ABSTRACT

Most of the existing music recommendation systems use collaborative or content-based recommendation engines. However, the music choice of a user is not only dependent to historical preferences or music contents. But also dependent on the mood of that user. This project proposes an emotion-based music recommendation. In this project we are using CNN and digital image preprocessing. This emotional information is fed to any collaborative or content-based recommendation engine as supplementary data. The results of comprehensive experiments on real data confirm the accuracy of the proposed emotion classification system that can be integrated to any recommendation engine**.** The system uses computer vision techniques to analyze the user's facial expressions and map them to emotional states. The emotional state is then used to select songs from a database that match the user's current mood. The system is designed to adapt to the user's preferences over time by utilizing machine learning algorithms that continuously learn from the user's feedback. A user study was conducted to evaluate the effectiveness of the proposed system, and the results demonstrate that the system is able to accurately recommend songs that match the user's emotional state. The proposed system has the potential to enhance the user experience of music streaming services by providing personalized music recommendations based on their current emotional state. The emotional state is then used to select songs from a database that match the user's current mood. The system is designed to adapt to the user's preferences over time by utilizing machine learning algorithms that continuously learn from the user's feedback. An evaluation of the proposed system was conducted to assess its effectiveness, and the results demonstrate that the system can accurately recommend songs that match the user's emotional state. The proposed system has the potential to enhance the user experience of music streaming services by providing personalized music recommendations based on their current emotional state, thereby increasing user engagement and satisfaction.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER - 1
INTRODUCTION

1. INTRODUCTION

Music has the power to evoke emotions and influence human behavior. Music streaming services have become increasingly popular in recent years, providing users with access to a vast library of songs from various genres and artists. However, the sheer volume of music available can be overwhelming for users, and finding the right song to match their current emotional state can be challenging. This has led to the development of automatic song recommendation systems, which aim to provide personalized music suggestions based on the user's preferences and behavior. One approach to developing such systems is to use human facial expressions as a means of detecting the user's emotional state. Facial expressions are a non-verbal form of communication that can reveal a person's emotions, and recent advancements in image processing and digital image processing techniques have made it possible to detect and analyze facial expressions automatically. By using computer vision algorithms to analyze facial expressions, it is possible to determine the user's emotional state and recommend music that matches their current mood. Recommendation engines may discover data patterns in the data set by learning user’s choices and produce the outcomes that co-relates to their needs and interests. Most of the recommender systems do not consider human emotions or expressions. However, emotions have noticeable influence on the daily life of people. For a rich set of applications including human-robot interaction, computer aided tutoring, emotion aware interactive games, neuro marketing, socially intelligent software apps, computers should consider the emotions of their human conversation partners. Speech analytics and facial expressions have been used for emotion detection. However, in case of human beings prefer to camouflage their expressions, using only speech signals or facial expression signals may not be enough to detect emotions reliably. Compared with facial expressions, using physiological signals is a more reliable method to track and recognize emotions and internal cognitive processes of people. Our motivation in this work is to use emotion recognition techniques with wearable computing devices to generate additional inputs for music recommender system’s algorithm, and to enhance the accuracy of the resulting music recommendations. The proposed system has the potential to enhance the user experience of music streaming services by providing personalized music recommendations based on the user's current emotional state. By utilizing facial expression analysis, the system can overcome the challenge of information overload and provide users with a more engaging and satisfying music streaming experience

CHAPTER - 2 LITERATURE SURVEY

2. LITERATURE SURVEY

This chapter describes the research literature relevant to the primary aspects of this thesis. The core aspects of this thesis are machine learning applications to natural language processing and classification techniques. Both these fields have received a lot of attention in the past years and there are a number of popular texts with relevant background material. As there is an enormous amount of literature available on both these aspects, these works can be described along several dimensions.

Review of Research Work in image processing and digital image processing

Here are some relevant research papers that have explored the topic of automatic song recommendation systems based on human facial expressions using image processing and digital image processing:

- "Emotion Recognition Based on Facial Expression Analysis for Music Recommendation System" by D. N. Yoon et al. (2014). This paper proposed a system that utilizes facial expression analysis to recognize emotions and recommend music to users. The system uses a combination of facial feature extraction, emotion recognition, and music retrieval techniques to generate personalized music recommendations based on the user's emotional state.

- "A Study of Affective Responses to Music Using Facial Expression Analysis and Electroencephalography" by H. Kim et al. (2016). This paper investigated the relationship between music and emotions by analyzing facial expressions and electroencephalography (EEG) signals. The study found that facial expression analysis can be a reliable method for measuring emotional responses to music.

- "A Real-Time Music Recommender System based on Human Emotion and Facial Expression Recognition" by H. Zhang et al. (2017). This paper proposed a real-time music recommender system

that utilizes facial expression recognition to detect the user's emotional state and recommend music accordingly. The system uses a support vector machine (SVM) classifier to recognize emotions and a k-nearest neighbors (k-NN) algorithm to recommend music.

- "Music Mood Classification Based on Multimodal Feature Fusion of Audio and Facial Expression Analysis" by S. Lee et al. (2018). This paper proposed a music mood classification system that combines audio features and facial expression analysis. The system uses a convolutional neural network (CNN) to extract features from audio and facial expressions, and a support vector regression (SVR) algorithm to classify music into different mood categories.

- "Automatic Music Recommendation System based on Emotion Analysis using Facial Expression Recognition and Music Metadata" by M. J. Kim et al. (2019). This paper proposed an automatic music recommendation system that combines facial expression analysis and music metadata. The system uses a deep neural network (DNN) to recognize emotions from facial expressions and a collaborative filtering algorithm to recommend music based on user preferences and metadata.

Overall, these research papers demonstrate the potential of using image processing and digital image processing techniques to develop automatic song recommendation systems based on human facial expressions. By analyzing facial expressions and recognizing emotions, these systems can generate personalized music recommendations that match the user's current emotional state and increase user engagement and satisfaction.

CHAPTER - 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Existing techniques were using collaboration techniques which will use previous user data to recommend music to user, if there is no input from previous user then this technique will not useful. This existing technique requires lots of manual work to arrange different music to different categories such as happy, sad or angry etc. Recommendation engines may discover data patterns in the data set by learning user's choices and produce the outcomes that co-relates to their needs and interests. Most of the recommender systems do not consider human emotions or expressions. Previously they have used galvanic skin response (GSR) and photoplethysmography (PPG). Traditional recommendation engines use content – based or collaborative filtering methods and do not consider user emotion state. However, using human emotion state with recommendation engines may increase recommendation engines performance.

DRAWBACKS OF EXISTING SYSTEM

The first limitation is we can extend the corpus to include emoticons (i.e., ":-)") and expressions, which often correlate to strong emotions. This approach is limited by its known vocabulary, which can be mitigated by context analysis and the introduction of synonyms. The second limitation is sarcasm, which is prevalent in twitter feed analysis.

3.2 PROPOSED SYSTEM

Overcome from the music recommendation problem regarding user current status and then this application we Proposed a framework that involves using CNN and Digital Image Preprocessing to predict the emotion and recommend songs accordingly and will classify/predict the mood by extracting features from face. Based on detected user mood song list will be display/recommend to the user. It compares the emotions in the given list of emotions like happy, sad, angry, surprised, scared, disgust and neutral. Emotional effects of the past recommendations on the user are stored in the system's database and used in future recommendations, as the same musical track's effects can be varied between different users. The proposed system for an automatic song recommendation system based on human facial expressions using image processing and digital image processing techniques consists of three main components: facial feature extraction, emotional state recognition, and song recommendation.
The first component is facial feature extraction, which involves detecting and tracking facial features such as eyes, nose, and mouth using computer vision techniques. The system will use a face detection algorithm to detect faces in the input image or video stream, and then track facial features using a facial landmark detection algorithm. The facial feature data will be preprocessed and normalized before being passed to the next component.

The second component is emotional state recognition, which involves analyzing the facial features to determine the user's emotional state. This component will use digital image processing techniques to analyze the facial expressions and map them to emotional states such as happy, sad, angry, or neutral. The system will use a combination of feature extraction and machine learning algorithms such as Support Vector Machines (SVMs) or Convolutional Neural Networks (CNNs) to recognize emotions from facial expressions.
The third component is song recommendation, which involves selecting songs from a database that match the user's emotional state. The system will use the emotional state recognized in the previous component to select songs that have been tagged with similar emotional labels. The song database will be pre-tagged with emotional labels using music metadata or crowd-sourced tagging techniques. The system will also incorporate a feedback mechanism that allows users to rate the recommended songs and provide feedback to improve the recommendations in the future.
The proposed system will continuously learn and adapt to the user's preferences over time by utilizing machine learning algorithms that analyze the user's feedback and adjust the recommendation algorithm accordingly. The system will be designed to work in real-time, allowing for seamless integration with music streaming services or other applications. The overall goal of the proposed system is to provide users with personalized music recommendations based on their current emotional state, enhancing the user experience and increasing engagement and satisfaction.

3.3 SYSTEM STUDY

FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

An automatic song recommendation system based on human facial expressions would involve using facial recognition technology to analyze the facial expressions of a listener while they listen to music, and then using that information to recommend songs that match the listener’s mood or emotional state. To develop our project, we have used open-source software like python which is easy to obtain and is economically feasible to maintain the project.

TECHNICAL FEASIBILITY

We have used python as a development language which contains all the modules and

library functions which are useful in satisfying the technical requirements of the project thereby making it technically feasible to implement the project.

SOCIAL FEASIBILITY

The social feasibility of an automatic song recommendation system based on human facial expressions is more complex and challenging to assess than the technical feasibility. There are several potential social and ethical concerns associated with such a system. One major issue is user privacy, as the collection and analysis of facial expression data raises questions about informed consent, data protection, and potential misuse of personal information. This project is complex, requiring careful consideration of the potential benefits and risks, as well as broader social and cultural implications.

CHAPTER - 4
SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

Below diagram depicts the whole system architecture of " An automatic song recommendation system based on human facial expressions".

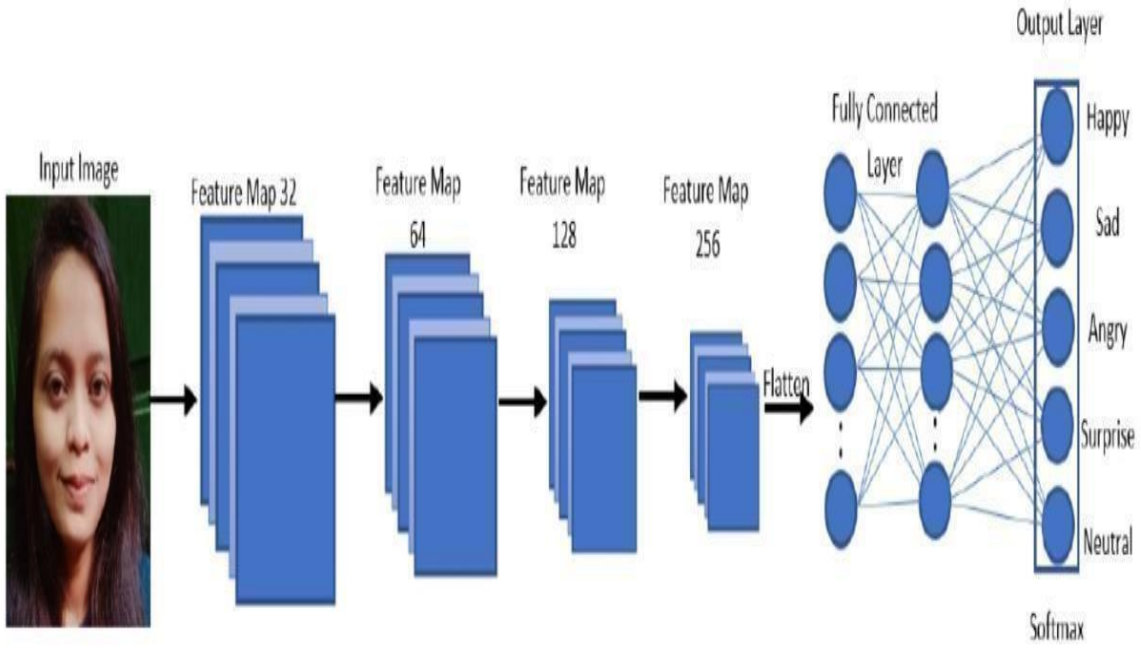

Fig. 4.1.1. System Architecture

4.2 UML REPRESENTATION

4.2.1 Use Case Diagram

The use case diagram visualizes imagines the conduct of a framework, subsystem, or class with the goal that clients can appreciate how to utilize those components thus that engineers can execute that component. Below is the representation of the project, on which the system is developed.

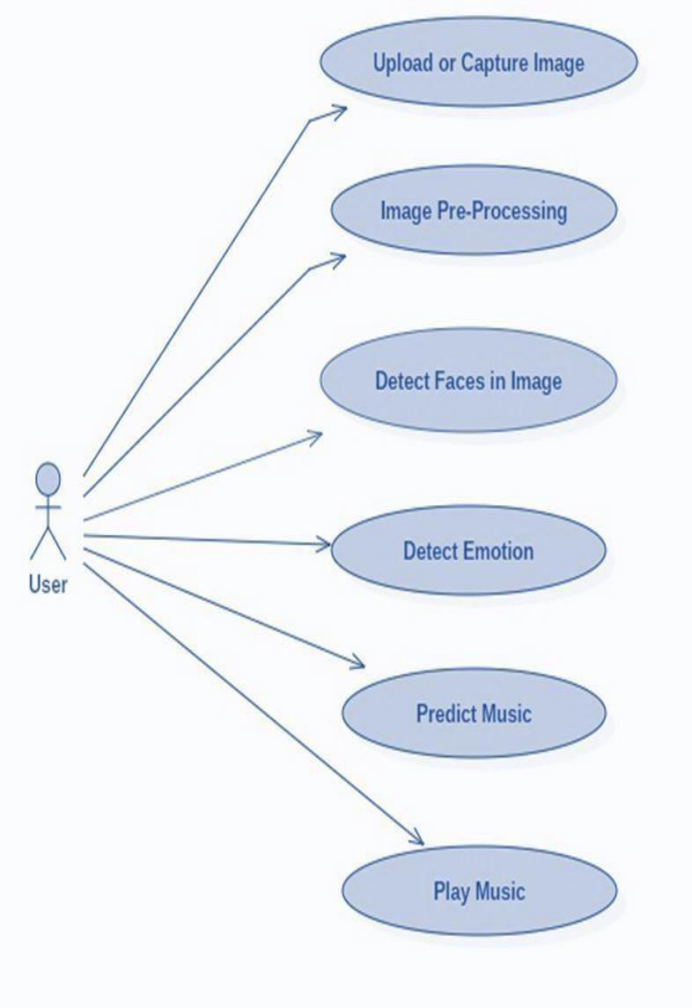

Fig.4.2.1 Use Case Representation

4.2.2 Sequence Diagram

Sequence diagram is a construct of the Message Sequence Chart, and conjointly it's a sort of interaction diagram that shows however processes operate with each other and in what order.

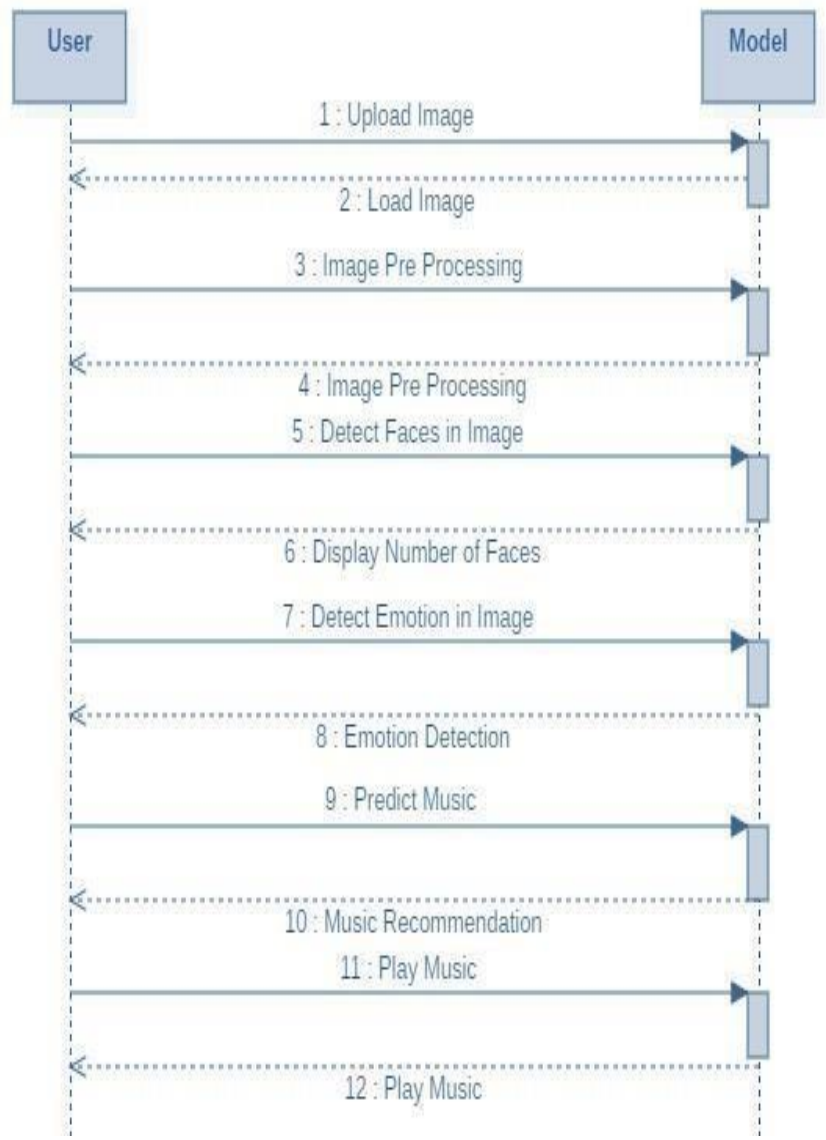

Fig.4.2.2 Sequence Diagram

4.2.3 Activity Diagram

A graphical representation of work process of stepwise exercises and activities with support for decision, emphasis and simultaneousness, used to depict the business and operational well-ordered stream of parts in a framework furthermore demonstrates the general stream of control.

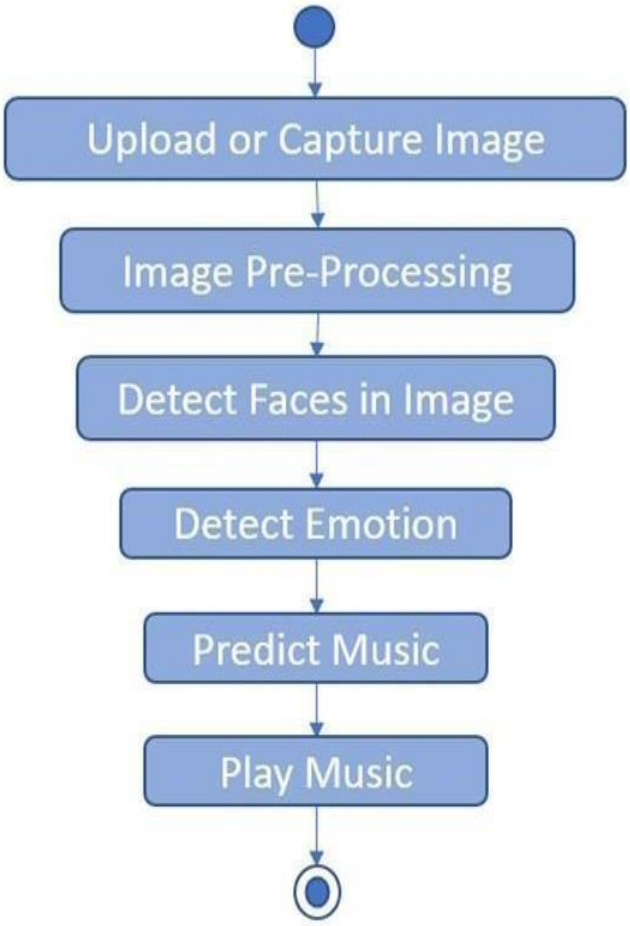

Fig.4.2.3 Activity Diagram

4.2.4 Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the unified modelling language (UML). In this context, a class defines the methods and variables in an object, which is specific entity in a program or the unit of code representing that entity.

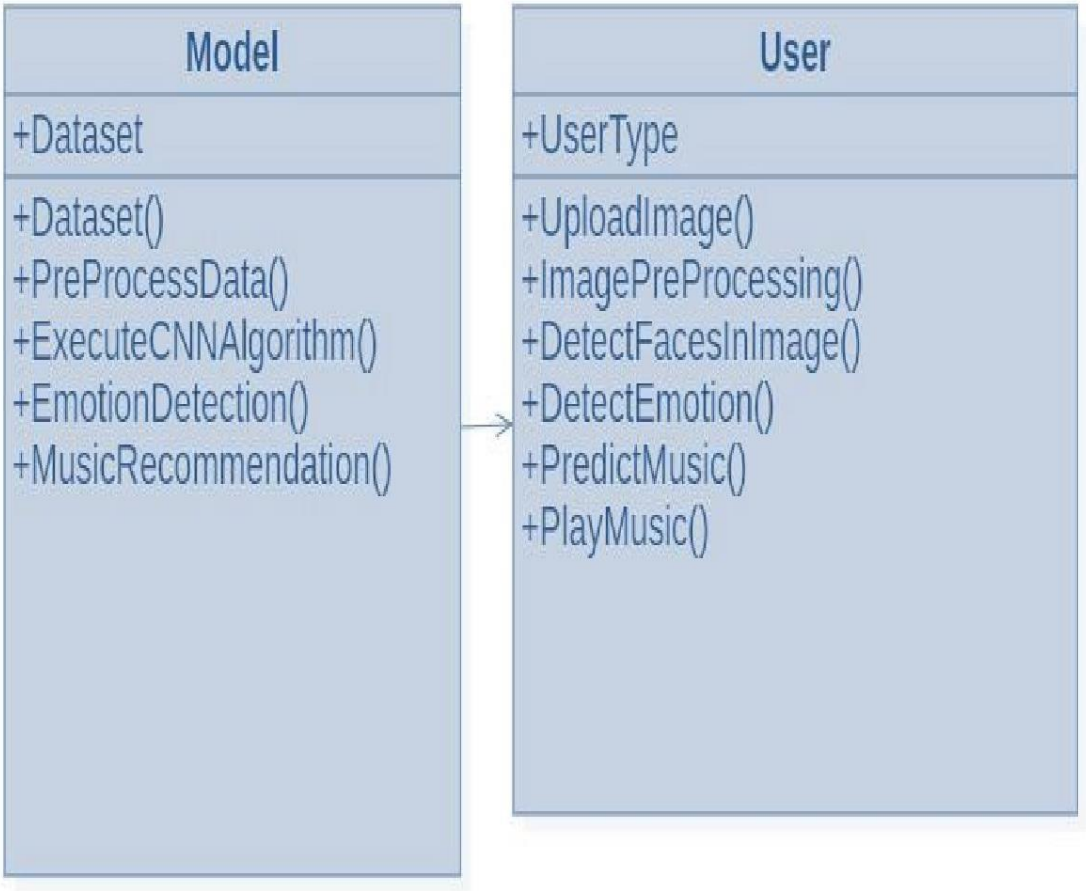

Fig.4.2.4 Class Diagram

4.2.5 Component Diagram

This diagram describes the organization and wiring of the physical components in a system. It shows the structure of the software system, which describes the software components, their interfaces, and their dependencies.

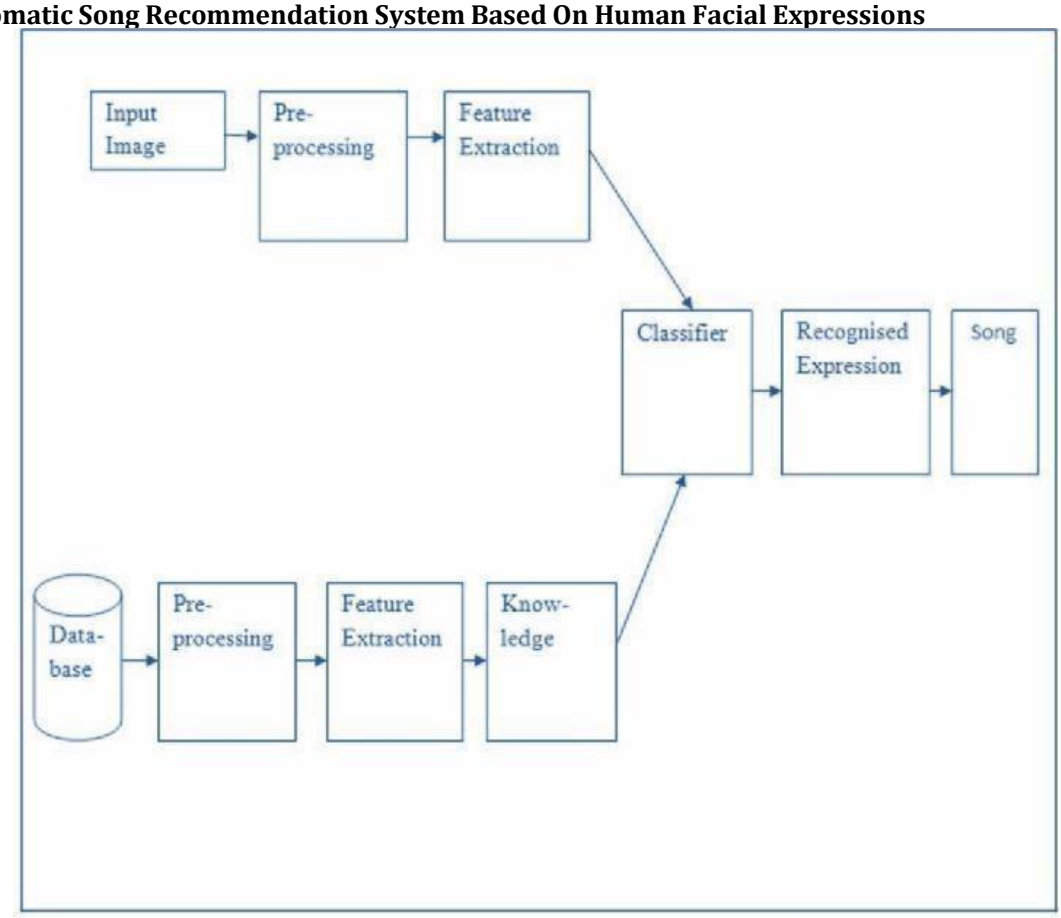

Fig.4.2.5 Component Diagram

CHAPTER - 5
SYSTEM IMPLEMENTATION

5. SYSTEM IMPLEMENTATION

5.1 MODULES

1. IMAGE DETECTION
2. IMAGE RECOGNITION
3. EMOTION DETECTION
4. MUSIC RECOMMENDATION

5.1.1 Image Detection

In this module an image is the input through camera or file selection process. This module detects images of people and displays its count.

5.1.2 Image Recognition

In this module image with faces will be recognized using image processing and haar cascade image classifier, which classifies image into positive and negative.

5.1.3 Emotion Detection

In this module emotions from the input image are detected like happy, sad, angry, surprised, scared and disgusted.

5.1.4 Music Recommendation

In this module music is recommended by the application based on the emotion detected.

5.2 SYSTEM REQUIREMENTS

5.2.1 HARDWARE REQUIREMENTS

RAM	:	4 GB
Processor	:	Intel i3 and above
Hard Disk	:	128 GB

5.2.2 SOFTWARE REQUIREMENTS

OS	:	Windows10
Programming Language	:	Python 5.2.1
IDE	:	VSCode 22

5.3 SOFTWARE ENVIRONMENT

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Tkinter:

Tkinter is the standard GUI library for python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI tool kit. Developing desktop-based applications with python Tkinter is not a complex task. An empty Tkinter top-level window can be created by using the following steps.

Steps for creating buttons and window:

1. import the Tkinter module.
2. Create the main application window.
3. Add the widgets like labels, buttons, frames, etc. to the window. 4. Call the main event loop so that the actions can take place on the user's computer screen.

VScode:

- Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favourite platform - Linux, macOS, and Windows.
- Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

- First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit- build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.
- Working with Python in Visual Studio Code is simple, fun, and productive. The extension makes VS Code an excellent Python editor, and works on any operating system with a variety of Python interpreters. It leverages all of VS Code's power to provide auto complete and IntelliSense, linting, debugging, and unit testing, along with the ability to easily switch between Python environments, including virtual and conda environments.

TensorFlow:

TensorFlow is an open-source software library. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

- TensorFlow library is used in many companies in the industries like Airbnb. This company applies machine learning using TensorFlow to detect objects and classify the images.
- In Python, if you want to move data through a graph then you can easily use the TensorFlow library for creating dataflow graphs. Basically, tensor is a matrix of n dimensions that represents the input type and flows works on the basis of flow graphs that have edges and nodes.

5.4 STUDY OF THE SYSTEM

1. NumPy
2. Pandas
3. Keras
4. IMUTILS

1. NumPy:

NumPy is a general-purpose array-processing package. It provides a High-Performance array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi- dimensional container of generic data.

2. **Pandas:**

Panda is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

3. **Keras:**

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Keras is the high-level API of TensorFlow 2: an approachable, highly- productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.

4. **IMUTILS:**

A series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, displaying Matplotlib images, sorting contours, detecting edges, and much more easier with OpenCV and both Python 2.7 and Py

CHAPTER - 6
TESTING

6. 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 tests. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTS

6.1.1 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. There are 4 units in the application and each unit is tested individually. The units are Image detection, Image Recognition, Emotion Detection, Music recommendation. After conducting unit testing for each and every individual module, all the given testcases are functionally passed. All the 4 modules are connected with each other and the same has been verified as well as validated with valid inputs. In this way, integration testing is successfully conducted for the project.

6.1.2 Functional Test

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 : Image (JPG, PNG, JPEG).
Invalid Input : Text, Docx, PPT.
Functions : Window, Upload, Preprocess, Detect emotion, Image to array, Play sound.
Output : Song Recommended.

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.

6.1.3 System Test

In system testing we evaluated the overall functionality and performance of a complete and fully integrated software solution. We tested if the system meets the specified requirements and if it is suitable for delivery to the end-users and we also uninstalled everything and checked if it working. System testing is done perfectly

All the requirements are satisfied and working in condition.

6.1.4 White Box Testing

In the whole project there are 4 if, else-if, else blocks. We have exercised all the if blocks with correct/true condition and verified the execution & control of the if-block. Similarly, we have exercised all the else blocks also by forcing execution & control in such directions. In this way, we conducted White Box testing successfully for the project.

6.1.5 Black Box Testing

S. No	Testcase ID	Testcase Scenario	Test Input	Expected Output	Actual Output	Test Result
1.	T_001	Image Recognition	Image	Image Detected:1	Image Detected:1	Pass
2.	T_002	Emotion Detection	Image	Emotion Detected: Happy	Emotion Detected: Happy	Pass
3.	T_003	Song Recommendation	Emotion as text	Song Recommended: Happy	Song Recommended: Happy	Pass
4.	T_004	Play Song	Song Recommend	Song Played	Song Played	Pass
5.	T_005	Emotion Detection	Image	Emotion Detected: Sad	Emotion Detected: Happy	Fail
6.	T_006	Emotion Detection	Image	Emotion Detected: Angry	Emotion Detected: Surprised	Fail
7.	T_007	Emotion Detection	Image	Emotion Detected: Surprised	Emotion Detected: Surprised	Pass

6.2 TEST STRATEGY AND APPROACH

Our aim was to uncover maximum number of bugs in minimum amount of time. We succeeded in this objective.

Test objectives

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format

No Duplicate entries should be allowed

All links should take the user to the correct page.

6.2.1 Integration Testing

All the 4 modules are connected with each other and the same has been verified as well as validated with valid inputs. In this way, Integration testing is successfully conducted for the project. The task of the integration test is to check that components or software applications

TEST RESULTS: All the test cases mentioned above passed successfully. All the failed cases are rectified. No defects encountered.

CHAPTER - 7
RESULTS

7. RESULTS

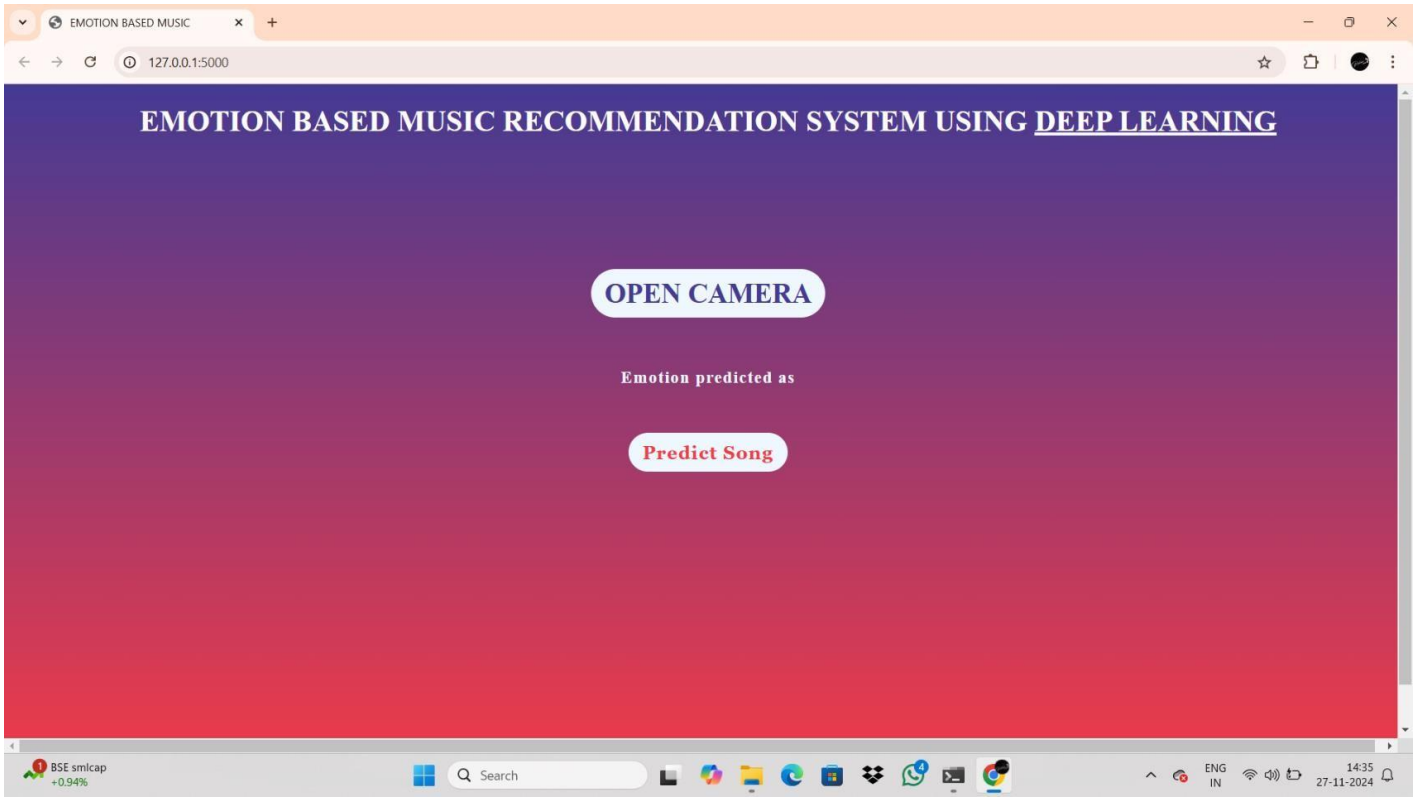

Fig. 7.1 Home page

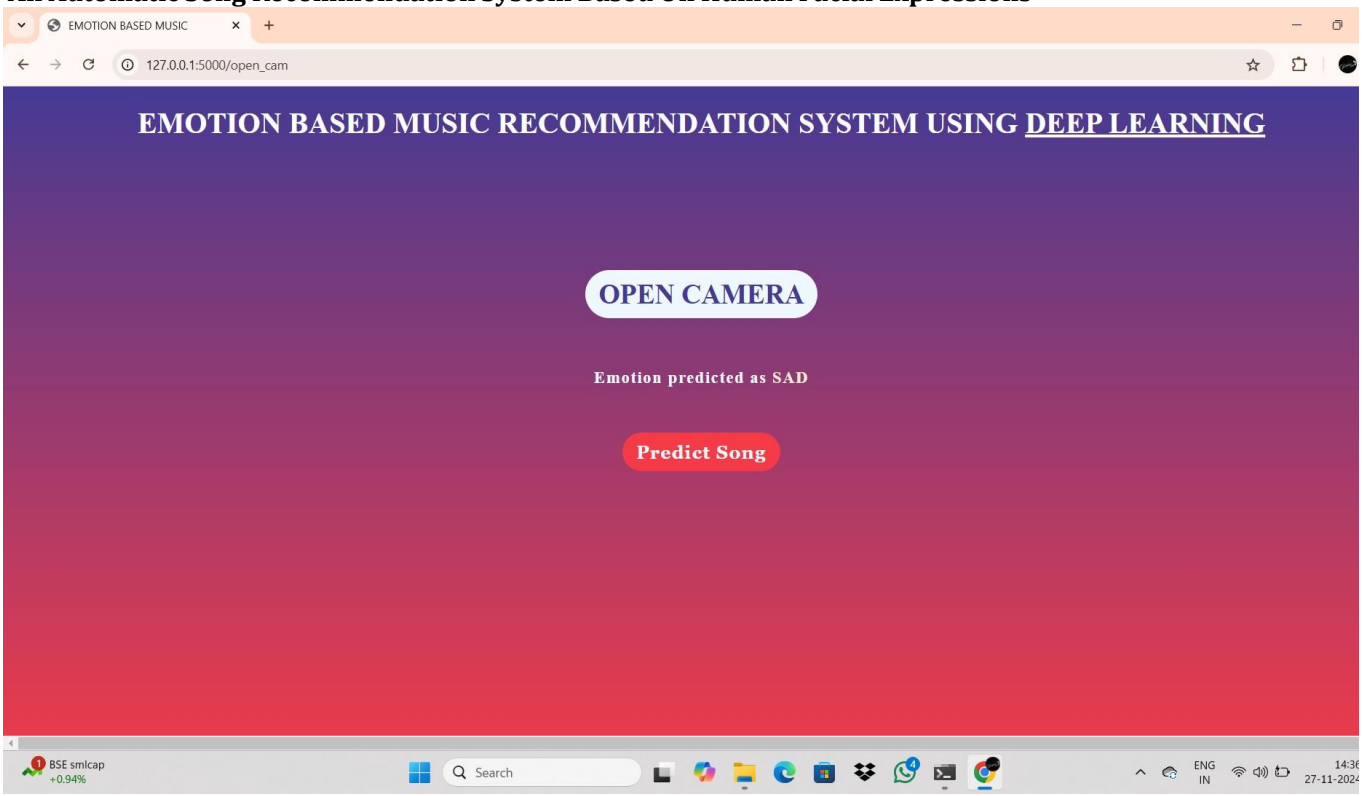

Fig. 7.2 Emotion Detected

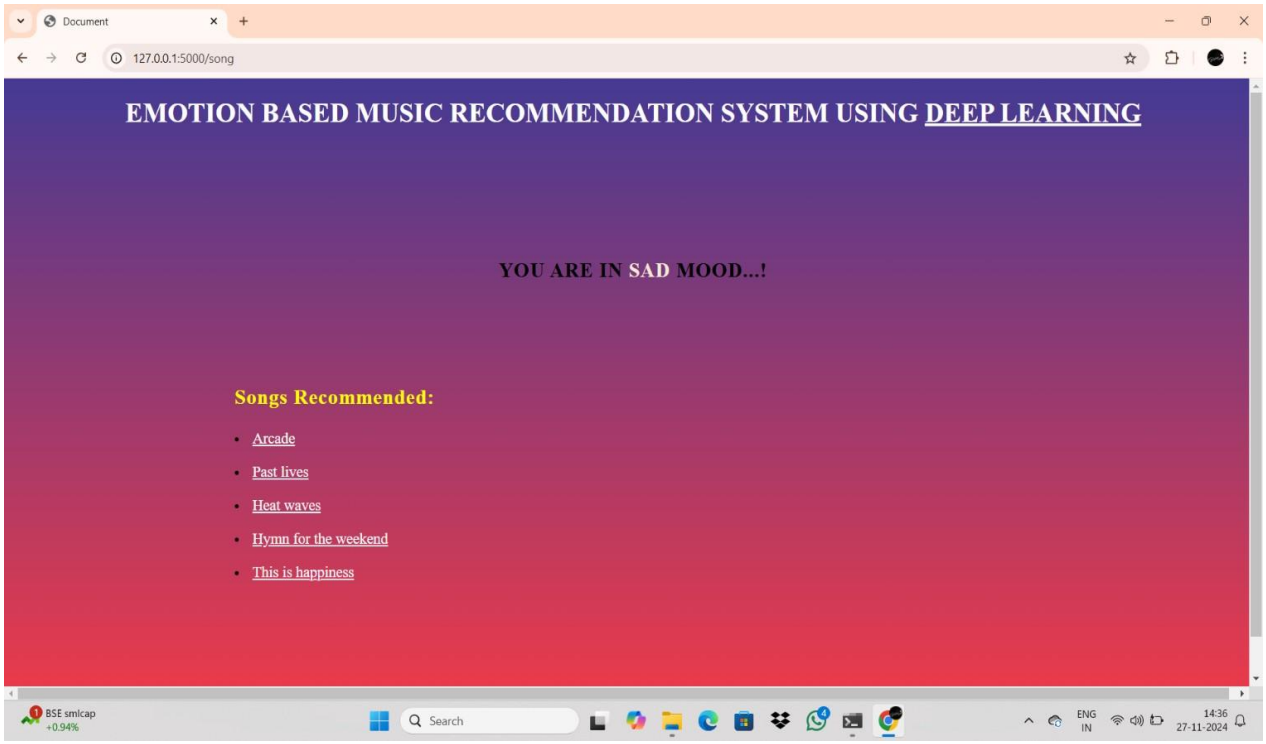

Fig. 7.3 SONGS RECOMMENDED

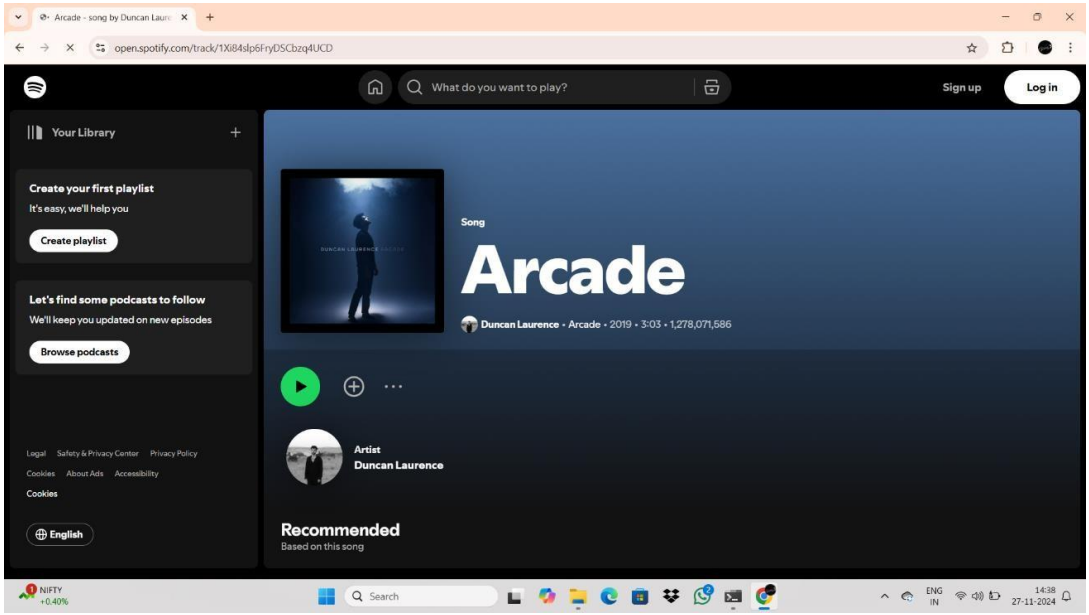

Fig. 7.4 Click on „Play Song" to play song

CHAPTER - 8
CONCLUSION
&
FUTURE WORK

8. CONCLUSION AND FUTURE WORK

In this project, a framework for enhancing music recommendation engines performance via physiological signals has been introduced. Emotion recognition from multi-channel physiological signals was performed, data fusion techniques were applied to combine data from GSR and PPG sensors and FLF has been implemented. Considering emotion state of the listener improves the performance of recommendations. Recognizing arousal and valence values directly from only GSR and PPG signals is a challenging task. We have showed that there is relationship between GSR and PPG signals and emotional arousal and valence dimensions. For GSR only signal, we have obtained 71.53% and 71.04% accuracy rate for arousal and valence prediction respectively. For photoplethysmography only signal, we have obtained 70.93% and 70.76% accuracy rate for arousal and valence prediction respectively. Fusing GSR and PPG signals we have obtained the results, 72.06% and 71.05% accuracy rate for arousal and valence prediction respectively. Although there is only slight improvement using fusion in emotion recognition accuracy, the proposed framework is promising for music recommendation engines in terms of adding multi modal emotion phenomenon into music recommendation logic. Performance can be improved with the advancement of wearable sensor technologies and using different type of sensors. Using more than one sensor may also help for failure management. As future work, we will consider different combination of sensors that handle the failures of wearable sensors and additional sensors usage to increase performance. The results of this study can be used to increase user experience of multimedia tools and music recommendation engines. Since there is high correlation between physiological GSR and PPG data and affective state and cognitive state of a person multimedia recommendation engines can benefit from physiological computing systems.

REFERENCES.

REFERENCES

1. S. Jhajharia, S. Pal, and S. Verma, "Wearable computing and its application," Int. J. Comp. Sci. and Inf. Tech., vol. 5, no. 4, pp. 5700– 5704, 2014.
2. K. Popat and P. Sharma, "Wearable computer applications: A feature perspective," Int. J. Eng. and Innov. Tech., vol. 3, no. 1, 2013.
3. P. Melville and V. Sindhwani, "Recommender systems," in Encyc. of mach. learn. Springer, 2011, pp. 829–838.
4. N. Sebe, I. Cohen, T. S. Huang et al., "Multimodal emotion recognition," Handbook of Pattern Recognition and Computer Vision, vol. 4, pp. 387– 419, 2005.
5. R. W. Picard, E. Vyzas, and J. Healey, "Toward machine emotional intelligence: Analysis of affective physiological state," IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 10, pp. 1175–1191, 2001.
6. D. Ayata, Y. Yaslan, and M. Kamasak, "Emotion recognition via galvanic skin response: Comparison of machine learning algorithms and feature extraction methods," IU J. of Elect. & Elect. Eng., vol. 17, no. 1, pp. 3129–3136, 2017.
7. P. Ekman, R. W. Levenson, and W. V. Friesen, "Autonomic nervous system activity distinguishes among emotions." Am. Assoc. for Adv. of Sci., 1983.
8. I.-h. Shin, J. Cha, G. W. Cheon, C. Lee, S. Y. Lee, H.-J. Yoon, and H. C. Kim, "Automatic stress- relieving music recommendation system based on photoplethysmography-derived heart rate variability analysis," in IEEE Int. Conf. on Eng. in Med. and Bio. Soc. IEEE, 2014, pp. 6402–6405.
9. S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao, "Musicalheart: A hearty way of listening to music," in Proc. of ACM Conf. on Emb. Netw. Sens. Sys. ACM, 2012, pp. 43–56.
10. H. Liu, J. Hu, and M. Rauterberg, "Music playlist recommendation based on user heartbeat and music preference," in Int. Conf. on Comp. Tech. and Dev., vol. 1. IEEE, 2009, pp. 545– 549.

APPENDIX

SOURCE CODE

```
from flask import Flask,redirect,url_for,render_template,request
import numpy as np
from keras.models import model_from_json
import cv2

app=Flask(__name__)

key = cv2.waitKey(1)

json_file = open("facialemotionmodel.json", "r")
model_json = json_file.read()
json_file.close()
model = model_from_json(model_json)

model.load_weights("facialemotionmodel.h5")
haar_file=cv2.data.haarcascades + 'haarcascade_frontalface_default.xml'
face_cascade=cv2.CascadeClassifier(haar_file)

def extract_features(image):
    feature = np.array(image)
    feature = feature.reshape(1,48,48,1)
    return feature/255.0
```

```
"""@app.route('/',methods=['GET','POST'])
def home():
        return render_template('login.html')

@app.route('/logincheck',methods=['GET','POST'])
def logincheck():
    if request.method=='POST':
        username = request.form["username"]
        password = request.form["password"]

        if username == "emotion" and password == "emotion":
            return render_template("prediction.html")
        else:
            return render_template("index.html")
    return render_template('index.html')


@app.route('/sign',methods=['GET','POST'])
def sign():
    return render_template('login.html')"""

@app.route('/',methods=['GET','POST'])
def prediction():
    return render_template('prediction.html')

@app.route('/happy',methods=['GET','POST'])
def happy():
```

```
    return render_template('happy.html')


@app.route('/sad',methods=['GET','POST'])
def sad():
    return render_template('sad.html')


@app.route('/angry',methods=['GET','POST'])
def angry():
    return render_template('angry.html')


@app.route('/nutral',methods=['GET','POST'])
def nutral():
    return render_template('neutral.html')


@app.route("/open_cam")
def open_cam():
    webcam=cv2.VideoCapture(0)
    labels = {0 : 'angry', 1 : 'disgust', 2 : 'fear', 3 : 'happy', 4 : 'neutral', 5 : 'sad', 6 :
'surprise'}
    while True:
        i,im=webcam.read()
        gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
        faces=face_cascade.detectMultiScale(im,1.3,5)
        try:
            for (p,q,r,s) in faces:
                image = gray[q:q+s,p:p+r]
                cv2.rectangle(im,(p,q),(p+r,q+s),(255,0,0),2)
                image = cv2.resize(image,(48,48))
```

```
            img = extract_features(image)
            pred = model.predict(img)
            prediction_label = labels[pred.argmax()]
            filename = prediction_label
            print("Predicted Output:", prediction_label)

            # cv2.putText(im,prediction_label)
            cv2.putText(im, '% s' %(prediction_label), (p-10, q-
10),cv2.FONT_HERSHEY_COMPLEX_SMALL,2, (0,0,255))
            cv2.waitKey(27)
            return render_template("prediction.html" ,e = prediction_label)
        except cv2.error:
            pass

@app.route('/song',methods=['GET','POST'])
def song():
    if request.method=='POST':
        emotion = request.form["emotion"]
        if(emotion == "happy"):
            return render_template("happy.html" , em = emotion)
        elif(emotion == "sad"):
            return render_template("sad.html" , em = emotion)
        elif(emotion == "angry"):
            return render_template("angry.html" , em = emotion)
        elif(emotion == "neutral"):
            return render_template("neutral.html" , em = emotion)
        elif(emotion == "fear"):
            return render_template("fear.html" , em = emotion)
```

```
        elif(emotion == "surprise"):
            return render_template("surprise.html" , em = emotion)
        elif(emotion == "disgust"):
            return render_template("disgust.html" , em = emotion)
    return render_template("index.html")

if__name__== '__main__':
    #DEBUG is SET to TRUE. CHANGE FOR PROD
    app.run(port=5000,debug=True)
```

PAPER PUBLICATION