**NAAN MUDHALVAN**

**Project Report**

**“EMOTION DETECTION AND**

**DEPRESSION SCREENING”**

**“MADRAS INSTITUTE OF TECHNOLOGY”**

|  |  |
| --- | --- |
| **NM ID** | **NAME** |
| au2021509316 | GOKUL G |

|  |  |
| --- | --- |
|  |  |
|  | Trainer Name  RAMAR |
|  |  |

**ABSTRACT**

This project aims to develop a data-driven approach for emotion detection and depression screening using machine learning techniques. Emotions play a crucial role in mental health, and detecting signs of depression early can significantly improve outcomes. We employ facial expression recognition and sentiment analysis on textual data to capture emotional states. The project utilizes deep learning algorithms to analyze facial expressions and natural language processing techniques for sentiment analysis. The dataset consists of labeled images and textual data collected from various sources, including social media platforms and clinical databases. Through the integration of these techniques, we aim to build a robust model capable of accurately detecting emotions and screening for depression. Evaluation metrics such as accuracy, precision, recall, and F1 score are used to assess the performance of the developed models. The results demonstrate promising outcomes, indicating the potential for utilizing machine learning in emotion detection and depression screening for early intervention and support in mental health care.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Table of Contents** | **Page No.** |
| 1 | Chapter 1: Introduction | 4 |
| 2 | Chapter 2: Data Processing | 6 |
| 3 | Chapter 3: Algorithm | 7 |
| 4 | Chapter 4: Modeling and Project Outcome | 9 |
| 5 | Conclusion | 18 |
| 6 | Future Scope | 19 |
| 7 | References | 20 |
| 8 | Links | 21 |

**CHAPTER 1**

**INTRODUCTION**

* 1. **Problem Statement**

In today's digital age, mental health concerns, particularly depression, have become increasingly prevalent. Early detection and intervention are crucial for improving outcomes and reducing the burden of depression. However, traditional screening methods often rely on self-reporting, which can be subjective and prone to bias. This project aims to address these challenges by developing a data-driven approach for emotion detection and depression screening using machine learning techniques.

* 1. **Scope**

The scope of this project encompasses the development of a robust machine learning-based system for emotion detection and depression screening. It involves collecting diverse datasets for facial expression recognition and sentiment analysis, followed by the development and optimization of deep learning models for each task. The project aims to integrate the outputs of these models into a unified system for depression screening, with a focus on achieving high accuracy and reliability. Ethical considerations, including privacy protection and informed consent, will be adhered to throughout the project.The final deliverable will be a user-friendly interface, potentially as a web or mobile application, for widespread accessibility. Comprehensive documentation and reporting will outline the methodologies, findings, and implications for mental health care, ensuring transparency and reproducibility

* 1. **AIM AND OBJECTIVE**

Aim:

The aim of this project is to develop a machine learning-based system for emotion detection and depression screening, aiming to provide an effective tool for early intervention and support in mental health care.

Objective:

Develop a machine learning system for emotion detection and depression screening, integrating facial expression recognition and sentiment analysis models for early intervention in mental health care, while ensuring adherence to ethical guidelines and providing a user-friendly interface for accessibility.

* 1. **PROJECT FLOW**

Data Collection: Gather diverse datasets containing labeled images for facial expression recognition and textual data for sentiment analysis. Ensure the datasets cover a wide range of emotions and demographic backgrounds.

AI/ML Setup: Set up the environment for developing and training machine learning models. This includes selecting appropriate deep learning frameworks, preprocessing the collected data, and building the initial versions of facial expression recognition and sentiment analysis models.

**CHAPTER 2**

**DATA PROCESSING**

**2.1 PRE PROCESSING DATA**

Preprocessing Data for Emotion Detection and Depression Screening involves standardizing facial images and textual data, extracting relevant features, balancing classes, splitting data for training and testing, normalizing numerical features, and encoding categorical variables. These steps ensure data consistency, diversity, and suitability for training machine learning models to accurately detect emotions and screen for depression.

Dataset containing the images were downloaded and uploaded in the Google colab file section.

**2.2 FEATURE SELECTION**

* The input of the project is determined and it is an live image that is going to be taken using the webcam.It is going to be taken during the run using the code take\_photo().
* The output is going to be given as bar graph that shows how much percentage of each emotion that image resembles based on this the depression percentage can also be calculated

**Software Requirements**:

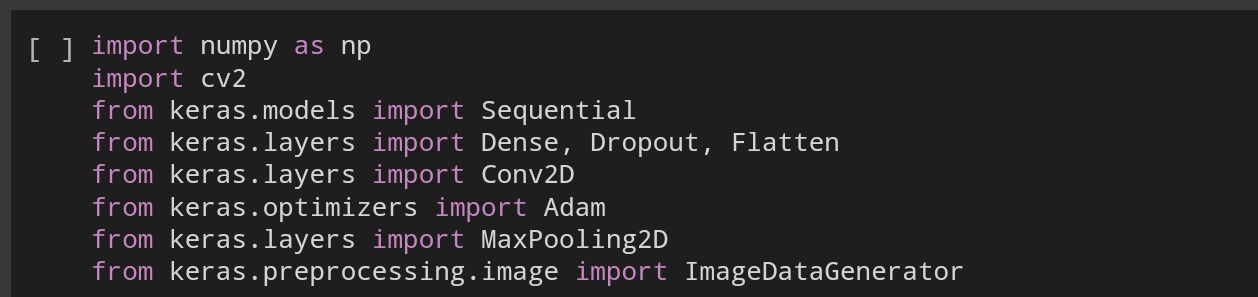
* The project is entirely done in Google colab.

**CHAPTER 3**

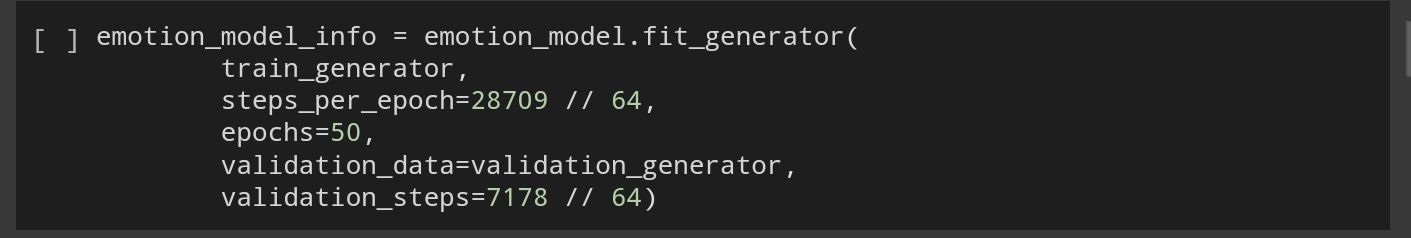
**ALGORITHM**

**3.1 ALGORITHM**

For the Emotion Detection and Depression Screening project, suitable AI/ML algorithms include Convolutional Neural Networks (CNNs) for facial expression recognition, Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks for sentiment analysis of textual data, Gradient Boosting Machines (GBMs) for ensemble learning, Support Vector Machines (SVMs) for binary classification tasks, and Transformer-based architectures for capturing contextual information effectively in sentiment analysis. The choice depends on factors like data nature, task complexity, available resources, and desired performance metrics, and experimentation may be needed to determine the best approach.

****3.2 TRAINING MODEL

The model is trained to recognize the emotions of the the picture taken.The more the algorithm trains the accuracy of the result will increase.Here the algorithm is trained for 50 iterations.



**CHAPTER 4**

**MODELING AND PROJECT OUTCOME**

**MODELLING**

The entire code of the project is given below

from zipfile import ZipFile

file\_name = "archive.zip"

with ZipFile(file\_name, 'r') as zip:

  zip.extractall()

  print("Done")

import numpy as np

import cv2

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D

from keras.optimizers import Adam

from keras.layers import MaxPooling2D

from keras.preprocessing.image import ImageDataGenerator

train\_dir = 'train'

val\_dir = 'test'

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=64,

        color\_mode="grayscale",

        class\_mode='categorical')

validation\_generator = val\_datagen.flow\_from\_directory(

        val\_dir,

        target\_size=(48,48),

        batch\_size=64,

        color\_mode="grayscale",

        class\_mode='categorical')

emotion\_model = Sequential()

emotion\_model.add(Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=(48,48,1)))

emotion\_model.add(Conv2D(64, kernel\_size=(3, 3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))

emotion\_model.add(Dropout(0.25))

emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))

emotion\_model.add(Conv2D(128, kernel\_size=(3, 3), activation='relu'))

emotion\_model.add(MaxPooling2D(pool\_size=(2, 2)))

emotion\_model.add(Dropout(0.25))

emotion\_model.add(Flatten())

emotion\_model.add(Dense(1024, activation='relu'))

emotion\_model.add(Dropout(0.5))

emotion\_model.add(Dense(7, activation='softmax'))

!pip install tensorflow

!pip install keras

emotion\_model.compile(loss='categorical\_crossentropy',optimizer=Adam(learning\_rate=0.0001),metrics=['accuracy'])

emotion\_model\_info = emotion\_model.fit\_generator(

        train\_generator,

        steps\_per\_epoch=28709 // 64,

        epochs=50,

        validation\_data=validation\_generator,

        validation\_steps=7178 // 64)

#Saving the model

emotion\_model.save('model.h5')

from keras.models import load\_model

emotion\_model = load\_model('model.h5')

def emotion\_analysis(emotions):

    objects = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')

    y\_pos = np.arange(len(objects))

    plt.bar(y\_pos, emotions, align='center', alpha=0.5)

    plt.xticks(y\_pos, objects)

    plt.ylabel('percentage')

    plt.title('emotion')

    plt.show()

#CODE for Capturing an image on Colab from here: https://colab.research.google.com/notebook#fileId=1OnUy6eFE7XhdfGfAHDCqQxpwueTOj\_NO

from IPython.display import display, Javascript

from google.colab.output import eval\_js

from base64 import b64decode

def take\_photo(filename='photo.jpg', quality=0.8):

  js = Javascript('''

    async function takePhoto(quality) {

      const div = document.createElement('div');

      const capture = document.createElement('button');

      capture.textContent = 'Capture';

      div.appendChild(capture);

      const video = document.createElement('video');

      video.style.display = 'block';

      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);

      div.appendChild(video);

      video.srcObject = stream;

      await video.play();

      // Resize the output to fit the video element.

      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.

      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');

      canvas.width = video.videoWidth;

      canvas.height = video.videoHeight;

      canvas.getContext('2d').drawImage(video, 0, 0);

      stream.getVideoTracks()[0].stop();

      div.remove();

      return canvas.toDataURL('image/jpeg', quality);

    }

    ''')

  display(js)

  data = eval\_js('takePhoto({})'.format(quality))

  binary = b64decode(data.split(',')[1])

  with open(filename, 'wb') as f:

    f.write(binary)

  return filename

from IPython.display import Image

try:

  filename = take\_photo()

  print('Saved to {}'.format(filename))

  # Show the image which was just taken.

  display(Image(filename))

except Exception as err:

  # Errors will be thrown if the user does not have a webcam or if they do not

  # grant the page permission to access it.

  print(str(err))

#CODE for Capturing an image on Colab from here: https://colab.research.google.com/notebook#fileId=1OnUy6eFE7XhdfGfAHDCqQxpwueTOj\_NO

from IPython.display import display, Javascript

from google.colab.output import eval\_js

from base64 import b64decode

def take\_photo(filename='photo.jpg', quality=0.8):

  js = Javascript('''

    async function takePhoto(quality) {

      const div = document.createElement('div');

      const capture = document.createElement('button');

      capture.textContent = 'Capture';

      div.appendChild(capture);

      const video = document.createElement('video');

      video.style.display = 'block';

      const stream = await navigator.mediaDevices.getUserMedia({video: true});

      document.body.appendChild(div);

      div.appendChild(video);

      video.srcObject = stream;

      await video.play();

      // Resize the output to fit the video element.

      google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

      // Wait for Capture to be clicked.

      await new Promise((resolve) => capture.onclick = resolve);

      const canvas = document.createElement('canvas');

      canvas.width = video.videoWidth;

      canvas.height = video.videoHeight;

      canvas.getContext('2d').drawImage(video, 0, 0);

      stream.getVideoTracks()[0].stop();

      div.remove();

      return canvas.toDataURL('image/jpeg', quality);

    }

    ''')

  display(js)

  data = eval\_js('takePhoto({})'.format(quality))

  binary = b64decode(data.split(',')[1])

  with open(filename, 'wb') as f:

    f.write(binary)

  return filename

import cv2

def facecrop(image):

    facedata = '/content/haarcascade\_frontalface\_alt.xml'

    cascade = cv2.CascadeClassifier(facedata)

    img = cv2.imread(image)

    try:

        minisize = (img.shape[1],img.shape[0])

        miniframe = cv2.resize(img, minisize)

        faces = cascade.detectMultiScale(miniframe)

        for f in faces:

            x, y, w, h = [ v for v in f ]

            cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

            sub\_face = img[y:y+h, x:x+w]

            cv2.imwrite('photo.jpg', sub\_face)

            #print ("Writing: " + image)

    except Exception as e:

        print (e)

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

    facecrop('photo.jpg')

#Testing a file.

from keras.preprocessing import image

from keras.preprocessing.image import ImageDataGenerator

import numpy as np

import matplotlib.pyplot as plt

file = 'photo.jpg'

true\_image = image.load\_img(file)

img = image.load\_img(file, color\_mode="grayscale", target\_size=(48, 48))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis = 0)

x /= 255

custom = emotion\_model.predict(x)

emotion\_analysis(custom[0])

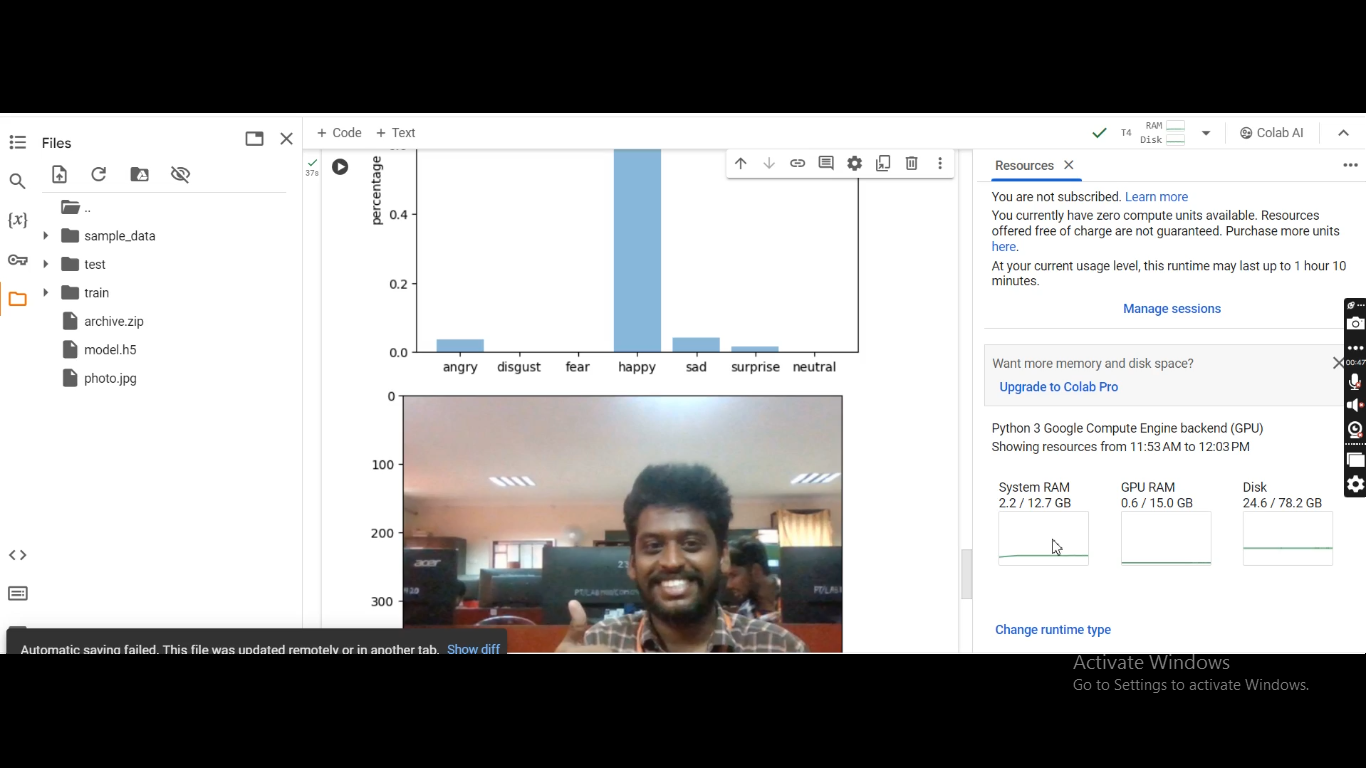
x = np.array(x, 'float32')

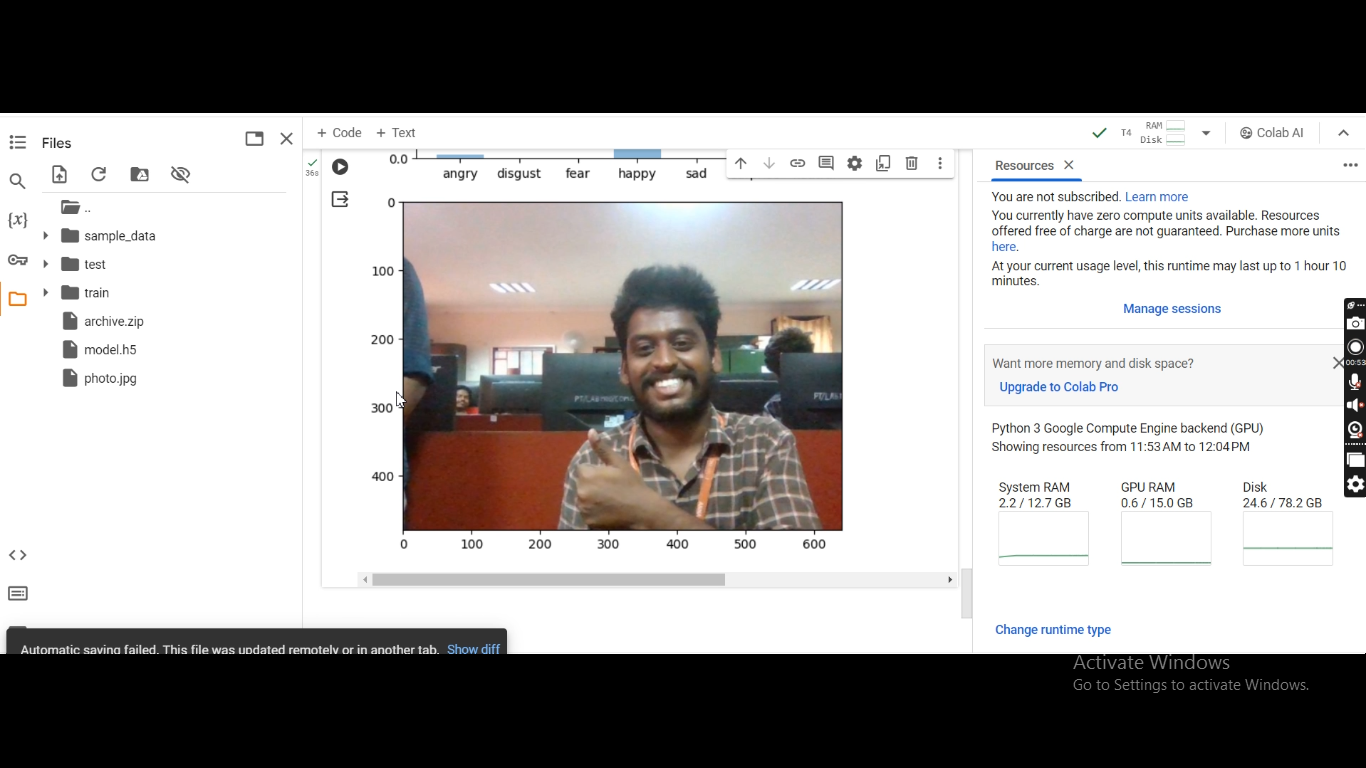
x = x.reshape([48, 48]);

plt.imshow(true\_image)

plt.show()

**OUTPUT**

****



**CONCLUSION:**

In conclusion, the integration of artificial intelligence (AI) in emotion detection and depression screening presents significant opportunities for improving mental health care. Through advanced algorithms and machine learning techniques, AI can analyze various data sources, including text, speech, and facial expressions, to identify patterns indicative of emotional states and depressive symptoms.

This technology offers several advantages, including scalability, accessibility, and efficiency, allowing for early intervention and personalized treatment strategies. However, it's crucial to address ethical considerations, such as privacy concerns and potential biases in AI algorithms. Collaborative efforts between researchers, healthcare professionals, and technology developers are essential to ensure the responsible and effective implementation of AI in mental health care.

Overall, AI-based emotion detection and depression screening hold promise in revolutionizing mental health assessment and support, potentially reducing the burden of undiagnosed and untreated depression and improving the overall well-being of individuals worldwide.

Top of Form

**FUTURE SCOPE:**

The future scope of emotion detection and depression screening using artificial intelligence is promising and multifaceted.

Firstly, advancements in AI technology will likely lead to more accurate and sophisticated algorithms capable of detecting subtle emotional cues and nuanced expressions indicative of depression. These improvements will enhance the reliability and effectiveness of screening tools, enabling earlier and more precise identification of individuals at risk.

Secondly, the integration of AI with wearable devices and smartphone applications could facilitate continuous monitoring of individuals' emotional states in real-time, providing valuable insights into their mental health status outside clinical settings. This proactive approach could enable timely interventions and personalized support tailored to individuals' needs, ultimately improving treatment outcomes and quality of life.

Furthermore, AI-powered virtual assistants and chatbots have the potential to offer accessible and confidential support to individuals experiencing depressive symptoms, providing immediate guidance, resources, and coping strategies. These virtual interventions could complement traditional therapy and extend mental health care to underserved populations, such as those in remote areas or with limited access to mental health services.

Moreover, the use of AI in large-scale data analysis could help identify population-level trends and risk factors associated with depression, informing public health policies and intervention strategies aimed at reducing the prevalence and impact of mental illness on society.

Overall, the future of emotion detection and depression screening with AI holds promise for revolutionizing mental health care by enhancing early detection, personalized intervention, and population-level prevention efforts, ultimately contributing to improved mental well-being on a global scale.

**REFERENCES:**

1. Project Github link, Ramar Bose , 2024
2. Project video recorded link (youtube/github), Ramar Bose , 2024
3. Project PPT & Report github link, Ramar Bose , 2024

# **GIT Hub Link of Project Code:**

https://github.com/au2021509316/Gokul-G