Emotion Based Music Recommendation System using Machine Learning

PROJECT REPORT

Submitted by

ARVIND SRINIVAS K G	**2007006**
HARESH V	**2007015**
SHRUTIKA PAWAR S	**2107206**

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

COIMBATORE INSTITUTE OF TECHNOLOGY, COIMBATORE-641014

(Government Aided Autonomous Institution Affiliated to Anna University)

ANNA UNIVERSITY, CHENNAI 600025

JUNE 2023

COIMBATORE INSTITUTE OF TECHNOLOGY

(A Govt. Aided Autonomous Institution Affiliated to Anna University)

COIMBATORE – 641014

BONAFIDE CERTIFICATE

Certified that this project report titled "**EMOTION BASED MUSIC RECOMMENDATION SYSTEM**" is the bonafide work of **ARVIND SRINIVAS K G (2007006), HARESH V (2007015)** and **SHRUTIKA PAWAR (2107206)** in partial fulfillment for the award of the Degree of Bachelor of Technology in Information Technology of Anna University, Chennai during the academic year 2020-2021 under my supervision.

Prof.N.K.KARTHIKEYAN,
HEAD OF THE DEPARTMENT,
Department of Information Technology,
Coimbatore Institute of Technology,
Coimbatore - 641014.

Mrs.POORNIMA S,
ASSISTANT PROFESSOR,
Department of Information Technology,
Coimbatore Institute of Technology,
Coimbatore - 641014.

Certified that the candidates were examined by us in the project work viva-vice examination held on …………………

Internal Examiner **External Examiner**

Place:

Date:

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Our project **"EMOTION BASED MUSIC RECOMMENDATION SYSTEM"** has been the result of motivation and encouragement from many, whom we would like to thank. We express our sincere thanks to our Secretary **Dr.R.Prabhakar** and our Principal **Dr.RAJESWARI** for providing us a great opportunity to carry out our work. The following words are rather very meagre to express our gratitude to them. This work is the outcome of their inspiration and product of plethora of their knowledge and rich experience.

We record the deep sense of gratefulness to **Dr.N.K.Karthikeyan**, Head of the Department of Information Technology, for his encouragement and support during this tenure. We equally tender my sincere gratitude to our project guide **Mrs.S.Poornima,** Department of Information Technology, for her valuable suggestions and guidance during this course.

During the entire period of study, the entire staff members of the Department of Computer Science and Engineering & Information Technology have offered ungrudging help. It is also a great pleasure to acknowledge the unfailing help we have received from our friends, parents and family members for their constant support and cooperation in the pursuit of this Endeavour.

ABSTRACT

The proposed abstract highlights the intricate relationship between music, human emotions, and brain function, with a focus on understanding how individuals respond to music based on their mood and interests. The abstract introduces the concept of an Emotion Detection Music Recommendation System (EDMRS), which aims to utilize facial expressions as a means to determine a person's emotional state. To implement the EDMRS, the project suggests employing computer vision technology along with a camera to capture a live stream of the user's face. Within the realm of computer vision and machine learning, significant research is being conducted to train machines in recognizing and understanding various human emotions or moods, employing diverse algorithms for this purpose. The proposed system primarily relies on computer vision components to analyze the user's facial expressions. By extracting relevant features from facial landmarks such as eye movements, mouth shape, and brow position, the system can discern the user's emotional state. This process of feature extraction utilizing facial landmarks significantly contributes to determining the user's mood or level of emotion. By leveraging computer vision techniques within the EDMRS, the project aims to delve into the profound impact of music on human emotions. It seeks to analyze facial expressions and extract emotional cues from facial landmarks in real-time, allowing for a better understanding of the user's emotional state. This research holds significant potential in providing valuable insights into how music affects our emotions and may have applications in fields such as psychology, music therapy, and user experience design. The ultimate goal of the EDMRS is to establish a strong connection between music and emotions, enabling personalized music recommendations that align with the user's emotional needs. This personalized approach can offer a transformative and engaging music listening experience, enhancing the user's emotional well-being and satisfaction.

In summary, the proposed research project explores the interconnectedness of music, human emotions, and brain function by employing computer vision techniques to analyze facial expressions. The EDMRS has the potential to revolutionize music recommendations by incorporating real-time emotional analysis, benefiting fields such as psychology, music therapy, and user experience design.

Keywords : *Computer vision, Facial Landmarks, Feature extraction, Emotion detection*

CHAPTER-1

INTRODUCTION

1.1 Need of Music in our day-to-day life

1.1.1 Music brings people together

One of the most important benefits of music is its ability to create a sense of belonging between individuals. Music has been linked to forming both social and familial bonds.

1.1.2 Music improves your health and well being

Music enhances mental and physical health, alleviating pain, reducing stress, and promoting relaxation, leading to improved overall well-being.

1.1.3 Music can improve confidence and resilience

Learning music builds discipline, perseverance, and self-confidence, while ensemble participation cultivates teamwork and leadership skills.

1.1.4 Music is a creative outlet

Music serves as a creative outlet for individuals from diverse backgrounds, enabling them to convey emotions, messages, and ideas.

1.1.5 Music is fun!

The sheer fun and enjoyment derived from music uplift spirits, fostering a positive atmosphere for all involved.

1.1.6 Understanding user's emotion

Music has the remarkable ability to evoke and amplify emotions within us. However, accurately discerning the emotional state of individual listeners has long been a challenge. ML comes to the forefront by providing advanced algorithms capable of analyzing facial expressions and recognizing patterns associated with various emotions. By combining ML with music, we can bridge the gap between emotions and music, enabling systems to detect and understand user emotions more effectively.

1.2 Machine learning:

Machine learning is one of the key application fields in thriving Artificial Intelligence (AI). Machine learning has included ready to learn and upgrade the execution of the system utilizing past experience. It doesn't need a particular programming to do as such.
Programming in machine learning focuses on getting to information and learning by utilizing the own information. It explicitly utilizes data or information like past models, direct insight, guidelines and required example in information is recognized to take better choices for development of framework in future. Automated learning, evasion of human intervention, performance improvement and suitable changes in activities of system are the key features what's more, attributes of machine learning. Deep learning is a subset of machine learning.

Performance measure in numerous useful applications, for example image recognition, sound recognition and so forth is extraordinarily improved when deep learning is used.

1.3 Image recognition:

Image or Object Detection is a computer technology that processes the image and identifies objects in it. Individuals frequently mistake Image Detection for Image Classification. If you need to classify image items, you use Classification. But if you just need to locate them, for example, find out the number of objects in the picture, you should use Image Detection. Image recognition is the ability of AI to detect the object, classify, and recognize it.
The last step is close to the human level of image processing. Image recognition is the ability of a system or software to identify objects, people, places, and actions in images. It uses machine vision technologies with artificial intelligence and trained algorithms to recognize images through a camera system.

CHAPTER 2
LITERATURE SURVEY

2.1 “Facial emotion detection using modified eye map–mouth map algorithm on an enhanced image and classification with TensorFlow”

This paper was proposed by Allen Joseph and P Geetha. They publisher it on 13 February 2019 at Springer-Verlag GmbH Germany, part of Springer Nature 2019.

The project "Facial emotion detection using modified eye map–mouth map algorithm on an enhanced image and classification with TensorFlow" focuses on developing a system for accurately detecting facial emotions. It utilizes a modified algorithm, the eye map–mouth map algorithm, to analyze specific facial regions and extract emotion-related features. Image enhancement techniques are applied to improve the quality of input images. The system employs TensorFlow for classification, enabling the training of a model for accurate emotion recognition. This project has potential applications in human-computer interaction, affective computing, and fields such as social robotics, virtual reality, and mental health monitoring.

2.2 “Deep Learning based Facial Expression Recognition using Keras”

Using this algorithm, up to five distinct facial emotions can be detected in real time. It runs on top of a Convolutional Neural Network (CNN) that is built with the help of Keras whose back end is TensorFlow in Python. The facial emotions that can be detected and classified by this system are Happy, Sad, Anger, Surprise and Neutral. OpenCV is used for image processing tasks where a face is identified from a live webcam feed which is then processed and fed into the trained neural network for emotion detection. Deep learning based facial expression recognition techniques bring down to a greater extent, the dependency on face-physics-based models and other pre- processing techniques by enabling lengthwise learning to occur in the pipeline directly from the input images.

2.3 “Facial Emotion Based Music Recommendation System using computer vision and machine learning techniques”

This paper was proposed by ShanthaShalini. K, Jaichandran. R , Leelavathy. S A, Raviraghul. R , Ranjitha. J and Saravanakumar. N on 5th April 2021 from Department of Computer Science and Engineering, AarupadaiVeedu Institute of Technology, Vinyaka Missions Research foundation (Deemed to be University), Paiyanoor, Tamil Nadu, India.

The "Facial Emotion Based Music Recommendation System" combines computer vision and machine learning to recommend music based on facial emotions. It analyzes facial expressions, extracts emotional cues, and classifies the user's emotional state using machine learning algorithms. The system then suggests music that aligns with the user's emotions by matching them with an annotated music database. This personalized and interactive system enhances the user experience by adapting the music selection in real-time. It has potential applications in personalized music streaming, mood-based playlists, and improving user experiences in entertainment and wellness domains.

2.4 “Human emotion based music player using ML”

The paper titled "Human Emotion based Music Player using ML" by Dhruv Patel, Heet Patel, Nilay Shah, and Shail Patel proposes a music player that plays music based on the user's current emotional state. The paper begins by discussing the importance of music in our daily lives and how it can impact our emotions. It then goes on to discuss the limitations of current music players that do not take into account the emotional state of the user.

The proposed music player utilizes machine learning to analyze facial expressions and determine the user's emotional state. It captures the user's face through a camera and applies a machine learning algorithm to detect emotions like happiness, sadness, anger, and fear. Based on the detected emotions, the music player selects suitable music for the user. The paper reviews related work in emotion recognition using facial expressions and discusses limitations in current systems. It suggests using physiological sensors alongside facial expression analysis for improved emotion detection. Overall, the paper presents an innovative approach to music player design that offers personalized music recommendations based on the user's emotions. Future work includes enhancing emotion recognition accuracy and integrating physiological sensors into the system.

2.5 “Automatic facial expression related emotion recognition using machine learning techniques.”

The paper titled "Automatic Facial Expression Related Emotion Recognition using Machine Learning Techniques" by V. Sathya and T. Chakravarthy presents a comprehensive survey of the state-of-the-art techniques for automatic facial expression recognition and emotion recognition using machine learning algorithms.

The paper discusses the significance of facial expression recognition in various domains and provides an overview of the recognition process. It reviews different techniques for facial expression and emotion recognition, including geometric features, appearance-based features, and machine learning algorithms. The authors compare the techniques based on accuracy, speed, and complexity, and highlight their limitations. They suggest future research directions to enhance the accuracy and integration of these techniques into real-world applications. Overall, the paper presents a comprehensive survey of facial expression and emotion recognition, emphasizing their importance and potential for advancements in fields like social robotics and human-computer interaction.

2.6 “Development of a Real-Time Emotion Recognition System Using Facial Expressions and EEG based on machine learning and deep neural network methods”

The paper titled "Development of a Real-Time Emotion Recognition System Using Facial Expressions and EEG based on Machine Learning and Deep Neural Network Methods" by Aya Hassouneh, A.M. Mutawa, and M. Murugappan presents a literature survey on the various techniques used for real-time emotion recognition using facial expressions and EEG signals.

The paper reviews the significance of emotion recognition in healthcare, entertainment, and human-computer interaction. It discusses techniques for facial expression recognition, physiological signal analysis, and multi-modal approaches. Various machine learning algorithms such as Support Vector Machines, Artificial Neural Networks, and Decision Trees are explored for classification. The paper also examines EEG signal analysis techniques, including time-domain, frequency-domain, and time-frequency domain features, along with machine learning algorithms. Multi-modal approaches combining facial expressions and EEG signals are discussed, including fusion techniques and classification algorithms. A comparative analysis of real-time emotion recognition techniques based on accuracy, speed, and complexity is provided. The paper concludes by suggesting future research directions and the integration of these techniques into practical applications.

2.7 “Emotion Based Music Player”

The paper titled "Emotion-Based Music Player" by Mr. M Sudhagar and Shamala Tejaswini presents a literature survey on the various techniques used for emotion-based music recommendation systems.

The paper discusses the significance of music in influencing emotions and presents an overview of techniques used in music recommendation systems. It explores emotion recognition techniques, including facial expression analysis and physiological signal analysis, along with machine learning algorithms employed in emotion recognition. The paper then reviews approaches for emotion-based music recommendation systems, such as rule-based systems and content-based and collaborative filtering methods. Factors impacting system accuracy are examined, including data quality, feature selection, and machine learning algorithms. The authors compare these techniques based on accuracy, speed, and complexity, acknowledging limitations and proposing future research directions. The paper emphasizes the importance of advancing emotion-based music recommendation systems for practical applications.

2.8“A real time face emotion classification and recognition using deep learning model”

The paper titled "A Real-Time Face Emotion Classification and Recognition Using Deep Learning Model" by Dr. Shaik Asif Hussain and Ahlam Salim Abdallah Al Balushi presents a literature survey on the various techniques used for real-time face emotion classification and recognition using deep learning models.

The paper discusses the significance of emotion recognition in various fields and provides an overview of techniques used, including facial expression recognition, physiological signals analysis, and multimodal approaches. It reviews different methods for facial expression recognition, such as geometric, appearance-based, and hybrid features, along with deep learning models like CNNs, RNNs, and LSTMs. The authors explore real-time face emotion recognition using deep learning models, highlighting challenges and models like Lightweight CNNs, MobileNet, and SqueezeNet. They analyze datasets used for training and evaluating models, emphasizing the need for more diverse and larger datasets. The paper concludes with a detailed comparison of techniques based on accuracy, speed, and complexity, along with suggestions for future research. Overall, it provides a comprehensive survey of real-time face emotion classification and recognition techniques using deep learning models.

2.9 “Facial Expression Classification Based on SVM, KNN, MLP”

The paper titled "Facial Expression Classification Based on SVM, KNN, MLP" by Hivi Ismat Dino and Maiwan Bahjat Abdulrazzaq presents a literature survey on the various techniques used for facial expression classification using machine learning algorithms.

The paper discusses the significance of facial expression recognition in various domains and provides an overview of techniques used in this area. It reviews different machine learning algorithms, including SVM, KNN, and MLP, for facial expression classification, highlighting their advantages and limitations. Feature extraction and selection methods like LBP, HOG, and PCA are also examined, along with their impact on algorithm performance. The paper concludes with a comparative analysis of these algorithms based on accuracy, speed, and complexity, emphasizing the need for further research. The authors suggest future directions for improving accuracy, robustness, and integration of facial expression recognition in real-world applications like human-computer interaction and security.

2.10 "FaceNet2ExpNet: Regularizing a Deep Face Recognition Net for Expression Recognition"

The paper "FaceNet2ExpNet: Regularizing a Deep Face Recognition Net for Expression Recognition" by Hui Ding, Shaohua Kevin Zhou, and Rama Chellappa presents a literature survey on the different techniques used for facial expression recognition using deep learning algorithms.

This paper provides an overview of facial expression recognition and explores deep learning techniques used in this field. It discusses different approaches and architectures, such as geometric-based, appearance-based, CNNs, RNNs, and DBNs. The authors introduce the FaceNet2ExpNet framework, which adapts the FaceNet architecture for expression recognition by incorporating regularization constraints. They highlight the framework's potential and suggest future directions, including multimodal data integration and improved regularization methods. In summary, the paper surveys facial expression recognition techniques, presents the FaceNet2ExpNet framework, and emphasizes the need for further advancements in deep learning architectures for accurate and efficient facial expression recognition.

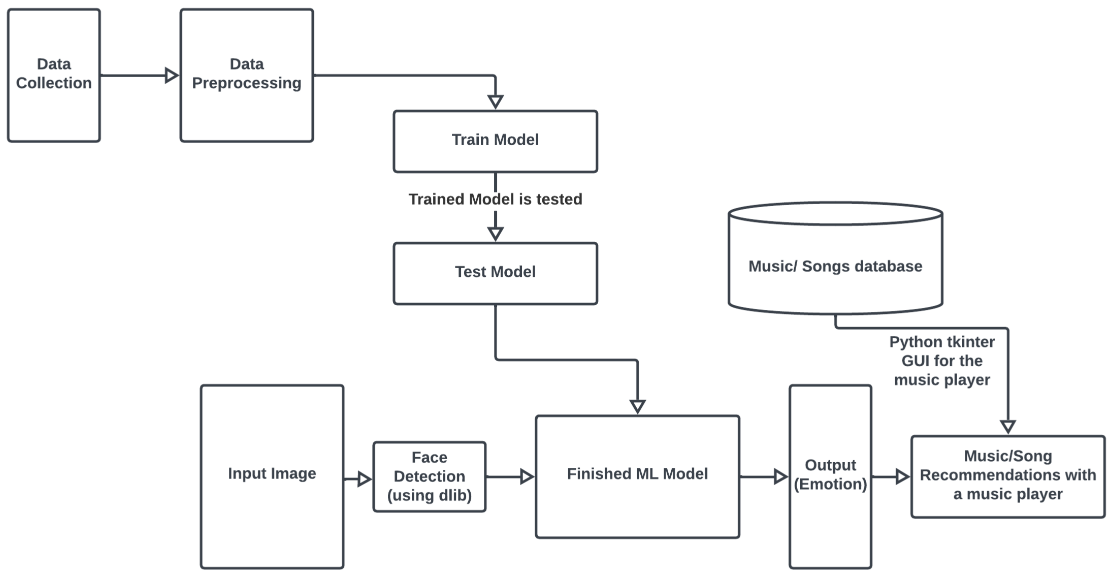
CHAPTER 3

SYSTEM ARCHITECTURE

3.1 PROPOSED SYSTEM

The emotion-based music recommendation system is an application that focusses on implementing real time emotion detection. It is a prototype which consists of two main modules: Facial expression recognition/emotion detection and Music Recommendation.

(The architecture of the system can be seen below)



Data Collection:

The Fer2013 dataset is a public dataset of facial expressions that was created for the purposes of facial expression recognition research. The dataset consists of 35,887 grayscale images, each of size 48x48 pixels, which represent facial expressions of seven different emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral. The images were collected from the Internet using keyword searches and then labeled by humans using Amazon's Mechanical Turk service.

Data Processing:

All the pre-processing required for the model has already been done on the dataset i.e., the images' sizes are 48x48 pixels, grayscale which are suitable for training with an ML model so the model will be more generalized.

Train Model:

The model was trained using the above dataset. The images were randomised and an 80:20 train-test split was used.

Test Model:

The testing of model involves feeding the testing data-set into the trained model and observing its predictions. The model's output is compared against the ground truth labels or values to assess its performance as well as the accuracy and loss plots were obtained.

Input Image:

Facial image of the user captured by a camera or provided as input for emotion detection and analysis.

Face Detection:

Face detection is the first step in EDMRS, the user's face is captured and co-ordinates of the face in the picture has been returned to the emotion detection module.

Finished ML Model:

The finished ML model takes in the co-ordinates of the boxes drawn around a detected face by the face detection algorithm and classifies it to its respective emotion.

Music/ song database:

EDMRS, for now uses the local storage for storing all the songs. The songs have been stored under the emotion labels of the characteristic emotion of that song.

Music/ song Recommendations with a music player:

A basic music player for the detected emotion, with basic functionalities like play, pause, next, prev.

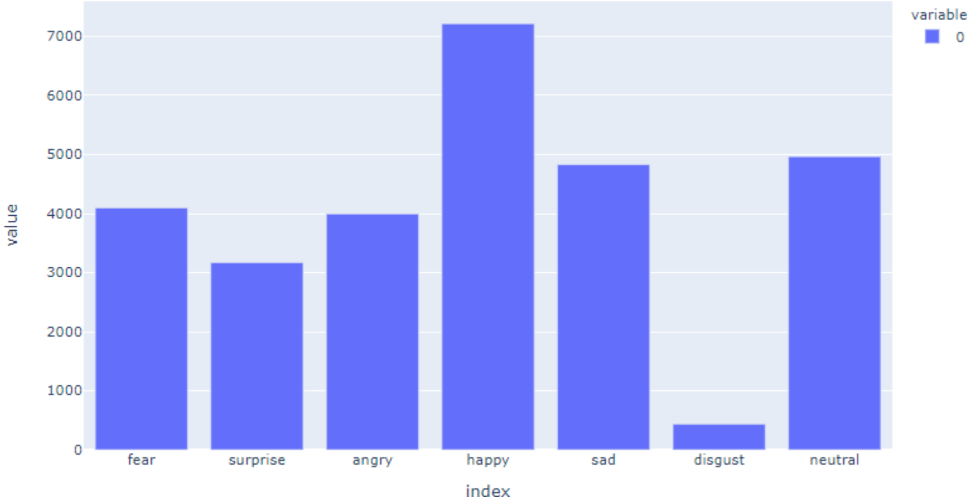
3.1.1 DATASET DESCRIPTION

The Fer2013 dataset is a public dataset of facial expressions that was created for the purposes of facial expression recognition research. The dataset consists of 35,887 grayscale images, each of size 48x48 pixels, which represent facial expressions of seven different emotions: anger, disgust, fear, happiness, sadness, surprise, and neutral.

The dataset is divided into three subsets: a training set with 28,709 images, a public test set with 3,589 images, and a private test set with 3,589 images. The private test set is only available to researchers who have submitted their results on the public test set to avoid over fitting.

(Visualization of dataset)

The images are presented in CSV format, where each row represents an image and its corresponding emotion label. The first column contains the emotion label, which is an integer value ranging from 0 to 6, representing the seven different emotions. The second column contains the pixels of the image, which are represented as a string of comma-separated integers. The pixel values range from 0 to 255, with 0 representing black and 255 representing white.The dataset has been used extensively in research on facial expression recognition and machine learning. However, some researchers have noted that the Fer2013 dataset has limitations, such as the lack of diversity in the age, ethnicity, and gender of the subjects, and the fact that the images are mostly frontal-facing and do not capture subtle facial expressions. Nonetheless, the dataset remains a valuable resource for researchers in this field.



(Distribution of images among various classes/emotions can be visualized)

CHAPTER 4
SYSTEM SPECIFICATION

This chapter includes the System Specification of our project.

The hardware and software for the system is selected by considering the factors such as CPU processing speed, peripheral channel speed, printer speed, seek time, relational delay of hard disk and communication speed etc. The hardware and software specifications are as follows.

4.1 Hardware Requirements:

RAM	4GB (minimum)
Memory	30MB (Approx)
Camera	Web Camera
Monitor	Laptop or System Monitor
Power Source	5 V

Table 4.1 Hardware requirements

4.2 Software Requirements:

Operating System	Any OS
IDE	Visual Studio Code
Language	Python 3.9
Library	SDL/ PyGame

Table 4.2 Software requirements

CHAPTER 5
DESIGN & IMPLEMENTATION

This chapter explains the design and implementation of various emotions.

The system is composed of the following modules

1. Emotion detection module

    1.1 Face Detection

    1.2 Emotion Detection

2. Music recommendation module

5.1 Emotion Detection Module

- Face Detection - Ability to detect the location of face in any input image or frame. The output is the bounding box coordinates of the detected faces. For this task, OpenCV is used along with the dlib library for face detection instead of the traditional Haar Cascade classifier because, dlib is faster and more accurate than Haar Cascade classifier.

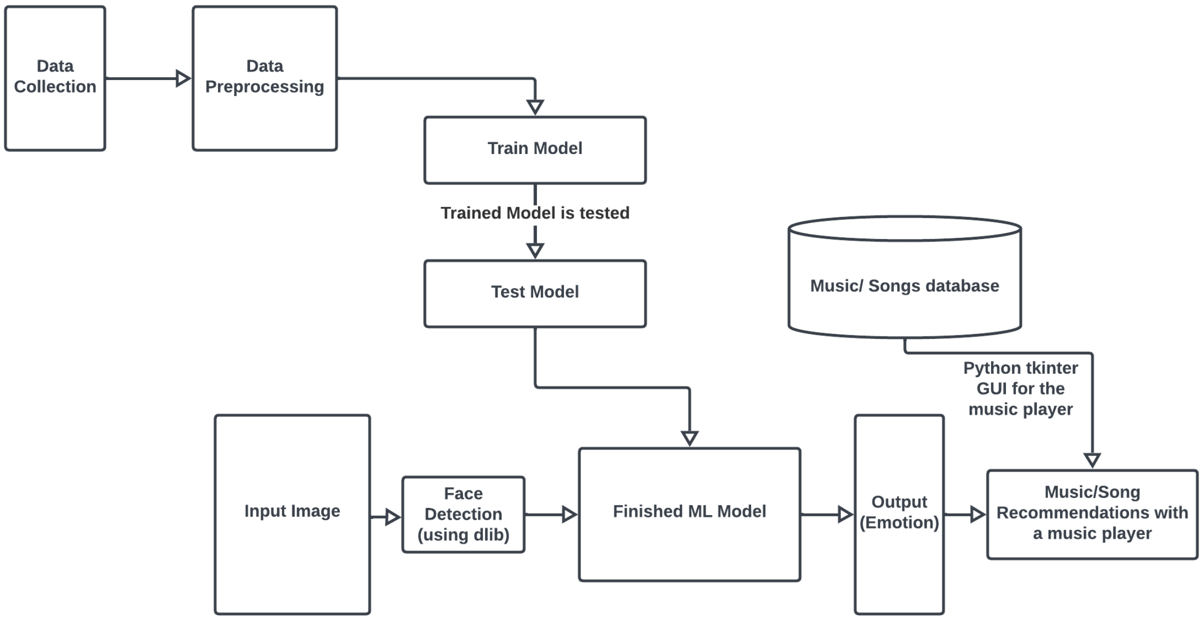
- Emotion Detection — Classification of the emotion on the face as happy, angry, sad, neutral, surprise, fear or disgust. For this task, the traditional Keras module of Python is used. low computational efficiency without compromising the accuracy of the results. It uses depth wise separable convolutions to build light weight deep neural networks. The dataset used for training was obtained by combining FER 2013 dataset from Kaggle. The FER 2013 dataset contained grayscale images of size 48x48 pixels. Keras was used to train and test our model for seven classes - happy, angry, neutral, sad, surprise, fear and disgust. We trained it for 15 epochs and achieved an accuracy of approximately 65%.

5.2 Music recommendation module

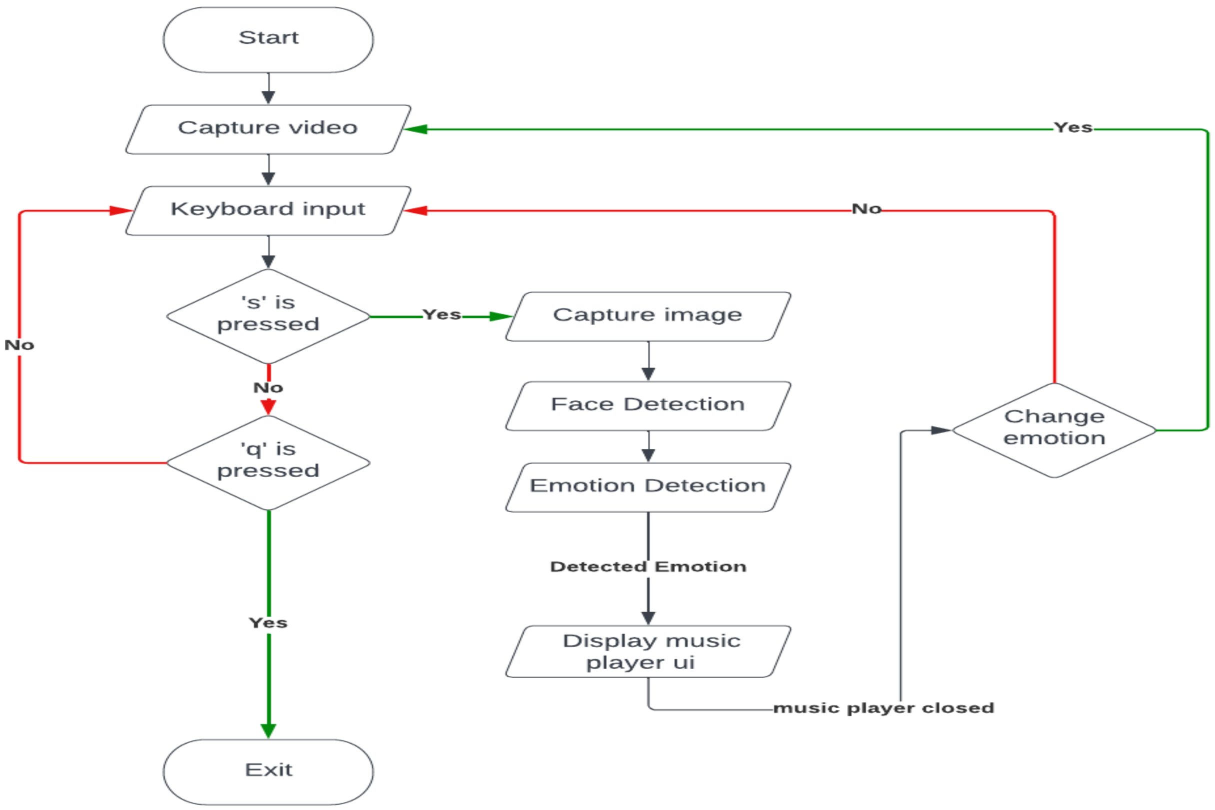
For this module, packages such as tkinter, pygame of Python are used to build a basic music player which plays music from the local storage based on the emotion label that is returned by the previous module.

5.3 Integration

For the integration of these two modules, the trained model architecture and the model weights are stored in .json and .h5 formats respectively, and it is served in the driver program. The driver program uses packages such as OpenCV, numpy, dlib, tkinter, pygame, PIL etc for various tasks. (Fig.1) shows the system architecture of our project. the Driver program captures the user’s video using OpenCV, dlib package is used for the face detection module and the faces are sent to the emotion detection module in order for emotion prediction. After the emotion is predicted, suitable songs from the local storage are shown as in the form of a Music Player GUI using the tkinter package.



(Fig.1)



(Fig.2)

Fig.2 illustrates our model's data flow diagram, explained step-wise below:

Step 1: The get_image() is called, this function is used to capture frames from the video feed and process them. The frames are converted to grayscale and faces are detected using Dlib's detector.

Step 2: The faces are then cropped and resized to match the input size of the per-trained Keras model. The model is used to predict the emotion for each face, and the predicted emotion is added to a list.

Step 3: The list of predicted emotions is then used to calculate the mode of the emotions, which is used as the final predicted emotion. The final predicted emotion is then displayed on the video feed and stored in the final_label variable.

Step 4: Set up Pygame mixer and define a folder containing songs for each emotion.

Step 5: Create a list of all the songs in the folder and initialize the current song index.

Step 6: Define several functions to play, pause, resume, skip to the next, and go back to the previous song.

Step 7: When a new emotion is predicted, the song folder is changed to the folder containing the songs to the appropriate folder based on the predicted emotion and plays the first song in the folder, also update the name of the song being played on the Pygame window.

5.4 Functions

5.4.1 Face Detection Module

Input: Image/live video camera feed

Output: Captured image with rectangle box drawn around the face

Begin

Step 1: Capture the image

Step 2: Convert the image to grayscale Step 3: Load the face cascade classifier

Step 4: Detect faces in the image

Step 5: Draw a rectangle around each face

Step 6: Display the image with the detected faces

End

5.3.2 Emotion Detection Module

Input: Images captured by the system with faces detected i.e., output of previous module

Output: Emotion Label

Begin

Step 1: Takes the in the gray-scaled face images from the previous module

Step 2: The machine learning model is already trained to predict the emotions from the person’s facial features.

Step 3: The images is analyzed and based on the matching features, the percentage of matching with every emotion label (i.e., Angry, Disgust, Fear, Happy, Neutral, Sad, Surprise) is returned in an np array format.

Step 4: The appropriate emotion label is returned i.e., the emotion label with the highest matching percentage is returned.

End

5.3.3 Music Recommendation Module

Input: Emotion Label from previous module

Output: Appropriate song based on emotion is played

Begin

Step 1: Get the emotion label from previous model.

Step 2: In the Songs directory, look for the directory with the same name as the emotion label.

Step 3: Open the directory and fetch the songs/music in that directory.

Step 4: Play the song with the functionality to play, pause, go to next song, go to previous song, etc..

End

CHAPTER 6

RESULTS

6.1 SNAP SHOTS OUTPUT

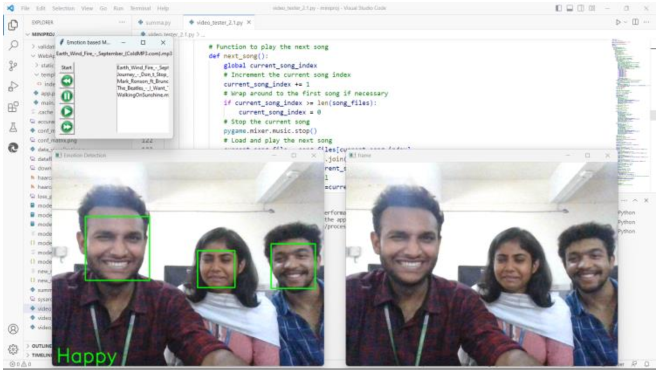


Figure 6.1 Group Emotion-- Happy

This image consists of 3 members where each of them exhibiting three different emotions. In which two of them are showing the same emotion while the other is exhibiting a different one. The algorithm finds the majority of the exhibited emotions and showcases it. --HAPPY

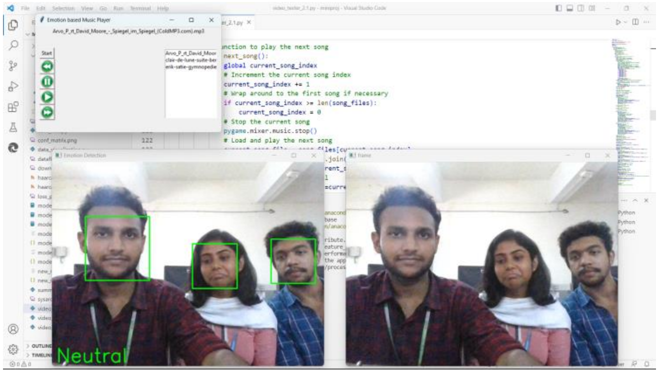


Figure 6.2 Group Emotion—Neutral

This image consists of 3 members where each of them exhibiting three different emotions. In which two of them are showing the same emotion while the other is exhibiting a different one. The algorithm finds the majority of the exhibited emotions and showcases it. -- NEUTRAL

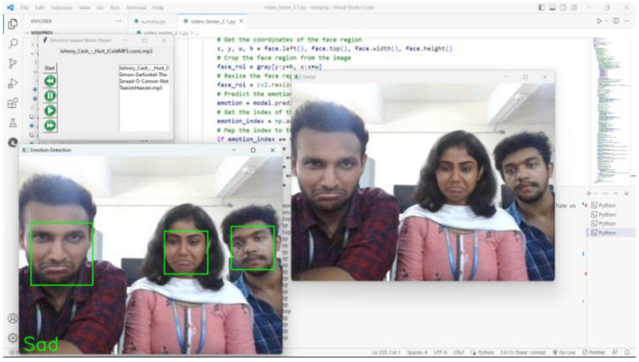


Figure 6.3 Group Emotion—Sad

This image consists of 3 members where each of them exhibiting three different emotions. In which two of them are showing the same emotion while the other is exhibiting a different one. The algorithm finds the majority of the exhibited emotions and showcases it. -- SAD

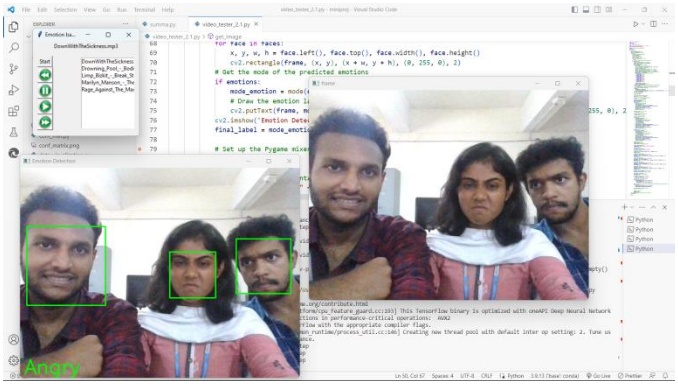


Figure 6.4 Group Emotion-- Angry

This image consists of 3 members where each of them exhibiting three different emotions. In which two of them are showing the same emotion while the other is exhibiting a different one. The algorithm finds the majority of the exhibited emotions and showcases it. -- ANGRY

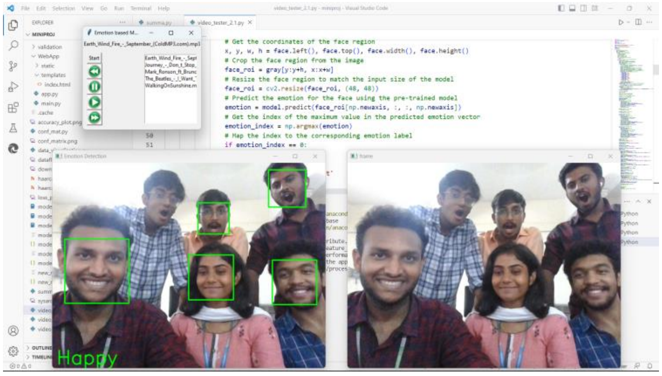


Figure 6.5 Group Emotion—Happy

This image consists of 5 members where each of them exhibiting three different emotions. In which three of them are showing the same emotion while the others are exhibiting a different one. The algorithm finds the majority of the exhibited emotions and showcases it. -- HAPPY

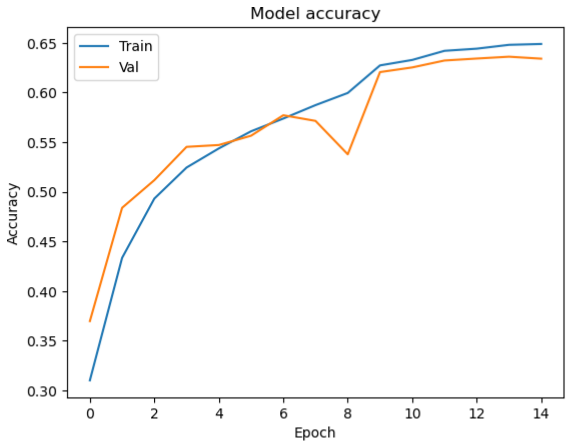


Fig 6.6 (a) Training and Validation Accuracy

The image illustrates the training and validation accuracy during the training process of a machine learning model. The x-axis represents the number of training epochs or iterations, while the y-axis represents the accuracy percentage. The model learns and adjusts its parameters, the training accuracy generally increases, indicating that the model is becoming more proficient at making predictions on the training data.

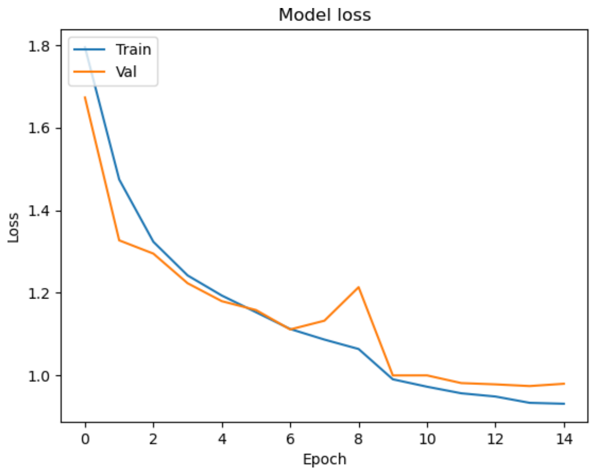


Fig 6.6 (b) Training and Validation Loss

The "training and validation loss" image provides a visual representation of the loss values during the training and validation phases of a machine learning model. The x-axis represents the number of training iterations or epochs, while the y-axis represents the loss value.

The image typically shows two lines or curves: one for the training loss and another for the validation loss. The training loss indicates how well the model is fitting the training data over time, while the validation loss reflects the model's performance on unseen validation data.

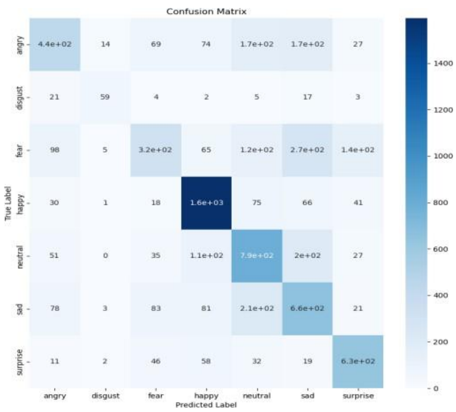


Fig 6.7 Confusion Matrix Model

The confusion matrix image is a square grid divided into four quadrants. The vertical axis represents the actual or true labels, while the horizontal axis represents the predicted labels. Each cell in the grid corresponds to the number of instances that fall into a particular combination of true and predicted labels.

CHAPTER 7

CONCLUSION AND FUTURE WORK

Our machine learning model is designed to identify various emotions based on facial expressions with an accuracy rate of around 65%. The model can recognize seven distinct emotions, including anger, disgust, fear, happiness, sadness, surprise, and neutrality. This accomplishment is noteworthy because human emotions are complex and subtle, making it difficult to discern them accurately. However, we recognize that the current model's accuracy can be improved, especially in detecting fear and disgust, which can be aided by incorporating additional parameters like heart rate or body temperature alongside facial expressions.

The primary objective of our project is to recommend music that aligns with the user's current emotional state. Based on the analysis of the facial expressions, the model suggests music that suits the user's mood accurately. However, finding appropriate music for certain emotions, like fear or disgust, poses a challenge, which presents opportunities for future development. Enhancing the model's ability to detect these emotions accurately will help identify suitable music for these emotions.

One of the main limitations of our current model is the likelihood of over fitting. Over-fitting occurs when the model's ability to recognize specific emotions is highly influenced by the dataset it was trained on, leading to occasional fluctuations in accurate detection. The "disgust" mood is often miss-classified as "anger" because the facial features of these two emotions, such as eyebrows and cheeks, are similar. To improve the model's precision, it is necessary to conduct further training with a larger dataset and an increased number of training epochs.

Expanding the scope of our project, we are considering the possibility of recommending movies and TV series based on mood detection. This prospect opens up exciting possibilities for future improvements. By analyzing the user's emotional state, the model can recommend movies and TV shows that align with their mood accurately. This feature could revolutionize the entertainment industry and help users find content that suits their emotional state.

CHAPTER 8

APPENDIX - I

8.1 SOURCE CODE

8.1.1 Dataset Visualization

```
import matplotlib.pyplot as plt
import os
from keras_preprocessing.image import load_img

#Displaying images
# size of the image: 48*48 pixels
pic_size = 48

# input path for the images
base_path = "C:\\Users\\arvin\\OneDrive\\Desktop\\Mini-Project\\images\\"

plt.figure(0, figsize=(12,20))
cpt = 0

for expression in os.listdir(base_path + "train/"):
    for i in range(1,6):
        cpt = cpt + 1
        plt.subplot(7,5,cpt)
        img = load_img(base_path + "train/" + expression + "/" +os.listdir(base_path + "train/"
+ expression)[i], target_size=(pic_size, pic_size))
        plt.imshow(img, cmap="gray")

plt.tight_layout()
plt.show()
```

8.1.2 Training the Model (ipynb)

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from keras.preprocessing.image import ImageDataGenerator
from keras.layers import Dense, Input, Dropout, Flatten, Conv2D
from keras.layers import BatchNormalization, Activation, MaxPooling2D
from keras.models import Model, Sequential
from keras.optimizers import Adam
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from keras.utils import plot_model

img_size = 48
batch_size = 64

base_path = "C:\\Users\\arvin\\OneDrive\\Desktop\\miniproj\\"

# Data generator to augment data for training
datagen_train = ImageDataGenerator(horizontal_flip=True)
train_generator = datagen_train.flow_from_directory(base_path+"train",
                                                    target_size=(img_size,img_size),
                                                    color_mode='grayscale',
                                                    batch_size=batch_size,
                                                    class_mode='categorical',
                                                    shuffle=True)

# Data generator to augment data for validation
datagen_validation = ImageDataGenerator(horizontal_flip=True)
```

```
validation_generator = datagen_train.flow_from_directory(base_path+"validation",
                                                          target_size=(img_size,img_size),
                                                         color_mode='grayscale',
                                                         batch_size=batch_size,
                                                         class_mode='categorical',
                                                         shuffle=False)
```

Found 28821 images belonging to 7 classes.
Found 7066 images belonging to 7 classes.

```
model = Sequential()

# Conv Block 1
model.add(Conv2D(64, (3,3), padding='same', input_shape=(48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# Conv Block 2
model.add(Conv2D(128,(5,5), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

# Conv Block 3
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(Dropout(0.25))

# Conv Block 4
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())

# Fully connected Block 1
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

# Fully connected Block 2
model.add(Dense(512))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(7, activation='softmax'))

opt = Adam(learning_rate=0.0005)
model.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 64)        640

 batch_normalization (BatchN  (None, 48, 48, 64)       256
 ormalization)

 activation (Activation)     (None, 48, 48, 64)        0

 max_pooling2d (MaxPooling2D  (None, 24, 24, 64)       0
 )

 dropout (Dropout)           (None, 24, 24, 64)        0

 conv2d_1 (Conv2D)           (None, 24, 24, 128)       204928

 batch_normalization_1 (Batc  (None, 24, 24, 128)      512
 hNormalization)

 activation_1 (Activation)   (None, 24, 24, 128)       0

 max_pooling2d_1 (MaxPooling  (None, 12, 12, 128)      0
 2D)
...
Total params: 4,478,727
Trainable params: 4,474,759
Non-trainable params: 3,968
_________________________________________________________________
```

```
epochs = 15
steps_per_epoch= train_generator.n//train_generator.batch_size
validation_steps = validation_generator.n//validation_generator.batch_size

checkpoint = ModelCheckpoint("model_weights.h5",monitor='val_accuracy',
                    save_weights_only=True, mode='max',verbose=1)
reduce_lr = ReduceLROnPlateau(monitor='val_loss' , factor=0.1, patience=2,
min_lr=0.00001,model='auto')

callbacks = [checkpoint, reduce_lr]
```

```
history = model.fit(
        x= train_generator,
        steps_per_epoch=steps_per_epoch,
        epochs=epochs,
        validation_data=validation_generator,
        validation_steps=validation_steps,
        callbacks=callbacks
)

model_json = model.to_json()
with open("model.json","w") as json_file:
    json_file.write(model_json)
```

8.1.3 Implementation of emotion-based music player

```
import cv2
import numpy as np
import dlib
from keras.models import model_from_json
from statistics import mode
import os
import tkinter as tk
import pygame
from PIL import Image, ImageTk

model = model_from_json(open('model.json',"r").read())
model.load_weights('model_weights.h5')

detector = dlib.get_frontal_face_detector()
emotions = []
final_label = ""
```

```
def get_image():
    # Initialize video capture object
    cap = cv2.VideoCapture(0)
    # Loop to continuously capture frames
    while True:
        # Read a frame from the video feed
        ret, frame = cap.read()
        # Display the frame
        cv2.imshow('frame', frame)
        # Wait for key press
        key = cv2.waitKey(1) & 0xFF
        # If 'q' key is pressed, exit the loop
        if key == ord('q'):
            break
        # If 's' key is pressed, take a picture and process it
        elif key == ord('s'):
            # Convert image to grayscale
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            # Detect faces in the image using dlib's detector
            faces = detector(gray)
            # Loop through each face in the image
            for face in faces:
                # Get the coordinates of the face region
                x, y, w, h = face.left(), face.top(), face.width(), face.height()
                # Crop the face region from the image
                face_roi = gray[y:y+h, x:x+w]
                # Resize the face region to match the input size of the model
                face_roi = cv2.resize(face_roi, (48, 48))
                # Predict the emotion for the face using the pre-trained model
                emotion = model.predict(face_roi[np.newaxis, :, :, np.newaxis])
                # Get the index of the maximum value in the predicted emotion vector
                emotion_index = np.argmax(emotion)
```

```
            # Map the index to the corresponding emotion label
            if emotion_index == 0:
                emotion_label = 'Angry'
            elif emotion_index == 1:
                emotion_label = 'Disgust'
            elif emotion_index == 2:
                emotion_label = 'Fear'
            elif emotion_index == 3:
                emotion_label = 'Happy'
            elif emotion_index == 4:
                emotion_label = 'Neutral'
            elif emotion_index == 5:
                emotion_label = 'Sad'
            else:
                emotion_label = 'Surprise'
            # Add the predicted emotion to the list
            emotions.append(emotion_label)
        # Display the captured image with bounding boxes around the detected faces
        for face in faces:
            x, y, w, h = face.left(), face.top(), face.width(), face.height()
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        # Get the mode of the predicted emotions
        if emotions:
            mode_emotion = mode(emotions)
            # Draw the emotion label on the image
            cv2.putText(frame, mode_emotion, (10, frame.shape[0]-10),
cv2.FONT_HERSHEY_SIMPLEX, 1.5, (0, 255, 0), 2, cv2.LINE_AA)
        cv2.imshow('Emotion Detection', frame)
        final_label = mode_emotion

        # Set up the Pygame mixer
        pygame.mixer.init()
```

```
# Define the folder containing the songs
songs_folder = "Songs/" + final_label

# Create a list of all the songs in the folder
song_files = os.listdir(songs_folder)

# Initialize the current song index
global current_song_index
current_song_index=0
# Function to play the current song
def play_song():
    # Define the current song index
    current_song_index = 0
    # Load the current song file
    current_song_file = song_files[current_song_index]
    current_song_path = os.path.join(songs_folder, current_song_file)
    pygame.mixer.music.load(current_song_path)
    # Update the song name label
    song_name_label.config(text=current_song_file)
    # Play the song
    pygame.mixer.music.play()

# Function to pause the current song
def pause_song():
    pygame.mixer.music.pause()

# Function to resume the current song
def resume_song():
    pygame.mixer.music.unpause()

# Function to play the next song
```

```
def next_song():
    global current_song_index
    # Increment the current song index
    current_song_index += 1
    # Wrap around to the first song if necessary
    if current_song_index >= len(song_files):
        current_song_index = 0
    # Stop the current song
    pygame.mixer.music.stop()
    # Load and play the next song
    current_song_file = song_files[current_song_index]
    current_song_path = os.path.join(songs_folder, current_song_file)
    pygame.mixer.music.load(current_song_path)
    # Update the song name label
    song_name_label.config(text=current_song_file)
    pygame.mixer.music.play()

# Function to play the previous song
def prev_song():
    global current_song_index
    # Decrement the current song index
    current_song_index -= 1
    # Wrap around to the last song if necessary
    if current_song_index < 0:
        current_song_index = len(song_files) - 1
    # Stop the current song
    pygame.mixer.music.stop()
    # Load and play the previous song
    current_song_file = song_files[current_song_index]
    current_song_path = os.path.join(songs_folder, current_song_file)
    pygame.mixer.music.load(current_song_path)
    # Update the song name label
```

```
    song_name_label.config(text=current_song_file)
    pygame.mixer.music.play()

# Create the main window
window = tk.Tk()
# window.geometry('400x350')
window.title("Emotion based Music Player")

# Create the song name label
song_name_label = tk.Label(window, text="No song playing")
song_name_label.pack()

# create leftframe
left_frame = tk.Frame(window)
left_frame.pack(side="left", padx=10, pady=10)

# Create the play button
play_button = tk.Button(left_frame,text="Start", command=play_song)
play_button.pack()

# Create the previous button
file4 = Image.open("gui\prev.jpg").resize((30,30))
buttonimg4 = ImageTk.PhotoImage(file4)
prev_button = tk.Button(left_frame, image = buttonimg4, text="Previous",
command=prev_song)
prev_button.pack()

# Create the pause button
file2 = Image.open("gui\pause.jpg").resize((30,30))
buttonimg2 = ImageTk.PhotoImage(file2)
pause_button = tk.Button(left_frame, image = buttonimg2, text="Pause",
command=pause_song)
```

```
    pause_button.pack()

    # Create the resume button
    file1 = Image.open("gui\play.jpg").resize((30,30))
    buttonimg1 = ImageTk.PhotoImage(file1)
    resume_button = tk.Button(left_frame, image = buttonimg1, text="Resume",
command=resume_song)
    resume_button.pack()

    # Create the next button
    file3 = Image.open("gui\\next.jpg").resize((30,30))
    buttonimg3 = ImageTk.PhotoImage(file3)
    next_button = tk.Button(left_frame, image = buttonimg3, text="Next",
command=next_song)
    next_button.pack()

    # Create a frame for the right side of the window
    right_frame = tk.Frame(window)
    right_frame.pack(side="right", padx=10, pady=10)

    # Create the listbox widget
    song_listbox = tk.Listbox(right_frame)
    song_listbox.pack()

    # Populate the listbox with the song names
    for song_file in song_files:
        song_listbox.insert(tk.END, song_file)

    # Highlight the currently playing song
    def highlight_current_song():
        current_song_index = pygame.mixer.music.get_busy()
        if current_song_index != 0:
```

```
            song_listbox.selection_clear(0, tk.END)
            song_listbox.selection_set(current_song_index - 1)

        # Add a callback function to update the song name label and highlight the currently
playing song
        def update_ui():
            highlight_current_song()
            current_song_file = song_files[current_song_index]
            song_name_label.config(text=current_song_file)

        # Call the update_ui function periodically
        window.after(100, update_ui)



        # Start the main loop
        window.mainloop()
        emotions.clear()
    # Release the video capture object and close all windows
    cap.release()

get_image()
```

CHAPTER 9

REFERENCES

[1] Dr. Shaik Asif Hussain, Ahlam Salim Abdallah Al Balushi “A real time face emotion classification and recognition using deep learning model ”Department of Electronics and Communication Engineering, Middle East College, Muscat. Journal of Physics: Conference Series, ICE4CT 2019.

[2] Hivi Ismat Dino, Maiwan Bahjat Abdulrazzaq “Facial Expression Classification Based on SVM, KNN and MLP Classifiers” 2019 International Conference on Advanced Science and Engineering (ICOASE), University of Zakho, Duhok Polytechnic University, Kurdistan Region, Iraq

[3] ShanthaShalini. K, Jaichandran. R , Leelavathy. S A, Raviraghul. R , Ranjitha. J and Saravanakumar. N “Facial Emotion Based Music Recommendation System using computer vision and machine learning techiniques” Turkish Journal of Computer and Mathematics Education Vol.12 No.1 (2021), 912-917

[4] Aya Hassouneh, A.M. Mutawa, M. Murugappan “Development of a Real-Time Emotion Recognition System Using Facial Expressions and EEG based on machine learning and deep neural network methods” Informatics in Medicine Unlocked 20 (2020) 100372

[5] Hui Ding , Shaohua Kevin Zhou and Rama Chellappa “FaceNet2ExpNet: Regularizing a Deep Face Recognition Net for Expression Recognition” arXiv:1609.06591v2 [cs.CV] 22 Sep 2016

[6] V. Sathya, T.Chakravarthy “AUTOMATIC FACIAL EXPRESSION RELATED EMOTION RECOGNITION USING MACHINE LEARNING TECHNIQUES” International Journal of Computer Engineering & Technology (IJCET) Volume 8, Issue 5, Sep-Oct 2017, pp. 126–135, Article ID: IJCET_08_05_014 Patra, Braja & Das, Dipankar & Bandyopadhyay, Sivaji. (2013). Automatic Music Mood Classification of Hindi Songs.