A

Major Project

On

AN ENHANCED SYSTEM FOR DETECTING STRESS IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE LEARNING

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

A.V.Prathyusha(187R1A0501)

Saikat Kumar Kundu (187R1A0545)

D.Adarsh (187R1A0516)

Under the Guidance of

Dr. K.MAHESWARI

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)
Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**AN ENHANCED SYSTEM FOR DETECTING STRESS IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE LEARNING**" being submitted by **A.V.PRATHYUSHA (187R1A0501), SAIKAT KUMAR KUNDU (187R1A0545) & D.ADARSH (187R1A0516)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr.K.Maheswari
(Associate Professor)
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on _____

ACKNOWLEGDEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to our guide **Dr.K.MAHESWARI,** Associate Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Mr. A. Uday kiran , Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao , Mrs. G. Latha , Mr. A. Kiran Kumar ,** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

A.V.Prathyusha (187R1A0501)

D.Adarsh (187R1A0516)

Saikat Kumar Kundu (187R1A0545)

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 the personal counseling but this system comprises of 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 best out of them during working hours.

LIST OF FIGURES/TABLES

LIST OF SCREENSHOTS

TABLE OF CONTENT

1. INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as “STRESS DETECTION IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE PROCESSING”. 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. 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.

1.2 PROJECT PURPOSE

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. This method completely depends on the answers given by the individuals, people will be tremulous to say whether they are stressed or normal.

1.3 PROJECT FEATURES

Automated stress detection is made possible by several pattern recognition algorithms. 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.

2. SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

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. 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)

2.2 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.

2.2.1 ADVANTAGES OF THE 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)

2.3 FEASIBILITY STUDY

A feasibility study is an analysis that considers all of a project's relevant factors—including economic, technical, and social considerations—to ascertain the likelihood of completing the project successfully .Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.3.1 ECONOMIC FEASIBILITY

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success.

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

2.3.2 TECHNICAL FEASIBILITY

Technical feasibility is a set of techniques aimed at forecasting future prices of securities, currencies or raw materials based on the analysis of price formation in the past. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.3.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.4 HARDWARE & SOFTWARE REQUIREMENTS

2.4.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- Processor : Intel Dual Core i3 7th Gen
- HardDisk : 1TB
- RAM : 8GB or Above.
- Input Devices : Keyboard,mouse.

2.4.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system : Windows 10 64-bit OS
- Languages : Python, Django
- Backend : Machine Learning

3. ARCHITECTURE

3. ARCHITECTURE

3.1 PROJECT ARCITECTURE

This project architecture shows the procedure followed for breed detection using machine learning, starting from input to final prediction.

A system architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. A system architecture can consist of system components and the sub-systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture , collectively these are called architecture description languages (ADLs).

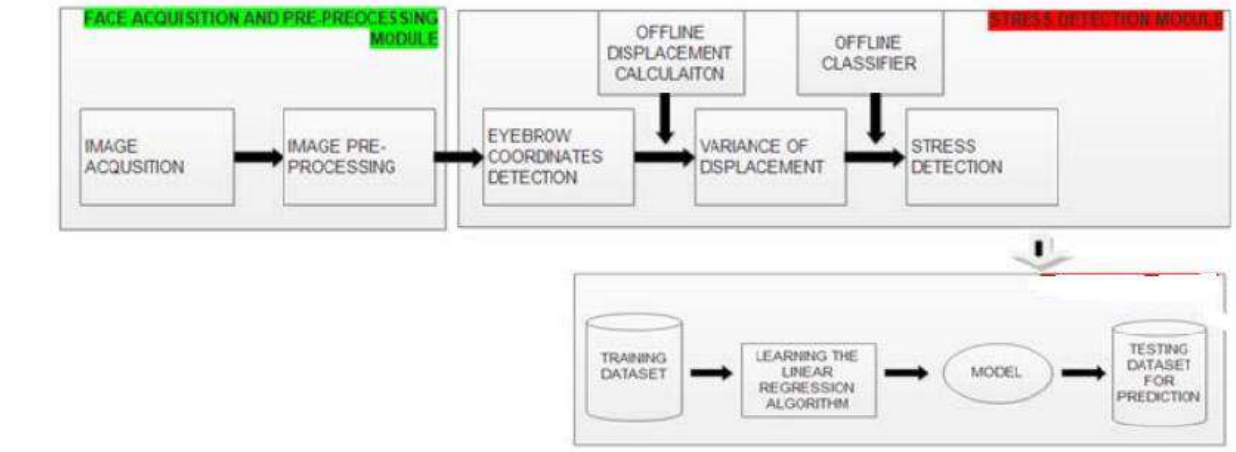


Fig 3.1: Project Architecture

3.2 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.

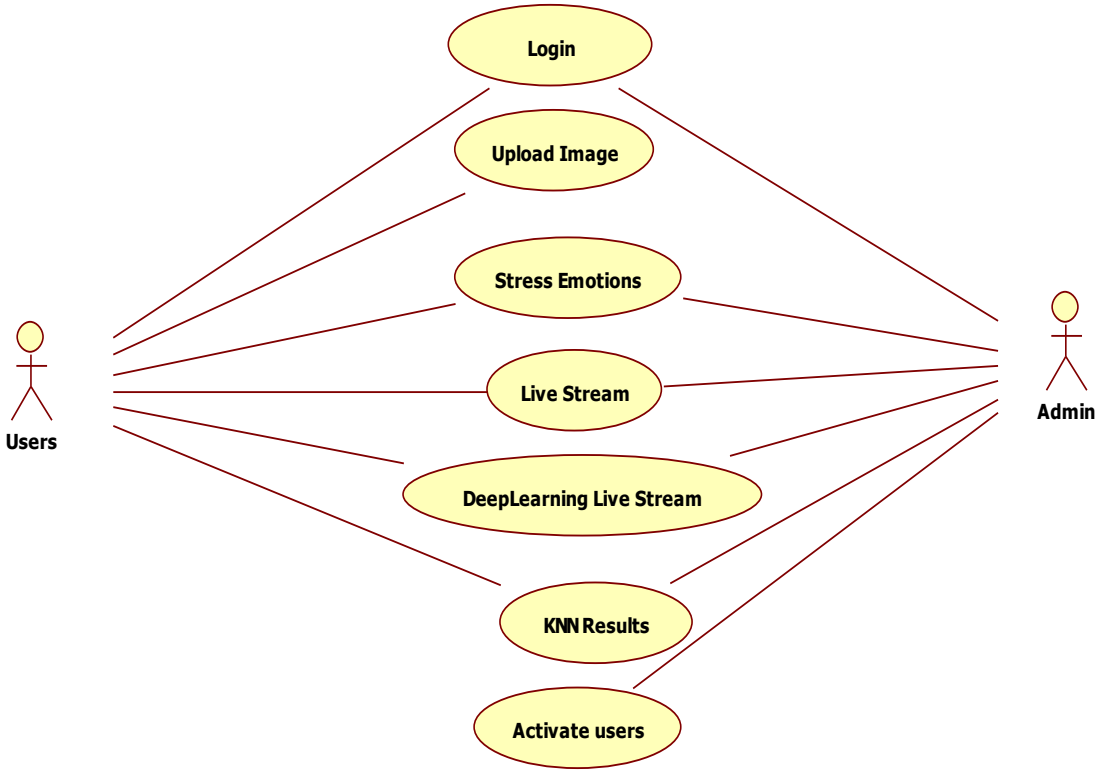


Figure 3.2: Use Case Diagram

3.3 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.

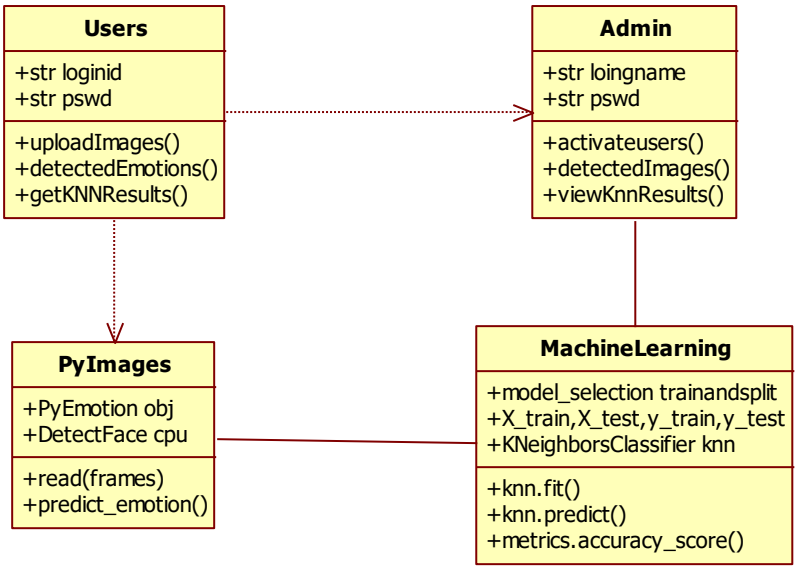


Figure 3.3: Class Diagram

3.4 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.

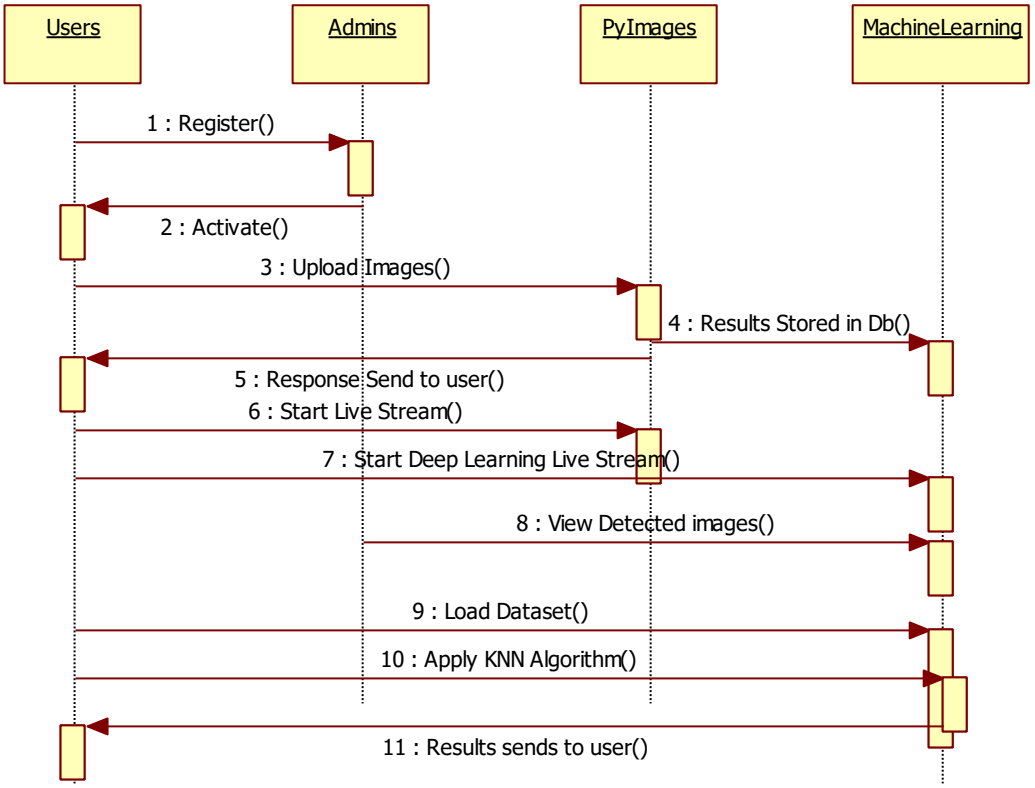


Figure 3.4: Sequence Diagram

3.5 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.

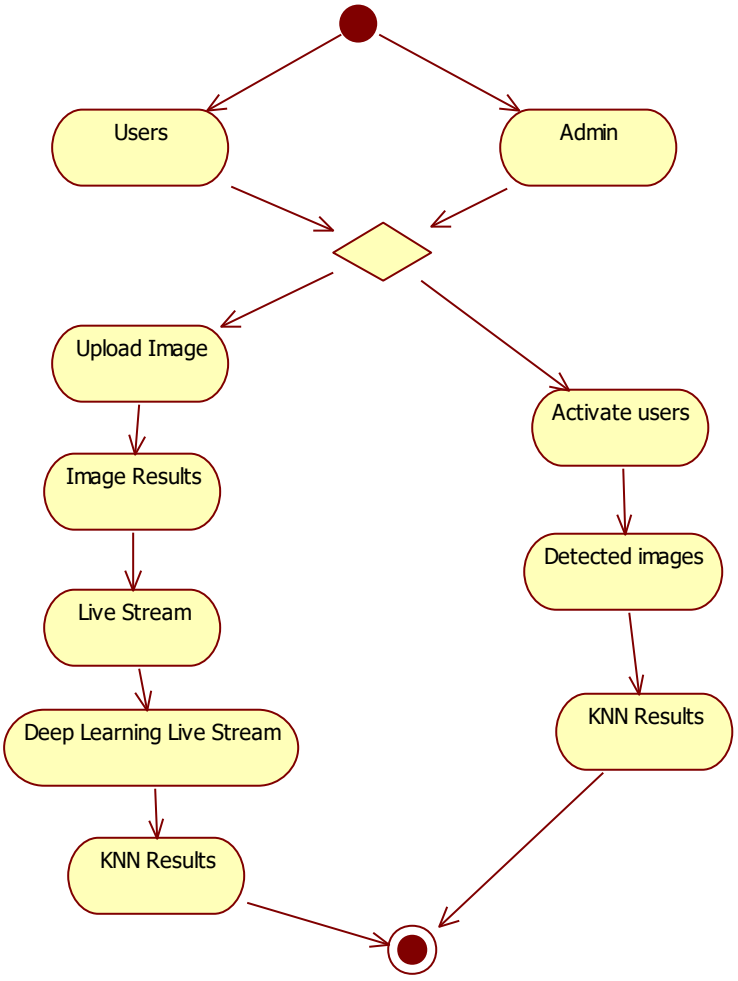


Figure 3.5: Activity Diagram

3.6 DATA FLOW DIAGRAM

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 the output data is generated by this system. 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. 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. 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.

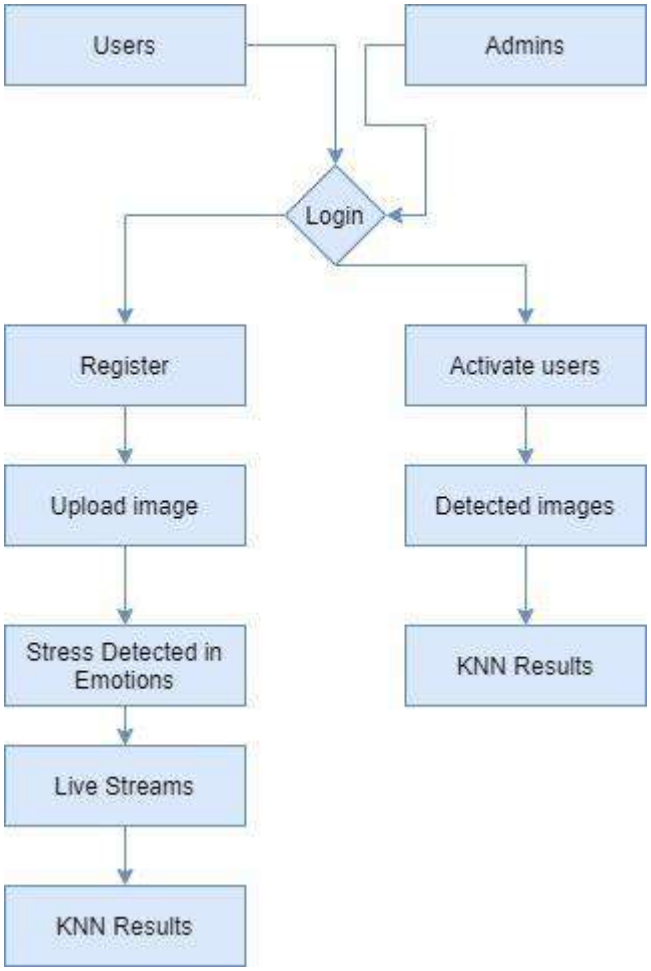


Figure 3.6: Data Flow Diagram

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

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')
            except Exception as e:
             print('Exception is ', str(e))
            pass
        messages.success(request, 'Invalid Login id and password')
    return render(request, 'UserLogin.html', {})
```

```
    def UserHome(request):
        return render(request, 'users/UserHome.html', {})

    def UploadImageForm(request):
        loginid = request.session['loginid']
        data = UserImagePredictinModel.objects.filter(loginid=loginid)
        return render(request, 'users/UserImageUploadForm.html', {'data': data})

    def UploadImageAction(request):
        image_file = request.FILES['file']


        if not image_file.name.endswith('.jpg'):
            messages.error(request, 'THIS IS NOT A JPG  FILE')

        fs = FileSystemStorage()
        filename = fs.save(image_file.name, image_file)

        uploaded_file_url = fs.url(filename)
        obj = ImageExpressionDetect()
        emotion = obj.getExpression(filename)
        username = request.session['loggeduser']
        loginid = request.session['loginid']
        email = request.session['email']

UserImagePredictinModel.objects.create(username=username,email=email,loginid=loginid,filename=filenam
e,emotions=emotion,file=uploaded_file_url)
        data = UserImagePredictinModel.objects.filter(loginid=loginid)
        return render(request, 'users/UserImageUploadForm.html', {'data':data})

    def UserEmotionsDetect(request):
        if request.method=='GET':
            imgname = request.GET.get('imgname')
            obj = ImageExpressionDetect()
            emotion = obj.getExpression(imgname)
            loginid = request.session['loginid']
            data = UserImagePredictinModel.objects.filter(loginid=loginid)
            return render(request, 'users/UserImageUploadForm.html', {'data': data})


    def UserLiveCameDetect(request):
        obj = ImageExpressionDetect()
        obj.getLiveDetect()
        return render(request, 'users/UserLiveHome.html', {})

    def UserKerasModel(request):

        subprocess.call("python kerasmodel.py --mode display")
        return render(request, 'users/UserLiveHome.html', {})
```

```
def UserKnnResults(request):


    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', }, inplace=True)
    data = df.to_html()
    return render(request,'users/UserKnnResults.html',{'data':data,'accuracy':accuracy,'classificationerror':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}), required=True, max_length=250)
    city = forms.CharField(widget=forms.TextInput(
        attrs={'autocomplete': 'off', 'pattern': '[A-Za-z ]+', 'title': 'Enter Characters Only '}), required=True,
                           max_length=100)
    state = forms.CharField(widget=forms.TextInput(
        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

    class Meta:
        db_table = 'UserRegistrations'
class UserImagePredictinModel(models.Model):
    username = models.CharField(max_length=100)
    email = models.CharField(max_length=100)
    loginid = models.CharField(max_length=100)
    filename = models.CharField(max_length=100)
    emotions = models.CharField(max_length=100000)
    file = models.FileField(upload_to='files/')
    cdate = models.DateTimeField(auto_now_add=True)

    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 getExpression(self,imagepath):
        filepath = settings.MEDIA_ROOT + "\\" + imagepath
```

```
        PyEmotion()
        er = DetectFace(device='cpu', gpu_id=0)


        frame, emotion = er.predict_emotion(cv.imread(filepath))
        cv.imshow('Alex Corporation', frame)
        cv.waitKey(0)
        print("Hola Hi",filepath,"Emotion is ",emotion)
        return emotion



    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
rom keras.layers.core import Dense, Dropout, Flatten
from keras.layers.convolutional import Conv2D
from keras.optimizers import Adam
from keras.layers.pooling import MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator
import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
import matplotlib as mpl
mpl.use('TkAgg')
import matplotlib.pyplot as plt


ap = argparse.ArgumentParser()
ap.add_argument("--mode",help="train/display")
a = ap.parse_args()
mode = a.mode
```

```
def plot_model_history(model_history):

    fig, axs = plt.subplots(1,2,figsize=(15,5))
    axs[0].plot(range(1,len(model_history.history['acc'])+1),model_history.history['acc'])
    axs[0].plot(range(1,len(model_history.history['val_acc'])+1),model_history.history['val_acc'])
    axs[0].set_title('Model Accuracy')
    axs[0].set_ylabel('Accuracy')
    axs[0].set_xlabel('Epoch')

axs[0].set_xticks(np.arange(1,len(model_history.history['acc'])+1),len(model_history.history['acc'])/
10)
    axs[0].legend(['train', 'val'], loc='best')
    # summarize history for loss
      axs[1].plot(range(1,len(model_history.history['loss'])+1),model_history.history['loss'])
    axs[1].plot(range(1,len(model_history.history['val_loss'])+1),model_history.history['val_loss'])
    axs[1].set_title('Model Loss')
    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.history['loss'])
/10)
    axs[1].legend(['train', 'val'], loc='best')
    fig.savefig('plot.png')
    plt.show()

# Define data generators
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)
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(48,48),
        batch_size=batch_size,
        color_mode="grayscale",
        class_mode='categorical')
        validation_generator = val_datagen.flow_from_directory(
        val_dir,
        target_size=(48,48),
        batch_size=batch_size,
        color_mode="grayscale",
        class_mode='categorical')
```

AN ENHANCED SYSTEM FOR DETECTING STRESS IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE LEARNING

```
# Create the model
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))

# If you want to train the same model or try other models, go for this
if mode == "train":
    model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-
6),metrics=['accuracy'])

    model_info = model.fit_generator(
            train_generator,
            steps_per_epoch=num_train // batch_size,
            epochs=num_epoch,
            validation_data=validation_generator,
            validation_steps=num_val // batch_size)

    plot_model_history(model_info)
    model.save_weights('model.h5')


elif mode == "display":
    model.load_weights('model.h5')
    cv2.ocl.setUseOpenCL(False)
    emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6:
"Surprised"}

    # start the webcam feed
    cap = cv2.VideoCapture(0)
    while True:
        # Find haar cascade to draw bounding box around face
        ret, frame = cap.read()
        facecasc = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = facecasc.detectMultiScale(gray,scaleFactor=1.3, minNeighbors=5)
```

```
        for (x, y, w, h) in faces:
            cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 0), 2)
                roi_gray = gray[y:y + h, x:x + w]
            cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray, (48, 48)), -1), 0)
            prediction = model.predict(cropped_img)
            maxindex = int(np.argmax(prediction))
            cv2.putText(frame, emotion_dict[maxindex], (x+20, y-60),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)



        # if the `q` key was pressed, break from the loop
        if key == ord("q"):
            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

# Create your views here.

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()
    return render(request, 'admins/RegisteredUsers.html', {'data': data})


def AdminActivaUsers(request):
    if request.method == 'GET':
        id = request.GET.get('uid')
        status = 'activated'
        print("PID = ", id, status)
        UserRegistrationModel.objects.filter(id=id).update(status=status)
        data = UserRegistrationModel.objects.all()
        return render(request, 'admins/RegisteredUsers.html', {'data': data})
        def AdminStressDetected(request):
    data = UserImagePredictinModel.objects.all()
    return render(request, 'admins/AllUsersStressView.html', {'data': data})

    def AdminKNNResults(request):
        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',


    'RESP(mV)': 'Frustration', },

            inplace=True)
        data = df.to_html()
        return render(request, 'admins/AdminKnnResults.html',
                      {'data': data, 'accuracy': accuracy, 'classificationerror': classificationerror,
                       'sensitivity': sensitivity, "Specificity": Specificity, 'fsp': fsp, 'precision': precision})
```

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
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from django.conf import settings

urlpatterns = [
```

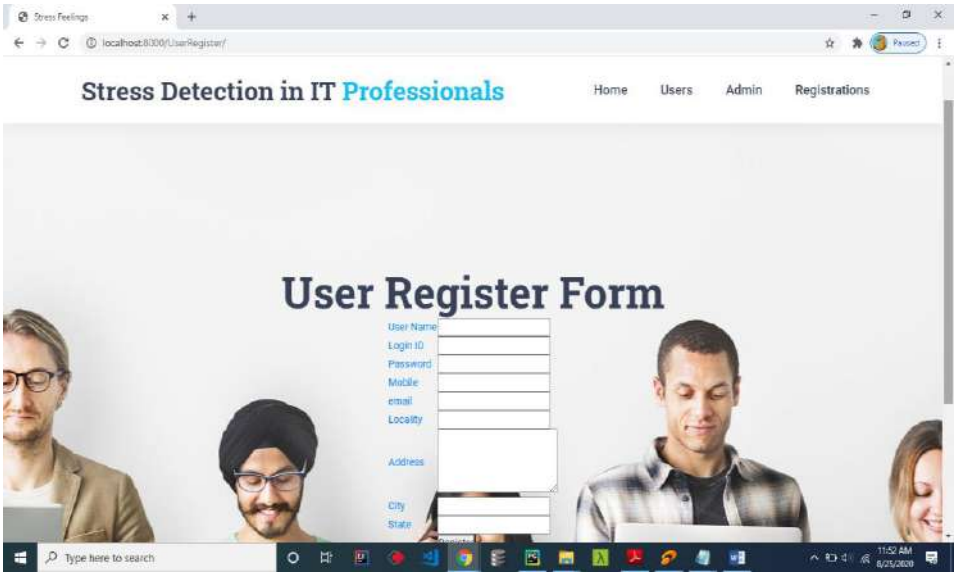
```
    ı('admin/', admin.site.urls),
    path("", mainView.index, name="index"),
    path("index/", mainView.index, name="index"),
    path("logout/", mainView.logout, name="logout"),
    path("UserLogin/", mainView.UserLogin, name="UserLogin"),
    path("AdminLogin/", mainView.AdminLogin, name="AdminLogin"),
    path("UserRegister/", mainView.UserRegister, name="UserRegister"),

    ### User Side Views
    path("UserRegisterActions/", usr.UserRegisterActions, name="UserRegisterActions"),
    path("UserLoginCheck/", usr.UserLoginCheck, name="UserLoginCheck"),
    path("UserHome/", usr.UserHome, name="UserHome"),
    path("UploadImageForm/", usr.UploadImageForm, name="UploadImageForm"),
    path("UploadImageAction/", usr.UploadImageAction, name="UploadImageAction"),
    path("UserEmotionsDetect/", usr.UserEmotionsDetect, name="UserEmotionsDetect"),
    path("UserLiveCameDetect/", usr.UserLiveCameDetect, name="UserLiveCameDetect"),
    path("UserKerasModel/", usr.UserKerasModel, name="UserKerasModel"),
    path("UserKnnResults/", usr.UserKnnResults, name="UserKnnResults"),

    ### Admin Side Views
    path("AdminLoginCheck/", admins.AdminLoginCheck, name="AdminLoginCheck"),
    path("AdminHome/", admins.AdminHome, name="AdminHome"),
    path("ViewRegisteredUsers/", admins.ViewRegisteredUsers, name="ViewRegisteredUsers"),
    path("AdminActivaUsers/", admins.AdminActivaUsers, name="AdminActivaUsers"),
    path("AdminStressDetected/", admins.AdminStressDetected, name="AdminStressDetected"),

    path("AdminKNNResults/", admins.AdminKNNResults, name="AdminKNNResults"),

urlpatterns += staticfiles_urlpatterns()
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

5. SCREENSHOTS

5. SCREENSHORTS

5.1 HOME PAGE

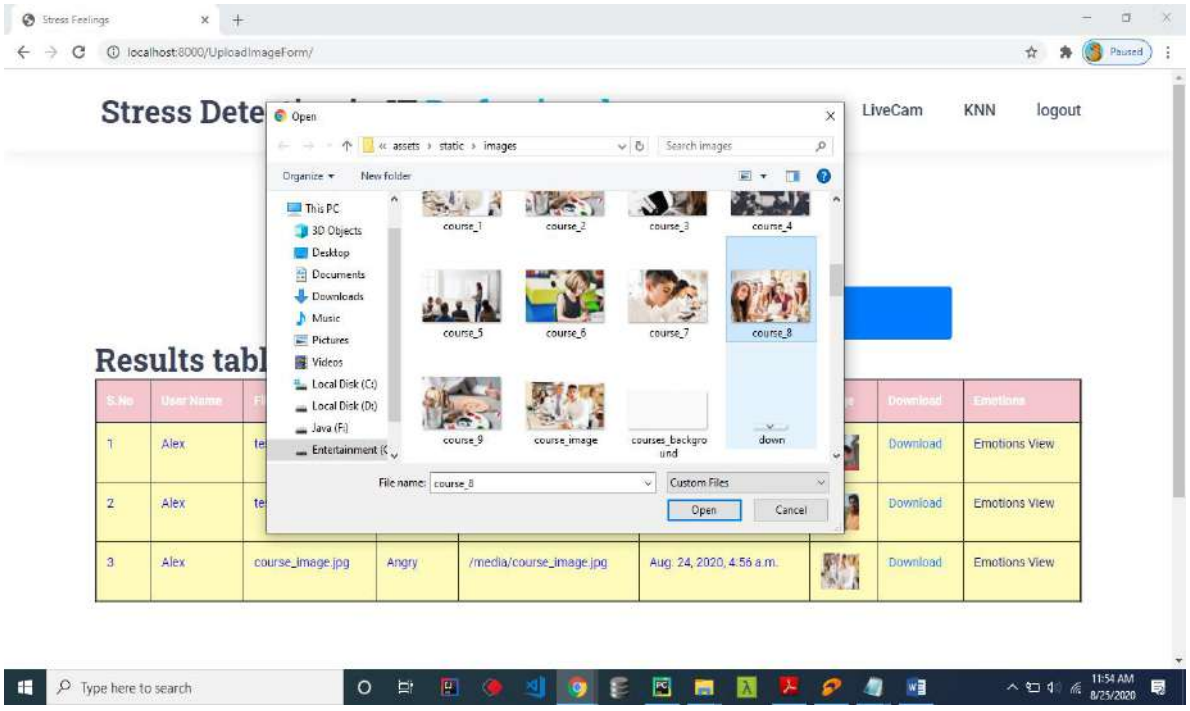
Screenshot 5.1.1: Home Page



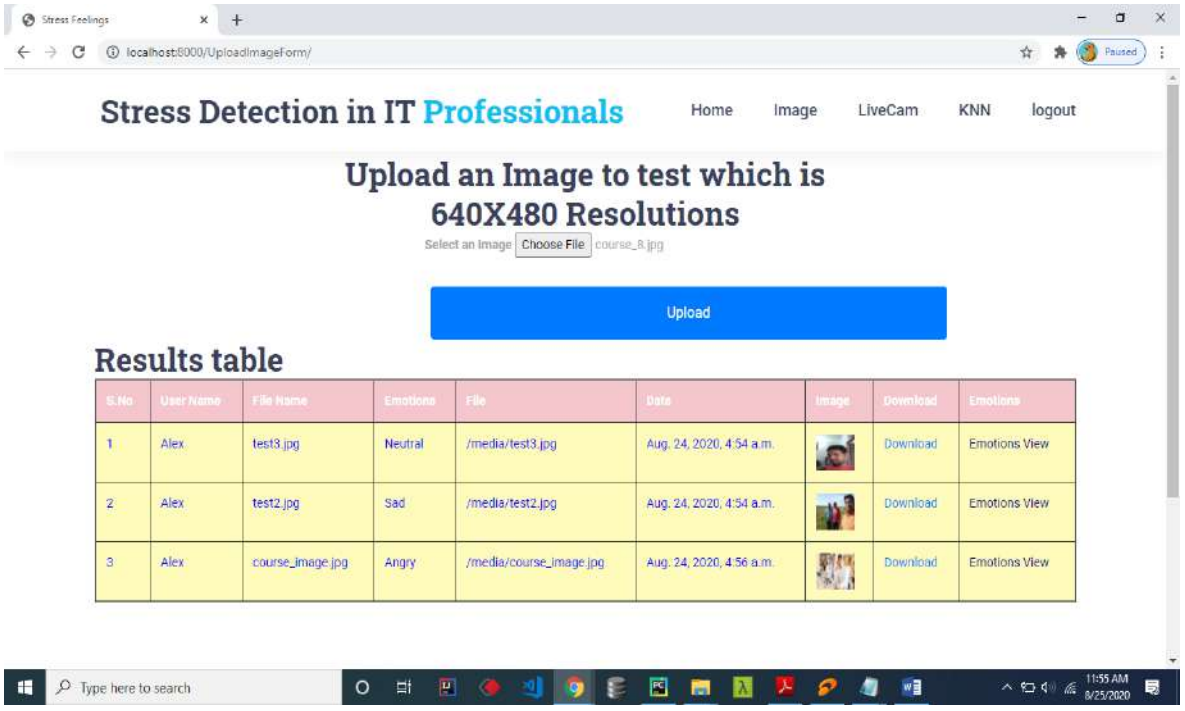
Screenshot 5.1.2: User Registration Page

Screenshot 5.1.3: User Login Form

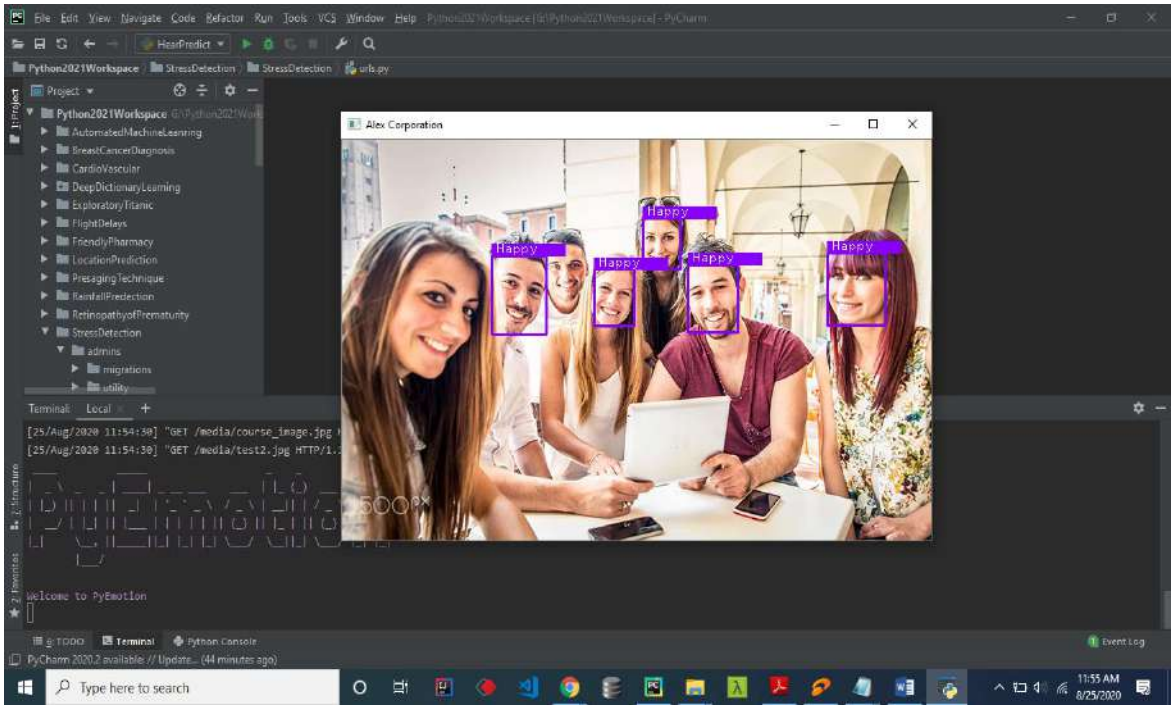
Screenshot 5.1.4: User Home Page



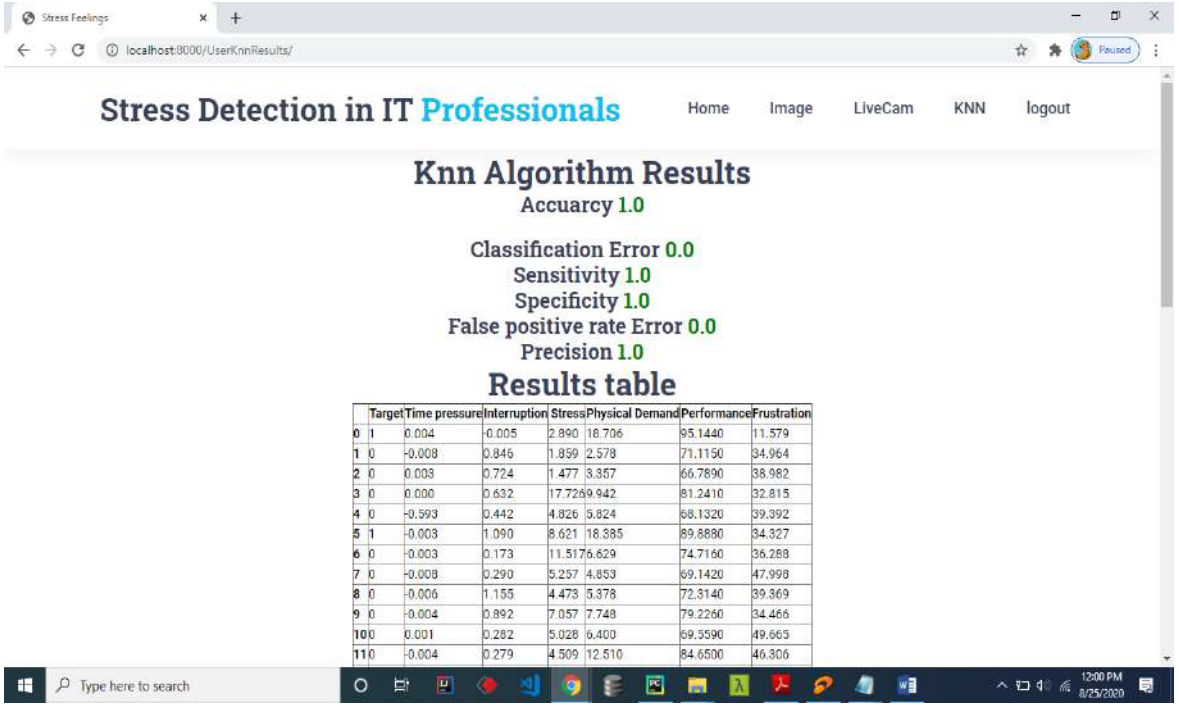
Screenshot 5.1.5: Giving Image as Input



Screenshot 5.1.6: Upload Image

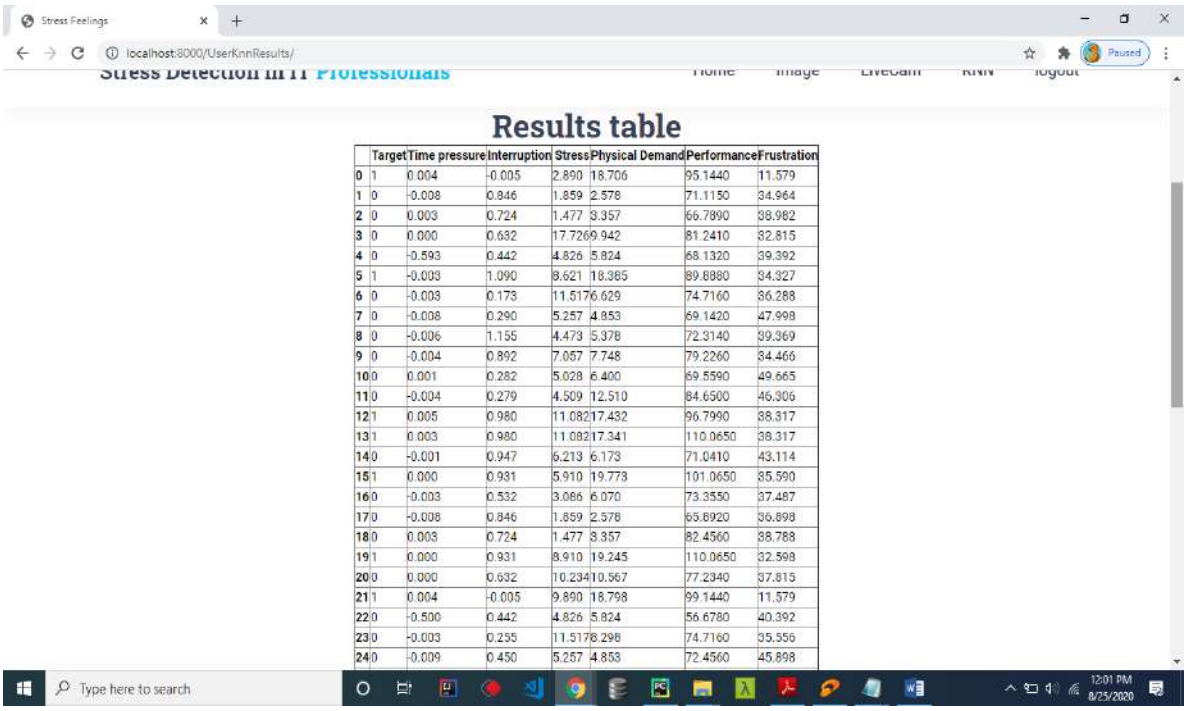


Screenshot 5.1.7: Response Image



	Target	Time pressure	Interruption	Stress	Physical Demand	Performance	Frustration
0	1	0.004	-0.005	2.890	18.706	95.1440	11.579
1	0	-0.008	0.846	1.859	2.578	71.1150	34.964
2	0	0.003	0.724	1.477	3.857	66.7890	38.982
3	0	0.000	0.632	17.726	9.942	81.2410	32.815
4	0	-0.593	0.442	4.826	5.824	68.1320	39.392
5	1	-0.003	1.090	8.621	18.385	89.8880	34.327
6	0	-0.003	0.173	11.517	6.629	74.7160	36.288
7	0	-0.008	0.290	5.257	4.853	69.1420	47.998
8	0	-0.006	1.155	4.473	5.378	72.3140	39.369
9	0	-0.004	0.892	7.057	7.748	79.2260	34.466
10	0	0.001	0.282	5.028	6.400	69.5590	49.665
11	0	-0.004	0.279	4.509	12.510	84.6500	46.306

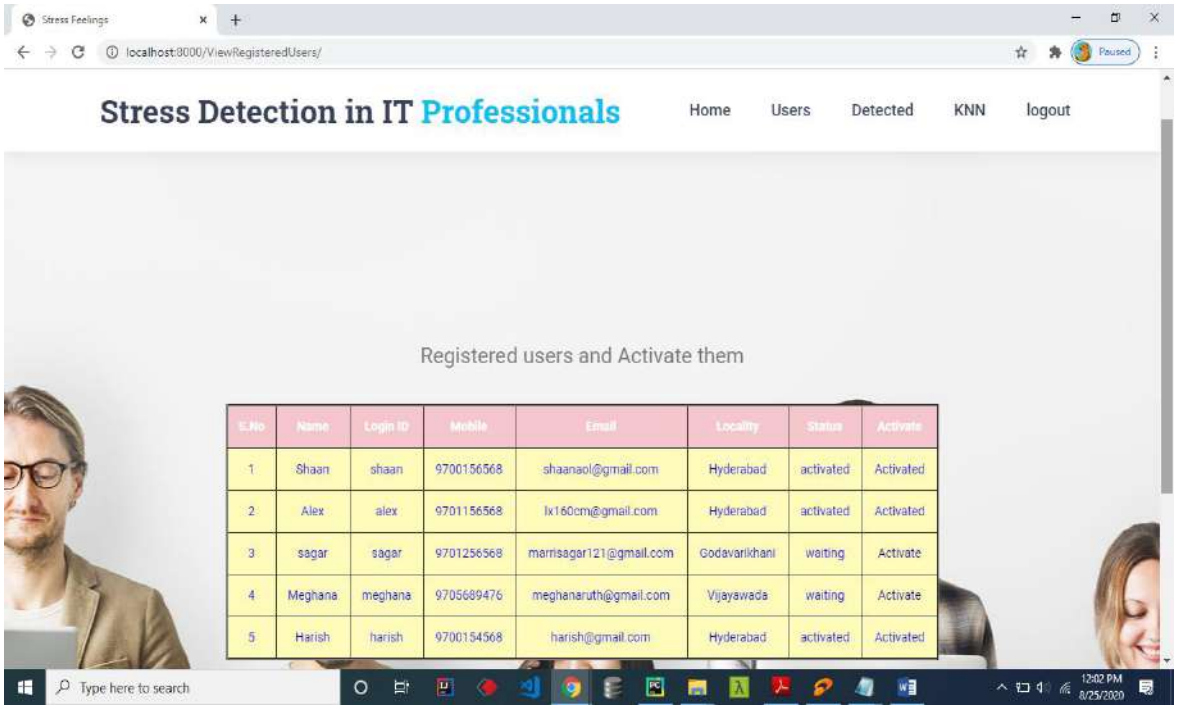
Screenshot 5.1.8: KNN Results



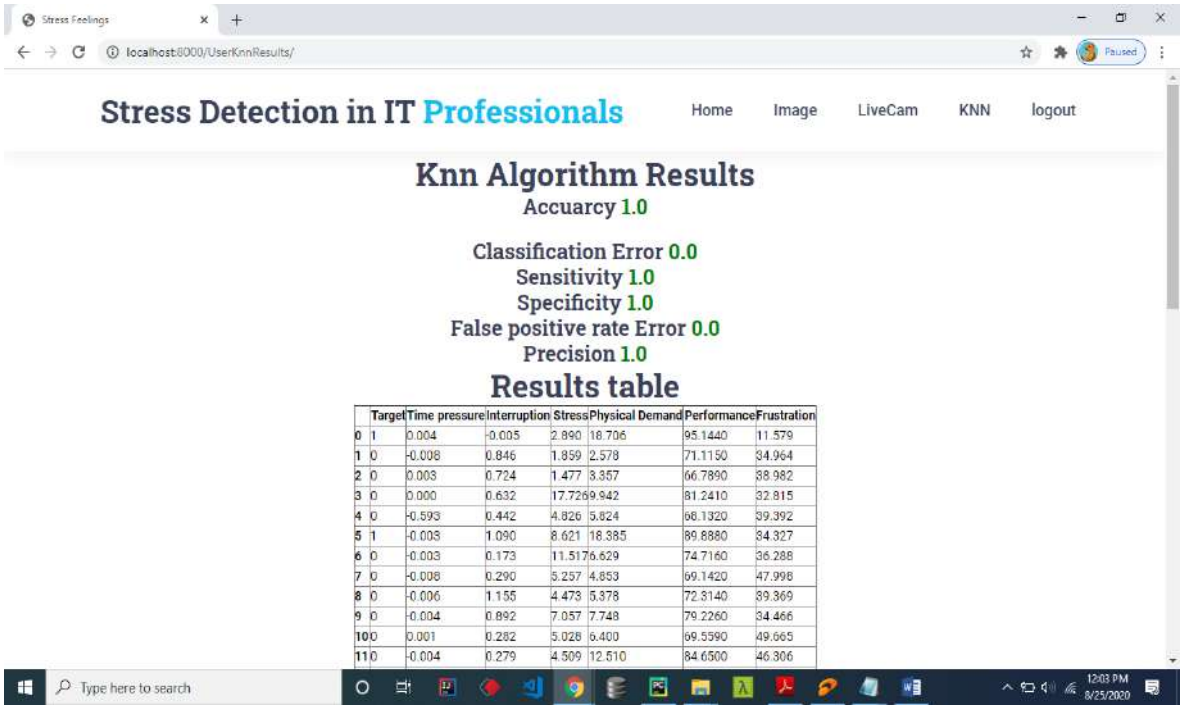
	Target	Time pressure	Interruption	Stress	Physical Demand	Performance	Frustration
0	1	0.004	-0.005	2.890	18.706	95.1440	11.579
1	0	-0.008	0.846	1.859	2.578	71.1150	34.964
2	0	0.003	0.724	1.477	3.357	66.7890	38.982
3	0	0.000	0.632	17.726	9.942	81.2410	32.815
4	0	-0.593	0.442	4.826	5.824	68.1320	39.392
5	1	-0.003	1.090	8.621	18.385	89.8880	34.327
6	0	-0.003	0.173	11.517	6.629	74.7160	36.288
7	0	-0.008	0.290	5.257	4.853	69.1420	47.998
8	0	-0.006	1.155	4.473	5.378	72.3140	39.369
9	0	-0.004	0.892	7.057	7.748	79.2260	34.466
10	0	0.001	0.282	5.028	6.400	69.5590	49.665
11	0	-0.004	0.279	4.509	12.510	84.6500	46.306
12	1	0.005	0.980	11.082	17.432	96.7990	38.317
13	1	0.003	0.980	11.082	17.341	110.0650	38.317
14	0	-0.001	0.947	6.213	6.173	71.0410	43.114
15	1	0.000	0.931	5.910	19.773	101.0650	35.590
16	0	-0.003	0.532	3.086	6.070	73.3550	37.487
17	0	-0.008	0.846	1.859	2.578	65.8920	36.898
18	0	0.003	0.724	1.477	3.357	82.4560	38.788
19	1	0.000	0.931	8.910	19.245	110.0650	32.598
20	0	0.000	0.632	10.234	10.567	77.2340	37.815
21	1	0.004	-0.005	9.890	18.798	99.1440	11.579
22	0	-0.500	0.442	4.826	5.824	56.6780	40.392
23	0	-0.003	0.255	11.517	8.298	74.7160	35.556
24	0	-0.009	0.450	5.257	4.853	72.4560	45.898

Screenshot 5.1.9: Dataset

Screenshot 5.1.10: Admin Login Page



Screenshot 5.1.11: Registered Users



Screenshot 5.1.12: Admin View KNN Results

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discovery conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, 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.

6.2 TYPES OF TESTING

6.2.1 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.

6.2.2 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

6.2.3 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.

6.3 TEST CASES

6.3.1 UPLOADING QUERIES

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.

Table No. 6.2.1.1 Test Cases

7.CONCLUSION

7.CONCLUSION & FUTURE SCOPE

7.1 PROJECT 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.

7.2 FUTURE SCOPE

Biomedical wearable sensors embedded with IoT technology is a proven combination in the health care sector. The benefits of using such devices have positively impacted the patients and doctors alike. Early diagnosis of medical conditions, faster medical assistance by means of Remote Monitoring and Telecommunication, emergency alert mechanism to notify the 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.

8. BIBILOGRAPHY

8.1 REFERENCES

[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] Villarejo, M.V., Zapirain, B.G., Zorrilla, A.M.. A stress sensor based on galvanic skin response (gsr) controlled by zigbee. Sensors 2012; 12(5):6075–6101.

[6] Liu, D., Ulrich, M.. Listen to your heart: Stress prediction using consumer heart rate sensors 2015.

[7] Nakashima, Y., Kim, J., Flutura, S., Seiderer, A., Andre, E.. Stress recognition in daily work. In: ´ International Symposium on Pervasive Computing Paradigms for Mental Health. Springer; 2015, p. 23–33.

[8] Palanisamy, K., Murugappan, M., Yaacob, S.. Multiple physiological signal-based human stress identification using non-linear classifiers.Elektronika ir elektrotechnika 2013;19(7):80–85.

8.2 GITHUB LINK

https://github.com/Prathyusha571/An-enhaced-system-to-detect-stress-in-IT-professionals-using-image-processing-and-machine-learning

V

8.3 PAPER PUBLICATION

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

AN ENHANCED SYSTEM FOR DETECTING STRESS IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE LEARNING

Dr. K. Maheswari[1], Prathyusha Addanki[2], Saikat Kumar Kundu[3] and Adarsh Dagde[4]

[1]Associate Professor,Dept of CSE,CMR Technical Campus,Kandlakoya,Medchal.

[2]Student,Dept of CSE,CMR Technical Campus,Kandlakoya, Medchal.

[3] Student,Dept of CSE,CMR Technical Campus,Kandlakoya,Medchal.

[4] Student,Dept of CSE,CMR Technical Campus,Kandlakoya,Medchal.

***Abstract*— The main motive of our design is to detect stress in the IT professionals using pictorial Machine literacy and Image processing. We've upgraded an old system interpretation of stress discovery systems which barred the live discovery and the particular comforting but this system comprises of live discovery and time to time analysis of their staff and detecting physical as well as internal stress situations in his/ her by furnishing them with proper remedies for managing stress position by furnishing check from time to time. Our system substantially focuses on trying to reduce the stress situations and making them work in a healthy terrain for the workers and to get the outgrowth of them during working hours.**

***Keywords*— Stress, Machine Learning, KNN , Image Preprocessing, Eyebrow Displacement**

I. INTRODUCTION

Stress operation systems play a significant part to detect the stress situations which disrupts our socio profitable life. As World Health Organization(WHO) says, Stress is a part life it's unwelcome state of emotional thrill that people witness in situations like working hours in front of computer. mortal stress leads to internal as well as socio- financial problems, lack of clarity in work, poor working relationship, depression and eventually commitment of suicide in severe cases. One of the traditional system to descry stress is grounded on questionnaire. This system fully depends on the answers given by the individualities, people will be tremulous to say whether they're stressed-out or normal. Monitoring the emotional status of a person who's working infront of a computer for longer duration is crucial for the safety of the person. A camera is used to capture the near anterior view of the person while is working on the computer.The camera is mounted facing a person.The image analysis includes the computation of the friction in the position of the eyebrows from its mean position.Maria Viqueira etal. describes the internal stress vaticination using a standalone stress seeing tackle by uniting GSR as the only physiological detector. David Liu et proposed a exploration to prognosticate stress situations solely from Electrocardiogram(ECG). This employs the detector data from detectors similar as pressure distribution, HR, Blood Volume palpitation(BVP) and Electro dermal exertion(EDA). An eye shamus detector is also used which totally analyses the eye movements with the stressors like Strop word test and information related to volleytasks. The authors of the performed perceived stress discovery by a set of non-invasive detectors which collects all the physiological signals similar as ECG, GSR, Electroencephalography(EEG), EMG, and Achromatism of supplemental oxygen(SpO2). nonstop stress situations are estimated using the physiological detector data similar as GSR, EMG, HR, Respirationin. Every detector data is compared with a stress indicator which is a threshold value used for detecting the stress position. The authors of collected data from 16 individualities under four stressor conditions which were tested with Bayesian Network, J48 algorithm and successional minimum Optimization(SMO) algorithm for prognosticatingstress.Various features are uprooted from the generally used physiological signals similar as ECG, EMG, GSR, BVPetc., measured using applicable detectors and named features are grouped into clusters for farther discovery of anxiety situations. In, it's concluded that lower clusters affect in better balance in stress discovery using the named General Retrogression Neural Network(GRNN) model. This results in the fact that different combinations of the uprooted features from the detector signals give better results to prognosticate the nonstop anxiety position. Frequency sphere features like LF power(low frequency power from0.04 Hz to0.15 Hz), HF power(High frequency power from0.15 Hz to Hz), LF/ HF(rate of LF to the HF). and time sphere features like Mean, Median, standard divagation of heart signal is considered for nonstop real time stress discovery in. Bracket using decision tree similar as PLDA is performed using two). Stressors videlicet pickup task and strop grounded word test wherein the authors concluded that the stressor grounded bracket proves wrong. In 2016, Gjoreski etal. Created laboratory grounded stress discovery classifiers from ECG signal and HRV features. Features of ECG are analyzed using GRNN model to measure the stress position. Heart rate variability(HRV) features and

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

RR(cycle length variability interval length between two consecutive Rs) interval features are used to classify the stress position. It's noticed that Support Vector Machine(SVM) was used as the bracket algorithm generally due to its conception capability and sound fine background colorful kernels were used to develop models using SVM and it's concluded in that a direct SVM on both ECG frequency features and HRV features performed stylish, outperforming other model choices. In this study, the stress situations in workers are also noticed to raise the bar high. Though there are numerous associations who give internal health related schemes for their workers but the issue is far from control. In this paper we try to go in the depth of this problem by trying to descry the stress patterns in the working hand in the companies we'd like to apply image processing and machine literacy ways to dissect stress patterns and to constrict down the factors that explosively determine the stress situations. Assaying and manipulating the image. Affair in which result is altered image or report that's grounded on image analysis. System gets the capability to automatically learn and ameliorate from tone- gests without being explicitly programmed using Machine literacy which is an operation of artificial intelligence(AI). Unequivocal programming to perform the task grounded on prognostications or opinions builds a fine model grounded on" training data" by using Machine literacy. The birth of retired data, association of image data and fresh 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 of all physical conditions are caused by stress. Stress is believed to be the top cause in cardiovascular conditions. Stress can place one at advanced threat for diabetes, ulcers, asthma, migraine headaches, skin diseases, epilepsy, and sexual dysfunction. Each of these conditions, and host of others, is psychosomatic(i.e., either caused or exaggerated by internal conditions similar as stress) in nature. Stress has three point goods private goods of stress include passions of guilt, shame, anxiety, aggression or frustration. Individualities also feel like tired, tense, nervous, perverse, temperamental, or lonely. Goods of behavioral stress are seen similar as increased accidents, use of medicines or alcohol, horselaugh out of environment, fantastic or argumentative gets , veritably hyper excitable moods, and/ or eating or drinking to excess. Dwindling internal capability, disabled judgment, rash opinions, obliviousness or acuity to review are some of the goods of Cognitive stress.

II. LITERATURE SURVEY

Tanev etal. Classified acute stress using direct and non-linear heart rate variability analysis deduced from sternal ECG. This work is an airman study with a focus on developing a system for detecting short- term psychophysiological changes through heart rate variability(HRV) features. The purpose of this study is to gain sapience on a set of features that could be used to descry psychophysiological changes that do during habitual stress. This study elicited four different types of thrill by images, sounds, internal tasks and rest, and classified them using direct andnon-linear HRV features from electrocardiograms(ECG) acquired by the wireless wearable ePatch ® archivist. The loftiest recognition rates were acquired for the neutral stage(90), the acute stress(80) and the birth(80) by sample entropy,detrended change analysis and regularized high frequency features.

Nidhi Raichur etal. Designed a model for stress discovery. Stress is a part of life it's an unwelcome state of emotional thrill that people witness in situations like working for long hours in front of computer. Computers have come a way of life and hence we're thus more affected by the ups and campo that they beget us. One cannot just fully avoid their work on computers but one can at least control his/ her operation when being scarified about him being stressed at certain point of time. Monitoring the emotional status of a person who's working in front of a computer for longer duration is pivotal for the safety of a person. In this work a real- timenon-intrusive vids are captured, which detects the emotional status of a person by analyzing the facialexpression.We employ a fashion that allows us to train a model and dissect differences in prognosticating the features. Theano is a python frame which aims at perfecting both the prosecution time and development time of the direct retrogression model which is used then as a deep literacy enhancement. The experimental results show that the advanced system is well on data with the general model of all periods.

S. Reddy etal. have experimented with approach for stress vaticination in working workers. Stress diseases are a common issue among working IT professionals in the assiduity moment. With changing life and work societies, there's an increase in the threat of stress among the workers. In this paper, we'd like to apply machine literacy methodology to dissect stress patterns in working grown-ups and to constrict down the factors that explosively determine the stresslevels.Boosting had the loftiest delicacy among the models enforced. By using Decision Trees, features that impact stress were linked as gender, family history and vacuity of health benefits in the plant.

G.Giannakakis etal. developed a model for the discovery and analysis of stress emotional countries through facial cues. A thorough experimental protocol was established to induce methodical variability in affective countries through a variety of external and internal stressors. Features of the frame included eye- related events, mouth exertion, head stir parameters and heartrate.The results indicated that, specific facial cues, deduced from eye exertion, mouth exertion, head movements and camera grounded heart exertion achieve good delicacy and are suitable as discriminational pointers of stress and anxiety.

III. PROPOSED SYSTEM

In Proposed System Machine Learning algorithms like KNN classifiers are applied to classify stress. Image Processing is used at the first stage for discovery, the employee's image is given by the cyber surfer which serves as input. In order to get an ameliorate the value image or to prize some useful information from its image processing is used by converting image into digital form and performing some operations on it. By taking input as an image and affair may be image or characteristics

V

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

associated with those images. The emotion is displayed on the given box. The stress position indicating by Angry, repulsed, Fearful, Sad, happy, normal and on emotion.

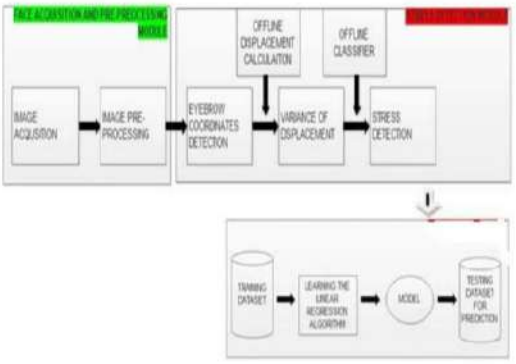


Fig.1. Diagram of Architecture

IV. METHODOLOGIES

A. *KNN Classifier*

K- Nearest Neighbor(KNN) is used for bracket as well as retrogression analysis. It's a supervised literacy algorithm which is used for prognosticating if a person needs treatment or not. KNN classifies the dependent variable grounded on how analogous it is; independent variables are to an analogous case from the formerly knowndata.Since the algorithm requires no training before making prognostications, new data can be added seamlessly.

B. *Image Pre-Processing*

G(i, j) = α. F(i, j) β, α> 0 and β are called as gain and bias parameters, these are used to bright and discrepancy the image. The face accession module processes the videotape sequences captured by the camera. The image frames are uprooted and the pre- processing of the images for the posterior analysis in the farther modules is done. Pre-processing of the images includes two metamorphoses of the uprooted frame. First one being the pixel metamorphosis and the other bone is double metamorphosis after pixel transformation. Pixel metamorphosis as an image processing function is a fashion of transubstantiating a pixel value. Each affair pixel value depends on only the corresponding input pixel value, which helps alter the brilliance and discrepancy of an input image. The two generally used point processing are addition and the addition with aconstant.G(i, j) = α. F(i, j) β, The parameters α> 0 and β are frequently called gain and the bias parameters, these parameters are said to control the brilliance and the discrepancy of the imagerespectively.F(i, j)-> source image pixel, G(i, j)-> affair image pixel. Where the(i, j) indicate that the pixel is located at i- th row and j- th column. The pixel metamorphosis of an image is done to make the model more general and more different. That's if a person is with a darker complexion the eyebrow and its co- ordinates discovery becomes delicate.

C. *Stress Detection*

Stress detection is grounded on the rigid metamorphoses shown by the eyebrows which are used as the major facial area if interest in its posterior process of analysis. The preliminarily followed processes used to pre-process the input image for the eyebrow discovery is further used in the stress. Discovery methodology, which involves sub-modules of offline relegation computation, variation of relegation and classifier which helps in stress detection. The eyebrow metamorphosis in every posterior image over a period of time is estimated by calculating its variability. And the set of images over a period of time that show friction above the threshold is classified by the sub-module as stressed over the particular time interval. By Property Extraction newly designed dataset appears which contains only numerical input variables as a result of Principal Component Analysis feature selection transforming to 6 principal components.

V

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

V. MODULES

A. *User*

The user can register the first. While registering he needed a valid user dispatch and mobile for farther dispatches. Once the user registers also admin can spark the client. Once admin actuated the client also user can login into our system. First user has to give the input as image to the system. The python library will prize the features and applicable emotion of the image. If given image contain further than one faces also possible to descry. The stress position we're going to indicate by facial expression like sad, angry etc. The image processing completed then we're going to start the live sluice. In the live sluice also we can get the facial expression more than one persons also. Compare to tensor flow live sluice the tesnorflow live sluice will gormandize and better results. Formerly done the we're loading the dataset to perform the knn bracket delicacy precession scores.

B. *Admin*

Admin can login with his credentials. Once he login he can spark the user. The actuated user only login in our operations. The admin can set the training and testing data for the design stoutly to the law. The admin can view all user detected results in hid frame. By clicking a hyperlink in the screen he can descry the feelings of the images. The admin can also view the knn bracket detected results. The dataset in the excel format. By authorized persons we can increase the dataset size according the imaginary values.

C. *Data Pre-process and Machine Learning*

Dataset contains grid view of formerly stored dataset conforming multitudinous parcels, by Property birth recently designed dataset appears which contains only numerical input variables as a result of star element Analysis point selection transubstantiating to 6 top factors which are Condition(No stress, Time pressure, Interruption), Stress, Physical Demand, Performance and Frustration.

K- Nearest Neighbor(KNN) is used for bracket as well as retrogression analysis. It's a supervised literacy algorithm which is used for prognosticating if a person needs treatment or not. KNN classifies the dependent variable grounded on how analogous it is; independent variables are to an analogous case from the formerly known data. The Knn Bracket can be called as a statistical model that uses a double dependent variable. In bracket analysis, KNN is estimating the parameters of a KNN model. Mathematically, a double KNN model has a dependent variable with two possible value, which is represented by an index variable, where the two values are labeled" 0" and" 1".

VI. RESULTS

A. *User Results*

Fig.2. User Registration Form

V

The International journal of analytical and experimental modal analysis ISSN NO: 0886-9367

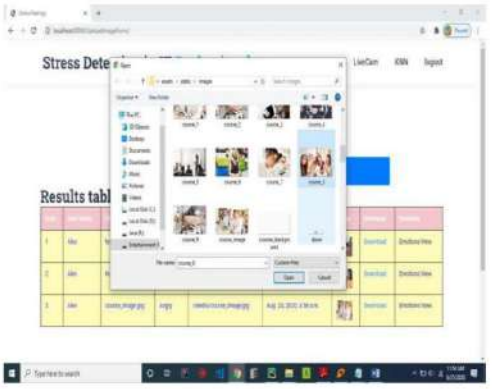


Fig.3. Uploading Image as an input

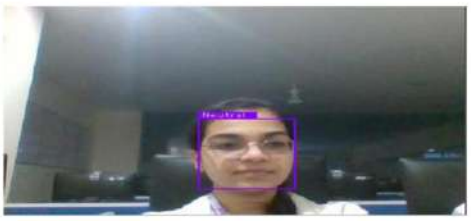


Fig.4. Neutral Emotion Detected

Fig.5. Sad Emotion Detected

Fig.6. Happy Emotion Detected

B. Admin Results

FIG.7. ADMIN LOGIN PAGE

V

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

Certificate of Publication

This is to certify that the paper entitled CERTIFICATE ID: IJAEMA/7490

"AN ENHANCED SYSTEM FOR DETECTING STRESS IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE LEARNING"

Authored by :

Saikat Kumar Kundu

From

CMR Technical Campus,Kandlakoya,Medchal.

Has been published in

IJAEMA JOURNAL, VOLUME XIV, ISSUE VI, JUNE- 2022

Michal A. Olszewski Editor-In-Chief

IJAEMA JOURNAL

http://ijaema.com/

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

Certificate of Publication

This is to certify that the paper entitled CERTIFICATE ID:IJAEMA/7490

“AN ENHANCED SYSTEM FOR DETECTING STRESS IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE LEARNING”

Authored by :

Prathyusha Addanki

From

CMR Technical Campus,Kandlakoya,Medchal.

Has been published in

IJAEMA JOURNAL, VOLUME XIV, ISSUE VI, JUNE- 2022

Michal A. Olszewski Editor-In-Chief
IJAEMA JOURNAL

http://ijaema.com/

V

The International Journal of Analytical and Experimental Modal analysis

An UGC-CARE Approved Group - II Journal

An ISO : 7021 - 2008 Certified Journal

ISSN NO: 0886-9367 / web : http://ijaema.com / e-mail: submitijaema@gmail.com

Certificate of Publication

This is to certify that the paper entitled CERTIFICATE ID: IJAEMA/7490

"AN ENHANCED SYSTEM FOR DETECTING STRESS IN IT PROFESSIONALS BY IMAGE PROCESSING AND MACHINE LEARNING"

Authored by :

Adarsh Dagde

From

CMR Technical Campus,Kandlakoya,Medchal.

Has been published in

IJAEMA JOURNAL, VOLUME XIV, ISSUE VI, JUNE- 2022

Michal A. Olszewski Editor-In-Chief

IJAEMA JOURNAL

6.3 IMPACT FACTOR

ISO International Organization for Standardization 7021-2008

UGC APPROVED

http://ijaema.com/