DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Report

"MUSIC RECOMMENDATION USING DEEP LEARNING FOR THERAPY"

By

Abhay Shreekant Shastry - ENG18CS0008
B. Antonio Mervyn - ENG18CS0056
Binish Zehra Rizvi - ENG18CS0060
Varun Menon - ENG18CS0314
Vasu Bhalodia - ENG18CS0316

Under the supervision of

Dr. Shaila S. G.
Professor, CSE Department

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

(2021-2022)

DAYANANDA SAGAR UNIVERSITY

School of Engineering

Department of Computer Science & Engineering

Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Major project work titled **"MUSIC RECOMMENDATION USING DEEP LEARNING FOR THERAPY"** is carried out by **Abhay Shreekant Shastry (ENG18CS0008), B. Antonio Mervyn (ENG18CS0056), Binish Zehra Rizvi (ENG18CS0060), Varun Menon (ENG18CS0314), Vasu Bhalodia (ENG18CS0316),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2021-2022**.

Dr. Shaila S G	**Dr Girisha G S**	**Dr. A Srinivas**
Professor Dept. of CS&E, School of Engineering Dayananda Sagar University Date:	Chairman CSE School of Engineering Dayananda Sagar University Date:	Dean School of Engineering Dayananda Sagar University Date:

Name of the Examiner **Signature of Examiner**

1.

2.

DECLARATION

We, **Abhay Shreekant Shastry (ENG18CS0008), B. Antonio Mervyn (ENG18CS0056), Binish Zehra Rizvi (ENG18CS0060), Varun Menon (ENG18CS0314), Vasu Bhalodia (ENG18CS0316),** are students of the eight semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major project titled **"Music Recommendation using Deep Learning for Therapy"** has been carried out by us and submitted in complete fulfillment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2021-2022**.

Student	**Signature**
Name 1 : Abhay Shreekant Shastry **USN : ENG18CS0008**	
Name 2 : B. Antonio Mervyn **USN : ENG18CS0056**	
Name 3 : Binish Zehra Rizvi **USN : ENG18CS0060**	
Name 4 : Varun Menon **USN : ENG18CS0314**	
Name 5 : Vasu Bhalodia **USN : ENG18CS0316**	
Place : Bangalore **Date :**	

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. A Srinivas. Dean**, **School of Engineering & Technology**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman**, **Computer Science, and Engineering**, **Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **Dr. Shaila S.G. Professor**, **Dept. of Computer Science and Engineering**, **Dayananda Sagar University**, for sparing his/her valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project.

We would like to thank our Project Coordinator **Dr. Meenakshi Malhotra** and **Dr. Bharanidharan N,** all the staff members of Computer Science and Engineering for their support.

We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work.

Signature of Students

Abhay Shreekant Shastry

B. Antonio Mervyn

Binish Zehra Rizvi

Varun Menon

Vasu Bhalodia

TABLE OF CONTENTS

LIST OF ABBRIVATIONS

1. **CNN:** A convolutional neural network (CNN) is a specific type of artificial neural network That uses perceptrons, a machine learning unit algorithm, for supervised learning, to analyze data. CNNs apply to image processing, natural language processing and other kinds of cognitive tasks. A convolutional neural network is also known as a ConvNet.
2. **LBP:** Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision.

LIST OF FIGURES

ABSTRACT

There is a colossus amount of music that is being generated every day. This music has no bounds in terms of instruments used, vocal arrangement, tunes, symphonies, and so on which results in a variety of genres. This often causes a major inconvenience to a user or a person in finding a song of his/ her preference among the vast existing music.

Our Project proposes a solution to the above dilemma by analyzing the music previously listened by the user and identifies a set of preferred genres. Upon identification of the user's preferences in music, our project then proceeds to recommend music that is similar to the music existing in the users list of preferred genres. This saves the user a large amount of time which is wasted in looking for a specific type or genre of music and could be spent actually listening to music.

Music has been scientifically proven to have a profound effect on the functioning of the human brain. According to recent studies, music has been shown to aid in many aspects of the brain including pain alleviation, stress relaxation, memory, brain damage and so on. Thus, our project aims to aid in the overall mental well-being of a person by using music to cope with anxiety, stress and so on.

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

Many of the studies in recent years admit that humans reply and react to music and this music has a high impression on the activity of the human brain. In one examination of the explanations why people hear music, researchers discovered that music played a crucial role in relating arousal and mood. Two of the most important functions of music are its ability as participants rated to help them achieve a good mood and become more self-aware. Musical preferences have been demonstrated to be highly related to personality traits and moods.

The meter, timbre, rhythm, and pitch of music are managed in areas of the brain that affect emotions and mood. Interaction between individuals may be a major aspect of lifestyle. It reveals perfect details and much data among humans, whether they are in the form of body language, speech, facial expression, or emotions.

Nowadays, emotion detection is considered the most important technique used in many applications such as smart card applications, surveillance, image database investigation, criminal, video indexing, civilian applications, security, and adaptive human-computer interface with multimedia environments. With the increase in technology for digital signal processing and other effective feature extraction algorithms, automated emotion detection in multimedia attributes like music or movies is growing rapidly and this system can play an important role in many potential applications like human-computer interaction systems and music entertainment.

We use facial expressions to propose a recommender system for emotion recognition that can detect user emotions and suggest a list of appropriate songs. The proposed system detects the emotions of a person, if the person has a negative emotion, then a certain playlist will be shown that includes the most related types of music that will enhance his mood. And if the emotion is positive, a specific playlist will be presented which contains different types of music that will inflate the positive emotions.

1.1. PURPOSE

Research suggests that three aspects of music—its emotional resonance, its lyrical content, and its unique way of synchronizing groups of people—may have the power to uplift people's mood and hence music can be used as therapy. Music therapy is used to treat many psychological, intellectual, and emotional issues. Music therapy helps a broad range of patients, from children to the elderly. Music therapy works with a range of health professionals to develop client management plans and achieve positive patient outcomes.

The purpose of this project is to develop a model based on detecting emotions of the person and in turn recommend music based on the person's mood. The project uses a custom CNN to perform emotion detection and the output of this model is fed to a music recommendation system which recommends music based on different moods like sad, neutral, angry.

1.2. SCOPE

The model can be used to uplift the mood of the user and hence has a scope in providing therapy to the user. It can be used to reduce stress, improve mood and self-expression. The model can be used as an alternative therapy that is a part of a much larger treatment plan. This system uses Deep Learning methods to process information and provide users with potentially more relevant items.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

Music is an integral part of everyone's life. It can lift us up when we're down or motivate us when we feel low. Therefore, it is therapeutic in nature hence making it an integral part of everyone's life. The therapeutic powers of music are mapped to whatever genre the listener prefers. But in a world where over 8,00,000 are released every year, it is often daunting navigating through all the genres to find songs that suit the user's needs. To combat this, we are proposing a Music Recommendation using Emotion Recognition that recommends music that the user prefers.

In comparison to other domains in which recommender systems are employed, such as products, movies, or hotels, recommendation in the music domain has certain specific characteristics that should be taken into account when creating Music Recommendation System. Some of these particularities in the music domain have implications on the use of recommender systems technology; and the use of Machine Learning approaches are directly motivated by them.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

[1] In this research, a latent component model for recommendation and estimate latent factors from music audio when consumption data is unavailable is used. On the Million Song Dataset, a standard approach based on a bag-of-words representation of audio data with deep convolutional neural networks is compared and the predictions are evaluated quantitatively and qualitatively. It is found that employing predicted latent components generates sensible recommendations.

[2] In this paper, a DTNMR which is a Deep Temporal Neural Music Recommendation model based on music characteristics and individuals' temporal preferences is used. Individuals' long-term and short-term preferences are learned from their listening histories using Long Short-Term Memory (LSTM) neural networks. DTNMR uses music metadata to solve the cold start problem on the item side and finds new users' preferences instantly after they listen to music. In terms of recall, precision, f-measure, MAP, user coverage, and AUC, the experimental findings reveal that DTNMR beats seven baseline approaches.

[3] Features obtained by normal short-term features are not as concise and detailed as the information you get from long term features. In this paper, LSTM is used with Zero Crossing Rate (ZCR) and Mel-frequency spectral coefficients (MFCC) which are time domain and frequency domain feature by passing the latter through LSTM network. Finally Support Vector Machine (SVM) and KNN are used to classify music into genres. LSTM is found to significantly increase accuracy over traditional methods

[4] This paper uses custom filtering called Tunes Recommendation System (T-RECSYS) a hybrid of content-based and collaborative filtering, as an input into a deep learning model to make a real time prediction and recommendation system. The result precision score was 88% at a balanced discrimination threshold.

[5] According to this paper, a solution should start with a questionnaire to generate a user profile. "The information from the questionnaire is compared with musical attributes to trim the initial 384,500 songs to 1000 songs that the user is more likely to love," they write. The questionnaire uses clustering, a type of machine learning that divides data into

clusters of comparable data points, to allocate individuals to groups with similar musical likes. The programme combines user profiles to create group profiles in order to obtain recommendations that are likely to be acceptable and appropriate for the entire group via collaborative filtering. These suggestions are centered on activities like group physical therapy, Zumba, yoga, and pilates. The User Profiling (UP) recommendation system's initial results are then refined using Reinforcement Learning (RL). On a five-point scale, users can rate the tempo, loudness, lively, positive, familiarity with the song, and prominence of the beat. This enables users to stick to strict preferences, which may be required during choice exercises. Despite the fact that this study incorporates many factors, collaborative filtering, and content-based filtering, it only relates to exercise music, which is not appropriate for everyday recommendations.

[6] This paper used a recurrent neural networks model (RNN) to compare the similarity of different songs, which aids recommendation systems in assigning ranking scores based on a variety of musical variables. RNN models have been described in other systems, but unlike earlier efforts, this paper examined the similarity between songs. Despite the fact that this programme considered musical aspects, there was no mention of real-time changes.

[7] This paper presented an indirect matching system for fast content-based music information retrieval. Representative inquiries from offline searches were employed in this framework. They used these offline searches as a foundation for quickly estimating similarity between internet sources. Their study incorporated content-based filtering components, similar to and. The algorithm, on the other hand, did not include collaborative filtering or real-time updates.

[8] This paper proposed that real-time data streams can be processed to make correct suggestions on the fly. In their work, they used Spark and machine learning frameworks to perform real-time changes in recommendation systems, focusing on TV channel suggestions in particular.

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

4.1. PROPOSED DESIGN

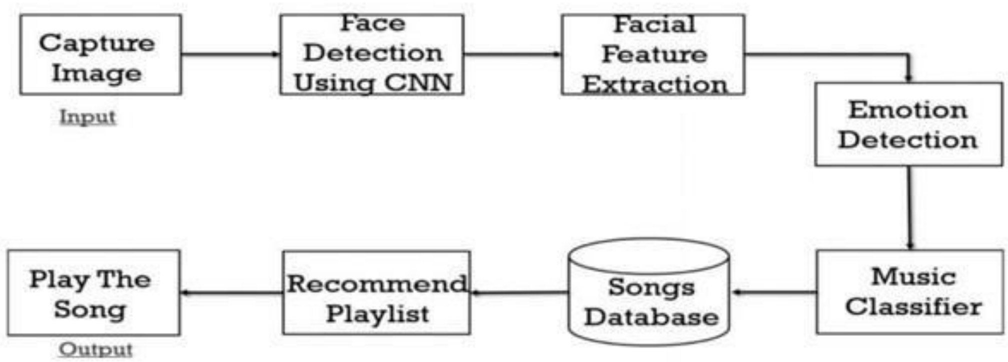

Fig 4.1. Data Flow Diagram

The proposed system benefits us to present interaction between the user and the music player. The purpose of the system is to capture the face properly with the camera. Captured images are fed into the Convolutional Neural Network which predicts the emotion. Then emotion derived from the captured image is used to get a playlist of songs. The main aim of our proposed system is to provide a music playlist automatically to change the user's moods, which can be happy, sad, natural, or surprised. The proposed system detects the emotions, if the topic features a negative emotion, then a selected playlist is going to be presented that contains the foremost suitable sorts of music that will enhance the mood of the person positively. Music recommendation based on facial emotion recognition contains four modules.

- **Real-Time Capture**: In this module, the system is to capture the face of the user correctly.
- **Face Recognition**: Here it will take the user's face as input. The convolutional neural network is programmed to evaluate the features of the user image.

- **Emotion Detection**: In this section extraction of the features of the user image is done to detect the emotion and depending on the user's emotions, the system will generate captions.
- **Music Recommendation**: Song is suggested by the recommendation module to the user by mapping their emotions to the mood type of the song.
- **Training:** Model is trained using all the training datasets which have been segregated after Data Preprocessing.
- **Testing:** Model is tested using all the testing datasets to ensure that the model works accurately and recommends perfectly.

4.2. ASSUMPTIONS AND DEPENDENCIES

- **Assumptions:**
  - User has a music sample that has a definitive genre.
  - Users must have a Google account to run the application on Google Collaborator.
  - The user knows basic Python.
- **Dependencies:**
  - Processing power from Google Collaborator should be GPU, in order to provide fast training and testing. Which helps in a quicker generation of models.
  - Stable internet connection which does not hinder with Google Collaborator while the model is training or testing.

CHAPTER 5

REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1. FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. Here, the system has to perform the following tasks:

- Has to accept input live image input and detects the emotion of the user.
- Model should run by taking features into account and recommend music accordingly.

5.2. NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that define specific behaviour or functions. The plan for implementing functional requirements is detailed in the system design. Other terms for non-functional requirements are “constraints", "quality goals", "quality of service requirements" and "non-behavioural requirements”. Some of the quality attributes are as follows:

Accessibility:

Accessibility is a general term used to describe the degree to which a product, device, service, or environment is accessible by as many people as possible. In our project any number of people can register and can access the service. User interface is simple and efficient and easy to use.

Maintainability:

In software engineering, maintainability is the ease with which a software product can be modified in order to:

- Correct defects
- Meet new requirements

New functionalities can be added in the project based on the user requirements just by adding the appropriate files to existing projects using python scripting languages.

Scalability:

System is capable of handling increased total throughput under an increased load when resources (typically songs) are added. System can work normally under situations such as low bandwidth and a large number of users.

Portability:

Portability is one of the key concepts of high-level programming. Portability is the software code base feature to be able to reuse the existing code instead of creating new code when moving software from an environment to another. Project can be executed under different operation conditions provided it meets its minimum configurations. Only system files and dependent assemblies would have to be configured in such a case.

5.3. SOFTWARE REQUIREMENTS

- Google Colab
- Jupyter Notebook
- Python 3.0 or above versions
- Windows 8 or higher OS

5.4. HARDWARE REQUIREMENTS

- Processor: Any processor above 500 MHz
- RAM: 512Mb
- Hard Disk: 10GB
- Input device: Standard Keyboard and Mouse
- Output device: High Resolution Monitor

CHAPTER 6
METHODOLOGY

CHAPTER 6 METHODOLOGY

6.1. DATABASE DESCRIPTION

We built the Convolutional Neural Network model using the Kaggle dataset. The database is FER2013 which is split into two parts: training and testing dataset. The training dataset consists of 24176 and the testing dataset contains 6043 images. There are 48x48 pixel grayscale images of faces in the dataset. Each image in FER-2013 is labelled as one of five emotions: happy, sad, angry, surprise, and neutral. The faces are automatically registered so that they are more or less centered in each image and take up about the same amount of space. The images in FER-2013 contain both posed and unposed headshots, which are in grayscale and 48x48 pixels. The FER-2013 dataset was created by gathering the results of a Google image search of every emotion and synonyms of the emotions. FER systems being trained on an imbalanced dataset may perform well on dominant emotions such as happy, sad, angry, neutral, and surprised but they perform poorly on the under-represented ones like disgust and fear. Usually, the weighted-SoftMax loss approach is used to handle this problem by weighting the loss term for each emotion class supported by its relative proportion within the training set. However, this weighted-loss approach is predicated on the SoftMax loss function, which is reported to easily force features of various classes to stay apart without listening to intra-class compactness. One effective strategy to deal with the matter of SoftMax loss is to use an auxiliary loss to coach the neural network. To treating missing and Outlier values we have used a loss function named categorical cross-entropy. For each iteration, a selected loss function is employed to gauge the error value. So, to treating missing and Outlier values, we have used a loss function named categorical cross-entropy.

Fig 6.1 Samples from FER2013 dataset

6.2. FACE DETECTION

Face detection is one of the applications which is considered under computer vision technology. This is the process in which algorithms are developed and trained to properly locate faces or objects in object detection or related systems in Images. This detection can be real-time from a video frame or images. Face detection uses such classifiers, which are algorithms that detect what's either a face (1) or not a face (0) in an image. Classifiers are trained to detect faces using numbers of images to get more accuracy. OpenCV uses two sorts of classifiers, LBP (Local Binary Pattern) and Haar Cascades. A Haar classifier is used for face detection where the classifier is trained with pre-defined varying face data which enables it to detect different faces accurately. The main aim of face detection is to spot the face within the frame by reducing external noises and other factors. It is a machine learning-based approach where the cascade function is trained with a group of input files. It is supported by the Haar Wavelet technique to research pixels inside the image into squares by function. This uses machine learning techniques to urge a high degree of accuracy from what's called "training data".

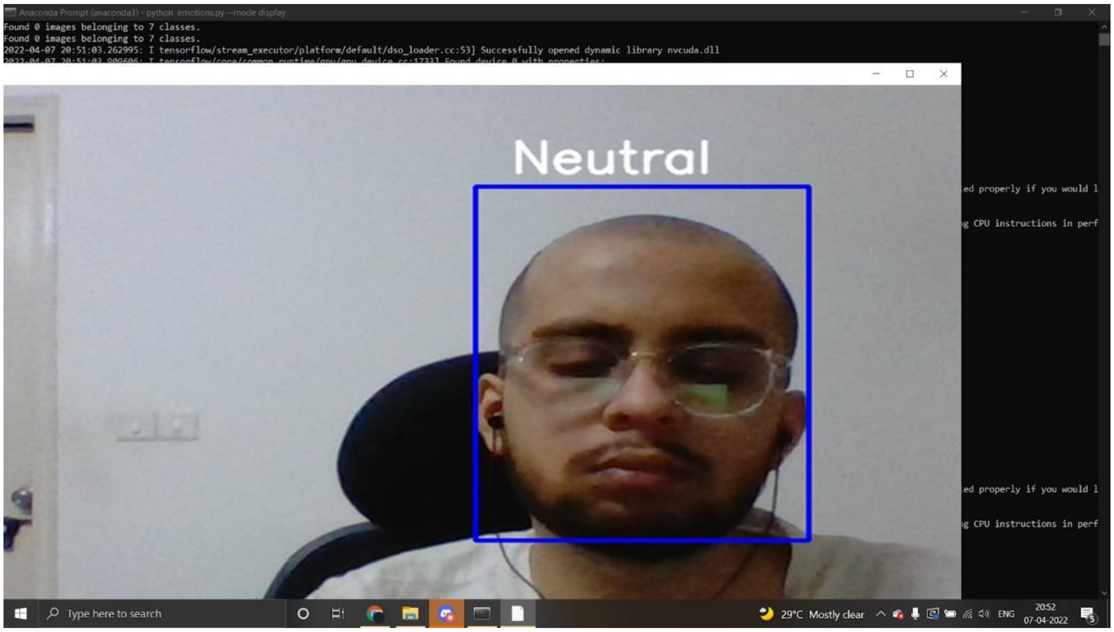

Fig 6.2. Sample Output of novel emotion recognition

CHAPTER 7
EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

```
emotion_dict= {0:"Angry", 1:"Disgusted", 2:"Fearful", 3:"Happy", 4:"Neutral", 5:"Sad",
6:"Surprised"}
```

In the above code snippet, a value is assigned to all the different emotions included and the emotion is classified into the appropriate class based on the value.

```
result = sp.search(name)

artist_uri = result['tracks']['items'][0]['artists'][0]['uri']

sp_albums = sp.artist_albums(artist_uri, album_type='album')

album_names = []
album_uris = []
for i in range(len(sp_albums['items'])):
    album_names.append(sp_albums['items'][i]['name'])
    album_uris.append(sp_albums['items'][i]['uri'])
```

In the above code snippet, we retrieve the artist preference from the user and then use SpotiPy to retrieve said artist's albums. We then use these playlists to retrieve their individual songs. Finally, the individual songs are collected in order for custom playlists to be generated.

```
def audio_features(album):
    # Add new key-values to store audio feature
    spotify_albums[album]['acousticness'] = []
    spotify_albums[album]['danceability'] = []
    spotify_albums[album]['energy'] = []
    spotify_albums[album]['instrumentalness'] = []
    spotify_albums[album]['liveness'] = []
    spotify_albums[album]['loudness'] = []
    spotify_albums[album]['speechiness'] = []
    spotify_albums[album]['tempo'] = []
    spotify_albums[album]['valence'] = []
    spotify_albums[album]['popularity'] = []
```

```
col_features = ['danceability', 'energy', 'valence', 'loudness']
X = MinMaxScaler().fit_transform(data1[col_features])
```

In the above code snippet, we are defining the factors or parameters that will be used inorder to create our custom playlists. These parameters are intentionally chosen to be continuous values that can be compared. This helps in differentiating songs.

```
client_credentials_manager =
SpotifyClientCredentials(client_id='39e3c9d1bf944d23b2594f90a0a8929f',
client_secret='14136c5f22964499984c59a25d87bb3b')
sp = spotipy.Spotify(client_credentials_manager=client_credentials_manager)
```

In the above code snippet, we are establishing a connection to the Spotify API (SpotiPy) through our unique credentials. This will help facilitate communication between our application and Spotify to retrieve songs and playlists.

```
def build_net(optim):
        net = Sequential(name='DCNN')

        net.add(Conv2D(filters=256,kernel_size=(5,5),input_shape=(img_width, img_height, img_depth),activation='elu',
                padding='same',kernel_initializer='he_normal', name='conv2d_1'))
        net.add(BatchNormalization(name='batchnorm_1'))
        net.add(Conv2D(filters=128,kernel_size=(5,5),activation='elu',padding='same',kernel_initializer='he_normal',
                name='conv2d_2'))
        net.add(BatchNormalization(name='batchnorm_2'))
        net.add(MaxPooling2D(pool_size=(2,2), name='maxpool2d_1'))
        net.add(Dropout(0.4, name='dropout_1'))
        net.add(Conv2D(filters=128,kernel_size=(3,3),activation='elu',padding='same',kernel_initializer='he_normal',
                name='conv2d_3'))
        net.add(BatchNormalization(name='batchnorm_3'))
        net.add(Conv2D(filters=128,kernel_size=(3,3),activation='elu',padding='same',kernel_initializer='he_normal',
                name='conv2d_4'))
        net.add(BatchNormalization(name='batchnorm_4'))
        net.add(MaxPooling2D(pool_size=(2,2), name='maxpool2d_2'))
        net.add(Dropout(0.4, name='dropout_2'))
        net.add(Conv2D(filters=256,kernel_size=(3,3),activation='elu',padding='same',kernel_initializer='he_normal',
                name='conv2d_5'))
        net.add(BatchNormalization(name='batchnorm_5'))
        net.add(Conv2D(filters=256,kernel_size=(3,3),activation='elu',padding='same',kernel_initializer='he_normal',
                name='conv2d_6'))
        net.add(BatchNormalization(name='batchnorm_6'))
        net.add(MaxPooling2D(pool_size=(2,2), name='maxpool2d_3'))
        net.add(Dropout(0.5, name='dropout_3'))
        net.add(Flatten(name='flatten'))
        net.add(Dense(128,activation='elu',kernel_initializer='he_normal',name='dense_1'))
        net.add(BatchNormalization(name='batchnorm_7'))
        net.add(Dropout(0.6, name='dropout_4'))
        net.add(Dense(num_classes,activation='softmax',name='out_layer'))

        net.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])

        net.summary()

        return net
```

The above code snippet shows the different layers of the custom CNN we have defined for recognizing emotions of the user.

Max-pooling is used after the convolution layer which selects the maximum element from the region of the feature map covered by the filter. Dropout is a regularization technique which is then applied to the feature map in order to reduce overfitting. The activation function that is used is softmax which scales numbers/logits into probabilities. Adam optimizer is used.

CHAPTER 8
TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

8.1. RESULTS

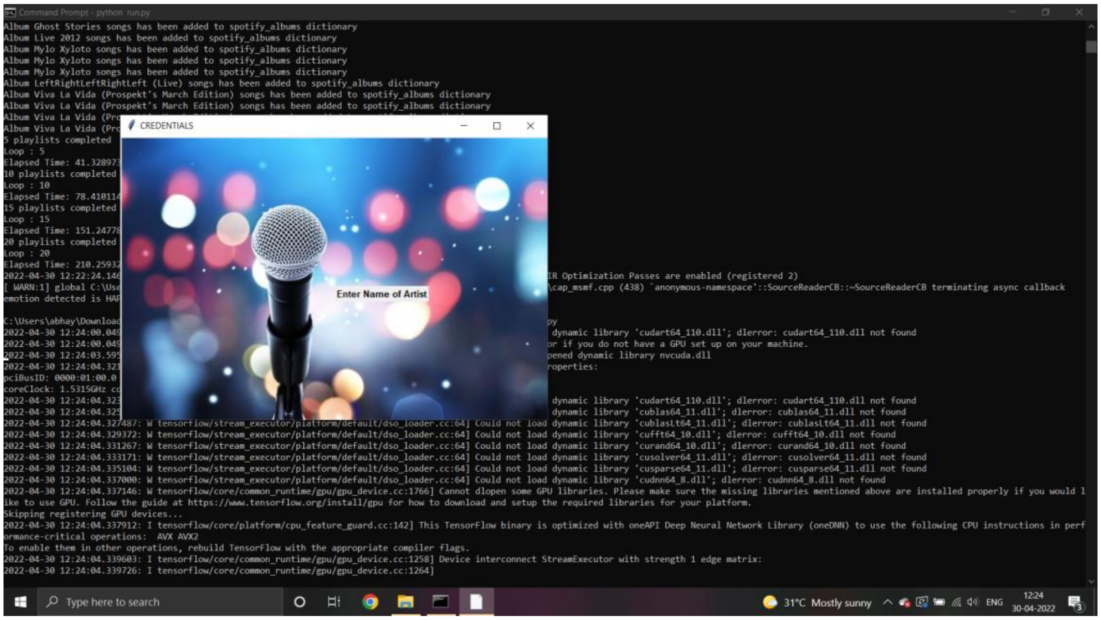

Fig 8.1.1. First Output Screen

The user is first greeted with an interface(Fig 8.1.1) where they need to enter the name of the artist of their liking. The playlists of different permutations and combinations start getting created(Fig 8.1.2). Each loop increments the number of playlists being generated by 5.

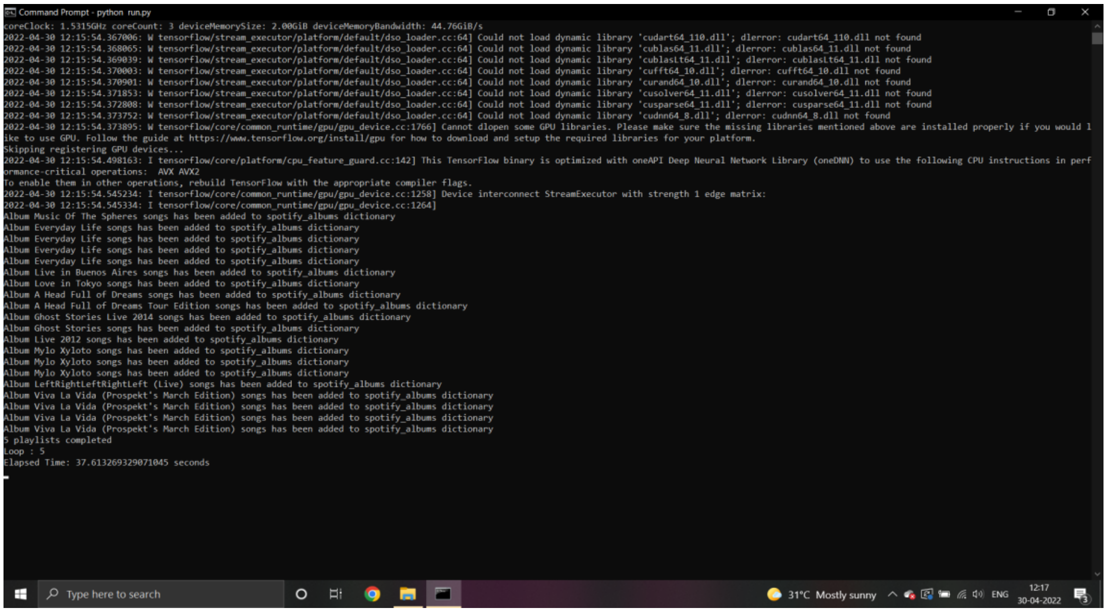

Fig 8.1.2. Second Output Screen

After the playlists are generated, the webcam turns on to capture the video feed of the user. The deep learning model then detects emotions and displays them in real time (Fig 8.1.3).

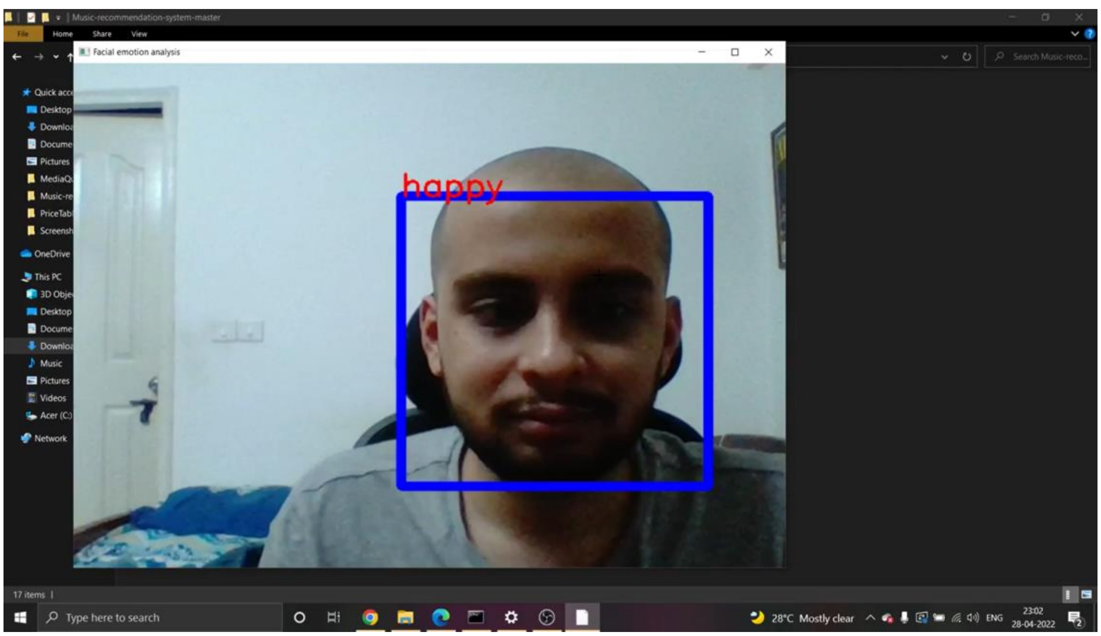

Fig 8.1.3. Emotion recognition Screen

We are then greeted by the second user interface(Fig 8.1.4) where we can either view the generated result by clicking “print” or quit the application by clicking “Quit”.

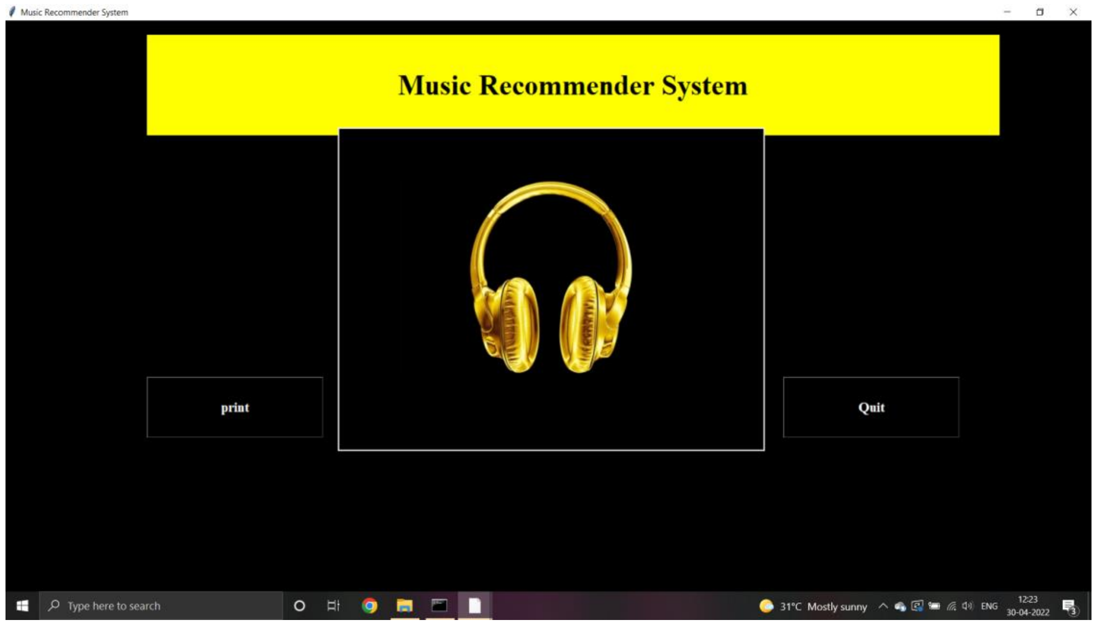

Fig 8.1.4. Third Output Screen

Finally, when we click print, we are greeted with a list of song recommendations (Figure 8.1.5)

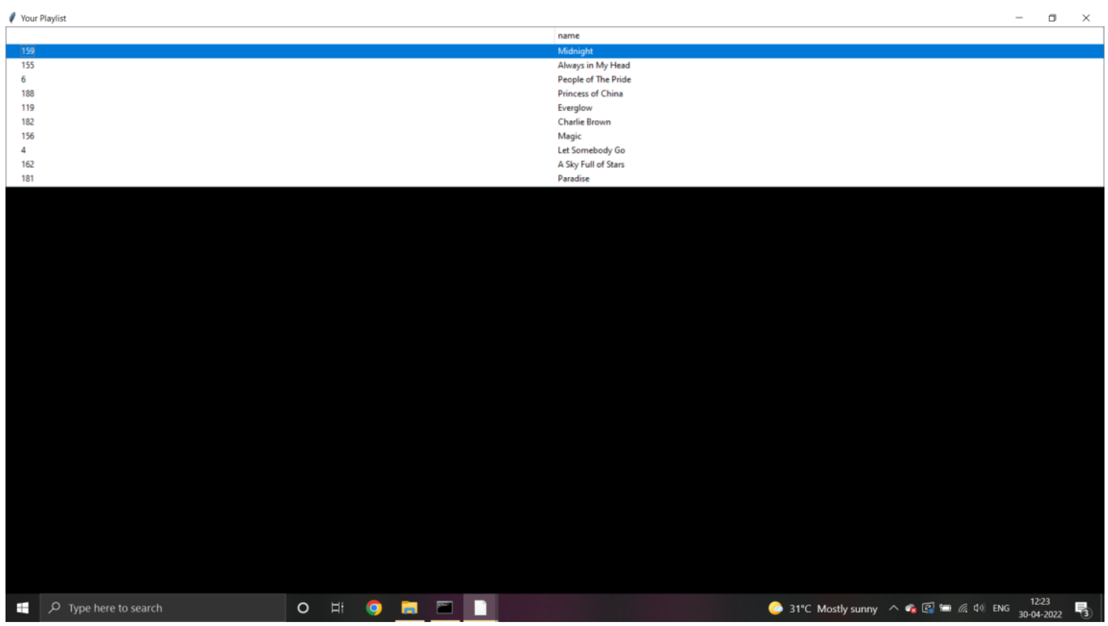

Fig 8.1.5. Song Recommendations

8.2. TEST CASES

Our first case involves the user invoking a happy emotion (8.2.1) which in turn gives us a recommendation of songs to suit the user's needs (8.2.2) for the artist of their liking(coldplay). The songs, if you observe, are cheerful and happy in nature which complements the emotion of the user. The list of songs are limited to 10 as we require a rich quality of a few songs as opposed to a huge list of songs which would be counter intuitive in our case.

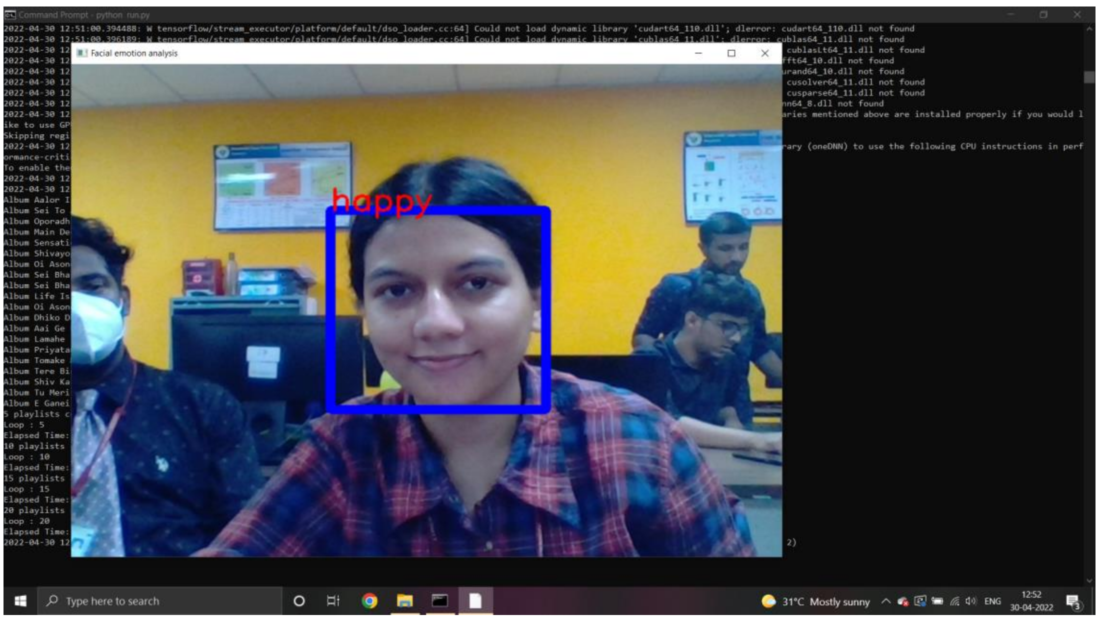

Fig 8.2.1. Sample Output of emotion recognition - Happy

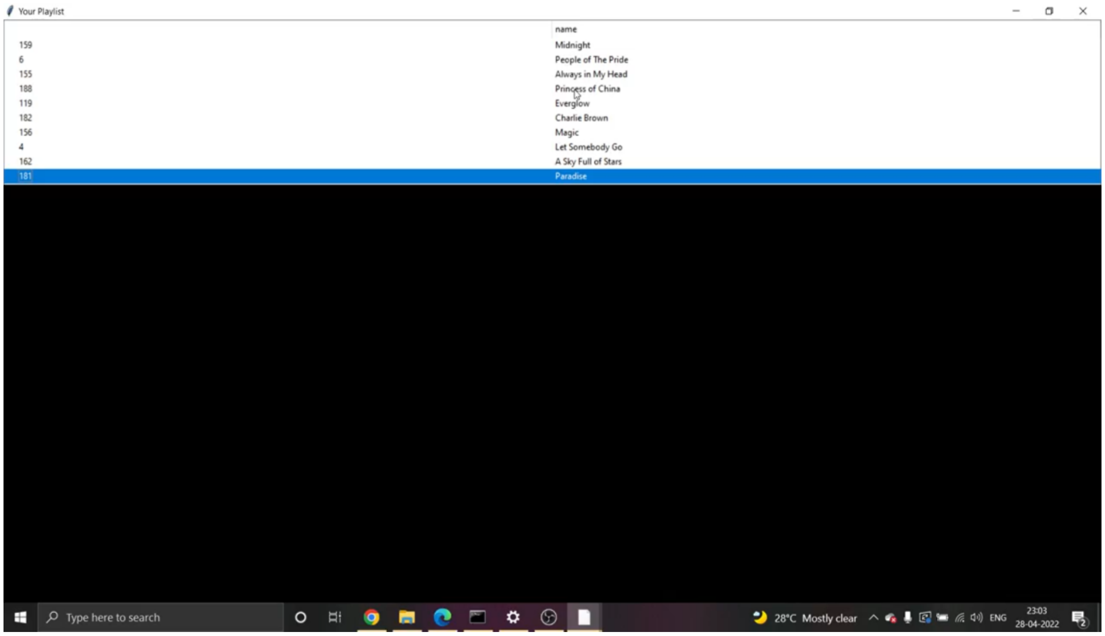

Fig 8.2.2. Song recommendations - Happy

We then have to move on to our next case where the user expresses the opposite emotion to the first one i.e sadness. This emotion with the user's artist preference (Nirvana) will in turn produce a playlist that will complement the user's emotion.

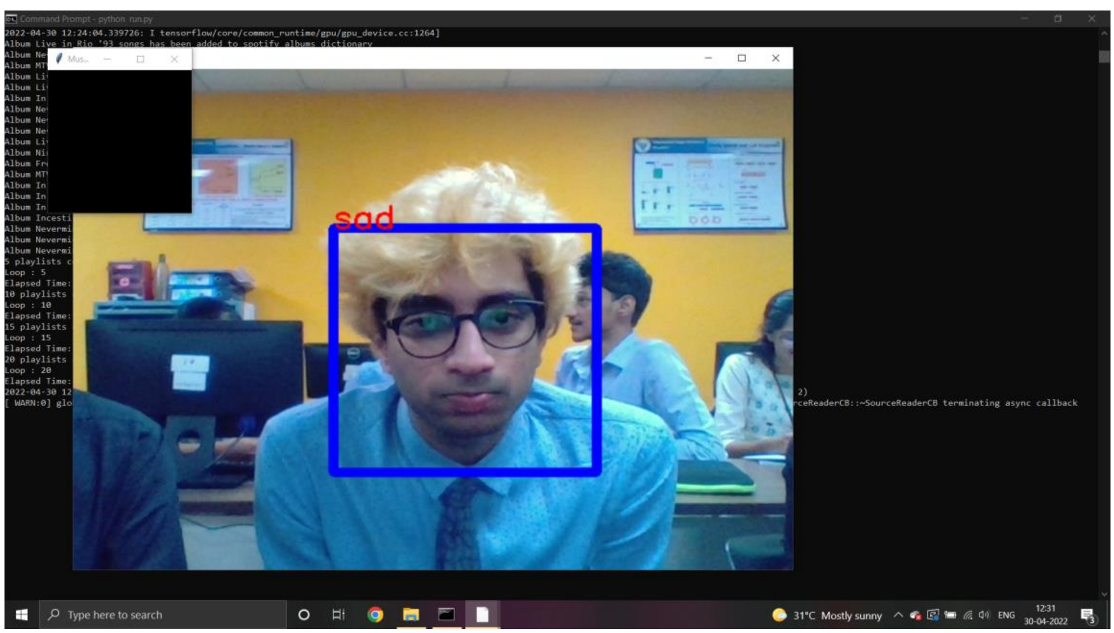

Fig 8.2.3. Sample Output of emotion recognition - Sad

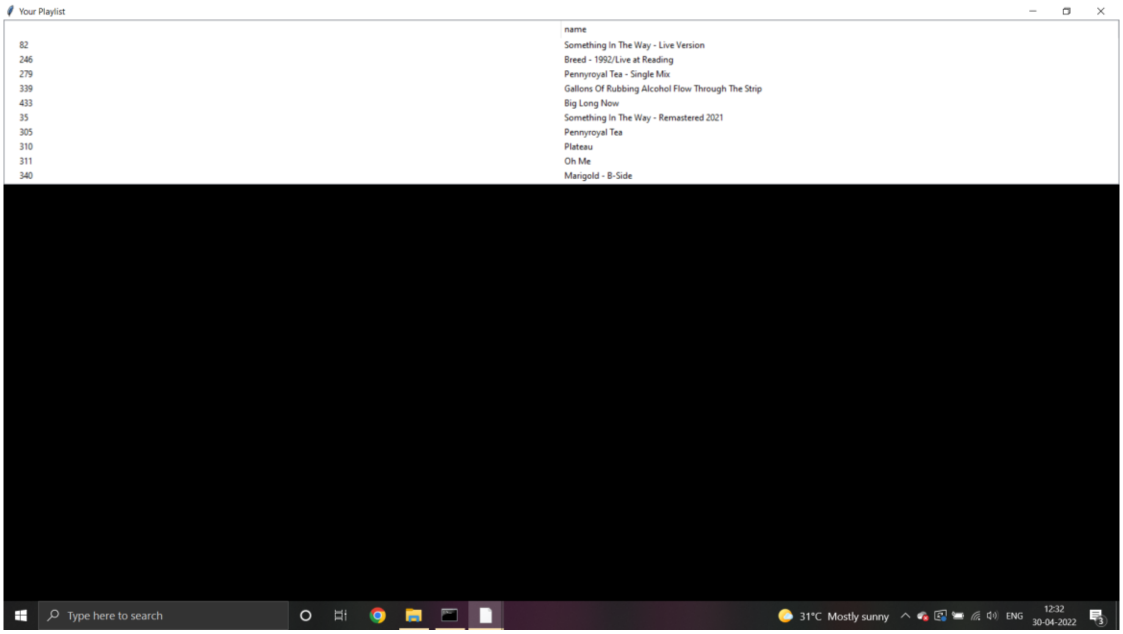

Fig 8.2.4. Song recommendations - Sad

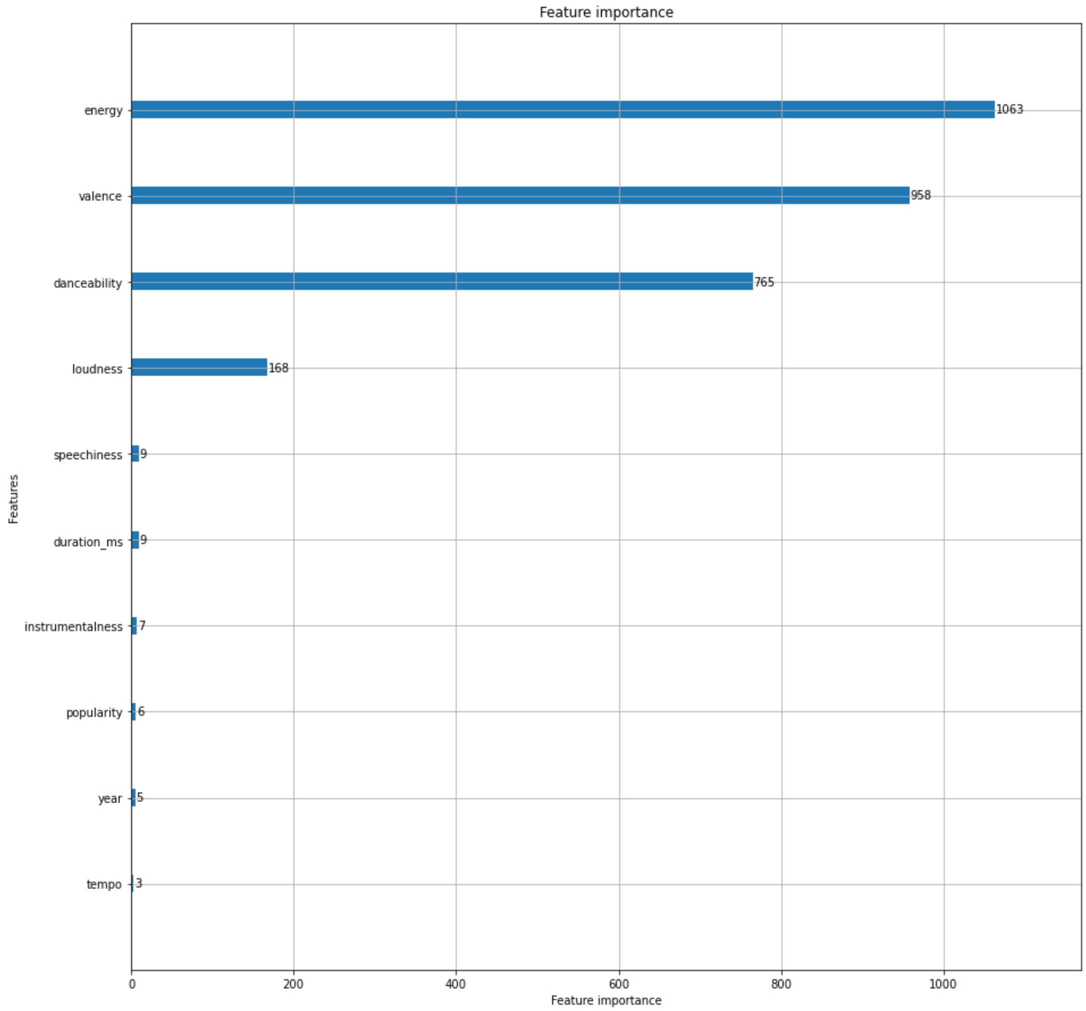

Fig 8.2.5. Feature Importance

The metric we give the most importance (Fig 8.2.5) to is Energy of the song in our case as the energy of the song directly tells us the connotation the song has in the first place. This gives us the right idea of how to then train a neural network. Therefore, feature importance is the main basis for our generation of playlists

Traditional test cases and metrics cannot really be applied to our project as the result only relies on whether the user is satisfied or not. There is no mathematical or statistical way we can figure out if the given playlist actually alleviates the user's feelings

CHAPTER 9

CONCLUSION AND FUTURE SCOPE

CHAPTER 9 CONCLUSION AND FUTURE SCOPE

9.1 CONCLUSION

A thorough review of the literature tells that there are many approaches to implement the Music Recommender System. A study of methods proposed by previous scientists and developers was done. Based on the findings, the objectives of our system were fixed. As the power and advantages of AI-powered applications are trending, our project will be a state-of-the-art trending technology utilization. In this system, we provide an overview of how music can affect the user's mood and how to choose the right music tracks to improve the user's moods. The implemented system can detect the user's emotions. The emotions that the system can detect were happy, sad, angry, neutral, or surprised. After determining the user’s emotion, the proposed system provided the user with a playlist that contains music matches that detected the mood. Processing a huge dataset is memory as well as CPU intensive. This will make development more challenging and attractive. The motive is to create this application in the cheapest possible way and also to create it under a standardized device. Our music recommendation system based on facial emotion recognition will reduce the efforts of users in creating and managing playlists.

9.2 FUTURE SCOPE

This system, although completely functioning, does have scope for improvement in the future. There are various aspects of the application that can be modified to produce better results and a smoother overall experience for the user. Some of these are an alternative method, based on additional emotions which are excluded in our system as disgust and fear. This emotion included supporting the playing of music automatically. The future scope within the system would style a mechanism that might be helpful in music therapy treatment and help the music therapist to treat the patients suffering from mental stress, anxiety, acute depression, and trauma. The current system does not perform well in extremely bad light conditions and poor camera resolution thereby provides an opportunity to add some functionality as a solution in the future.

REFERENCES

[1]Vanden Oord, Aaron and Dieleman, Sander and Schrauwen, Benjamin, Deep content based music recommendation, Curran Associates, Inc., Volume 26, 2013.

[2] Zheng H-T, Chen J-Y, Liang N, Sangaiah AK, Jiang Y, Zhao C-Z. A Deep Temporal Neural Music Recommendation Model Utilizing Music and User Metadata. Applied Sciences. 2019; 9(4):703.

[3] Y. Yi, X. Zhu, Y. Yue and W. Wang, "Music Genre Classification with LSTM based on Time and Frequency Domain Features," 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), 2021, pp. 678-682.

[4] Fessahaye, Ferdos & Perez, Luis & Zhan, Tiffany & Zhang, Raymond & Fossier, Calais & Markarian, Robyn & Chiu, Carter & Zhan, Justin & Gewali, Laxmi & Oh, Paul. (2019). T-RECSYS: A Novel Music Recommendation System Using Deep Learning. 1-6. 10.1109/ICCE.2019.8662028.

[5] J. Fang, D. Grunberg, S. Luit, and Y. Wang, “Development of a music recommendation system for motivating exercise,” in 2017 International Conference on Orange Technologies (ICOT), Dec 2017, pp. 83–86.

[6] M. Jiang, Z. Yang, and C. Zhao, “What to play next? a rnn-based music recommendation system,” in 2017 51st Asilomar Conference on Signals, Systems, and Computers, Oct 2017, pp. 356–358.

[7] T. Hayashi, N. Ishii, M. Ishimori, and K. Abe, “Stability improvement of indirect matching for music information retrieval,” in 2015 IEEE International Symposium on Multimedia (ISM), Dec 2015, pp. 229– 232.

[8] B. K. Sunny, P. S. Janardhanan, A. B. Francis, and R. Murali, “Implementation of a self-adaptive real time recommendation system using spark machine learning libraries,” in 2017 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), Aug 2017, pp. 1–7