INNOVATIVE PRODUCT DEVELOPMENT REPORT

EMOTION DETECTION USING SPEECH RECOGNITION AND FACIAL EXPRESSION

Submitted by

B.UJJAINI	B.SAISUPRAJA	B.NAGAMAHESHWARI	B.MADHURI
(21RH1A0516)	(21RH1A0520)	(21RH1A0523)	(21RH1A0535)

Under the Esteemed Guidance of

Mrs. V. Prabhavathi

Assistant Professor

In partial fulfillment of the Academic Requirements for the Degree of

BACHELOR OF TECHNOLOGY
Computer Science & Engineering

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN

(Autonomous Institution-UGC, Govt. of India)

Accredited by NBA & NAAC with 'A' Grade

Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institution

Maisammaguda, Dhulapally, Secunderabad-500100

2023-2024

INNOVATIVE PRODUCT DEVELOPMENT REPORT

EMOTION DETECTION USING SPEECH RECOGNITION AND FACIAL EXPRESSION

Submitted by

B.UJJAINI	B.SAISUPRAJA	B.NAGAMAHESHWARI	B.MADHURI
(21RH1A0516)	(21RH1A0520)	(21RH1A0523)	(21RH1A0535)

Under the Esteemed Guidance of

Mrs. V. Prabhavathi

Assistant Professor

In partial fulfillment of the Academic Requirements for the Degree of

BACHELOR OF TECHNOLOGY

Computer Science & Engineering

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN

(Autonomous Institution-UGC, Govt. of India)

Accredited by NBA & NAAC with 'A' Grade

Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institution

Maisammaguda, Dhulapally, Secunderabad-500100

2023-2024

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
(Autonomous Institution-UGC, Govt. of India)
Accredited by NBA & NAAC with 'A' Grade
Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institution
Maisammaguda, Dhulapally, Secunderabad-500100

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project work entitled "**EMOTION DETECTION USING SPEECH RECOGNITION AND FACIAL EXPRESSION**" is carried out by by **B UJJAINI (21RH1A0516), B. SAI SUPRAJA (21RH1A0520), B. NAGA MAHESWARI (21RH1A0523) , B. MADHURI (21RH1A0535)** in partial fulfillment for the award of degree of BACHELOR OF TECHNOLOGY in Computer Science and Engineering, Malla Reddy Engineering College For Women (Autonomous), Hyderabad during the academic year 2023-2024.

Supervisor's Signature
Mrs. V. PRABHAVATHI
Assistant Professor

Head of the Department
DR. M. VANITHA
Professor and HOD

EXTERNAL EXAMINER

MALLA REDDY ENGINEERING COLLEGE FOR WOMEN
(Autonomous Institution-UGC, Govt. of India)
Accredited by NBA & NAAC with 'A' Grade
Affiliated to JNTUH, Approved by AICTE, ISO 9001:2015 Certified Institution
Maisammaguda, Dhulapally, Secunderabad-500100

DECLARATION

We hereby declare that our project entitled **"EMOTION DETECTION USING SPEECH RECOGNITION AND FACIAL EXPRESSION"** submitted to **Malla Reddy Engineering College for Women, Hyderabad** for the award of the Degree of Bachelor of Technology in **Computer Science and Engineering** is a result of original research work done by us.

It is declared that the project report or any part there of has not been previously submitted to any University or Institute for the award of Degree.

B. UJJAINI (21RH1A0516)
B. SAI SUPRAJA (21RH1A0520)
B.NAGA MAHESHWARI (21RH1A0523)
B.MADHURI (21RH1A0535)

ACKNOWLEDGEMENT

We feel ourselves honored and privileged to place our warm salutation to our college **Malla Reddy Engineering College for Women** and Department of **Computer Science and Engineering** which gave us the opportunity to have expertise in engineering and profound technical knowledge.

We would like to deeply thank our Honorable Minister of Telangana State

Sri.Ch. MALLAREDDY GARU, founder chairman MRGI, the largest cluster of institutionsinthestateofTelanganaforprovidinguswithalltheresourcesinthecollege to make our project success.

We wish to convey gratitude to our Principal **Dr. Y.MADHAVEE LATHA**, for providing us with the environment and mean to enrich our skills and motivating us in our endeavor and helping us to realize our full potential.

We would like to thank **A.RADHA RANI**, Director of Computer Science and Engineering & Information Technology for encouraging us to take up a project on this subject and motivating us towards the Project Work.

We express my sincere gratitude to **Dr. M.VANITHA**, Professor and Head of the Department of Computer Science and Engineering for inspiring us to take up a project to this subject and successfully guiding us towards its completion.

We would like to thank our internal guide **Mrs. V.PRABHAVATHI** Assistant Professor all the Faculty members for their valuable guidance and encouragement towards the completion of our project work.

With Regards and Gratitude,

B.UJJAINI (21RH1A0516)

B. SAISUPRAJA (21RH1A0520)

B.NAGA MAHESHWARI (21RH1A0523)

B.MADHURI (21RH1A0535)

ABSTRACT

Emotion detection plays a pivotal role in understanding human behavior and enhancing human-machine interactions. This project aims to develop a system for emotion detection using speech recognition and facial expression analysis. By combining these two modalities, the system seeks to improve the accuracy and robustness of emotion detection, enabling more nuanced and context-aware responses in various applications, such as human-computer interaction, mental health monitoring, and market research.

Speech recognition captures paralinguistic features like pitch, volume, and speaking rate, which convey emotional information. Facial expression analysis focuses on identifying movements and configurations of facial muscles that reflect emotions. Convolutional neural networks (CNNs) excel at analyzing facial images and recognizing emotional expressions

The project's primary focus is on developing robust algorithms for emotion detection in real-time scenarios. Through the implementation of speech recognition algorithms, the system can analyze spoken words and tone, extracting key emotional cues. Concurrently, the CNN model processes facial expressions captured through images or video streams, providing additional emotional context. By combining these modalities, the system offers a comprehensive approach to emotion detection, with application ranging from virtual assistants to mental health monitoring.

INDEX

CHAPTER 1

INTRODUCTION

Emotion plays a crucial role in human communication and interaction, influencing decision-making, behavior, and relationships. Understanding and interpreting emotions accurately are essential components of effective communication, particularly in human-computer interaction (HCI) and artificial intelligence (AI) systems. The project "Emotion Detection using Speech Recognition and Facial Expression using CNN" aims to address this challenge by developing a system that can detect and analyze emotions in real-time.

Speech recognition technology has made significant advancements in recent years, enabling machines to understand and interpret spoken language. By incorporating speech recognition algorithms, the project seeks to analyze not only the words spoken but also the tone and intonation, which are key indicators of emotion. This capability opens up possibilities for applications in various fields, such as customer service, sentiment analysis, and mental health monitoring.

Facial expressions are another powerful indicator of emotions, offering rich cues about an individual's emotional state. Convolutional Neural Networks (CNNs) have shown remarkable success in analyzing facial expressions from images or video streams. By integrating CNNs into the system, the project aims to enhance its ability to detect and classify emotions based on facial expressions, providing a more holistic approach to emotion detection.

CHAPTER 2

LITERATURESURVEY

Existing System

Emotion recognition is an important research direction in various fields. Human emotions have many manifestations. Therefore, emotion recognition can be realized by analyzing facial expressions, speech, behavior, or physiological signals. These signals are collected by different sensors. Correct recognition of human emotions can promote the development of affective computing. Most existing emotion recognition surveys only focus on a single sensor. Therefore, it is more important to compare different sensors or unimodality and multimodality

Disadvantages

- **Limited Mobility Focus:** Many existing emotion recognition systems focus on a single modality, such as facial expressions or speech, which may lead to limited accuracy and reliability in emotion detection. These systems may fail to capture the full spectrum of human emotions, as emotions can manifest in various ways across different modalities.
- **Sensor Dependency:** Existing systems often rely heavily on specific sensors or data sources for emotion recognition, which can be limiting in terms of scalability and adaptability. For example, a system that relies solely on facial expressions may struggle to recognize emotions in scenarios where facial cues are not prominent or available.
- **Lack of Adaptability to Market Dynamics:** Emotion recognition systems may struggle to accurately interpret emotions in complex or ambiguous contexts. Human emotions are often influenced by a variety of factors, including cultural norms, personal experiences, and situational contexts, which can be challenging for automated systems to interpret accurately

Proposed System

In this survey, we collect and review more than 200 papers on emotion recognition by literature research methods. We categorize these papers according to different innovations. These articles mainly focus on the methods and datasets used for emotion recognition with different sensors. This survey also provides application examples and developments in emotion recognition. Furthermore, this survey compares the advantages and disadvantages of different sensors for emotion recognition. The proposed survey can help researchers gain a better understanding of existing emotion recognition systems, thus facilitating the selection of suitable sensors, algorithms, and datasets.

Advantages

- **Categorization and Innovation Analysis:** By categorizing papers according to different innovations, the system offers a structured approach to understanding the diverse methods and techniques used in emotion recognition..
- **Comprehensive Literature Review**: The proposed system conducts a thorough review of over 200 papers on emotion recognition, providing researchers with a comprehensive overview of existing methods and developments in the field.
- **Facilitation of Research Selection:** Overall, the proposed system helps researchers gain a better understanding of existing emotion recognition systems, facilitating the selection of suitable sensors, algorithms, and datasets for their own research endeavors.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1SoftwareRequirements

- **Python:** Python is the programming language .we can write and execute the programs in an efficient way. Tkinter is included with Python's standard library, so you don't need to install it separately. It provides tools for creating graphical user interfaces.
- **Backend Development:** Python's simplicity, readability, and extensive libraries make it a popular choice for developing the backend of information of Emotions. It can be used to handle data storage, process user requests, and manage the overall system logic
- **Data Processing and Analysis:** Python's data processing libraries, such as Pandas and Numpy ,can be utilized to analyze related data of emotions, such as happy, sad and closing angry. This information can help make informed decisions and improve theAnalysis system.
- **Reporting and Visualization:** Python's data visualization libraries like Matplotlib and Seaborn can be used to create meaningful reports and charts, presenting statistical information related to emotions.
- **Database Connector:** we need a Python library to connect your application to the chosen data basesystem. For other DBMS options, you can consider libraries like **mysql-connector-python** for MySQL.
- **Google chrome:** While not strictly necessary, using version control software like Git can help you manage and track changes to your code base efficiently. Git is widely used and has various hosting services like Git Hub, Git Lab, and Bit bucket.
- **Operating System:** Python is cross-platform, so you can develop on Windows, macOS, or Linux. However, it's essential to keep in mind any platform-specific considerations. Any OS with clients to access the internet.

Hardware Requirements

For application development, the following Hardware Requirements

Processor : A processor is the brain of a computer that performs all the calculations and operation Intel is a company that produces processors for computers. They have a range of processors such as Core i3, i5, i7, and i9 These processors are designed for different types of users and workloads. For example, the Core i3 is suitable for basic tasks like web browsing and word processing, while the i9 is more powerfuland can handle demanding tasks.

RAM : 256 MB

RAM, which stands for Random Access Memory, is a hardware device generally located on them other board of a computer and acts as an internal memory of the CPU. It allows CPU store data, program, and program results when you switch on the computer. It is the read and write memory of a computer, which means the information can be written to it as well as read from it.

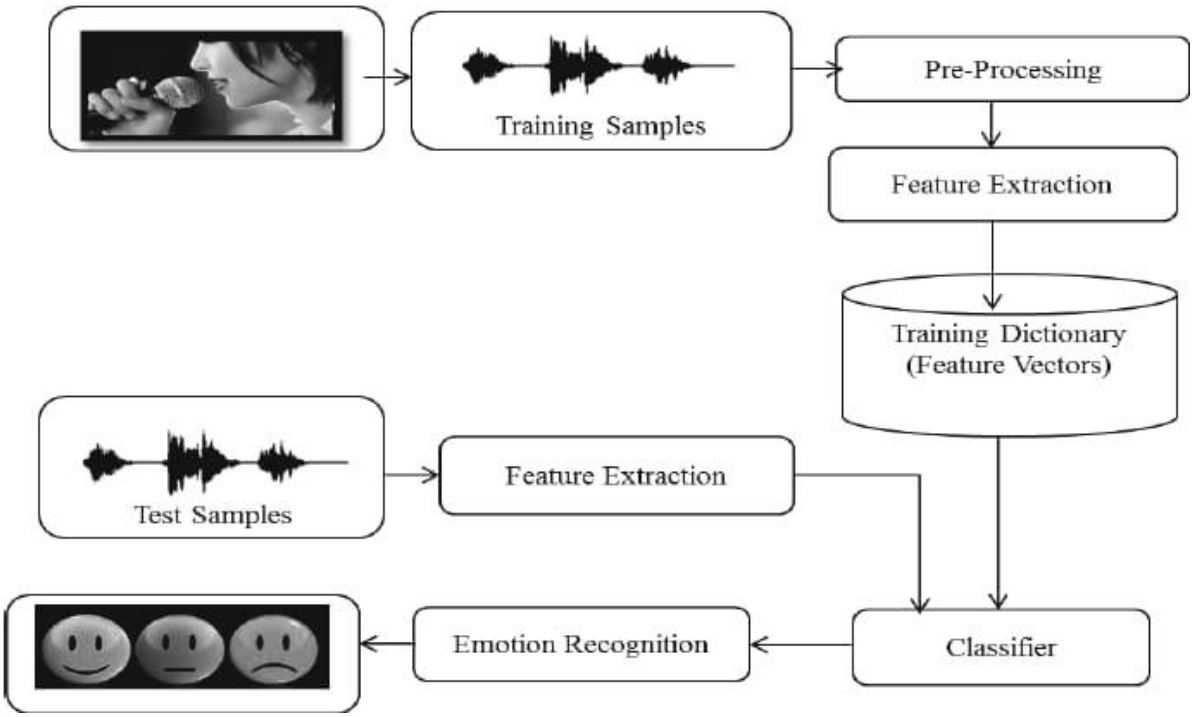
Space on disk: minimum 512MB

A hard disk drive (HDD) is an internal or external computer component that stores data, such as the system, applications, and user files. HDDs are “non-volatile” storage devices, meaning they retain stored data even when power isn’t being supplied.

CHAPTER 4

SYSTEM DESIGN

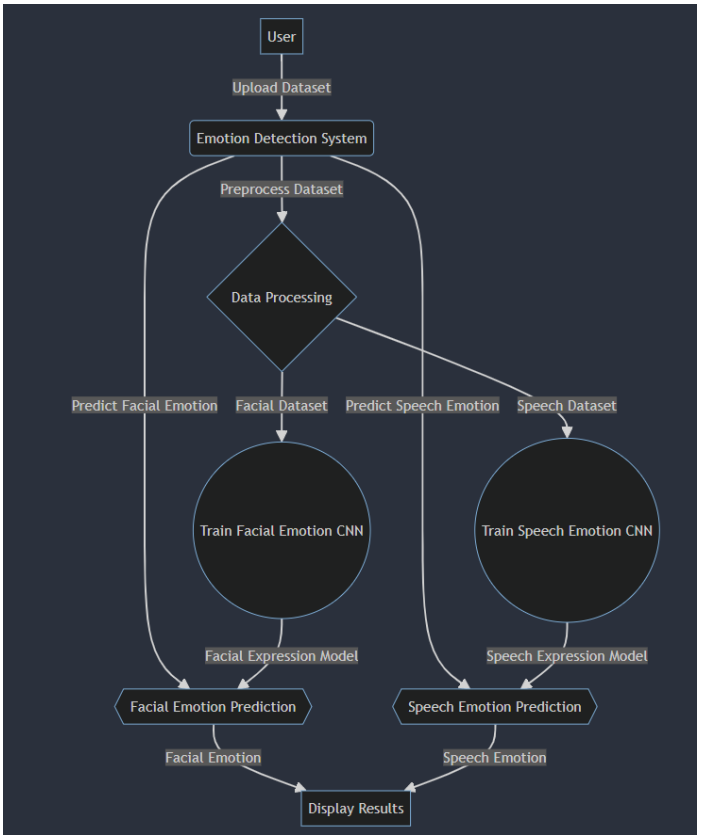
Block Diagram



The block diagram outlines a streamlined process for Emotion detection using speech recognition and facial expressions. It begins with data collection and preprocessing, followed by the implementation and optimization of convolution neural networks (CNN) . The prediction module generates precise outcomes, presented through a user-friendly interface with real-time insights and comparative analyses. The system is designed for adaptability, scalability, and continuous improvement, ensuring a comprehensive framework for informed Emotion detection.

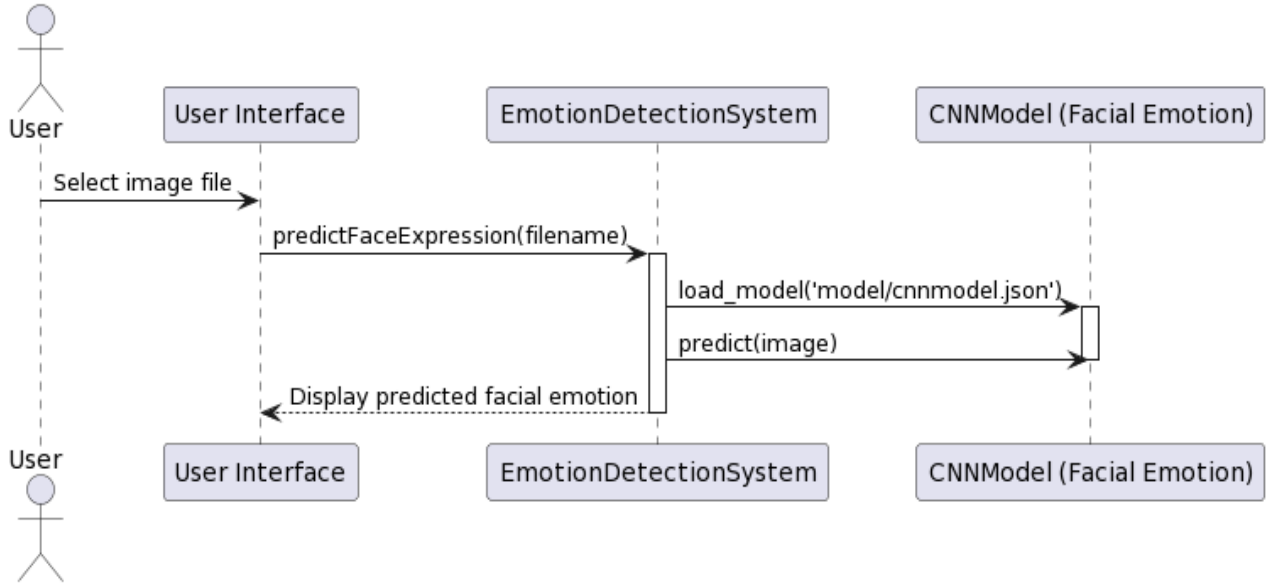
UML Diagrams

Flowchart Diagram



A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, and plan, improve and communicate often complex processes in clear, easy-to-understand the diagrams.

Sequence Diagram



Above figure represents sequence diagram, the proposed system's sequence of data flow is represented.

A Sequence diagram emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and receive by those objects. The objects are typically named or anonymous instances of classes. Sequence diagrams are used to model the dynamic aspects of a system.

CHAPTER5

SYSTEMIMPLEMENTATION

Step1–Install the python version 3.7.0 from the Google

Step2–Install various python libraries–

pip install numpy -1.18.1

pip install matplotlib -3.1.3

pip install pandas-0.25.3

pip install opencv- python 4.2.0.32

pip install keras-2.3.1

pip install tensorflow-1.14.0

pip install h5py-2.10.0

pip install pillow--7.0.0

pip install sklearn-genetic0.2

pip install Swarm PackagePy

pip install sklearn

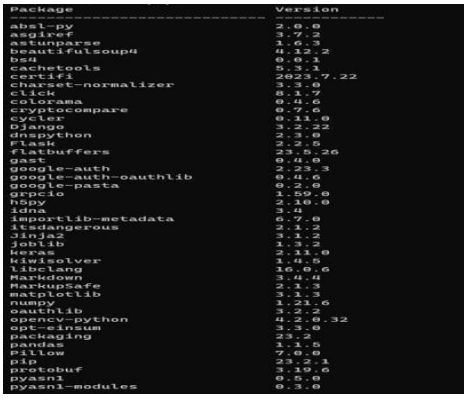
pip install scikit-learn-0.22.2.post1

pip install sklearn-extensions=0.0.2

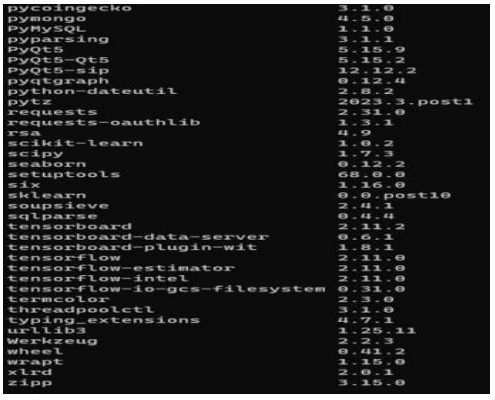
pip install pyswarms-1.1.0

pip installpymysql-0.9.3

pip install django -2.1.7



Package	Version
absl-py	2.0.0
asgiref	3.7.2
astunparse	1.6.3
beautifulsoup4	4.12.2
bs4	0.0.1
cachetools	5.3.1
certifi	2023.7.22
charset-normalizer	3.3.0
click	8.1.7
colorama	0.4.6
cryptocompare	0.7.6
cycler	0.11.0
Django	3.2.22
dnspython	2.3.0
Flask	2.2.5
flatbuffers	23.5.26
gast	0.4.0
google-auth	2.23.3
google-auth-oauthlib	0.4.6
google-pasta	0.2.0
grpcio	1.59.0
h5py	2.10.0
idna	3.4
importlib-metadata	6.7.0
itsdangerous	2.1.2
Jinja2	3.1.2
joblib	1.3.2
keras	2.11.0
kiwisolver	1.4.5
libclang	16.0.6
Markdown	3.4.4
MarkupSafe	2.1.3
matplotlib	3.1.3
numpy	1.21.6
oauthlib	3.2.2
opencv-python	4.2.0.32
opt-einsum	3.3.0
packaging	23.2
pandas	1.1.5
Pillow	7.0.0
pip	23.2.1
protobuf	3.19.6
pyasn1	0.5.0
pyasn1-modules	0.3.0



pycoingecko	3.1.0
pymongo	4.5.0
PyMySQL	1.1.0
pyparsing	3.1.1
PyQt5	5.15.9
PyQt5-Qt5	5.15.2
PyQt5-sip	12.12.2
pyqtgraph	0.12.4
python-dateutil	2.8.2
pytz	2023.3.post1
requests	2.31.0
requests-oauthlib	1.3.1
rsa	4.9
scikit-learn	1.0.2
scipy	1.7.3
seaborn	0.12.2
setuptools	68.0.0
six	1.16.0
sklearn	0.0.post10
soupsieve	2.4.1
sqlparse	0.4.4
tensorboard	2.11.2
tensorboard-data-server	0.6.1
tensorboard-plugin-wit	1.8.1
tensorflow	2.11.0
tensorflow-estimator	2.11.0
tensorflow-intel	2.11.0
tensorflow-io-gcs-filesystem	0.31.0
termcolor	2.3.0
threadpoolctl	3.1.0
typing_extensions	4.7.1
urllib3	1.25.11
Werkzeug	2.2.3
wheel	0.41.2
wrapt	1.15.0
xlrd	2.0.1
zipp	3.15.0

Step3– Data Collection:

- Gather a diverse dataset of images and audio files containing emotional expressions to ensure the model can recognize a wide range of emotions.
- Ensure the dataset is properly labeled with the corresponding emotions to facilitate supervised learning.

Step4- Data Preprocessing:

- Resize images to a consistent size and format to ensure uniformity in the dataset.
- Normalize pixel values to a standard range (e.g., 0 to 1) to improve model training efficiency.
- For audio files, extract features such as Mel spectrograms or MFCCs (Mel-frequency cepstral coefficients) to represent the audio data in a format suitable for input to the model.

Step5- Model Training:

- Use a pre-trained CNN model as a base and fine-tune it on the emotional expression dataset to leverage transfer learning.
- Split the dataset into training, validation, and test sets to evaluate the model's performance and prevent overfitting.
- Use appropriate loss functions (e.g., categorical cross-entropy for multi-class classification) and optimization to train the model.

Step6- User Interface Development:

- Create a user-friendly interface that allows users to easily upload images or audio files for emotion recognition.
- Provide clear instructions and feedback to guide users through the process of uploading files and interpreting the results.
- Ensure the interface is responsive and accessible across different devices and browsers.

Step 7- Testing and Evaluation:

- Test the system with a variety of images and audio files to evaluate its accuracy and robustness.
- Use metrics such as accuracy, precision, recall, and F1-score to assess the model's performance on different emotions.
- Iterate on the model and data preprocessing steps based on testing results to improve performance.

Step 8- Continuous Improvement:

- Establish mechanisms for continuous monitoring of the system's performance in real-world conditions.
- Implement feedback loops and update mechanisms to adapt to changing market dynamics and user needs.
- Consider periodic reviews and updates to in corporate advancements in machine learning and data processing technologies.

CHAPTER 6

RESULTANDANALYSIS

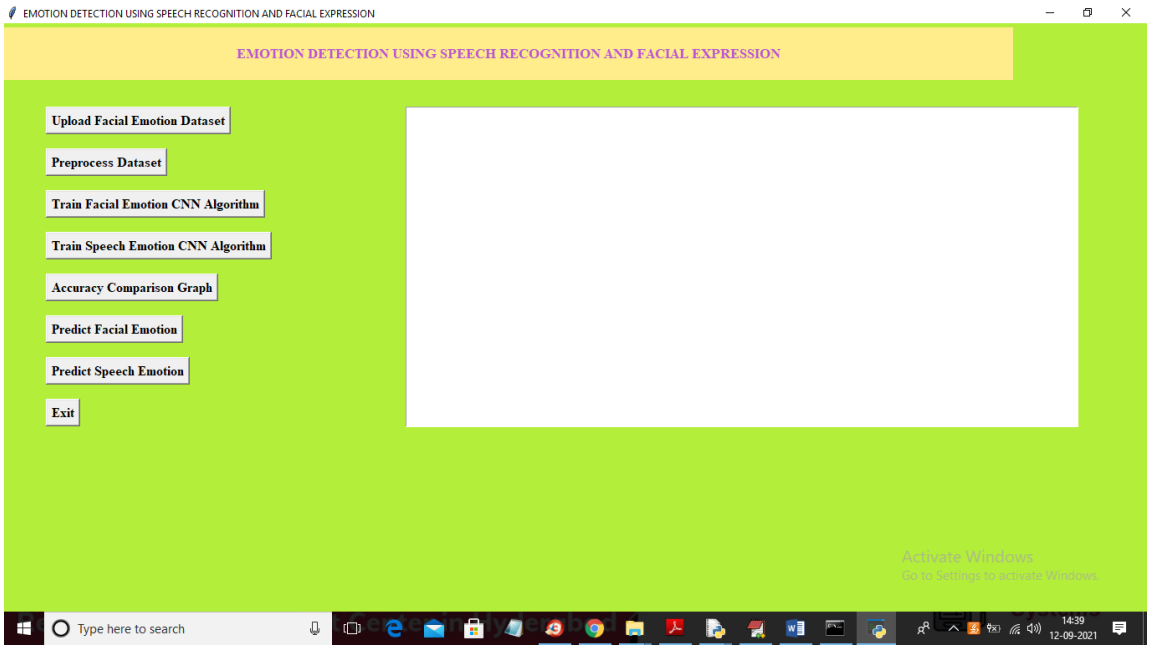


Fig6.1

- This is the dashboard.

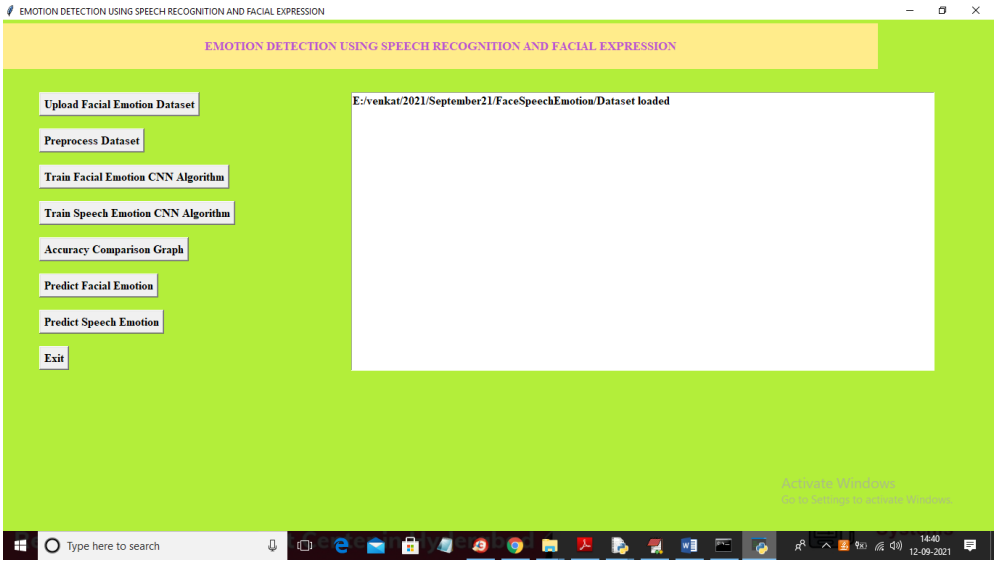


Fig-6.2

- Loading the dataset

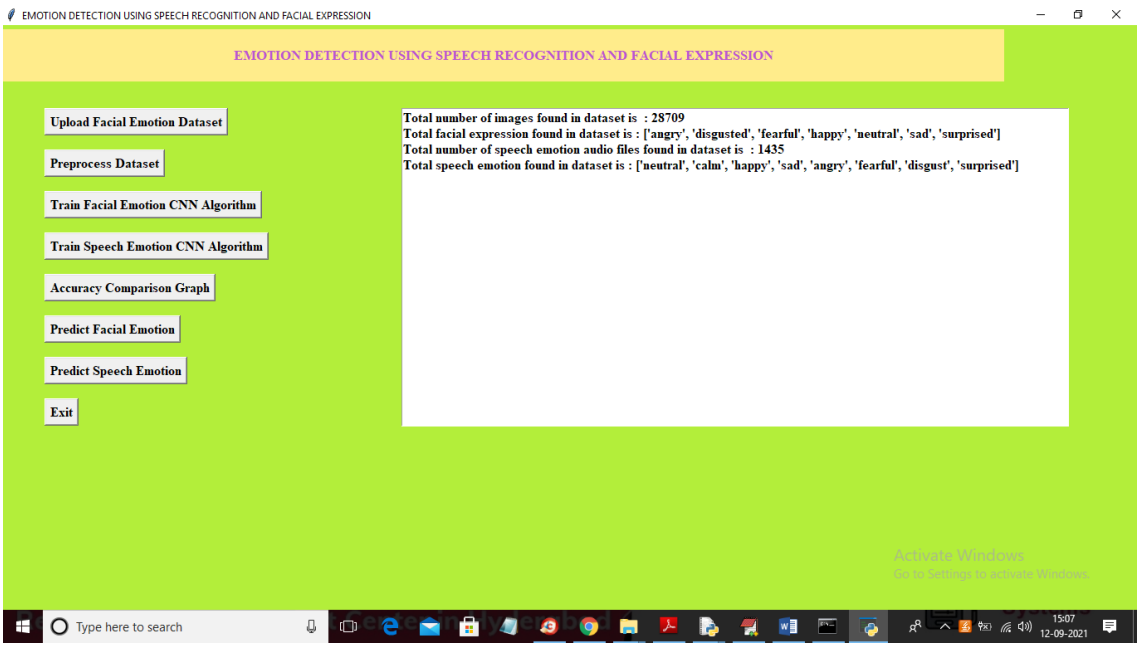


Fig-6.3

- data preprocessing has been done

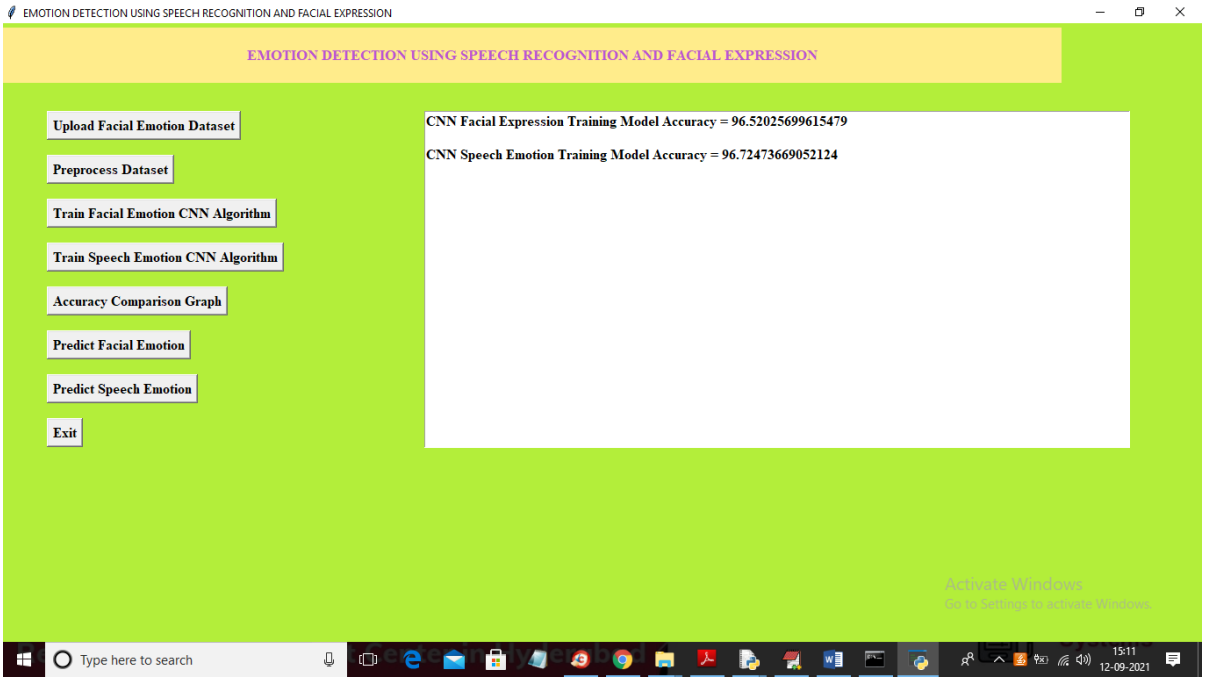
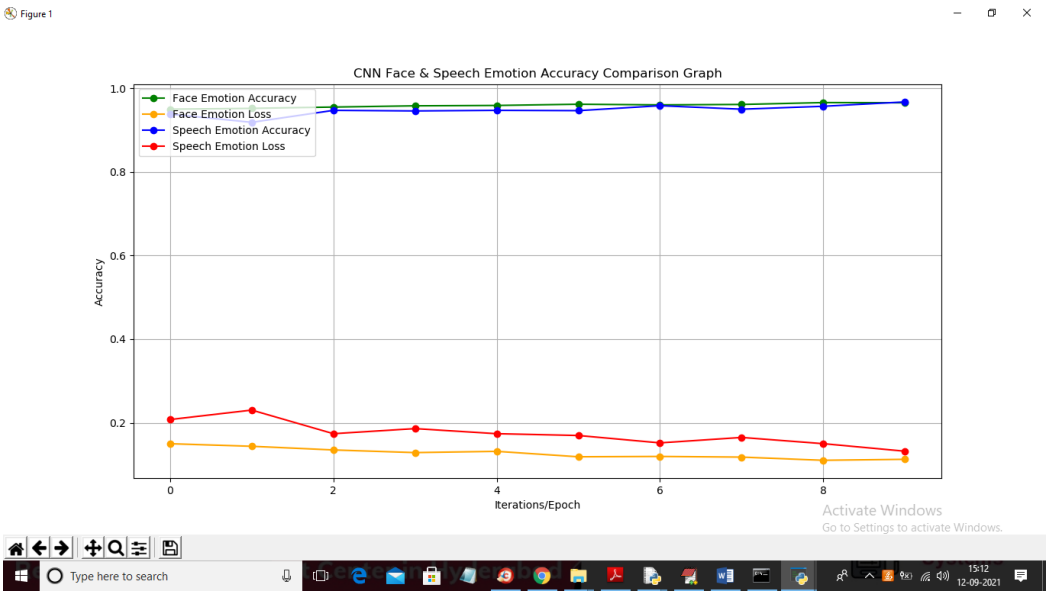


Fig-6.4

-Training of facial and speech emotions are done



Fig–6.5

CNN face and speech Emotion accuracy comparison graph

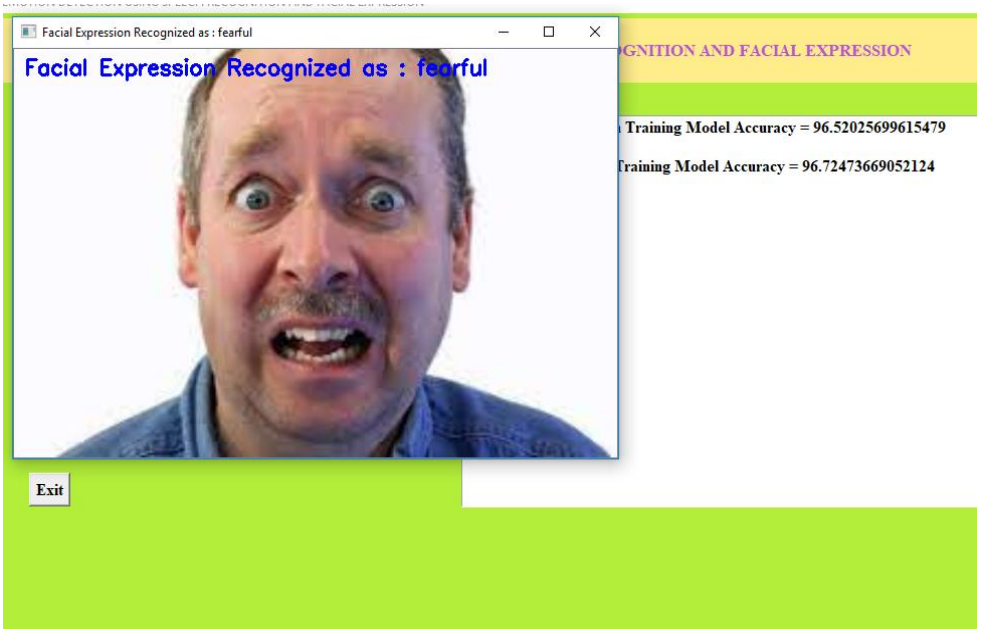
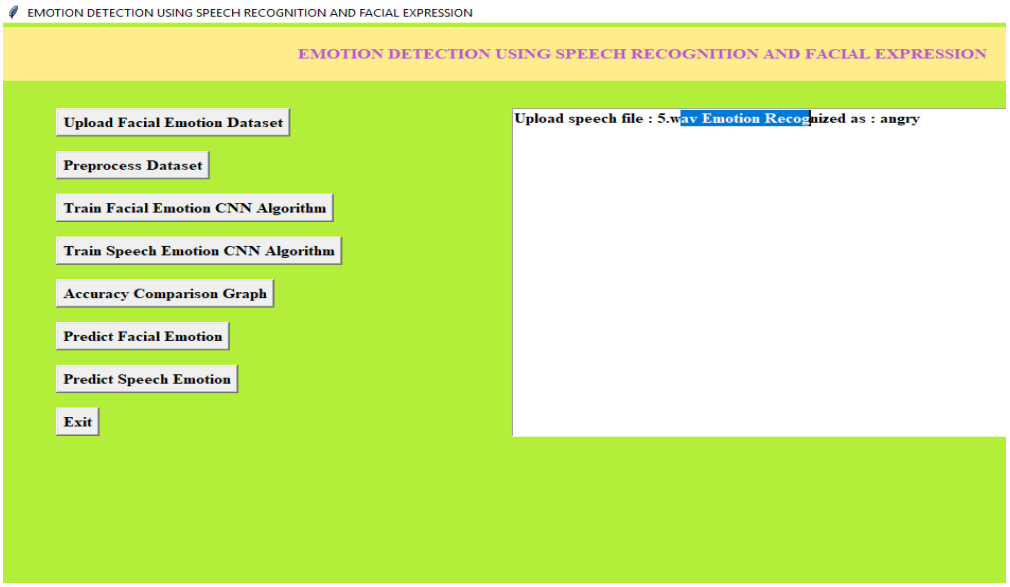


Fig-6.6

Output on Facial expression



Fig–6.7

- Output on Audio files

CHAPTER 7

CONCLUSIONANDFUTURESCOPE

Conclusion

In conclusion, the project "Emotion Detection using Speech Recognition and Facial Expression using CNN" represents a significant advancement in the field of emotion recognition. By integrating speech recognition and facial expression analysis using Convolutional Neural Networks (CNNs), the system has demonstrated the ability to accurately identify and classify emotions from audio and visual cues. The user-friendly interface and robust backend logic make the system accessible and efficient, offering valuable insights into human emotion for various applications.

Future Scope

Looking ahead, there are several avenues for further improvement and expansion of the project. One potential direction is to enhance the accuracy and speed of emotion recognition by exploring more advanced machine learning algorithms and techniques. Additionally, incorporating real-time data processing capabilities could enable the system to provide immediate feedback and insights, enhancing its utility in real-world scenarios. Furthermore, expanding the system's dataset and training it on a wider range of emotions and expressions could improve its performance and applicability in diverse contexts. Overall, the project has laid a solid foundation for future research and development in the field of emotion recognition, with the potential to impact various industries and domains.

CHAPTER 8

SAMPLECODE

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from tkinter import simpledialog
from tkinter import filedialog
import os
import cv2
import numpy as np
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_json
import pickle
from sklearn.model_selection import train_test_split
import soundfile
import librosa


main = tkinter.Tk()
main.title("EMOTION DETECTION USING SPEECH RECOGNITION AND FACIAL
EXPRESSION") #designing main screen
```

```
main.geometry("1300x1200")

global filename
global X, Y
global face_classifier
global speech_X, speech_Y
global speech_classifier

face_emotion = ['angry','disgusted','fearful','happy','neutral','sad','surprised']
speech_emotion = ['neutral', 'calm', 'happy', 'sad', 'angry', 'fearful', 'disgust', 'surprised']

def getID(name):
    index = 0
    for i in range(len(names)):
        if names[i] == name:
            index = i
            break
    return index
   def upload():
    global filename
    filename = filedialog.askdirectory(initialdir=".")
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");
    def processDataset():
    text.delete('1.0', END)
    global X, Y
    global speech_X, speech_Y
    '''
    X = []
    Y = []
    for root, dirs, directory in os.walk(filename):
        for j in range(len(directory)):
```

```
        name = os.path.basename(root)
        print(name+" "+root+"/"+directory[j])
        if 'Thumbs.db' not in directory[j]:
            img = cv2.imread(root+"/"+directory[j])
            img = cv2.resize(img, (32,32))
            im2arr = np.array(img)
            im2arr = im2arr.reshape(32,32,3)
            X.append(im2arr)
            Y.append(getID(name))

X = np.asarray(X)
Y = np.asarray(Y)
print(Y)

X = X.astype('float32')
X = X/255
test = X[3]
test = cv2.resize(test,(400,400))
cv2.imshow("aa",test)
cv2.waitKey(0)
indices = np.arange(X.shape[0])
np.random.shuffle(indices)
X = X[indices]
Y = Y[indices]
Y = to_categorical(Y)
np.save('model/X.txt',X)
np.save('model/Y.txt',Y)
'''
X = np.load('model/X.txt.npy')
Y = np.load('model/Y.txt.npy')
speech_X = np.load('model/speechX.txt.npy')
speech_Y = np.load('model/speechY.txt.npy')
```

```
    text.insert(END,"Total number of images found in dataset is  : "+str(len(X))+"\n")
    text.insert(END,"Total facial expression found in dataset is : "+str(face_emotion)+"\n")
    text.insert(END,"Total number of speech emotion audio files found in dataset is  : "+str(speech_X.shape[0])+"\n")
    text.insert(END,"Total speech emotion found in dataset is : "+str(speech_emotion)+"\n")
def trainSpeechCNN():
    global speech_classifier
    if os.path.exists('model/speechmodel.json'):
        with open('model/speechmodel.json', "r") as json_file:
            loaded_model_json = json_file.read()
            speech_classifier = model_from_json(loaded_model_json)
        json_file.close()
        speech_classifier.load_weights("model/speech_weights.h5")
        speech_classifier._make_predict_function()
    else:
        speech_classifier = Sequential()
        speech_classifier.add(Convolution2D(32, 1, 1, input_shape = (speech_X.shape[1], speech_X.shape[2], speech_X.shape[3]), activation = 'relu'))
        speech_classifier.add(MaxPooling2D(pool_size = (1, 1)))
        speech_classifier.add(Convolution2D(32, 1, 1, activation = 'relu'))
        speech_classifier.add(MaxPooling2D(pool_size = (1, 1)))
        speech_classifier.add(Flatten())
        speech_classifier.add(Dense(output_dim = 256, activation = 'relu'))
        speech_classifier.add(Dense(output_dim = speech_Y.shape[1], activation = 'softmax'))
        print(speech_classifier.summary())
        speech_classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
        hist = speech_classifier.fit(speech_X, speech_Y, batch_size=16, epochs=10, shuffle=True, verbose=2)
        speech_classifier.save_weights('model/speech_weights.h5')
        model_json = speech_classifier.to_json()
        with open("model/speechmodel.json", "w") as json_file:
```

```
            json_file.write(model_json)
        json_file.close()
        f = open('model/speechhistory.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
    print(face_classifier.summary())
    f = open('model/speechhistory.pckl', 'rb')
    data = pickle.load(f)
    f.close()
    acc = data['accuracy']
    accuracy = acc[99] * 100
    text.insert(END,"CNN Speech Emotion Training Model Accuracy = "+str(accuracy)+"\n\n")
def trainFaceCNN():
    global face_classifier
    text.delete('1.0', END)
    if os.path.exists('model/cnnmodel.json'):
        with open('model/cnnmodel.json', "r") as json_file:
            loaded_model_json = json_file.read()
            face_classifier = model_from_json(loaded_model_json)
        json_file.close()
        face_classifier.load_weights("model/cnnmodel_weights.h5")
        face_classifier._make_predict_function()
    else:
        face_classifier = Sequential()
        face_classifier.add(Convolution2D(32, 3, 3, input_shape = (32, 32, 3), activation = 'relu'))
        face_classifier.add(MaxPooling2D(pool_size = (2, 2)))
        face_classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))
        face_classifier.add(MaxPooling2D(pool_size = (2, 2)))
        face_classifier.add(Flatten())
        face_classifier.add(Dense(output_dim = 256, activation = 'relu'))
        face_classifier.add(Dense(output_dim = 7, activation = 'softmax'))
        print(face_classifier.summary())
```

```
        face_classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
        hist = face_classifier.fit(X, Y, batch_size=16, epochs=10, shuffle=True, verbose=2)
        face_classifier.save_weights('model/cnnmodel_weights.h5')
        model_json = face_classifier.to_json()
        with open("model/cnnmodel.json", "w") as json_file:
            json_file.write(model_json)
        json_file.close()
        f = open('model/cnnhistory.pckl', 'wb')
        pickle.dump(hist.history, f)
        f.close()
    print(face_classifier.summary())
    f = open('model/cnnhistory.pckl', 'rb')
    data = pickle.load(f)
    f.close()
    acc = data['accuracy']
    accuracy = acc[29] * 100
    text.insert(END,"CNN Facial Expression Training Model Accuracy = "+str(accuracy)+"\n\n")

def predictFaceExpression():
    global face_classifier
    filename = filedialog.askopenfilename(initialdir="testImages")
    image = cv2.imread(filename)
    img = cv2.resize(image, (32,32))
    im2arr = np.array(img)
    im2arr = im2arr.reshape(1,32,32,3)
    img = np.asarray(im2arr)
    img = img.astype('float32')
    img = img/255
    preds = face_classifier.predict(img)
    predict = np.argmax(preds)

    img = cv2.imread(filename)
```

```
    img = cv2.resize(img, (600,400))
    cv2.putText(img, 'Facial Expression Recognized as : '+face_emotion[predict], (10, 25),
cv2.FONT_HERSHEY_SIMPLEX,0.7, (255, 0, 0), 2)
    cv2.imshow('Facial Expression Recognized as : '+face_emotion[predict], img)
    cv2.waitKey(0)
def extract_feature(file_name, mfcc, chroma, mel):
    with soundfile.SoundFile(file_name) as sound_file:
        X = sound_file.read(dtype="float32")
        sample_rate=sound_file.samplerate
        if chroma:
            stft=np.abs(librosa.stft(X))
        result=np.array([])
        if mfcc:
            mfccs=np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
            result=np.hstack((result, mfccs))
        if chroma:
            chroma=np.mean(librosa.feature.chroma_stft(S=stft, sr=sample_rate).T,axis=0)
            result=np.hstack((result, chroma))
        if mel:
            mel=np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T,axis=0)
            result=np.hstack((result, mel))
    sound_file.close()
    return result
def predictSpeechExpression():
    global speech_classifier
    filename = filedialog.askopenfilename(initialdir="testSpeech")
    fname = os.path.basename(filename)
    test = []
    mfcc = extract_feature(filename, mfcc=True, chroma=True, mel=True)
    test.append(mfcc)
    test = np.asarray(test)
    test = test.astype('float32')
```

```
test = test/255
test = test.reshape((test.shape[0],test.shape[1],1,1))
predict = speech_classifier.predict(test)
predict = np.argmax(predict)
print(predict)
emotion = speech_emotion[predict-1]
text.delete('1.0', END)
text.insert(END,"Upload speech file : "+fname+" Emotion Recognized as : "+emotion+"\n")
def graph():
f = open('model/cnnhistory.pckl', 'rb')
cnn_data = pickle.load(f)
f.close()
face_accuracy = cnn_data['accuracy']
face_loss = cnn_data['loss']
f = open('model/speechhistory.pckl', 'rb')
cnn_data = pickle.load(f)
f.close()
speech_accuracy = cnn_data['accuracy']
speech_loss = cnn_data['loss']
sa = []
sl = []
for i in range(90,100):
    sa.append(speech_accuracy[i])
    sl.append(speech_loss[i])
fa = []
fl = []
for i in range(20,30):
    fa.append(face_accuracy[i])
    fl.append(face_loss[i])
plt.figure(figsize=(10,6))
plt.grid(True)
plt.xlabel('Iterations/Epoch')
```

```
    plt.ylabel('Accuracy')
    plt.plot(fa, 'ro-', color = 'green')
    plt.plot(fl, 'ro-', color = 'orange')
    plt.plot(sa, 'ro-', color = 'blue')
    plt.plot(sl, 'ro-', color = 'red')
    plt.legend(['Face Emotion Accuracy', 'Face Emotion Loss','Speech Emotion Accuracy','Speech
Emotion Loss'], loc='upper left')
    plt.title('CNN Face & Speech Emotion Accuracy Comparison Graph')
    plt.show()
def exit():
    main.destroy()
font = ('times', 13, 'bold')
title = Label(main, text='EMOTION DETECTION USING SPEECH RECOGNITION AND FACIAL
EXPRESSION')
title.config(bg='LightGoldenrod1', fg='medium orchid')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=100)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=480,y=100)
text.config(font=font1)
font1 = ('times', 12, 'bold')
uploadButton = Button(main, text="Upload Facial Emotion Dataset", command=upload)
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)
processButton = Button(main, text="Preprocess Dataset", command=processDataset)
processButton.place(x=50,y=150)
processButton.config(font=font1)
```

```
cnnButton = Button(main, text="Train Facial Emotion CNN Algorithm", command=trainFaceCNN)
cnnButton.place(x=50,y=200)
cnnButton.config(font=font1)
rnnButton = Button(main, text="Train Speech Emotion CNN Algorithm", command=trainSpeechCNN)
rnnButton.place(x=50,y=250)
rnnButton.config(font=font1)
graphButton = Button(main, text="Accuracy Comparison Graph", command=graph)
graphButton.place(x=50,y=300)
graphButton.config(font=font1)
predictfaceButton = Button(main, text="Predict Facial Emotion", command=predictFaceExpression)
predictfaceButton.place(x=50,y=350)
predictfaceButton.config(font=font1)
predictspeechButton = Button(main, text="Predict Speech Emotion",
command=predictSpeechExpression)
predictspeechButton.place(x=50,y=400)
predictspeechButton.config(font=font1)
exitButton = Button(main, text="Exit", command=exit)
exitButton.place(x=50,y=450)
exitButton.config(font=font1)
main.config(bg='OliveDrab2')
main.main()
```

REFERENCES

[1] The problem statement is taken from **IEEEExplorer**
https://iopscience.iop.org/article/10.1088/1757-899X/928/3/032007

[2] The **GeeksForGeeks** provided the platform to learn how to create a graph using python

https://www.geeksforgeeks.org/bar-plot-in-matplotlib/

[3] The **Datagen** provided platform to learn about the Convolutional Neural Network

https://datagen.tech/guides/computer-vision/cnn-convolutional-neural-network