POST.get('loginname')
        pswd = request.POST.get('pswd')
        print("Login ID = ", loginid, ' Password = ', pswd)
        try:
            check = UserRegistrationModel.objects.get(loginid=loginid, password=pswd)
            status = check.status
            print('Status is = ', status
if status == "activated":
 request.session['id'] = check.id

request.session['loggeduser'] = check.name
                request.session['loginid'] = loginid
                request.session['email'] = check.email
                print("User id At", check.id, status)
                return render(request, 'users/UserHome.html', {})
            else:
                messages.success(request, 'Your Account Not at activated')
                return render(request, 'UserLogin.html')
        'Interruption', 'Foot GSR(mV)': 'Stress', 'Hand GSR(mV)': 'Physical Demand',
'HR(bpm)': 'Performance', 'RESP(mV)': 'Frustration', }, inplace=True)
    data = df.to_html()
    return
render(request,'users/UserKnnResults.html',{'data':data,'accuracy':accuracy,'classificatio
nerror':classificationerror,
'sensitivity':sensitivity,"Specificity":Specificity,'fsp':fsp,'precision':precision})
```

user side forms.py

```
from django import forms
from .models import UserRegistrationModel

class UserRegistrationForm(forms.ModelForm):
    name = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-zA-Z]+'}),
required=True, max_length=100)
    loginid = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-zA-Z]+'}),
```

```
required=True, max_length=100)
   password = forms.CharField(widget=forms.PasswordInput(attrs={'pattern':
'(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}',
                                                'title': 'Must contain at least one number and
one uppercase and lowercase letter, and at least 8 or more characters'}),
                          required=True, max_length=100)
   mobile = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[56789][0-
9]{9}'}), required=True,
                         max_length=100)
   email = forms.CharField(widget=forms.TextInput(attrs={'pattern': '[a-z0-9._%+-
]+@[a-z0-9.-]+\.[a-z]{2,}$'}),
                         required=True, max_length=100)
   locality = forms.CharField(widget=forms.TextInput(), required=True,
max_length=100)
   address = forms.CharField(widget=forms.Textarea(attrs={'rows': 4, 'cols': 22}),
attrs={'autocomplete': 'off', 'pattern': '[A-Za-z ]+', 'title': 'Enter Characters Only '}),
required=True,
                         max_length=100)
   status = forms.CharField(widget=forms.HiddenInput(), initial='waiting',
max_length=100)
   class Meta():
      model = UserRegistrationModel
      fields = '__all__'
```

user side Models.py

```
from django.db import models
class UserRegistrationModel(models.Model):
   name = models.CharField(max_length=100)
   loginid = models.CharField(unique=True, max_length=100)
   password = models.CharField(max_length=100)
   mobile = models.CharField(unique=True, max_length=100)
```

```
email = models.CharField(unique=True, max_length=100)
    locality = models.CharField(max_length=100)
    address = models.CharField(max_length=1000)
    city = models.CharField(max_length=100)
    state = models.CharField(max_length=100)
    status = models.CharField(max_length=100)
 def __str__(self):
        return self.loginid
    def __str__(self):
        return self.loginid
    class Meta:
        db_table = "UserImageEmotions"
```

Image Classification:

```
from django.conf import settings
from PyEmotion import *
import cv2 as cv
class ImageExpressionDetect:
    def getLiveDetect(self):
        print("Streaming Started")
        PyEmotion()
        er = DetectFace(device='cpu', gpu_id=0)


     cap = cv.VideoCapture(0)
        while (True):
            ret, frame = cap.read()
            frame, emotion = er.predict_emotion(frame)
            cv.imshow('Press Q to Exit', frame)
            if cv.waitKey(1) & 0xFF == ord('q'):
                break
```

```
cap.release()
        cv.destroyAllWindows()
```

Deeplearning Model:

```
import numpy as np
import argparse
import cv2
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Flatten
from keras.layers.convolutional import Conv2D
from keras.optimizers import Adam
    axs[1].set_ylabel('Loss')
    axs[1].set_xlabel('Epoch')
axs[1].set_xticks(np.arange(1,len(model_history.history['loss'])+1),len(model_history.hi
story['loss'])/10)
    axs[1].legend(['train', 'val'], loc='best')
    fig.savefig('plot.png')
    plt.show()
train_dir = 'data/train'
val_dir = 'data/test'
num_train = 28709
num_val = 7178
batch_size = 64

num_epoch = 50
train_datagen = ImageDataGenerator(rescale=1./255)

          break

cap.release()
    cv2.destroyAllWindows()
```

Admin side Views.py

```
from django.shortcuts import render
from django.contrib import messages
from users.models import UserRegistrationModel,UserImagePredictinModel
from .utility.AlgorithmExecutions import KNNclassifier
def AdminLoginCheck(request):
    if request.method == 'POST':
        usrid = request.POST.get('loginid')
        pswd = request.POST.get('pswd')
        print("User ID is = ", usrid)
        if usrid == 'admin' and pswd == 'admin':
            return render(request, 'admins/AdminHome.html')

 elif usrid == 'Admin' and pswd == 'Admin':
            return render(request, 'admins/AdminHome.html')
        else:
            messages.success(request, 'Please Check Your Login Details')
    return render(request, 'AdminLogin.html', {})
def AdminHome(request):
    return render(request, 'admins/AdminHome.html')
def ViewRegisteredUsers(request):
    data = UserRegistrationModel.objects.all()
    obj = KNNclassifier()


    df, accuracy, classificationerror, sensitivity, Specificity, fsp, precision =
obj.getKnnResults()
    df.rename(
        columns={'Target': 'Target', 'ECG(mV)': 'Time pressure', 'EMG(mV)':
'Interruption', 'Foot GSR(mV)': 'Stress',
                 'Hand GSR(mV)': 'Physical Demand', 'HR(bpm)': 'Performance', 'RESP(mV)':
'Frustration', },
```

All urls.py

```
from django.contrib import admin
from django.urls import path
from StressDetection import views as mainView
from users import views as usr
from admins import views as admins
from django.contrib.staticfiles.urls import static

    path('admin/', admin.site.urls),
    path("", mainView.index, name="index"),
    path("index/", mainView.index, name="index"),
    path("logout/", mainView.logout, name="logout"),
    path("UserLogin/", mainView.UserLogin, name="UserLogin"),
    path("AdminLogin/", mainView.AdminLogin, name="AdminLogin"),
    path("UserRegister/", mainView.UserRegister, name="UserRegister")
name="AdminStressDetected"),
    path("AdminKNNResults/", admins.AdminKNNResults,
name="AdminKNNResults"),
]
urlpatterns += staticfiles_urlpatterns()
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Base.html

```
<!DOCTYPE html>
{%load static%}
<html lang="en">
<head>
<title>Stress Feelings</title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="description" content="Unicat project">

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href="{%static
'styles/bootstrap4/bootstrap.min.css'%}">


 </div>

<script src="{%static 'styles/bootstrap4/bootstrap.min.js'%}"></script>
<script src="{%static 'plugins/greensock/TweenMax.min.js'%}"></script>
<script src="{%static 'plugins/greensock/TimelineMax.min.js'%}"></script>
script>
<script src="{%static 'js/custom.js'%}"></script>
</body>
</html>
```

Index.html

```
{%extends 'base.html'%}
{%load static%}
{%block contents%}
<div class="home">
    <div class="home_slider_container">


   <div class="home_slider_background" style="background-image:url({%static
'images/home_slider_1.jpg'%})"></div>


                    <div class="col text-center">
                <div class="home_slider_title">Stress Detection in IT Professionals
```

```
</div>
<div class="home_slider_subtitle">by Image Processing and Machine Learning</div>
  </div>
{%endblock%}
```

User login .html

```
{%extends 'base.html'%}
{%load static%}
{%block contents%}
<div class="home">
    <div class="home_slider_container">
      <div class="owl-carousel owl-theme home_slider">
        <div class="owl-item">
 <div class="home_slider_background" style="background-image:url({%static
'images/home_slider_1.jpg'%})"></div>
          <div class="home_slider_content">
            <div class="container">
              <div class="row">
                <div class="col text-center">
                  <div class="home_slider_title">User Login Form </div>
                  <div class="home_slider_subtitle"></div>
<div class="home_slider_form_container">
                        <p>
                        <center>
                      <form action="{%url 'UserLoginCheck'%}" method="POST"
class="text-primary" style="width:100%">
              {% csrf_token %}      </div>
  </div>
{%endblock%}
```

KNN Results.html

```
{%extends 'users/userbase.html'%}
{%load static%}

{%block contents%}
<div class="features">
    <div class="container">
      <div class="row">
        <div class="col">
          <div class="section_title_container text-center">
            <h2 class="section_title">Knn Algorithm Results</h2>
            <h3>Accuarcy <font color="Green">{{accuracy}}</font></h3> <br/>
            <h3>Classification Error <font
color="Green">{{classificationerror}}</font></h3>
            <h3>Sensitivity <font color="Green">{{sensitivity}}</font></h3>
            <h3>Specificity <font color="Green">{{Specificity}}</font></h3>
            <h3>False positive rate Error <font color="Green">{{fsp}}</font></h3>
            <h3>Precision <font color="Green">{{precision}}</font></h3>
```

CHAPTER - 7

\

SYSTEM TESTING

7.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

7.2 TESTING METHODOLOGIES

The following are the Testing Methodologies

- Unit Testing.
- Integration Testing.
- Functional Testing.
- System Testing.
- Acceptance Testing.

7.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests

ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.3 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.

7.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

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

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2.5 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully.

TEST CASES:

S.no	Test Case	Excepted Result	Result	Remarks(IF Fails)
1.	User Register	If User registration successfully.	Pass	If already user email exist then it fails.
2.	User Login	If User name and password is correct then it will getting valid page.	Pass	Un Register Users will not logged in.
3.	Upload An Image	Image uploaded to server and strating process to detetct	Pass	Image must be 640X480 resolution will get better results
4.	Draw Squares in images	Detected images draw square and writing stress emotions	Pass	Images must be clearly to detect facial expression
5.	Start live Stream	PyImage libaray will load the process and start the live	Pass	If library not available then failed
6.	Start Deep learning live stream	If tensorflow not installed then it will fail	Pass	Depends on system configuration and tensorflow library
7.	Knn Results	Load the dataset and process the KNN Algorithm	Pass	The dataset must be media folder
8.	Predict Train and Test data	Predicted and original salary will be displayed	Pass	Trains and test size must be specify otherwise failed
9.	Admin login	Admin can login with his login credential. If success he get his home page	Pass	Invalid login details will not allowed here
10.	Admin can activate the register users	Admin can activate the register user id	Pass	If user id not found then it won’t login.

CHAPTER - 8

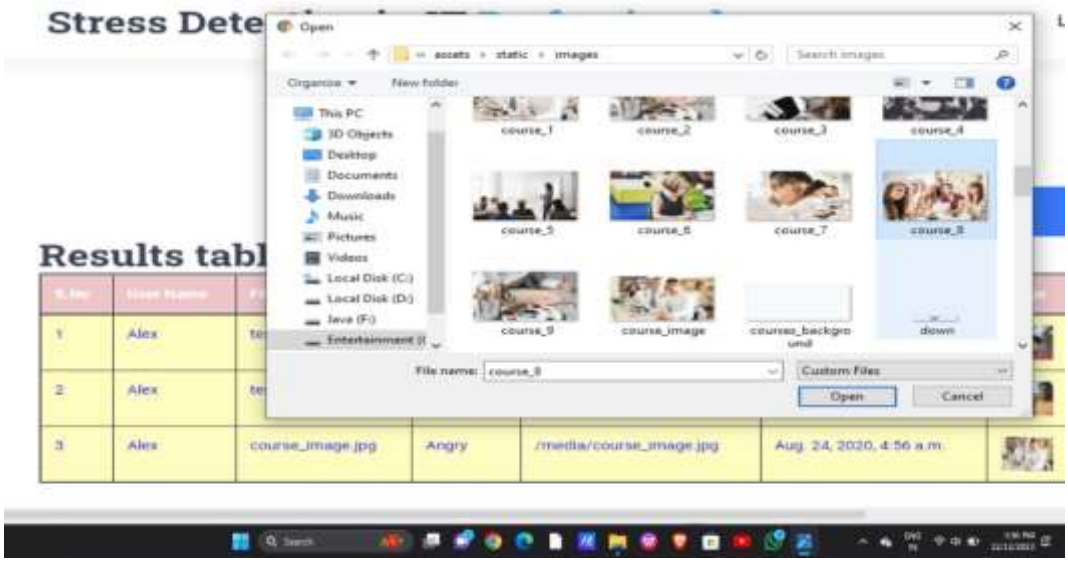
OUTPUT SCREENS

OUTPUT SCREEN: 8.1 HOMEPAGE

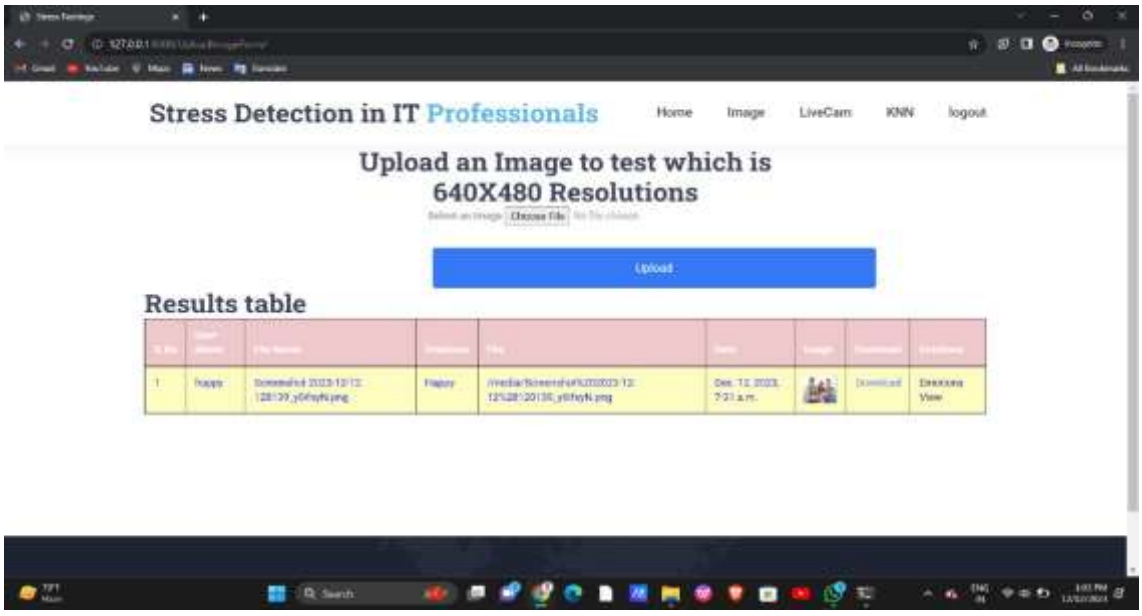
OUTPUT SCREEN:8.2 USER REGISTER PAGE

OUTPUT SCREEN:8.3 USER LOGIN FORM

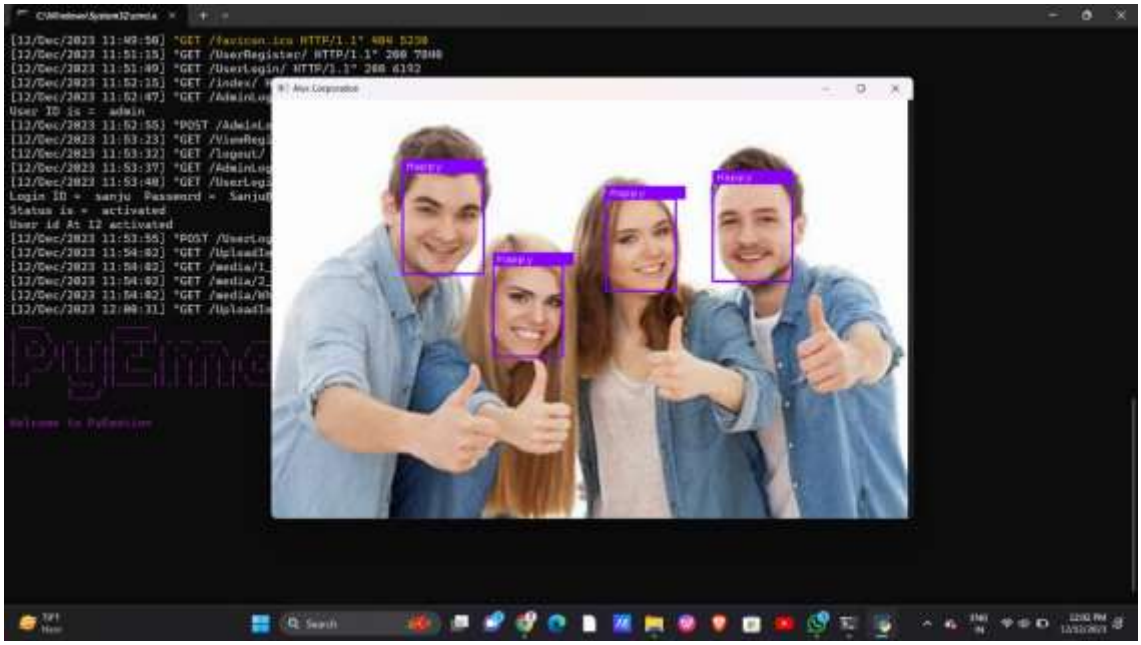
OUTPUT SCREEN:8.4 USER HOME PAGE



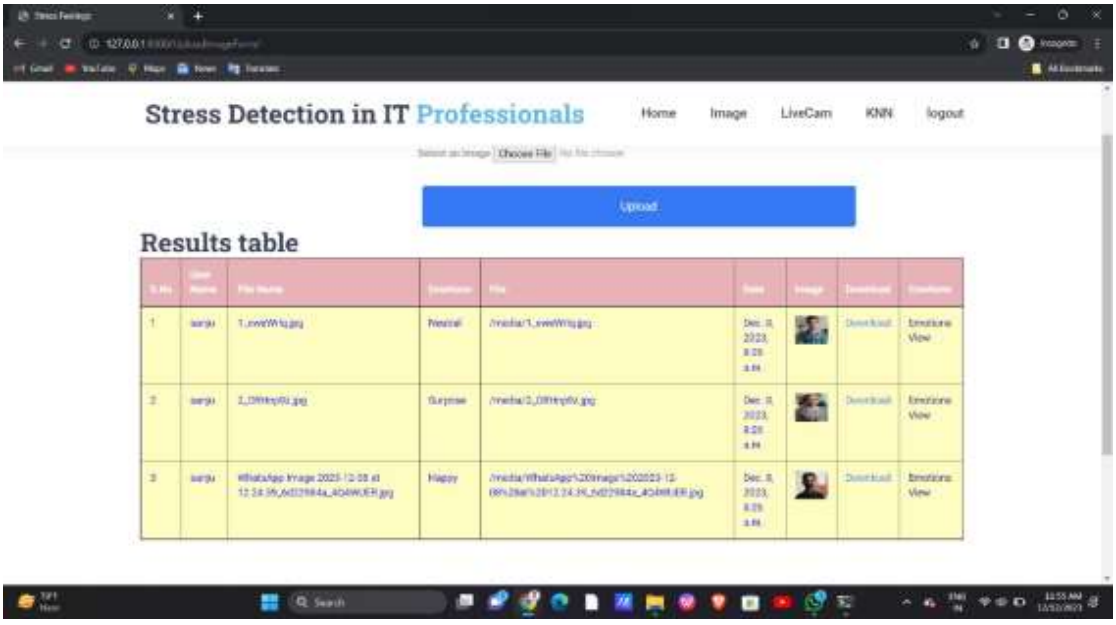
OUTPUT SCREEN: 8.5 GIVING IMAGE AS INPUT



OUTPUT SCREEN: 8.6 UPLOAD IMAGE



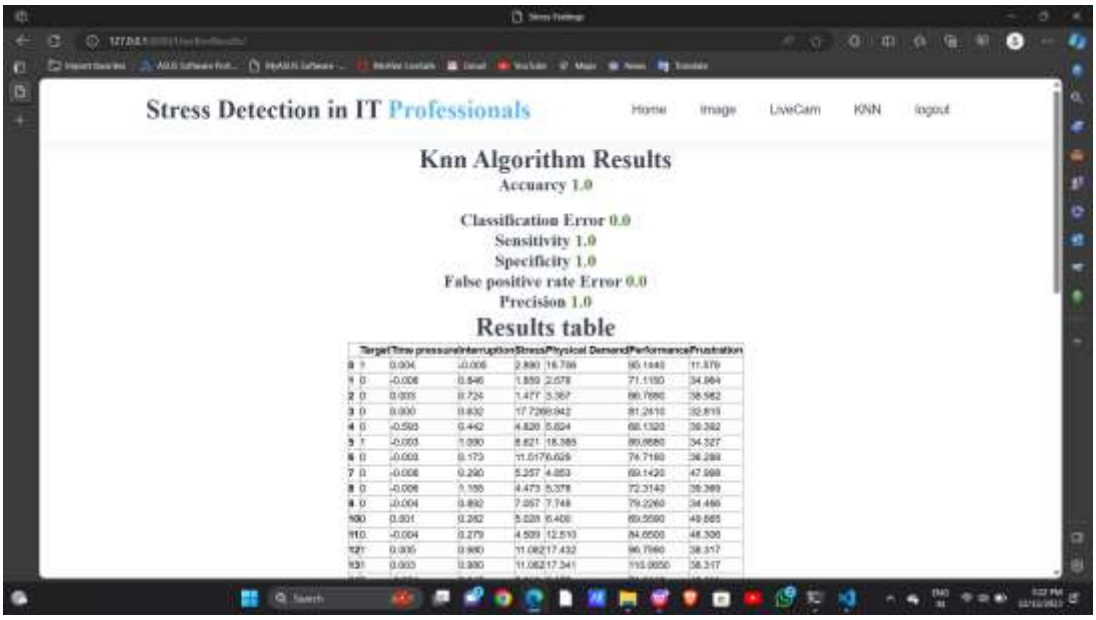
OUTPUT SCREEN:8.7 RESPONSE IMAGE



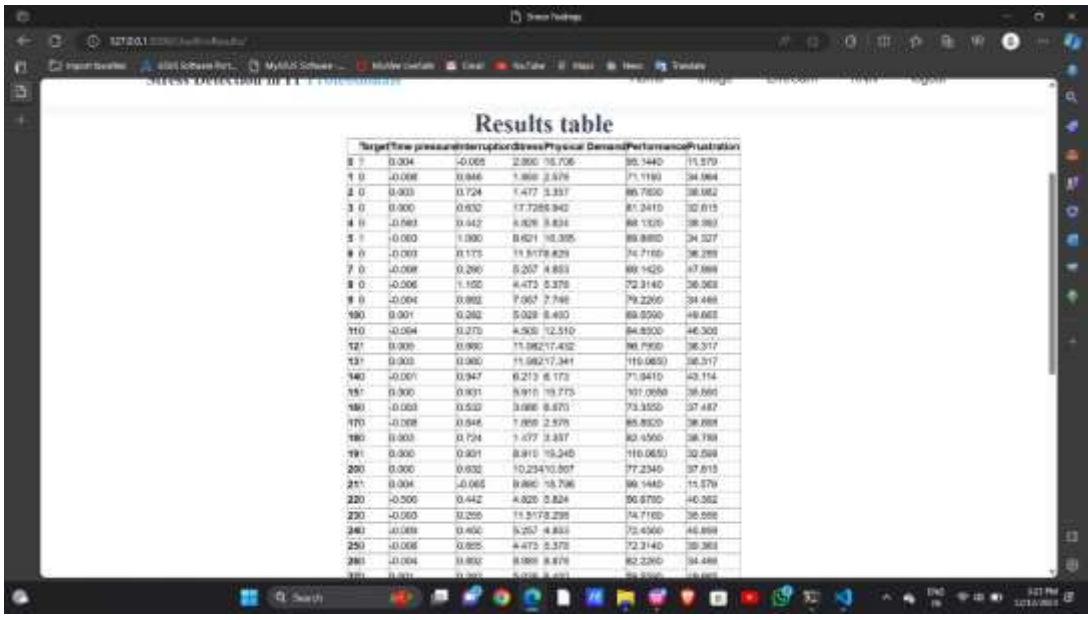
OUTPUT SCREEN:8.8 RESULTS

OUTPUT SCREEN:8.9 LIVE STREAM

OUTPUT SCREEN: 8.10 DEEP LEARNING PAGE



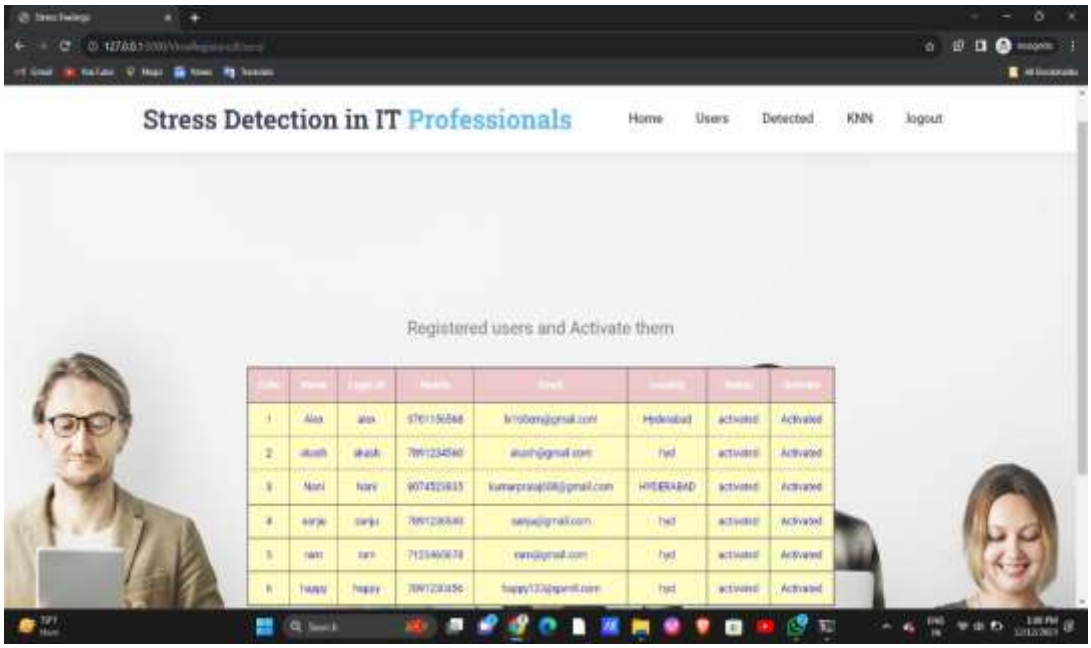
OUTPUT SCREEN:8.11 KNN RESULTS



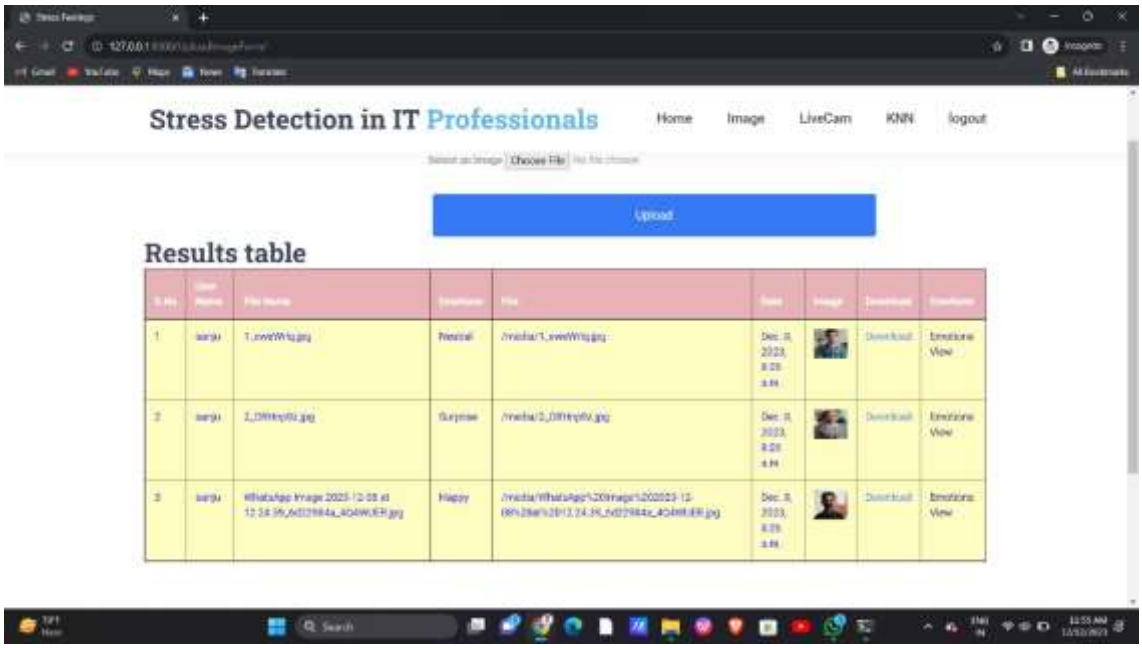
OUTPUT SCREEN:8.12 DATASET VIEW

OUTPUT SCREEN:8.13 ADMIN PAGE

OUTPUT SCREEN: 8.14 ADMIN HOMEPAGE

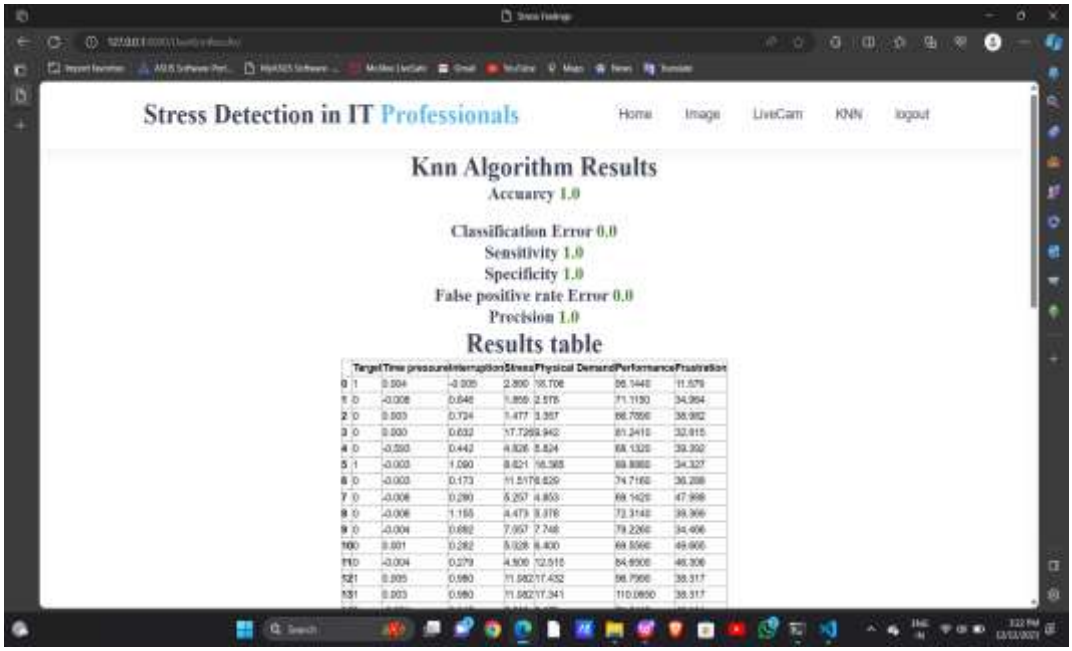


OUTPUT SCREEN:8.15 ACTIVATE USERS



OUTPUT SCREEN:8.16 DETECTED IMAGES

OUTPUT SCREEN:8.17 ADMIN SIDE RESULTS



OUTPUT SCREEN:8.18 ADMIN VIEW RESULTS

CHAPTER – 9

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 analyze the stress levels by using Machine Learning algorithms which generates the results that are more efficient.

CHAPTER – 10

FUTURE ENHANCEMENTS

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 caretaker 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.

CHAPTER – 11

BIBLIOGRAPHY

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

[2] T. Jick and R. Payne, “Stress at work,” Journal of Management Education, vol. 5, no. 3, pp. 50-56, 1980.

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

[4] Bhattacharyya, R., & Basu, S. (2018). Retrieved from ‘The Economic Times’.

[5] OSMI Mental Health in Tech Survey Dataset, 2017

[6] U. S. Reddy, A. V. Thota and A. Dharun, "Machine Learning Techniques for Stress Prediction in Working Employees," 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, 2018, pp. 1-4.

[7] https://www.kaggle.com/qiriro/stress

[8] Communications, N.. World health report. 2001. URL: http://www.who.int/whr/2001/media_centre/press_release/en/.

[9] Bakker, J., Holenderski, L., Kocielnik, R., Pechenizkiy, M., Sidorova, N.. Stess@ work: From measuring stress to its understanding, prediction and handling with personalized coaching. In: Proceedings of the 2nd ACM SIGHIT International health informatics symposium. ACM; 2012, p. 673–678.

[10] Deng, Y., Wu, Z., Chu, C.H., Zhang, Q., Hsu, D.F.. Sensor feature selection and combination for stress identification using combinatorial fusion. International Journal of Advanced Robotic Systems 2013;10(8):306.