STRESS DETECTION OF AN IT-EMPLOYEE USING MACHINE LEARNING AND IMAGE PROCESSING

MINI PROJECT REPORT

SUBMITTED IN FULFILMENT OF THE REQUIREMENTS TO

RGUKT-SRIKAKULAM

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

D. Vandana (S180247)

V. KeerthiChavla (S180657)

SK.V.Shareef (S180258)

Under the Esteemed Guidance of

Mr. N. Sesha Kumar,

Assistant Professor,

Department Of Computer Science And Engineering,

RGUKT-SRIKAKULAM, ETCHERL

CERTIFICATE OF COMPLETION

This is to certify that the mini project report titled "**STRESS DETECTION OF AN IT-EMPLOYEE USING MACHINE-LEARNING AND IMAGE-PROCESSING**" was successfully completed by D.Vandana (S180247), V.KeerthiChavala(S180657), SK.V.Shareef(S180258) under the guidance of Mr.Sesha Kumar In partial fulfilment of the requirements for the Mini Project in Computer Science And Engineering of Rajiv Gandhi University of Knowledge Technologies under my guidance and output of the work carried out is satisfactory.

Project Guide

Mr.N.Sesha Kumar ,

Assistant Professor

BONA FIDE CERTIFICATE

This is to certify that report entitled "STRESS DETECTION OF AN IT EMPLOYEE" was submitted by vandana bearing ID no.S180247,shareef bearing ID no. S180258,keerthi chavla bearing ID no.S180657, in partial fulfilment of the requirements for the award of bachelor of technology in computer science in a bonafede work carried out by under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other university or institution to award any degree or diploma.

Mr. N. Sesha Kumar
Assistant Professor
RGUKT-Srikakulam

Mr.N.Sesha Kumar ,
Head of the Department(C.S.E)
RGUKT-Srikakulam

ABSTRACT

The detection of stress among It-Employee is crucial for ensuring their well-being and productivity in the workplace. Traditional methods of assessing stress levels rely on subjective self-reporting, which is prone to biases. In this study, we propose a novel approach that leverages machine learning and image processing techniques to analyze facial emotions and accurately classify stress-related expressions. Our methodology involves training Convolutional Neural Network (CNN) algorithms using a dataset of annotated facial expressions to develop a robust stress detection model. By utilizing this trained model, we can predict stress levels based on facial emotions captured through images. This approach provides an objective and non-intrusive means of assessing stress, thereby enabling timely interventions to support employees and improve their overall work environment.

KEYWORDS: CNN, Facial emotions, Keras

ACKNOWLEDGMENT

We would like to articulate my profound gratitude and indebtedness to our project guide Asst. Prof. Sri N.Sesha Kumar sir Assistant Professor who has always been a constant motivation and guiding factor throughout the project time. it has been a great pleasure for us to get an opportunity to work under his guidance and complete the work successfully. We wish to extend our sincere thanks to Asst. Prof. Sri N.Sesha Kumar sir Head of the CSE {Computer science and engineering} Department, for his constant encouragement throughout the project.we are also to grateful to other members of the department without their support our work would have been carried out so successfully. We thank one and all who have rendered help to us directly or indirectly in the completion of our the work.

Project Associate

D.Vandana - S180247

V.Keerthi Chavla – S180657

SK.V.Shareef –S180258

Table of Contents

CHAPTER – 1

INTRODUCTION

1.1 Introduction

The main aim of our project is an instant detection of stress of an employee using machine learning and image processing techniques. The system will use computer vision algorithms to analyze physiological and behavioural changes in employee's facial expressions to determine their stress. In this by using the webcam we have to take the picture of a person, after uploading that picture by using the CNN Algorithms we can detect stress of a person. The system aims to provide early detection of stress in IT employee, which could lead to interventions that improve employee well-being.

1.2 Statement of the problem

Stress is a state of worry or mental tension caused by a difficult situation. The aim of this project is to design and develop a stress detection system specifically targeted towards IT- Employees. The system will utilize the CNN (Convolutional neural network) algorithm to enhance accuracy and performance. The stress detection system will analyze various visual cues and facial expressions captured through images or real time webcam streams. This tool will contribute to improving employee well – being and enhancing overall productivity within IT organizations.

1.3 Scope of the project

The scope of the project is to develop a stress detection system for IT employees, which allows users to capture images of employees. The system employees Convolutional Neural Network (CNNs) to analyze these images obtained through a webcam and determine whether the person is experiencing stress or not. The focus is on utilizing image analysis and CNN algorithms to provide accurate stress detection for IT employees.

1.4 Objectives

- Enable users to upload images of employees.
- Utilize Convolutional Neural Network (CNNs) to analyze the images.
- Based on the analysis of the images, determine whether the person is experiencing stress or not.

1.5 Definitions

- CNN: Convolutional Neural Networks
    - Convolutional Neural Network is a network architecture which is used to process the data which is stored as pixel data.
- Different classes of facial emotions:
  - Happy: This class represents positive emotion such as joy,pleasure and satisfaction.
  - Sad: This represents negative emotions such as sadness or disappointment.
  - Angry: This represents emotions of anger, irritations or frustration.
  - Surprised: This is represents emotions of surprise or astonishment
  - Disgusted: This class represents emotions of disgust or aversion.
  - Fearful: This represents emotions of fear or anxiety
  -Neutral: This represents Neutral state.

Keras:

Keras provides a user-friendly and efficient framework for developing and training deep learning models for stress detection in IT employees. Its simplicity, compatibility with TensorFlow, and extensive community support make it a popular choice for researchers and developers in this domain.

CHAPTER-2

LITETATURE SURVEY

- Several studies have been conducted in recent years on the use of machine learning and image processing techniques to detect stress in IT-Employees. One such study by Alghowinem utilized self-reporting and machine learning algorithms to detect stress in individuals. The study reported a 78% accuracy rate in detecting stress using self-reporting.
- Research on stress detection in the existing system relies on digital signal processing, which takes into account parameters such as galvanic skin response, blood volume, pupil dilation and skin temperature.
- Our proposed system utilizes facial emotion analysis to determine whether a person is experiencing stress or not.

2.1 Related Work

This section narrates some most related works on Stress detection using different techniques for detecting the stress.

2.1.1 Machine Learning Techniques

Compare algorithms used for the detection of stress:

Claaification techniques such as K-Nearest Neighbour Algorithm, Decision tree and Logistic regression

2.2 Existing Systems

- The existing system relies on self-reporting for data collection and information gathering.
- The existing system utilizes a decision tree as its underlying framework for decision-making analysis.
- The existing system incorporates both naïve Bayes and logistic regression algorithms for classification and prediction tasks.

2.2.1 Disadvantages

- Employee should feel uncomfortable to sharing their concerns with fear of negative consequences.
- It have less accuracy

2.3 Study

Key Features:

- ✓ Detecting stress

2.4 Benefits

- ✓ Early detection of stress
- ✓ High Accuracy

Summary

By utilizing CNN(Convolutional Neural Network), We can develop a highly accurate stress detection model.

CHAPTER – 3

PROPOSED SYSTEM

3.1 Proposed System

- Our proposed system utilizes facial emotion analysis to determine whether a person is experiencing stress or not
- Here, we employ trained data to analyze and determine whether an individual is experiencing stress or not.
- Picture processing is utilized enhance an image extract relavant information. By inputing an image and generating an output indicating whether the person is stressed or not.

3.2 Advantages

- It gives high accuracy
- The output of image analysis is transformed to determine whether the person is experiencing stress or not.

3.3 System Requirements

3.3.1 Software Requirements:

- CNN
- Google Colab
- Webcam

3.3.2 Hardware Requirements:

- I3 Processor system
- 4GB RAM or Higher

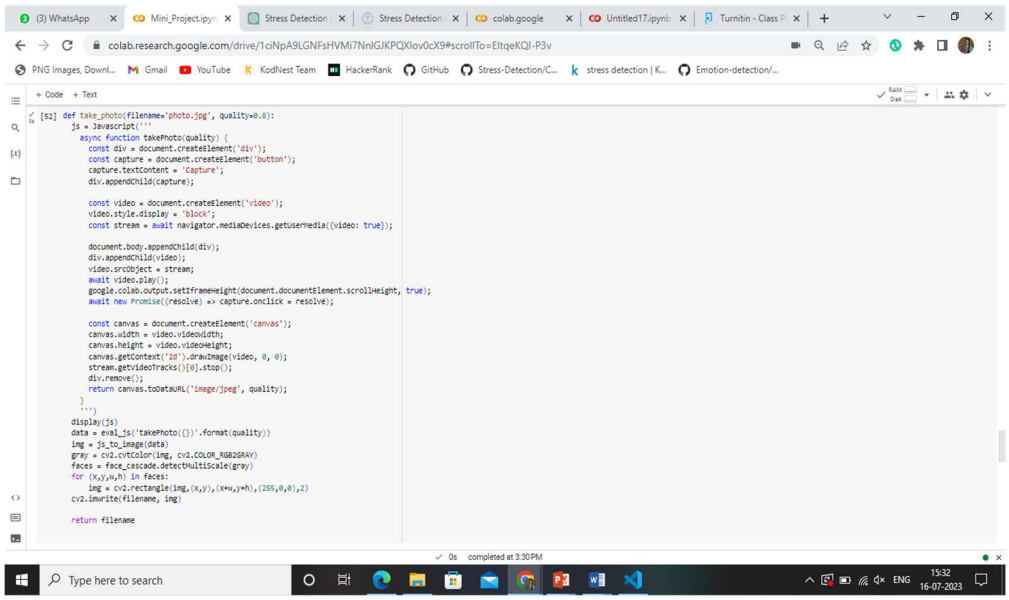
REFERENCES

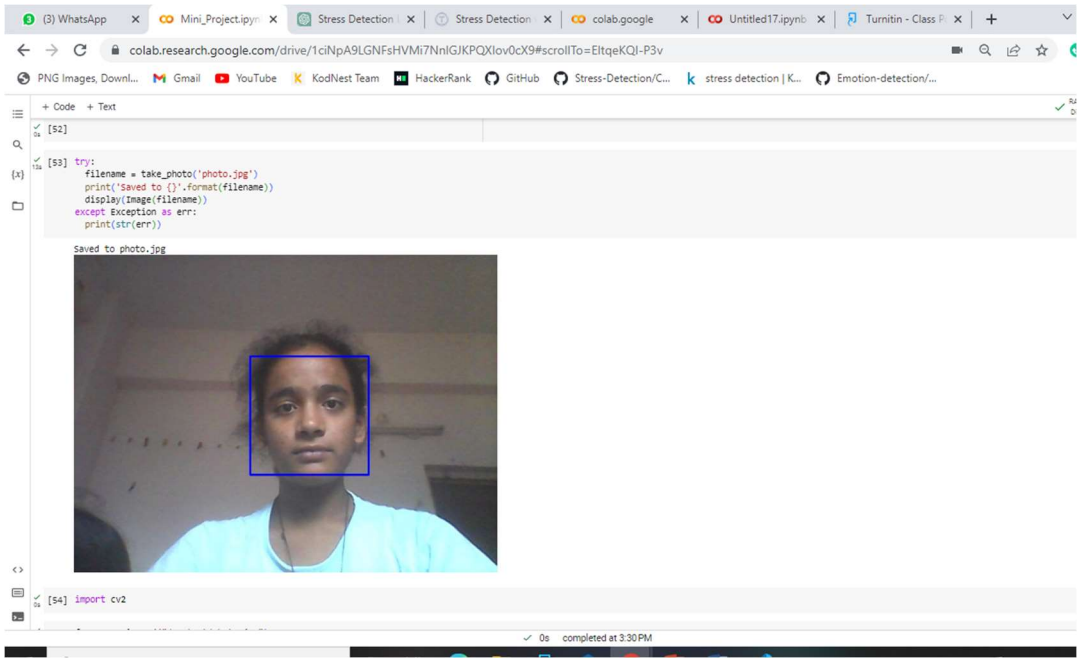
- Suresh Kumar Kanaparthi ; Surekha P; Lakshmi Priya Bellamkonda; Bhavya Kadiam; Beulah Mungara, “Detection of Stress in IT Employees using Machine Learning Technique”, 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), IEEE Conference, 2022.
- *IEEE Paper*
- *https://thecleverprogrammer.com/2021/12/20/stress-detection-with-machine-learning*

CHAPTER – 4

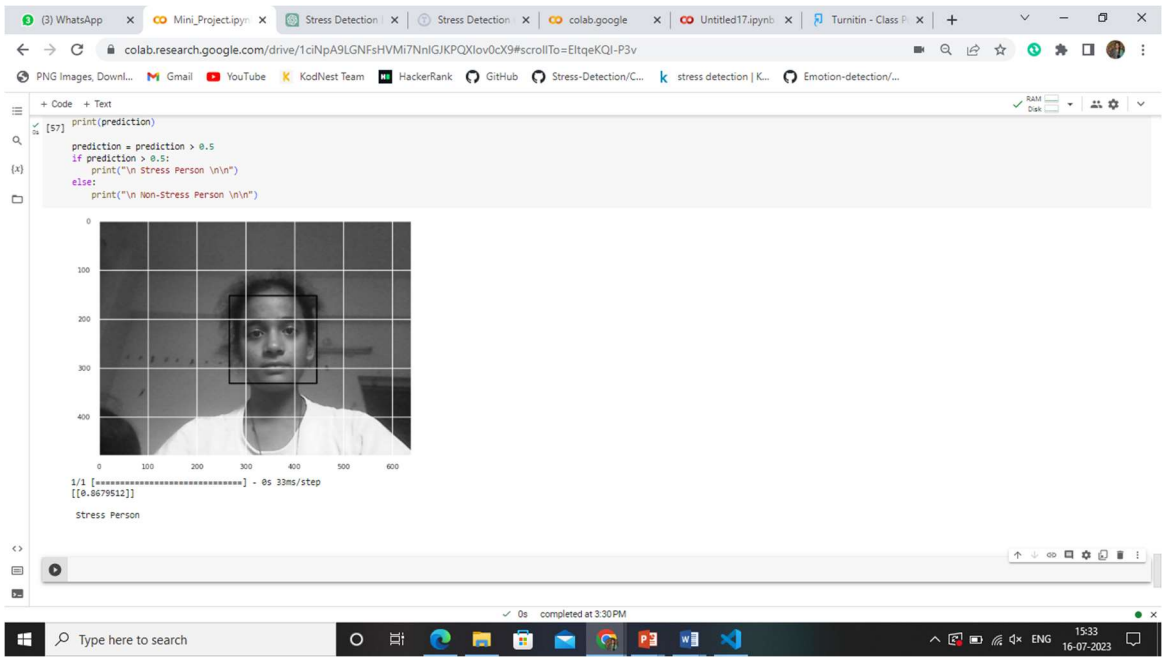
IMPLEMENTATION

4.1 CAPTURE IMAGE




colab.research.google.com/drive/1ciNpA9LGNFsHVMi7NnlGJKPQXIov0cX9#scrollTo=EltqeKQI-P3v
[52]
[53] try:
filename = take_photo('photo.jpg')
print('Saved to {}'.format(filename))
display(Image(filename))
except Exception as err:
print(str(err))
Saved to photo.jpg
[54] import cv2
completed at 3:30 PM

4.2 RESULT


colab.research.google.com/drive/1ciNpA9LGNFsHVMi7NnlGJKPQXIov0cX9#scrollTo=EltqeKQI-P3v
[57] print(prediction)
prediction = prediction > 0.5
if prediction > 0.5:
print("\n Stress Person \n\n")
else:
print("\n Non-Stress Person \n\n")
1/1 [==============================] - 0s 33ms/step
[[0.8679512]]
Stress Person
completed at 3:30 PM
Type here to search
ENG
15:33
16-07-2023

CHAPTER– 5

SOURCE CODE

5.1 STRESS DETECTION OF AN IT-EMPLOYEE USING MACHINE LEARNING AND IMAGE PROCESSING

```
From google.colab import drive

Drive.mount('/content/drive')

# Import Basic Libraries and modules

Import numpy as np

Import pandas as pd

Import matplotlib.pyplot as plt

Import random

From sklearn.model_selection import learning_curve, validation_curve

From sklearn.metrics import confusion_matrix

From sklearn.metrics import precision_score, recall_score, f1_score

# Loading dataset

Train_x=np.loadtxt('/content/drive/Mydrive/Stress_mini_project/project/Dataset/train_input.csv',delimeter="","")

Train_Y=np.loadtxt('/content/drive/Mydrive/Stress_mini_project/project/Dataset/train_lables.csv')

Test_X=np.loadtxt('/content/drive/Mydrive/Stress_mini_project/project/Dataset/test_input.csv',delimeter="" ,"")

Test_Y=np.loadtxt('/content/drive/Mydrive/Stress_mini_project/project/Dataset/test_lables.csv')

# Reshape

Train_X=train_X.reshape(10000,48,48,1)

Train_Y=train_Y.reshape(10000,1)

Test_X=test_X.reshape(2000,48,48,1)
```

```
Test_Y=test_Y.reshape(2000,1)
# Normalization
Train_X=train_X/225
Test_X=test_X/225
# Train images Display
Train_index=random.randint(0,9999)
Plt.imshow(train_X[train_index,:],cmp='gray')
Plt.show()
# Test images Display
Test_index=random.randint(0,1999)
Plt.imshow(test_X[test_index,:],cmap='gray')
Plt.show()
# CNN Model
# Importing modules
From keras.models import sequential
From keras.layers import Activation,Dropout,Conv2D,Dense
From keras.layers import BatchNormalization
From keras.layers import Maxpooling2D
From keras.layers import Flatten
#Model
Design=
design.add(Conv2D(filters=16, kernel_size=(7,7_, padding='same'))
```

```
# Model
design=Sequential()
design.add(Conv2D(filters=16, kernel_size=(7,7_, padding='same'))
input
design.add(Conv2D(filters=16, kernel_size=(7, 7), padding='same',
input_shape=(48,48,1)))  # features extraction
design.add(BatchNormalization())
design.add(Conv2D(filters=16, kernel_size=(7, 7), padding='same'))
design.add(BatchNormalization())
design.add(Activation('relu'))
```

```
design.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
design.add(Dropout(.5))

design.add(Conv2D(filters=32, kernel_size=(5, 5), padding='same'))
design.add(BatchNormalization())
design.add(Conv2D(filters=32, kernel_size=(5, 5), padding='same'))
design.add(BatchNormalization())
design.add(Activation('relu'))
design.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
design.add(Dropout(.5))

design.add(Conv2D(filters=64, kernel_size=(3, 3), padding='same'))
design.add(BatchNormalization())
design.add(Conv2D(filters=64, kernel_size=(3, 3), padding='same'))
design.add(BatchNormalization())
design.add(Activation('relu'))
design.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
design.add(Dropout(.5))

design.add(Conv2D(filters=128, kernel_size=(3, 3), padding='same'))
design.add(BatchNormalization())
design.add(Conv2D(filters=128, kernel_size=(3, 3), padding='same'))
design.add(BatchNormalization())
design.add(Activation('relu'))
design.add(MaxPooling2D(pool_size=(2, 2), padding='same'))
design.add(Dropout(.5))

design.add(Flatten())
design.add(Dense(units=256,activation='relu'))
design.add(Dense(units=128,activation='relu'))
design.add(Dense(units=1,activation='sigmoid'))

design.compile(loss="binary_crossentropy",optimizer="adam",metrics=["accuracy"])
design.summary()
```

CNN model Training

```
history=design.fit(train_X,train_Y,epochs=50,batch_size=64,validation_data=(test_X,test_Y))
```

Save model

design.save(‘/content/drive/Mydrive/Stress_mini_project/project/model/cnn_stress_model.h5’)

Load model

From keras.models import load_model

Loaded_model=load_model(‘/content/drive/Mydrive/Stress_mini_project/project/model /cnn_stress_model.h5’)

Model Testing

Cnn_acc=loaded_model.evaluate(test_X,test_Y)[1]

Print(“CNN Accuracy:”, cnn_acc)

Make prediction from CNN model

```
index=random.randint(0,1999)
plt.imshow(test_X[index,:],cmap='gray')
plt.show()
prediction_s=load_design.predict(test_X[index,:].reshape(1,48,48,1))
print(prediction_s)
#prediction=prediction>0.5
#if prediction==1:
if prediction_s >0.5:
  print("\n Stress Person \n\n")
else:
  print("\n Non-Stress Person \n\n")

# Confusion Matrix
import seaborn as sns
# Compute the confusion matrix
cm = confusion_matrix(test_Y, y_pred, labels=[0, 1])

# Visualize confusion matrix using heatmap
sns.set(font_scale=0.7)
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Predicted labels')
plt.ylabel('True labels')
plt.title('Confusion Matrix of Convolutional neural network')
# Fix x and y axis tick labels
tick_labels = ['No Stress', 'Stress']
plt.xticks(np.arange(len(tick_labels))+0.5, tick_labels)
plt.yticks(np.arange(len(tick_labels))+0.5, tick_labels)
plt.show()
```

Webcam

From Ipython.display import display, javascript, Image

From google.colab.output import eval_js

From base64 import b64decode, b64encode

Import cv2

```
Import numpy as np
Import PIL
Import io
Import html
Import time
Def js_to_image(js_reply):
Image_bytes = b64decode(js_reply.split(‘,’)[1])
Jpg_as_np = np.frombuffer(image_bytes, dtype=np.uint8)
Img=cv2.imdecode(jpg_as_np, flags=1)
Return img
Def bbox_to_bytes(bbox_array):
Bbox_PIL = PIL.image.fromarray(bbox_array,’RGBA’)
Iobuf = io.BytesIO()
Bbox_PIL.save(iobuf, format=’png’)
Bbox_bytes=’{};.format((str(b64encode(iobuf.getvalue()), ‘utf-8’)))
Return bbox_bytes
Face_cascade=cv2.cascadeClassifier(cv2.samples.findFile(cv2.data.harrcascades + ‘haarcascade_frontalface_defalut.xml’))
Def take_photo(filename=’photo.jpg’,quality=0.8):
Data=eval_js(‘takephoto({})’.format(quality))
Img=js_to_image(data)
Gray=cv2.cvtcolor(img,cv2.COLOR_RGBA2GRAY)
Print(gray.shape)
Faces= face_cascade.detectMultiScale(gray)
For(x,y,w,h) in faces:
Img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
Cv2.imwrite(filename,img)
Return filename
Try:
```

```
Filename=take_photo('photo.jpg')
Print('saved to {}'.format(filename))
Display(Image(filename))
Except Exception as err:
Print(str(err))
Import cv2
Frame=cv2.imread("/content/photo.jpg)
Gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
New=gray/255
Import matplotlib.pyplot as plt
Plt.imshow(new, cmap='gray')
Plt.show()
New_resized= cv2.resize(new,(48,48), interpolation=cv2.INTER_LINEAR)
New=np.reshape(new_resized,(1,48,48,1))
prediction=loaded_model.predict(new)
Print(prediction)
Prediction = prediction > 0.5
If prediction > 0.5:
Print("\n Stress person \n\n")
Else:
Print("\n Non-Stress person \n\n")
```

CHAPTER– 6

CONCLUSION

This Project “STRESS DETECTION OF AN IT-EMPLOYEE USING MACHINE LEARNING AND IMAGE PROCESSING Uses Convolutional Neural Network(CNN)”, our system provides efficient and effective result when compared to existing system. Assist organizations in monitoring the stress of their employees and implementing appropriate measures to enhance their well-being and performance. The stress detection system aims to assist IT Employees in determining whether they are experiencing stress or not.