A

Mini Project Report

On

SENTIMENTAL ANALYSIS USING MACHINE LEARNING ON MULTIPLE MODALITIES

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Gangaram Sravika Reddy	(217R1A0590)
Eluka Sai Kumar	(217R1A0586)
Gagireddy Bhavish Reddy	(217R1A0587)

Under the Guidance of

Raheem Unnisa

(Assistant 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 UGC Act.1956, Kandlakoya(V), Medchal Road, Hyderabad-501401.

2021-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**SENTIMENT ANALYSIS USING MACHINE LEARNING WITH MUILTIPLE MODALITIES**" being submitted by **G.SRAVIKA REDDY (217R1A0590), E.SAIKUMAR (217R1A0586), G.BHAVISH REDDY (217R1A0587)** 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 2024-25.

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

Raheem Unnisa
Assistant Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. Nuthanakanti Bhaskar
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ________________________

ACKNOWLEDGEMENT

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 my guide

Ms. Raheem Unnisa, Assistant 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) Coordinators: **Dr. K. Maheshwari, Dr. J. Narasimha Rao, Ms. K. Shilpa, Mr. K. Ranjith Reddy** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. Nuthanakanti Bhaskar**, 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 would like to 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.

G. SRAVIKA REDDY (217R1A0590)

E. SAI KUMAR (217R1A0586)

G. BHAVISH REDDY (217R1A0587)

ABSTRACT

Sentiment analysis has evolved significantly with the rise of digital communication, where textual content alone often fails to capture the full spectrum of user emotions. This project introduces a novel approach to sentiment analysis by incorporating multiple modalities—text, images, and audio—using advanced machine learning techniques. Our framework integrates convolutional neural networks (CNNs) to process visual data, recurrent neural networks (RNNs) to analyse textual information, and specialized RNNs with attention mechanisms for audio signals. By combining these diverse data sources, our approach offers a more nuanced understanding of sentiment, addressing the limitations of traditional unimodal methods. We demonstrate the effectiveness of our multimodal model through comprehensive experiments on a diverse dataset of social media posts, product reviews, and multimedia content, achieving notable improvements in sentiment classification accuracy. This approach not only enhances the depth of sentiment analysis but also paves the way for future advancements in affective computing, providing richer insights into user emotions and opinions across various digital platforms.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1.INTRODUCTION

1. INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as “Sentiment Analysis Using Machine Learning with Multiple Modalities”. It collects and preprocess data from multiple sources and Implement machine learning models (CNN, LSTM, Transformers) for each modality. Fuse modalities using early, late, or hybrid fusion techniques. Evaluate performance using accuracy, precision, recall, F1-score, and MAE metrics.

1.2 PROJECT PURPOSE

The primary purpose of this project is to develop a comprehensive multimodal sentiment analysis system using machine learning, capable of accurately analyzing and interpreting human emotions and opinions expressed through various forms of media, including text, images, audio, and video. By integrating multiple modalities, this system aims to provide a more nuanced and realistic understanding of public sentiment, surpassing the limitations of single-modality analysis.

1.3 PROJECT FEATURES

The main features of this project are multi-class sentiment classification. The project integrates text, images, audio, and video analysis, leveraging deep learning models (CNN, LSTM, Transformers) and multimodal fusion techniques. It detects emotions (happiness, sadness, anger, fear, surprise), mines opinions, and analyzes aspect-based sentiment. Technical features include scalability, flexibility, high accuracy, and interpretability. Advanced features include contextual understanding, sarcasm and irony detection, multilingual support, and transfer learning.

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. Sentiment analysis, also known as opinion mining, involves determining the emotional tone behind a body of text, typically through machine learning (ML) techniques. The system can combine features from these modalities to improve accuracy and understand the sentiment expressed. This system analysis ensures a robust, efficient, and effective Sentiment Analysis system, leveraging machine learning and multiple modalities to provide actionable insights.

2.1 PROBLEM DEFINITION

This project explores the development and application of a multimodal sentiment analysis framework that combines these machine learning techniques to analyse and interpret data from multiple sources. By evaluating this framework on diverse datasets, including social media posts, product reviews, and multimedia content, we aim to demonstrate its effectiveness in improving sentiment classification accuracy and providing deeper insights into user emotions. Our approach represents a significant step forward in the field of affective computing, addressing the need for more nuanced and holistic sentiment analysis in the age of multimedia communication.

2.2 EXISTING SYSTEM

Existing systems for sentiment analysis using machine learning across multiple modalities have made significant strides in enhancing the understanding of sentiment by integrating various types of data. For instance, Visual Sent advances multimodal analysis by combining textual and visual data through CNNs and LSTMs. Multi-Modal Deep Sentiment Analysis (MDSA) incorporates text, images, and audio, using deep learning models for each modality, but faces challenges related to high computational cost and complex model training. Sentiment Fusion

Network (SFN) employs multi-level fusion of features and decisions, presenting challenges in model complexity and training time.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

1.Visual Sent

Dependency on Image Quality: Performance is highly dependent on the quality and relevance of the images. Poor-quality images or irrelevant visuals can degrade sentiment classification.

2. Multi-Modal Deep Sentiment Analysis (MDSA)

High Computational Cost: Processing and integrating text, images, and audio require substantial computational power and memory, making the system expensive to train and deploy.

3. Sentiment Fusion Network (SFN)

Fusion Complexity: Multi-level fusion strategies can be complex to design and tune, potentially leading to difficulties in achieving optimal integration of features and decisions.

2.3 PROPOSED SYSTEM

The proposed system for sentiment analysis aims to enhance accuracy and depth by integrating textual, visual, and auditory data through an advanced multimodal framework. It employs a hybrid architecture that leverages Convolutional Neural Networks (CNNs) for image analysis, Long Short-Term Memory (LSTM) networks with attention mechanisms for text processing, and Recurrent Neural Networks (RNNs) for audio analysis. This system first processes each modality separately: textual data is analyzed to capture contextual and semantic nuances, images are evaluated for visual sentiment cues, and audio is examined for tonal and prosodic features. These modality-specific features are then fused using both early and late fusion techniques to create a comprehensive representation of sentiment.

2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- **Comprehensive Sentiment Understanding**: Integrates textual, visual, and auditory data to capture a broader and more nuanced emotional landscape.
- **Enhanced Accuracy**: Combines features from multiple modalities, improving sentiment classification by addressing ambiguities and providing context that single-modality systems might miss.
- **Effective Modality Integration**: Uses Convolutional Neural Networks (CNNs) for images, Long Short-Term Memory (LSTM) networks with attention mechanisms for text, and Recurrent Neural Networks (RNNs) for audio, leveraging the strengths of each data type.

2.4 FEASIBILITY STUDY

A feasibility study aims to evaluate the technical, economic and operational viability of a project or system. For sentiment analysis using machine learning with multiple modalities, the feasibility study will access whether it is practical and beneficial to implement such a system. The key considerations involved in the feasibility analysis are

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

2.4.1 TECHNICAL FEASIBILITY

High, leveraging existing machine learning libraries (TensorFlow, PyTorch) and multimodal frameworks. However, collecting and labeling multimodal data requires significant effort. Datasets like **CMU-MOSI** (for video, audio, and text) exist, but preparing and annotating multimodal datasets for specific domains can be challenging.

- **Textual Sentiment Models**: Natural Language Processing (NLP) models like RNNs, LSTMs, and Transformers (BERT, GPT) are mature and perform well in text-based sentiment analysis.
- **Visual Sentiment Models**: Convolutional Neural Networks (CNNs) such as ResNet and VGG can handle visual sentiment tasks by analysing features like facial expressions or object context.

2.4.2 ECONOMIC FEASIBILITY

High, addressing growing demand for sentiment analysis in industries like marketing, customer service, and healthcare. While economically viable in certain domains (e.g., large enterprises, tech firms, media companies), the initial investment in data collection, hardware, and talent is high. The potential long-term gains, especially in customer insights, may justify the costs for enterprises with large-scale operations.

2.4.3 OPERATIONAL FEASIBILITY

High, with scalable cloud infrastructure and containerization (Docker, Kubernetes). While operationally feasible for organizations with the necessary resources, real-time multimodal sentiment analysis introduces significant complexity in data handling, model training, and integration into existing workflows.

- **Data Storage and Management**: Storing large volumes of multimodal data can be operationally challenging. Text data is relatively lightweight, but storing and managing video, audio, and image data is far more demanding in terms of both storage capacity and retrieval speed.
- **Data Security and Privacy**: Handling multiple data sources introduces additional privacy concerns. Video and audio data are particularly sensitive, requiring strong security measures and compliance with data protection regulations such as GDPR or HIPAA.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARWARE REQUIREMENTS

- Processor - Pentium –IV
- RAM - 4 GB (min)
- Hard Disk - 20 GB

2.5.2 SOFTWARE REQUIREMENTS

- Operating system : Windows 7 Ultimate.
- Coding Language : Python.

3. ARCHITECTURE

3. ARCHITECTURE

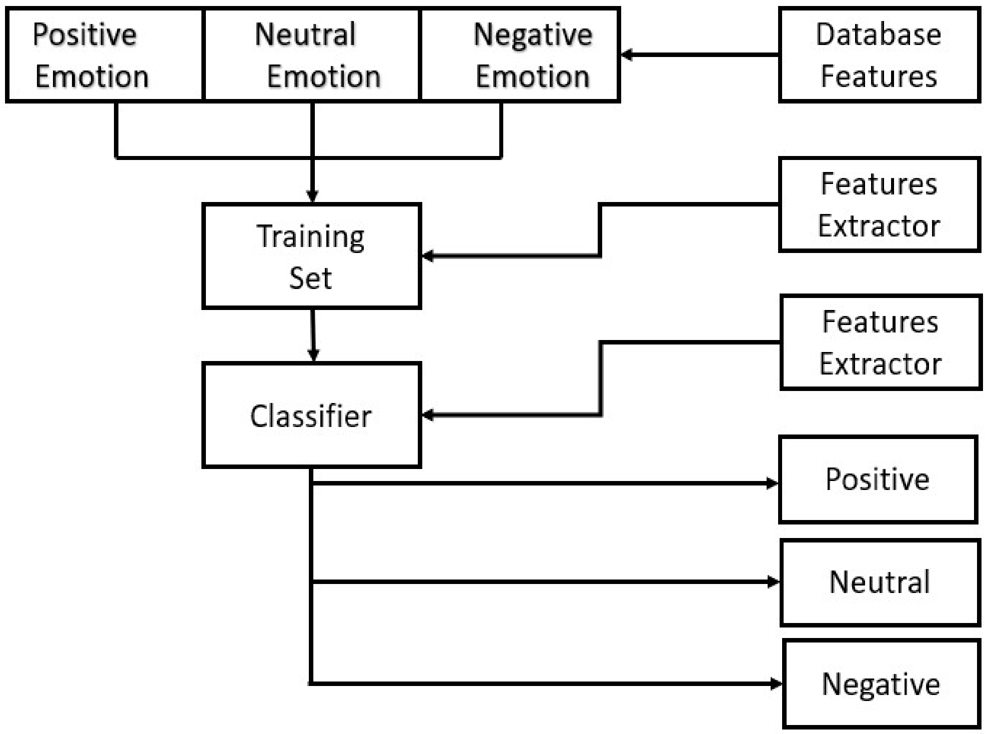


Figure 3.1: Project Architecture

3.2 DESCRIPTION

Data Collection: Gather datasets from multimodal sources like publicly available datasets for text, audio, and images.

Data Preprocessing: In this step the collected raw data is normalized, that is all the unwanted signals ,noise disturbances are removed by preprocessing techniques.

Feature Extraction & selection methods: By using different extraction and selection methods, it will extract required features.

Classification: By using convolutional neural network, it classifies the training data set.By this we will be able to know the accuracy, precision,recall.And also a comparision graph is formed.

3.3 USECASE 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.

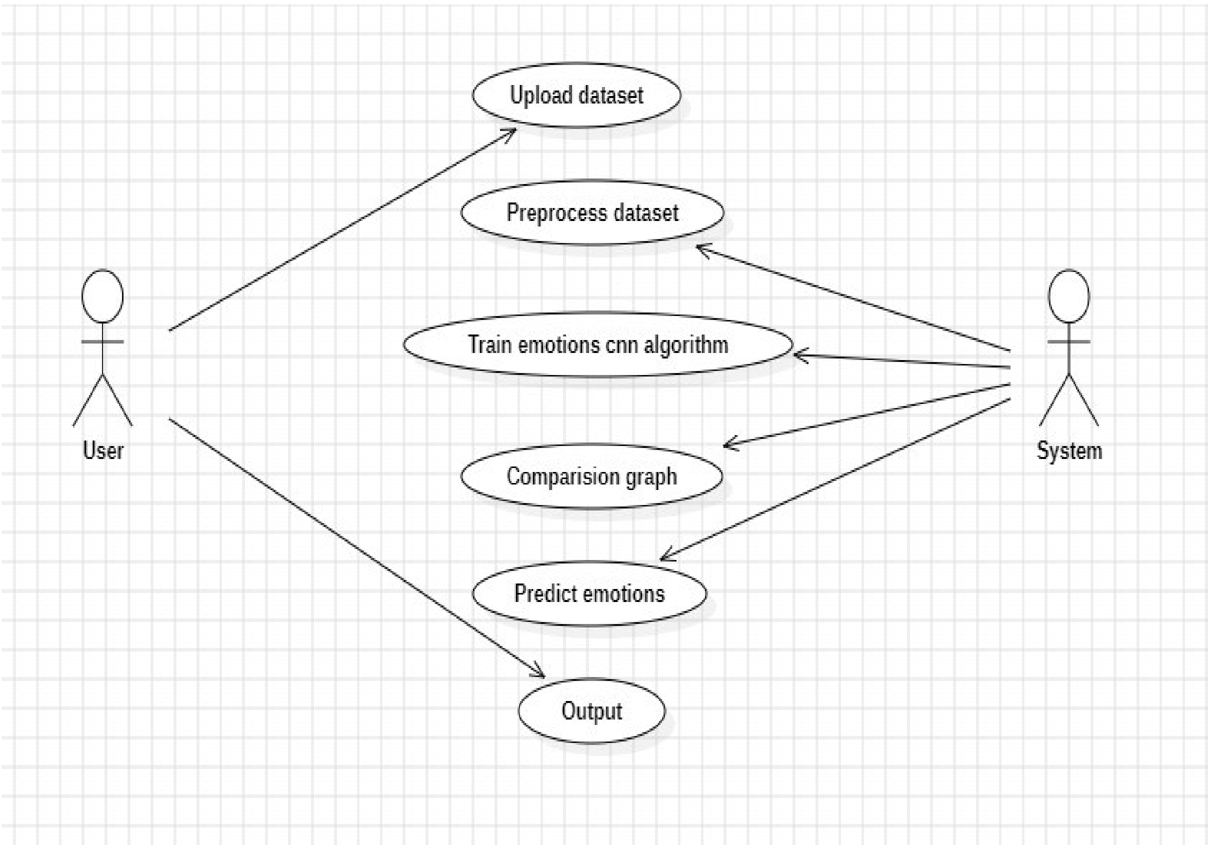


Figure 3.2: Use Case Diagram

3.4 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.

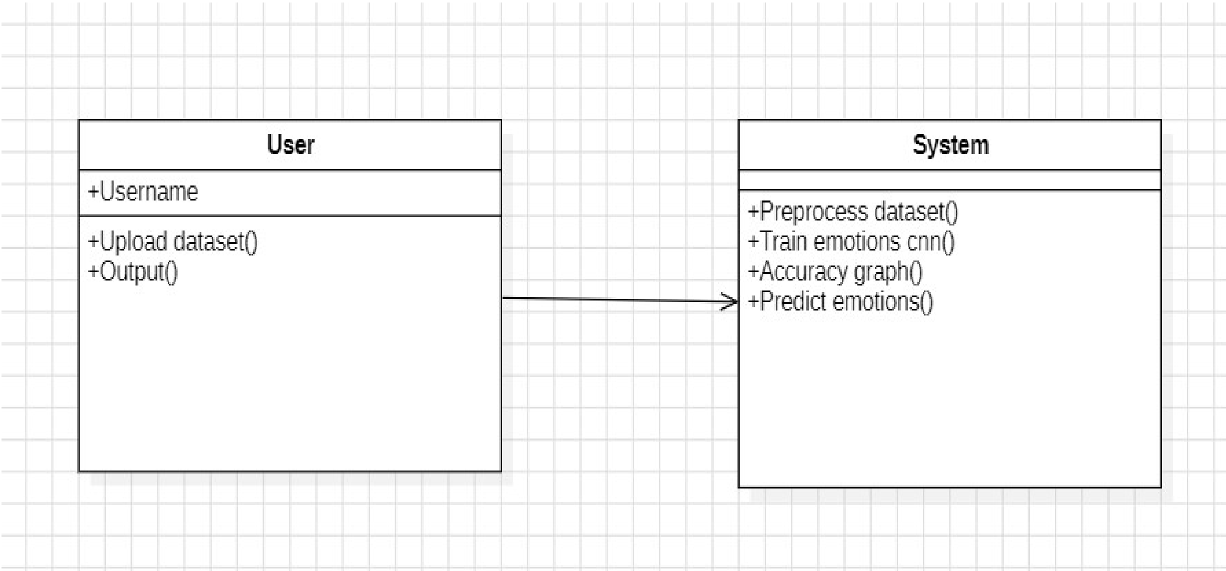


Figure 3.3: Class Diagram

3.5 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.

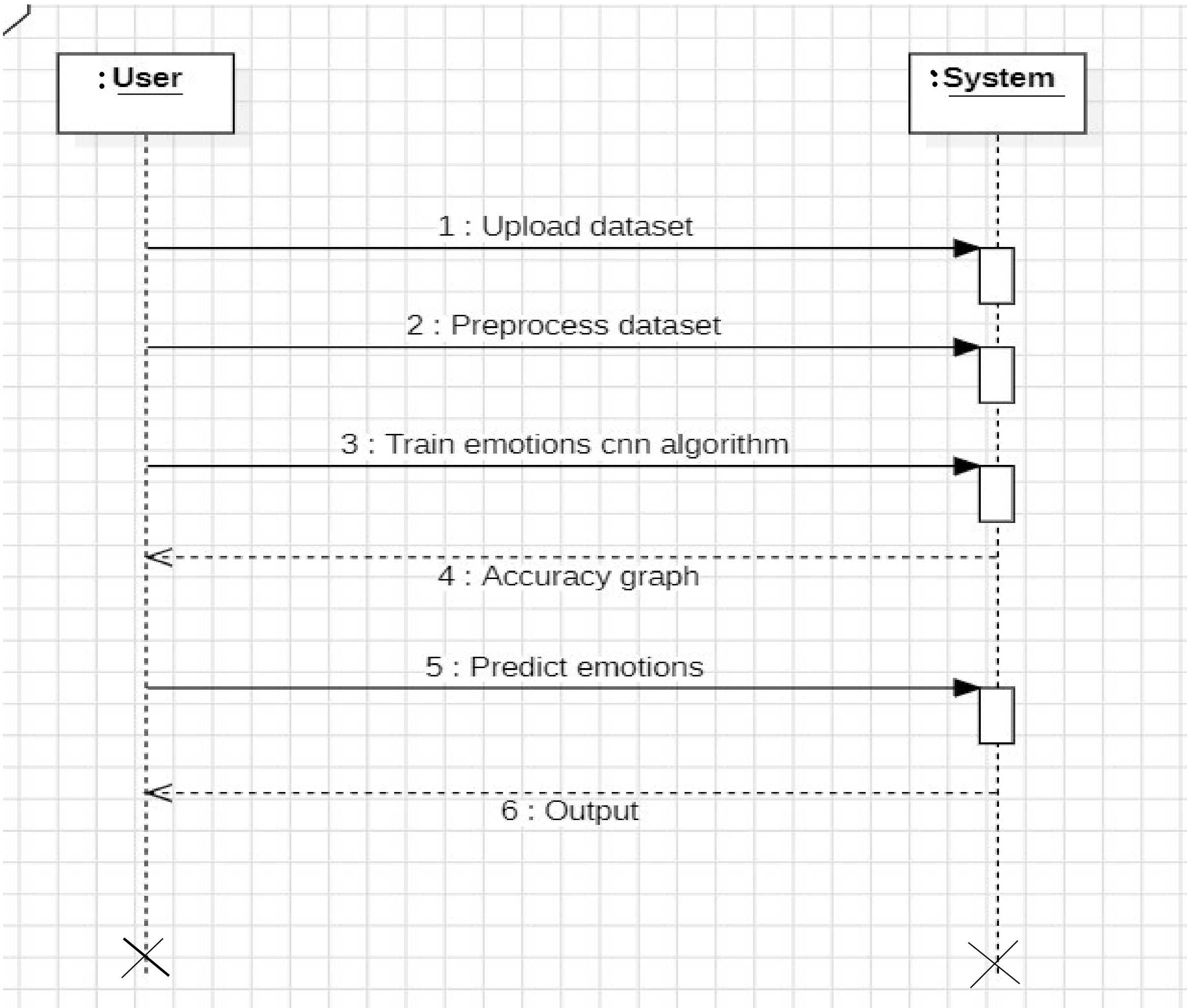


Figure 3.4: Sequence Diagram

3.6 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.

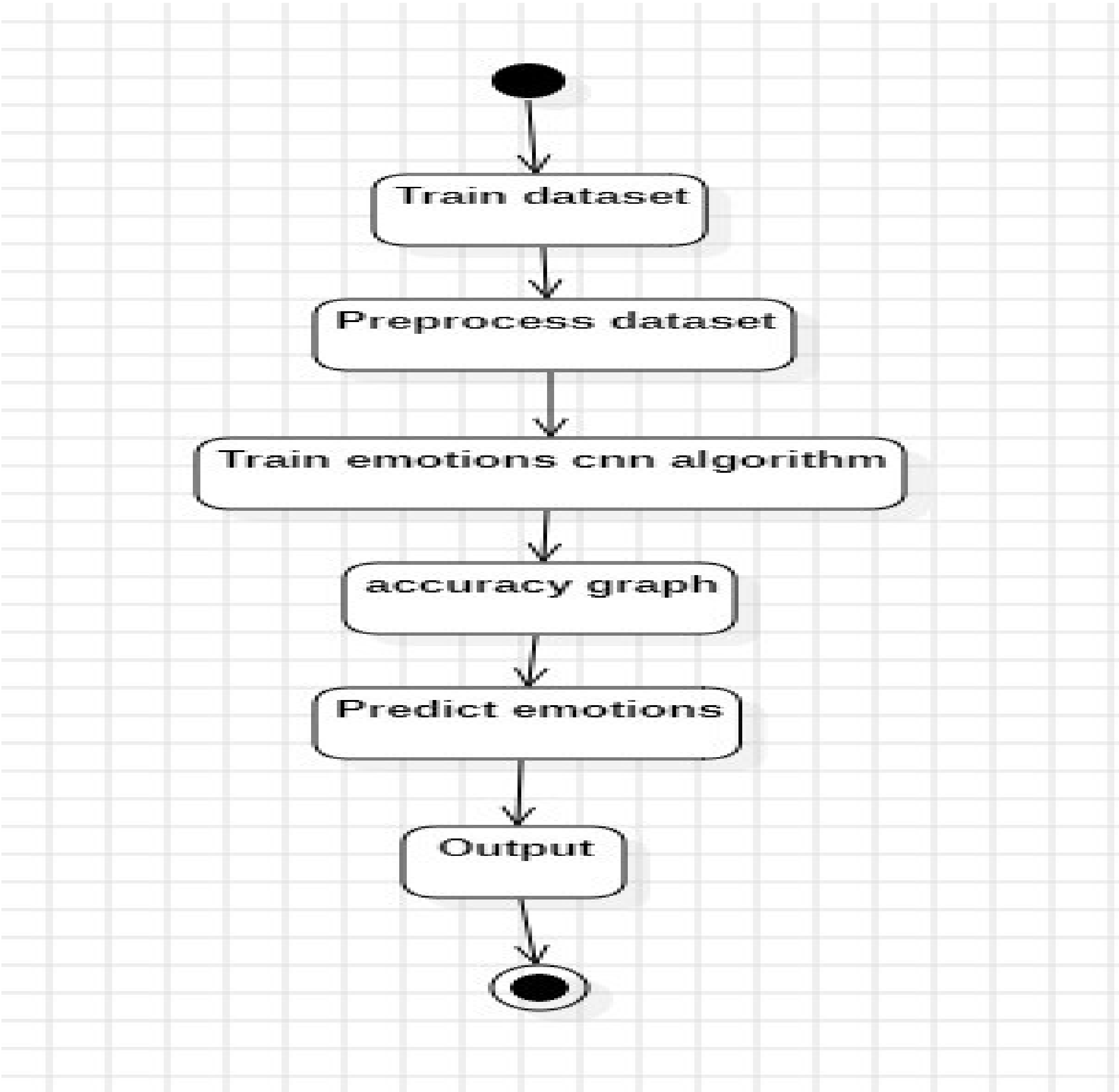


Figure 3.5: Activity Diagram

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

```
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
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
 main = tkinter.Tk()
main.title("Sentiment analysis using Machine Learning on multiple modalities") #designing
main screen
main.geometry("1300x1200")
 sid = SentimentIntensityAnalyzer()
```

```
 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 trainSpeechRNN():
```

```
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,"RNN 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)
```

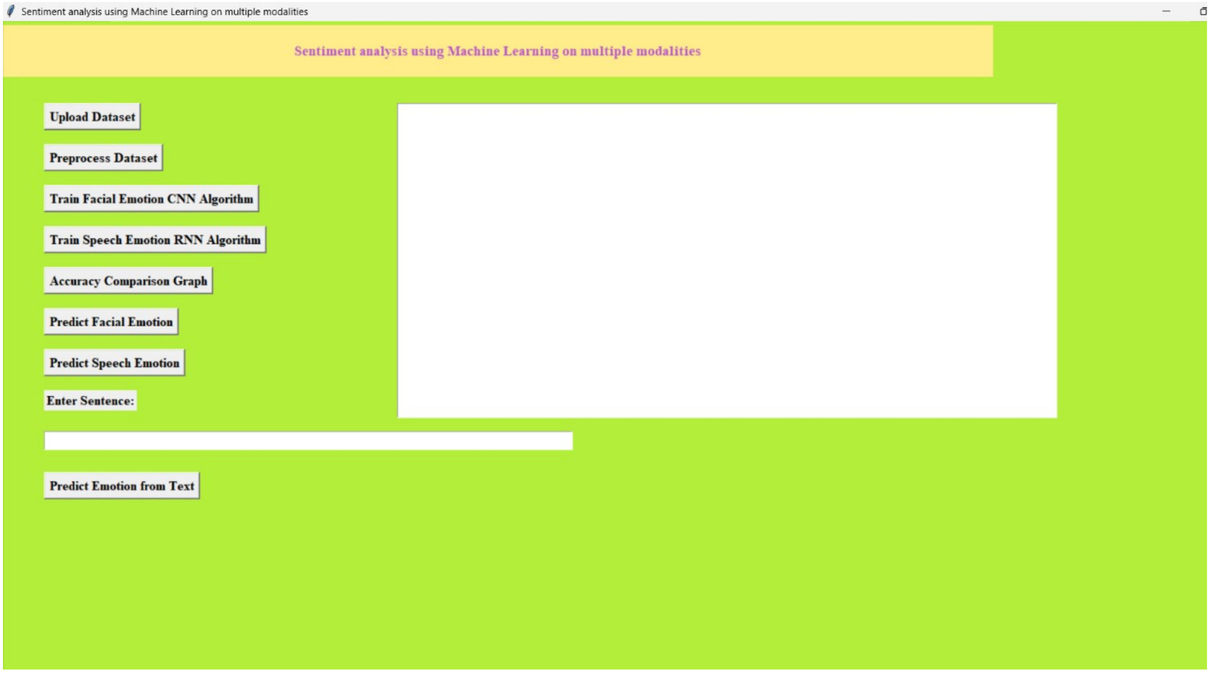
```
  text.insert(END,"Upload speech file : "+fname+" Emotion Recognized as : "+emotion+"\n")
  def graph():
  f = open('model/cnnhistory.pckl', 'rb')
  cnn_data = pickle.load(f)
  f.close()
  face_accuracy = cnn_data['accuracy']
  face_loss = cnn_data['loss']

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 textEmotion():
text.delete('1.0', END)
sentence = tf1.get()
sentiment_dict = sid.polarity_scores(sentence)
negative = sentiment_dict['neg']
positive = sentiment_dict['pos']
neutral = sentiment_dict['neu']
compound = sentiment_dict['compound']
result = ''
if compound >= 0.05:
   result = 'Happy'
elif compound <= -0.05 :
   result = 'Disgusted'
elif compound >= 0.03 and compound < -0.05:
   result = 'Sad'
elif compound >= 0.01 and compound < 0.03:
   result = 'Fearful'
else :
   result = 'Neutral'
text.insert(END,"Entered Text: "+sentence+"\n\n")
text.insert(END,"Predicted Emotion : "+str(result))
tf1.delete(first=0,last=500)
font = ('times', 13, 'bold')
 title = Label(main, text='Sentiment analysis using Machine Learning on multiple modalities')
title.config(bg='LightGoldenrod1', fg='medium orchid')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)
```

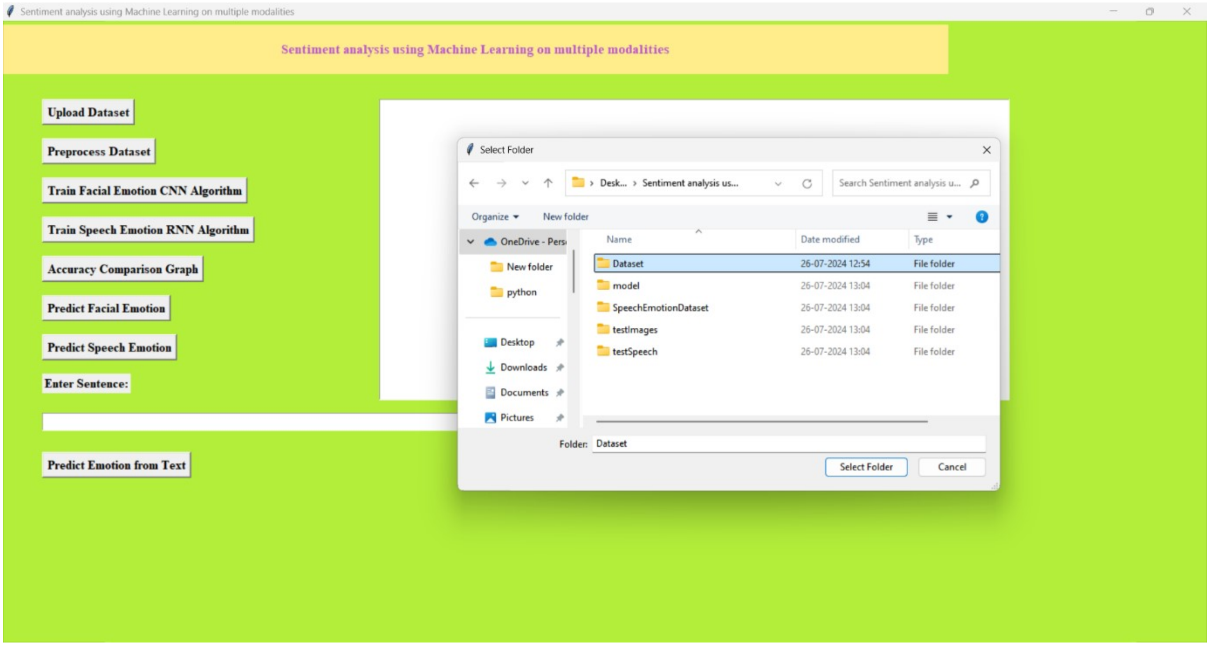
```
    font1 = ('times', 12, 'bold')
    text=Text(main,height=20,width=100)
    scroll=Scrollbar(text)
    text.configure(yscrollcommand=scroll.set)
    text.place(x=480,y=100)
    text.config(font=font1)
   font1 = ('times', 12, 'bold')
   uploadButton = Button(main, text="Upload Dataset", command=upload)
    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 RNN 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)
l1 = Label(main, text='Enter Sentence:')
l1.config(font=font1)
l1.place(x=50,y=450)
tf1 = Entry(main,width=80)
tf1.config(font=font1)
tf1.place(x=50,y=500)
exitButton = Button(main, text="Predict Emotion from Text", command=textEmotion)
exitButton.place(x=50,y=550)
exitButton.config(font=font1)
main.config(bg='OliveDrab2')
main.mainloop()
```

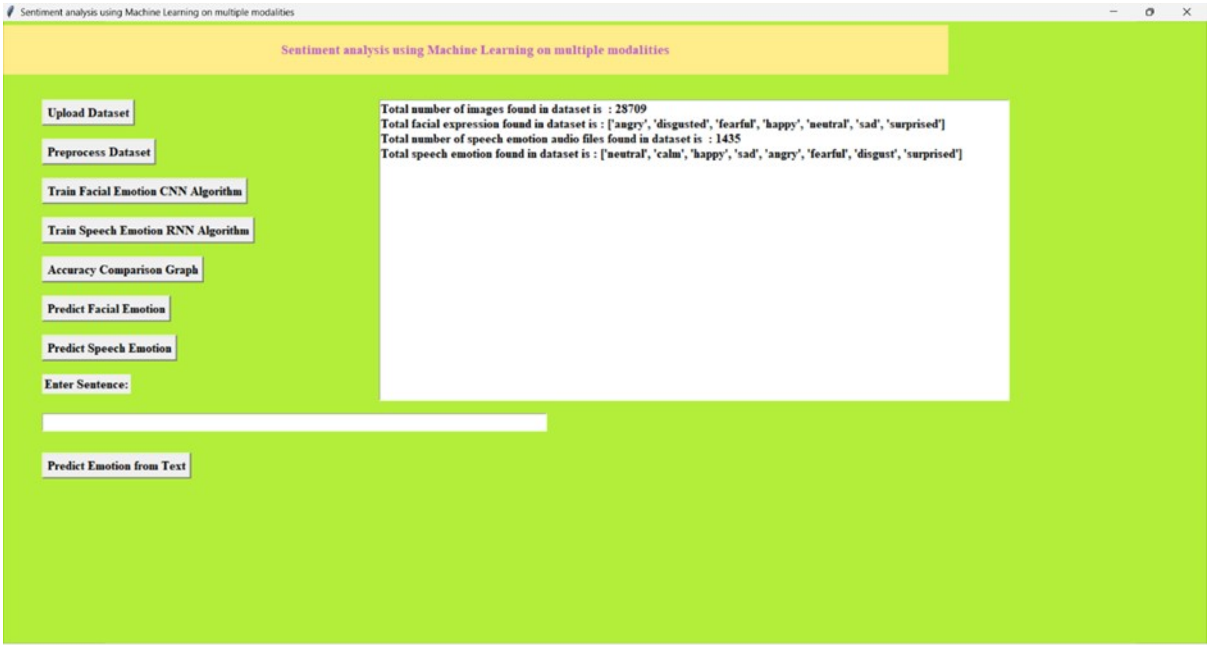
5. RESULTS



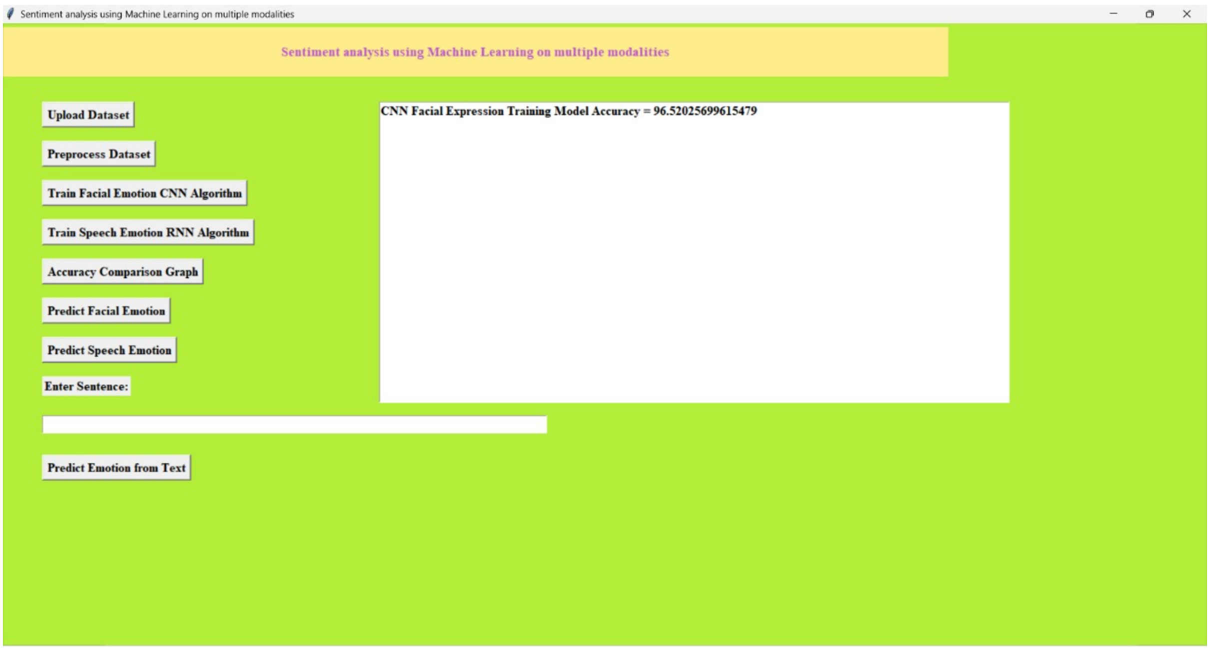
Screenshot 5.1 Upload Emotion Dataset



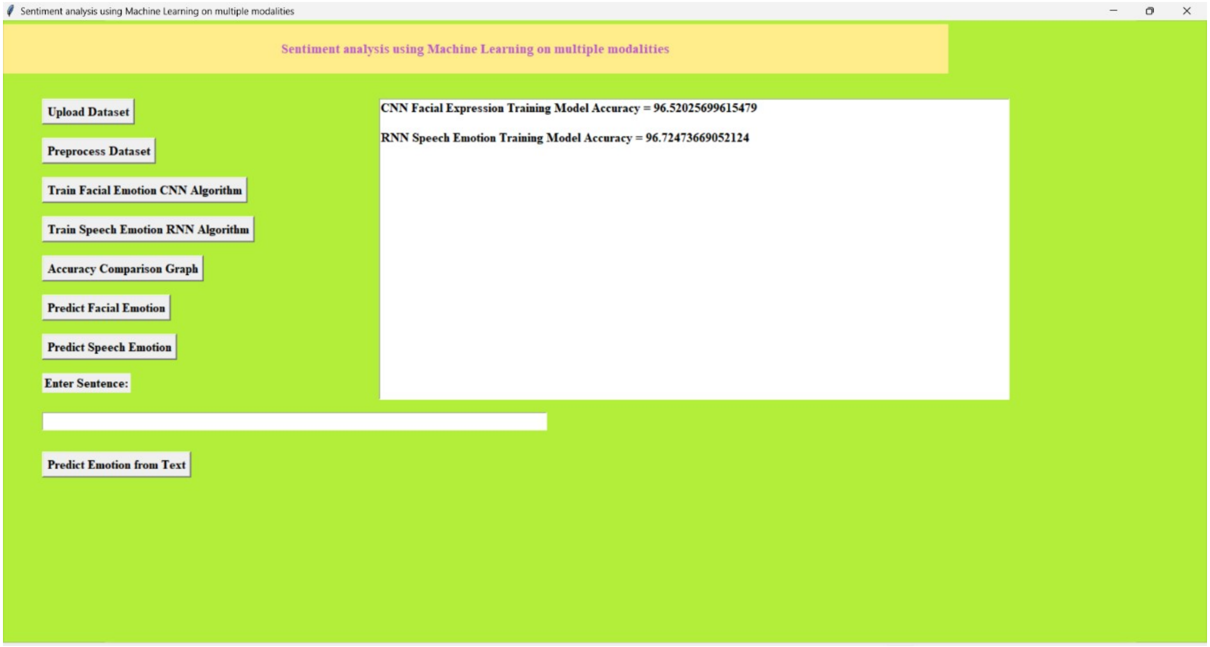
Screenshot 5.2 Uploading Dataset



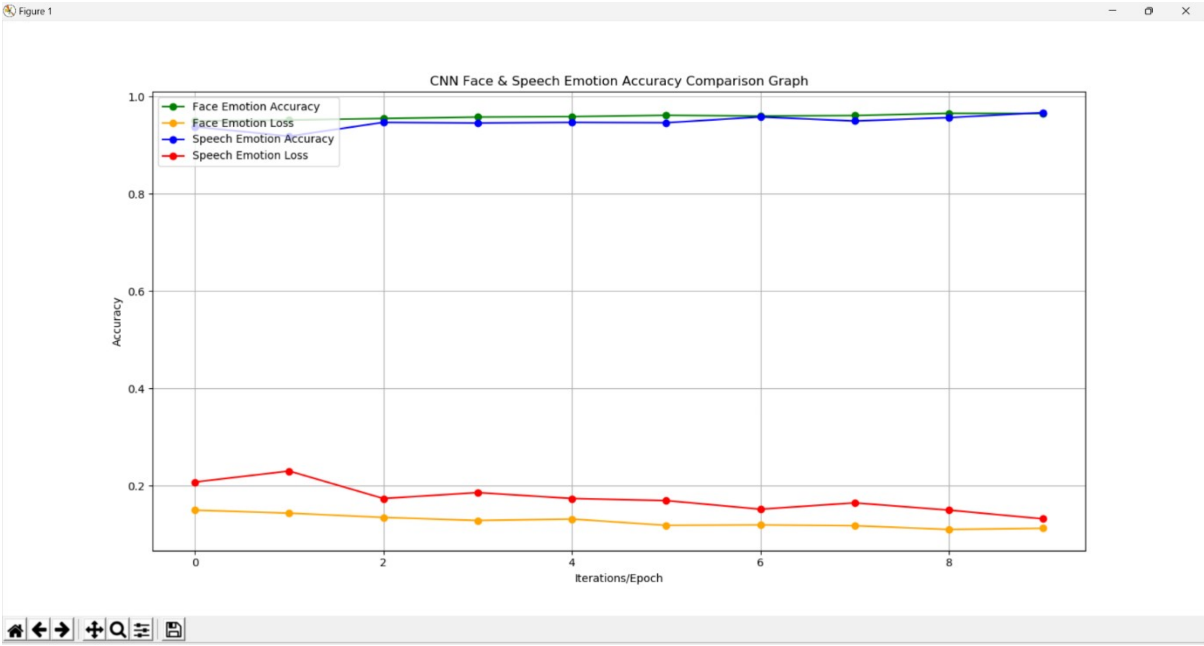
Screenshot 5.3 Preprocessing the Dataset



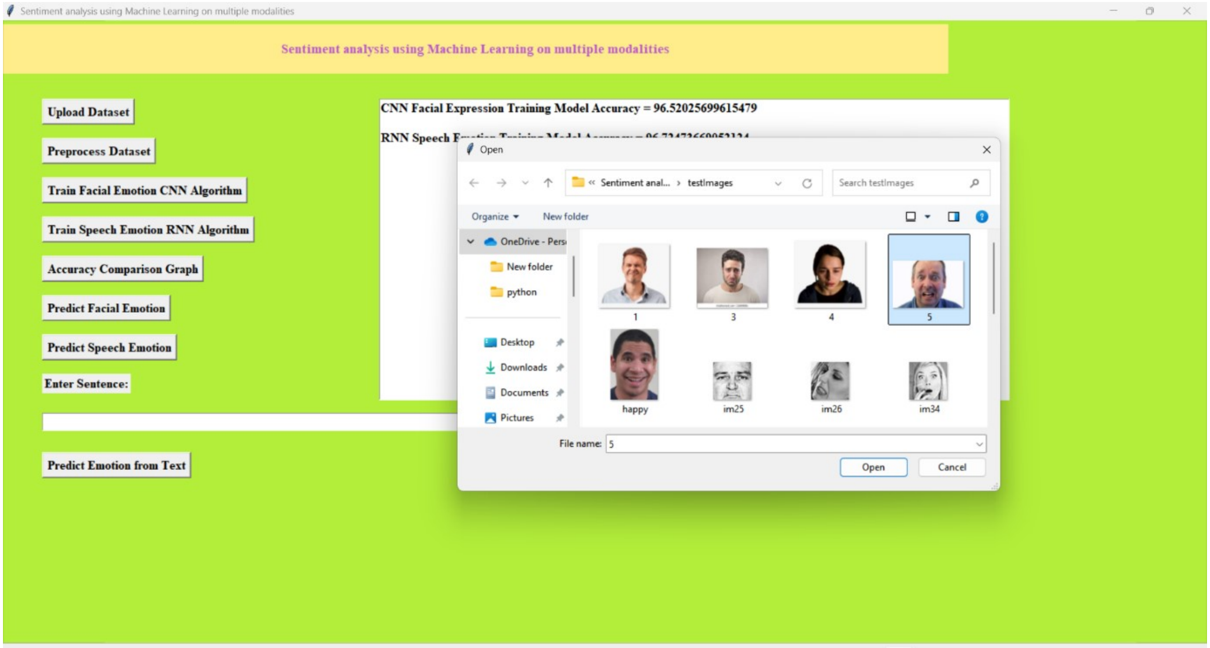
Screenshot 5.4 Train Facial Emotion



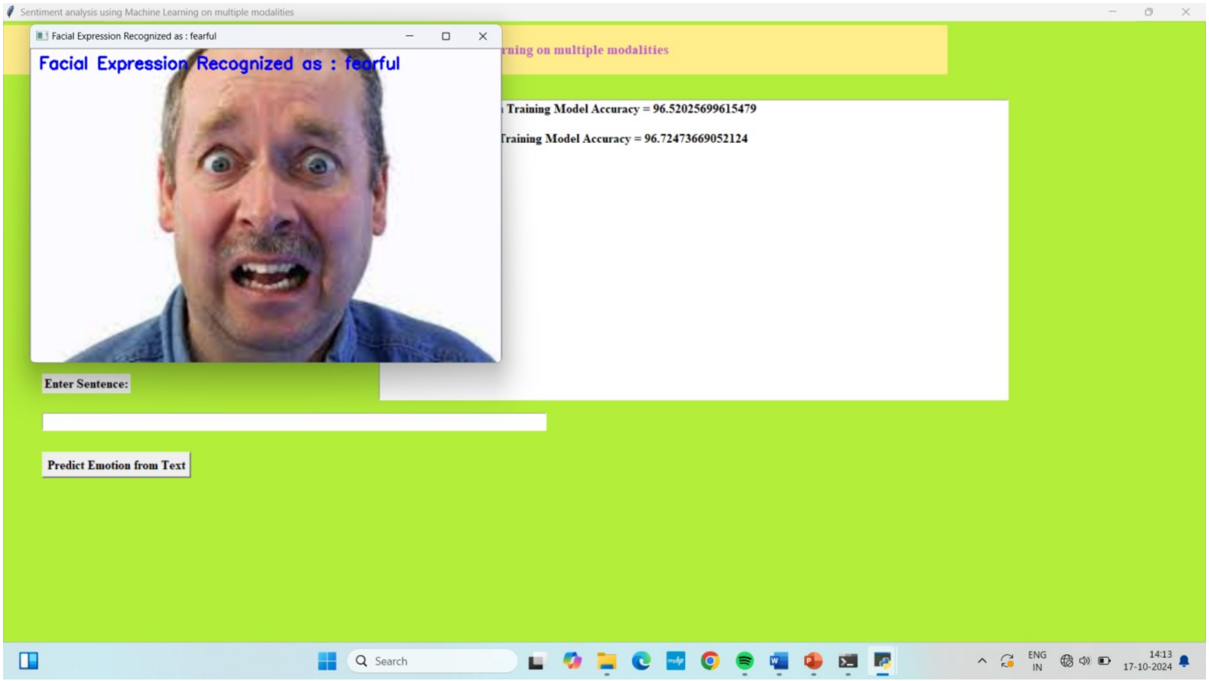
Screenshot 5.5 Train Speech Emotion



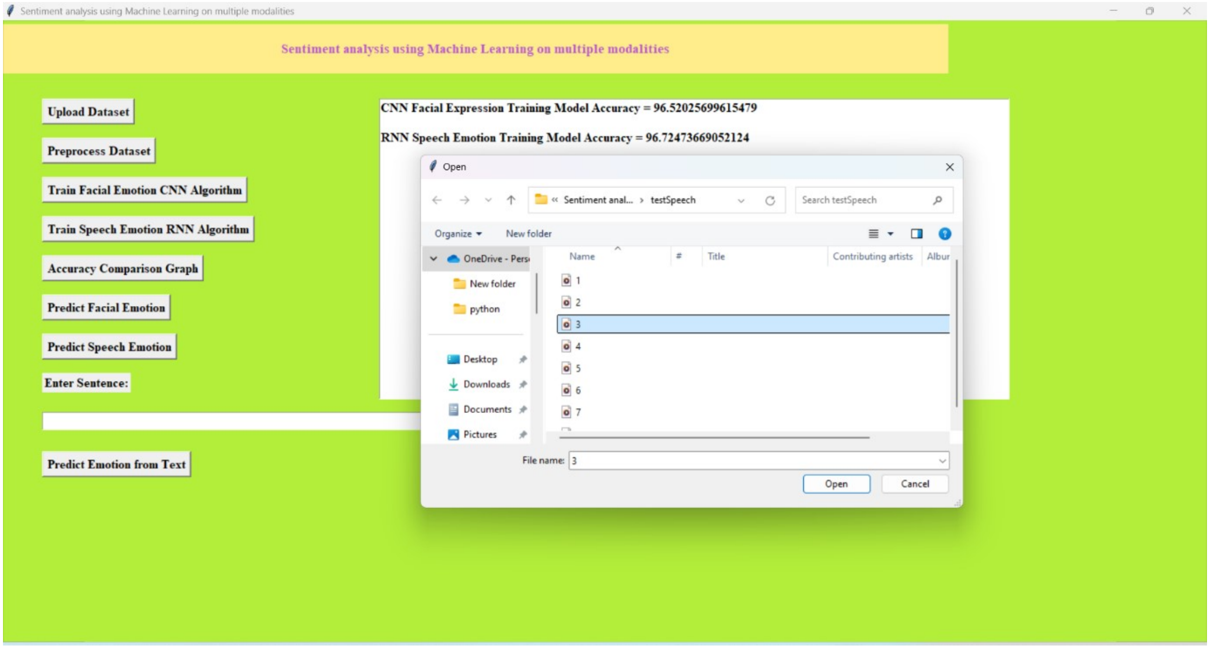
Screenshot 5.6 Accuracy Comparison Graph



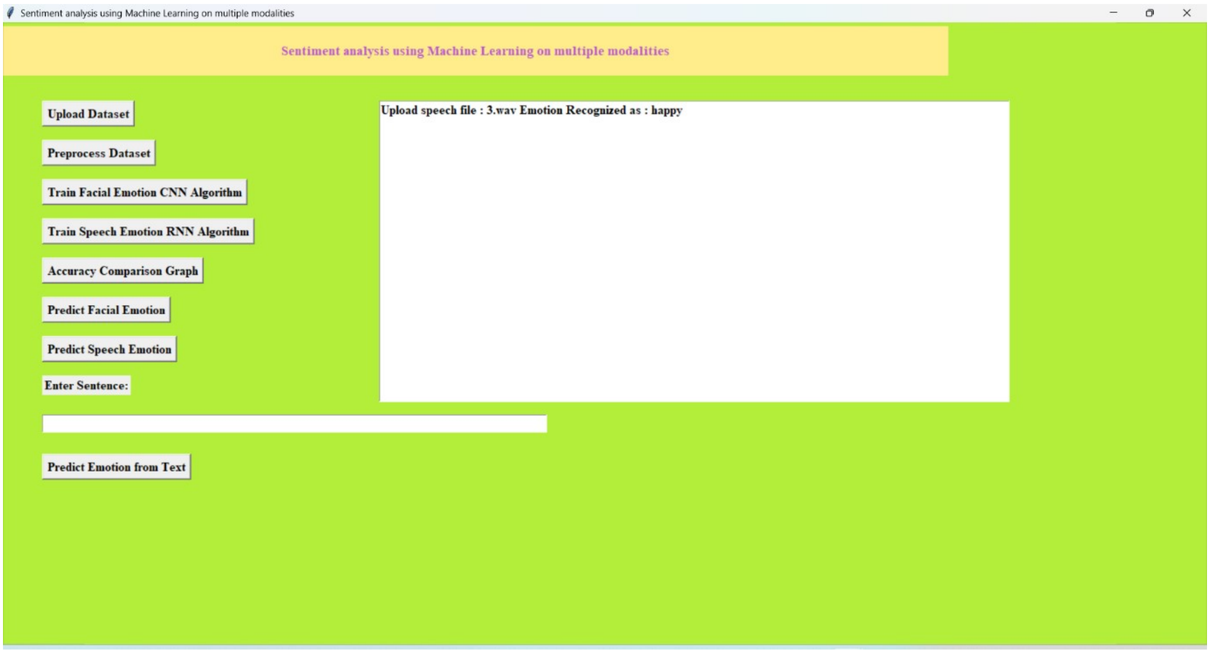
Screenshot 5.7 Uploading the image



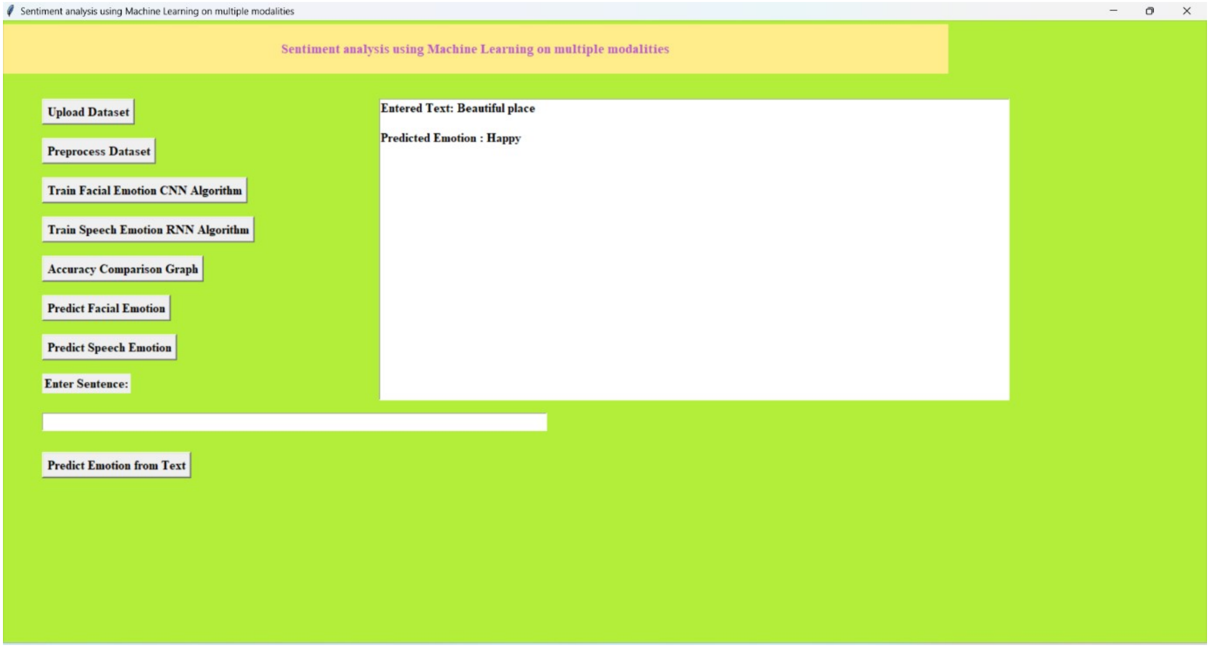
Screenshot 5.8 Facial emotion predicted



Screenshot 5.9 Uploading speech file



Screenshot 5.10 Speech emotion predicted



Screenshot 5.11 Emotion Predicted from Text

6.TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

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

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, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

6.2.4 SYSTEM TESTING

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

6.2.5 WHITE BOX TESTING

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

6.2.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.The test provides inputs and responds to outputs without considering how the software works.

6.3 TESTCASES

6.3.1 UPLOADING DATA

TESTCASE ID	TESTCASE NAME	PURPOSE	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
1	User uploads image	Use it to predict emotion from image	Emotion predicted as fearful from image	Image emotion is predicted as fearful	Pass
2	User uploads Audio clip	Use it to predict emotion from audio	Emotion predicted as happy from audio	Speech emotion is predicted as happy	Pass
3	User enter the text	Use it to predict emotion from text	Emotion predicted as happy from text	Text emotion is predicted as happy	Pass

Table 6.3.1 Uploading Training and Testing Dataset

7. CONCLUSION

7.CONCLUSION

7.1 CONCLUSION

In conclusion, the proposed multimodal sentiment analysis system represents a significant advancement in the field by integrating textual, visual, and auditory data to achieve a more nuanced and accurate understanding of sentiment. By leveraging Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks with attention mechanisms, and Recurrent Neural Networks (RNNs), the system capitalizes on the strengths of each modality while addressing their individual limitations. The use of early and late fusion techniques ensures that the contributions of each data type are effectively balanced and integrated, enhancing overall sentiment classification accuracy. This comprehensive approach not only improves contextual understanding and robustness against noisy or incomplete data but also broadens the range of applications where sentiment analysis can be applied, including social media, customer feedback, and multimedia content. The proposed system's ability to capture complex and subtle emotional cues from multiple sources underscores its potential to advance sentiment analysis and provide deeper insights into human emotions.

7.2 FUTURE SCOPE

Integration of Additional Modalities:

- **Video Data**: Incorporating video analysis to capture both visual and auditory cues simultaneously, enhancing the depth of sentiment understanding by integrating facial expressions, gestures, and body language.
- **Physiological Signals**: Exploring the integration of physiological data such as heart rate variability or skin conductance to provide additional insights into emotional states.

Real-Time Sentiment Analysis:

- **Scalability**: Developing algorithms and infrastructure to enable real-time processing of multimodal data streams, making the system applicable for live social media monitoring, customer service interactions, and live event analysis.

- **Optimization**: Implementing optimizations to reduce latency and computational overhead, allowing for efficient sentiment analysis in dynamic and high-throughput environments.

Enhanced Contextual Understanding:

- **Deep Contextual Models**: Expanding the use of advanced contextual models, such as Transformers, to improve the system's ability to understand and interpret complex and nuanced emotional contexts across different modalities.
- **Cross-Cultural and Multilingual Analysis**: Adapting the system to handle diverse linguistic, cultural, and regional expressions of sentiment, enhancing its applicability across different languages and cultural contexts.

Improved Model Interpretability:

- **Explainable AI**: Developing techniques to improve the interpretability of multimodal models, allowing users to understand and trust the reasoning behind sentiment predictions.
- **Visualization Tools**: Creating tools to visualize the contributions of different modalities and features in the sentiment analysis process, providing insights into how predictions are made.

Advanced Data Fusion Techniques:

- **Adaptive Fusion Strategies**: Implementing adaptive fusion strategies that dynamically adjust the integration of modalities based on their relevance and quality, optimizing the sentiment analysis process.
- **Hierarchical Fusion Models**: Exploring hierarchical fusion models that combine data at multiple levels, from feature extraction to decision-making, for more robust sentiment analysis.

Ethical and Privacy Considerations:

- **Data Privacy**: Ensuring that the system complies with data protection regulations and ethical standards, addressing concerns related to user privacy and data security.
- **Bias Mitigation**: Developing methods to identify and mitigate biases in sentiment analysis, ensuring fair and unbiased predictions across different demographic groups.

Application Expansion:

- **Healthcare and Well-being**: Applying the system to monitor and analyze emotional well-being in healthcare settings, providing insights into patient sentiments and mental health.
- **Entertainment and Media**: Enhancing sentiment analysis in entertainment and media by analyzing audience reactions to movies, music, and advertisements, enabling better content creation and targeting.

Continual Learning and Adaptation:

- **Model Updates**: Implementing mechanisms for continual learning and adaptation to keep the model up-to-date with evolving language, visual, and auditory patterns.
- **Feedback Loops**: Creating feedback loops that allow the system to learn from user interactions and corrections, improving its accuracy and relevance over time.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

1. **Zhang, L., & Zhou, H. (2020).** "Multimodal Sentiment Analysis with Dual Attention Networks." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 3500-3511.
2. **Zhao, Y., Zhang, Y., & Liu, S. (2019).** "Unified Multi-Modal Sentiment Analysis with Transfer Learning." *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 1134-1144.
3. **Hazarika, D., Poria, S., & Cambria, E. (2018).** "MISA: Multimodal Sentiment Analysis using Deep Attention." *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 1281-1293.

8.2 WEBSITES

1. **Zhang, X., Zhao, J., & LeCun, Y. (2015).** "Character-level Convolutional Networks for Text Classification." *Advances in Neural Information Processing Systems*, 649-657.

✓ **Link**: Character-level Convolutional Networks

2. **Chen, L., Zhang, T., & Li, Z. (2019).** "Cross-Modal Sentiment Analysis Framework with Attention Mechanism." *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2828-2838.

✓ **Link**: Cross-Modal Sentiment Analysis Framework

8.3 GITHUB LINK

https://github.com/GangaramSravikaReddy/Sentiment-Analysis-using-Machine-Learning-with-Multiple-Modalities