MOODSHOT SONG RECOMMENDER BASED ON FACIAL EXPRESSION

by

Nurhande Akyüz
Diala Jassem Mohammad Bader Jubeh
Münevver Sueda Kocatürk

CSE497 / CSE498 Engineering Project report submitted to Faculty of Engineering in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

Supervised by:
Assoc. Prof. Mustafa Borahan Tümer

Marmara University, Faculty of Engineering

Computer Engineering Department

2021

MOODSHOT SONG RECOMMENDER BASED ON FACIAL EXPRESSION

by

Nurhande Akyüz
Diala Jassem Mohammad Bader Jubeh
Münevver Sueda Kocatürk

CSE497 / CSE498 Engineering Project report submitted to Faculty of Engineering
in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE

Supervised by:
Assoc. Prof. Mustafa Borahan Tümer

Sign

Marmara University, Faculty of Engineering

Computer Engineering Department

2021

ABSTRACT

In this paper, we are going to propose an web-based application called MoodShot which recommends appropriate songs chosen from the playlists to the users based on their mood states captured from their facial expressions and provide an opportunity to users to share the songs they are listening to. The goal of the project is to uplift the mood of people, to increase social interaction among people and to present a new way of discovering music. This project consists of five phases; face detection, facial expression recognition, categorization of songs by moods, song recommendation according to facial expression and share of songs. Mainly, the project centers on two primary models which are facial expression recognition and song classification which were built using Python with the use of TensorFlow and OpenCV libraries, as well as Keras API with the data taken from Kaggle to extract facial emotion from the picture taken by the user, and from Spotify API to classify the songs according to emotions. For the user interface, Bootstrap framework was implemented with HTML coding. After defining the requirements and drawing the blueprint of the project, the implementation of the project finished and ready to be used.

ACKNOWLEDGEMENTS

Throughout the writing of this dissertation, we have received a great deal of support and assistance. We would first like to show our gratitude to all of the Department faculty members for their help and support. We also thank our parents for the unceasing encouragement, support and their wise counsel. Finally, we could not have completed this dissertation without the support of each other.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Music has become an omnipresent companion to everyday lives of people day to day and listening to music has been one of the most prominent leisure activities [1]. Distinct music genres have a vital influence on mood states and emotions. Moreover, it has long been considered as an effective and powerful method for regulating emotions and triggering moods [2]. Thus, we would like to establish the relation among mood states of people and music genres. The focus of the project was to create a web-based application to propose appropriate songs chosen from the playlists to the users based on their mood states captured from their facial expressions and provide an opportunity to users to share the songs they are listening to.

1.1 Problem Description and Motivation

Music is just one of the ways to express feelings. One may use music as a tool to reveal repressed emotions, so it is undeniable that music and mood are closely related. Our aim was to recommend songs to the users based upon their state of mood according to their captured facial expressions.

According to McCraty's research, studies have proven that music affects emotions and mood states. In that research, a group of people has been exposed to two different types of music and while they were listening to this music their mood swings were observed. For instance, with grunge rock music, while important increases were observed in hostility, sadness, tiredness, and tension, decreases were shown in relaxation, mental clearness, and energy [3]. The impact of music is also studied in medicine. Moris states that even before and during surgery, patients display lower anxiety levels when they listen to music [4]. To conclude, as scientifically proven, music has a significant correlation with states of mood. Within music, there are some elements such as sound, tempo, melody, harmony, rhythm, texture etc. Combination of these elements will help us to upraise people's mood states.

In our lives, we face many problems that affect our moods. In order to continue our social and professional lives successfully, it is important to keep our moods at high levels, so our project's main goal was to devise an efficient tool to upraise the user's mood by suggesting the right song for the user's current state of mood to play; moreover, it has the feature of being able to share it with other friends. Sharing any media is a need for social relations. To promote social relations between people using a new kind of song, we increased people's social needs.

Currently, there are a wide variety of recommender systems for different products such as film, e-commerce, or book recommenders. The motivation behind our project was the lack of such a recommendation system which suggests songs based upon the user's facial expressions detected uniquely and offers other features such as concurrently sharing the song played with others as well as having a rich library of songs in multiple languages. This is a new way of playing music on emotions. People are able to discover new music. People are able to express their moods verbally, but taking a selfie will give faster results about one's mood.

We built our project as a time and cost saver which shall constitute a successful example of human-computer interaction.

1.2 Aims of the Project

By virtue of studies that clearly show that faces are rich in information about one's mood and mental state [5], we implemented the technique "deep learning" in a user-interactive music platform, including the following aims:

The first aim of the project was to build a model that acts as a facial expression detector. For illustration, we will have ready data that includes images, with the help of openCV the program will detect the faces in the images and draw a box around them, soon we will use the neural network in keras API to detect facial expressions. Our goal here is to categorize the facial expression for each image in the data (happy, neutral, sad,…). This way the model will be trained and saved to be able to process real-time pictures.

The second aim of the project was to classify some music datasets. By setting a developer account on Spotify, we plan to obtain the audio features for each song[6]. Then a supervised machine learning technique, the k-nearest neighbors algorithm, will be applied (basically it takes a data point and finds the distance between other K labeled data points [7]).

The third aim of the project was to implement an interface to play a song; according to the facial expression model and the music classifier.

The fourth aim of the project was to implicate the feature of sharing the current song with other users.

2. DEFINITION OF THE PROJECT

2.1 Scope of the Project

The scope of our project centered on two primary models: facial expression recognition and song classification. They were built using Python with the use of TensorFlow, and OpenCV libraries (in which they detect a face in a picture and draw a box around it), as well as Keras API which is the wheel of our project; since the image processing will be done using one of this API's libraries (ImageDataGenerator), in the same time it detects the facial expression on the detected face from the previous libraries.

A user interface coded by HTML with the implementation of the Bootstrap framework. Through the interface, a webcam is used for capturing the photo of the user, and the recommendation system takes part in suggesting a song (using the k-nearest neighbour algorithm). Besides, an optional feature was added to send the song currently playing to another user.

The prerequisite of song selection from a playlist is facial feature detection and we accomplished this prerequisite in our project.

For illustration, the system needs:

- A stable internet connection to run,
- A webcam for function in the relevant steps (open app – take a photo – suggest a song),
- An accurate trained facial expression model, and
- An accurate music categorizer.

As soon as these criteria are satisfied, many software testing are done to let the project process successfully. We are assuming to get access to song datasets from social music platforms such as Spotify in a short period (setting the developer account). All in all, we expect the application to work properly and on time.

Some criteria are out of our scope, which mainly were, ability to send a song in which at that time the sender is not playing it, as well as, the ability to play a song without the camera detecting facial expression step.

2.2 Success Factors and Benefits

- Measurability / Measuring Success

Measuring factors of this project depended on the correct application of the used algorithms, we planned to use both *haar cascade algorithm* and *k-nearest algorithm*. The haar algorithm must take a huge amount of imaged data, apply it to extract features in a detection window, sum up the pixel in every locale, and figure the contrast between these sums. The result of the feature function will be positive of the feature found and negative if not. Taking it to the next step, each haar feature is only a ‘weak classifier’; this algorithm puts all the weak classifiers together to end up with a cascade classifier to form a strong classifier.

The cascade classifier has many stages and will result in:

- True positive: When a positive image is correctly categorized.
- False positive: When a negative image is wrongly categorized as positive.
- True negative: When a negative image is correctly categorized as negative.
- False negative: When a positive image is wrongly categorized as negative.

F1 score is an accuracy measurement, which incorporates more the sensitivity and specificity of the system.

$$F1 = \frac{number\ of\ true\ positives}{number\ of\ true\ positives + 0.5 \times (number\ of\ false\ positives + number\ of\ false\ negatives)}$$

The accuracy of this algorithm depends on two keys for a correct face detection: measuring the feature function correctly. Getting data that has images that have faces and other images that do not have faces [8].

The K-nearest neighbors algorithm uses the *Euclidean distance function* where we identify the K value, take a data point, and calculate the distance between the k number of labeled data points. The distance should be accurate; thus, the user will not be incorrectly directed (when the song is classified incorrectly to its genre) [9].

- ❑ Benefits / Implications

1. Enhance the user's temperament [10], by directly playing a song based on the specific mood detected.

   For instance; the program can detect that if a person is angry; it automatically proposes options for playing calming songs or other songs from the user's choice as a result.
2. Increase social contact while observing the 'keep the physical distance' rule, especially with the corona pandemic. Sharing a song with friends strengthens relations even if they are far from each other.
3. Helpful for (treatment by music) mechanism. As explained in "Music as Therapy" [11] music distracts patients from unpleasant symptoms. Moreover, most hospitals use a specific music type not considering patients' different moods. Our application is user centered for each user depending on their list of songs besides the mood detected.

2.3 Professional Considerations

- ❑ *Methodological considerations / engineering standards*

I. We applied the Scrum project management principles for managing our project.

II. We used UML Diagrams in order to help us regulate, plan, and visualize our project deeply.

III. Gantt chart was used for time management.

IV. We used GitHub for repository management while developing our software project.

V. We used the Python programming language as the main software since Python has rich libraries especially for machine learning that reduces development time and process.

- ❑ *Social / ethical considerations*

I. Ethical perspective: The project is strongly secured; each user has their private profile that includes their song lists.

   Another point, the project aimed to strengthen people's relations through sharing songs; thus, achieving social coherence.

II. Health and safety: The project concentrates on how to mood up the user; simply because of "Music Listening Releases Dopamine"(happiness hormone) [12].

III. Sustainability: Updating the music dataset is the key to sustaining the app for a long time.

- *Legal considerations*

  The project does not need any legal issues. We used public datasets taken from Kaggle and Spotify API.

2.4 Literature Survey

There exists some works in literature [13-17] for facial expression recognition systems applying different methods and music recommendation systems which use many different algorithms. Some of them are explained below.

2.4.1 Face Detection and Facial Expression Recognition System

Dhavalikar and Kulkarni proposed an *automatic facial expression recognition system* with three phases: face detection, feature extraction and facial expression recognition. In the first stage, face detection, they used RGB and lightning compensation for skin color detection, morphological operations for face detection. In the second stage, feature extraction; *active appearance model method* is applied since it includes and combines texture information and shape from facial images, also it locates points on the facial features such as mouth, eyes and eyebrows and it creates a data file including information about model points located on the detected face. Data is trained for facial expressions according to points on the face. In the last stage facial expression recognition; *Euclidean distance method and artificial neuro-fuzzy inference system* are used to analyze facial expression [13].

Our intended approach was to analyse the current facial expressions captured from a webcam and propose a list of songs related to the emotional status of users. However, this project uses images of users recorded a priori into hard disks for face detection and facial expression recognition process. Also, this project does not require a music recommendation system which was one of the main goals in our project.

2.4.2 Spotify

Spotify is a Swedish, digital, cloud-based music platform launched in 2008. It offers over 50 millions songs with an expected 5 million playlists and a rapidly-increasing number of podcasts [14]. Spotify applies not only one algorithm but three algorithms for the recommendation model: *collaborative filtering*,*natural language processing* and *audio models*.Collaborative filtering is a recommendation method which suggests songs to a user based on correspondence between the other users' similar taste of music preferences. Some of the common tasks in natural language processing (NLP) that consider the playlist itself as a document are speech recognition and segmentation, text-to-speech and tokenization. Spotify uses NLP in order to recommend songs based on similar lyrical patterns. Audio models algorithm is applied for examining data from audio tracks and classifying songs accordingly by using convolutional neural networks.

Our intended approach was to suggest songs to users according to their mood status. However, Spotify uses various methods for recommendation such as collaborative filtering, natural language processes etc. Therefore, the main difference between Spotify and our project is the distinct approaches for music recommendation systems. In other words, in our project we were going to use facial expressions where Spotify uses collaborative filtering.

2.4.3 An Accurate Algorithm for Generating a Music Playlist based on Facial Expressions

Dureha proposed an accurate algorithm which produces and recommends a list of songs from the playlists of users according to their mood states by detecting their facial expressions. This automated music recommendation system consists of three main modules which are facial expression recognition module, audio emotion recognition module and system integration module. In the first module facial expression recognition; RGB is applied using the image of a user either detected from a webcam or obtained from a hard disk in order to categorize emotions and form a class. In the second module which is audio emotion recognition; *Audacity* is used in order to classify songs taken from the users' playlists according to the related emotions with respect to their frequencies. Hence, the first two modules are mutually exclusive components; in order to relate both these two modules a system integration module is applied using intermediate mapping with the aid of the audio metadata [15].

Our intended approach was to combine both the facial expressions recognition system and the music recommendation system, also there is a feature for users where they can have connections with their friends and they are capable of sharing the songs they are currently listening with their friends. However, in this project, even though it is a music recommendation system based on users' emotions, this project does not require any feature for network among users.

2.4.4 Emotion Recognition from Facial Expression Based On Bezier Curve

Bansal and Nagar proposed a technique containing two steps for face detection and emotion recognition. In the first step, by using the Canny *edge detection algorithm,* the location of eyes and mouth is found and distinguished from a face and cropped. In the second step, recognition of facial expression is done in four stages which are *skin filtration*,*big connect operation*, *applying Bezier*, and recognition of facial emotion. Skin color is detected by utilizing the YCbCr model (Y for luminance, Cb for chroma blue and Cr for chroma red) which is a color model in order to compress colors. In the result of skin filtration, there are diverse shading other than skin shading and in order to discover the exact area containing eyes and mouth, a big connect operation is applied. They applied the Bezier curve after they detected the location of eyes and mouth and according to the training set containing four emotions with individuals' own bends of outward appearance results, they recognized the facial expression [16].

Our intended approach was to create an application which analyses the current facial expressions captured from a webcam and recommend a list of songs related to the emotional status of users. However, this project uses images of users and there are only four emotions. In addition, this project does not require a music recommendation system, which was one of the main goals in our project.

2.4.5 Emotion Based Music System

Rabashette et al. proposed a system which recommends a playlist of songs suitable to the mood after the user's emotion is detected. There are three phases of this system which are facial detection, feature point detection and recommending a playlist related to the mood of the user. For the facial detection phase, they used a *haar feature based cascade classifier,* in the training data there are positive images which are the images of faces and negative images which are without faces. In the feature point detection phase, they convert the images from red-green-blue color model (RGB) image to a binary image by taking average values in RGB. Then, the system scans the images and detects eyes, lips and nose. For extracting expression features, the system uses the distance between upper lip and lower lip which will give two moods containing happy and surprised. According to the expressions detected, the system will provide suitable playlists to a user [17].

Our intended approach was to recommend a playlist according to the facial expression of the user which is captured from the webcam. Even though this project has the same aim with our project, they used only lips for detecting facial expressions where we used both eyes and lips. Also, they have two moods where we have five moods.

3. SYSTEM DESIGN AND SOFTWARE ARCHITECTURE

3.1 Project Requirements

3.1.1 Functional Requirements

3.1.1.1 **Accurate Facial Detection**

- Description: The requirement of detecting a face via a well-trained classifier; in which to distinguish if a picture has a face or not.
- Inputs: A clear image from the webcam.
- Processing:
  a. Webcam opens (video capturing starts)
  b. Haar cascade algorithm functions on a defined cascade classifier.
  c. Looking for a face via applying the classifier on the real-time video.
  d. When a face is detected, a rectangle will be drawn around it.
  e. Captures a picture from the video at a specific moment (Is set to be when the rectangle is one; one face detected only, not more or less).
- Outputs:
  a. True for an input image with a face (with a rectangle drawn around it).
  b. False for an input image without a face.
- Error / Data Handling:
  a. An expected error can be detecting a face in a picture that does not include any face; this error will be handled by applying huge data that includes many without-faces images and many with-faces images.
  b. Another possible error is moving too much while capturing a picture from the live webcam; the possible considerations for this bug can be, at the moment that face is detected and the rectangle is drawn, the program captures the picture.

3.1.1.2 **Accurate facial emotion detection**

- Description: The requirement of detecting the emotion of the user's face is an engine for providing the user with a song accordingly.

- Inputs: A picture with a detected face (with a rectangle around the face; showing its features).
- Processing:

  a. Take the input picture.
  b. Imply the input picture to the Keras API's ImageDataGenerator to process it.
  c. After processing, apply the processed image to the pre-trained classifier. (the classifier is trained with many grey-scaled images; where each corresponds to a 0-6 range of emotions).
  d. Classifier detects the face's emotion.

- Outputs: an integer between the range 0-6; each number defines an emotion. (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).
- Error/Data Handling: An expected error can be, detecting a wrong emotion for the face; this error will be handled by training the classifier with enormous images that have different emotions. Thus, the classifier's accuracy increases, and the classifier's results purify.

3.1.1.3 **Functioning securely when playing music**

- Description: Playing a song is the main goal for the program, it is a requirement that functions in a secured way; for illustration, each user has his/her private list of songs in his/her private profile. Besides, each user can take part in adding a new song file to his/her list.
- Inputs:
  a. User's list of songs.
  b. Number of the emotion detected previously.
  c. Song file. (optional)
- Processing:

  a. Add a new song file to the own user’s list.(Optional)
  b. Check each song on the user's private list.

c. If this song is in the program's songs database, append it in a temporary list that defines its category.
d. If this song is not in the program's songs database, apply the song for the KNN categorizer. Then, get the matched category with the number of emotions.
e. Shuffle the category's list of this specific user and play a random song from the list.

- Outputs:

a. A song plays privately.
b. A new song added to the user's private list.(Only if the user has added a new song file)

- Error/Data Handling: An expected error can be not finding any song for the specific emotion in the user's list. This error will be handled by letting the user participate and permitting the program to suggest him/her a song, not from his/her private list.

3.1.1.4 **Successful p2p network**

- Description: A network is needed so that one user can send another user a song; whereas the user is listening to it at that time, and the receiver will listen to it concurrently.
- Inputs:

a. Sender's concurrent playing song.
b. Receiver's username.

- Processing:

a. Create a p2p network between sender and receiver with the use of their usernames.
b. Send the concurrent playing song, with the already played amount of time of this song.
c. Receiver receives a notification for acceptance.

d. Receiver listens to the song at the same time.

- Outputs: A song plays on sender and receiver concurrently.
- Error/Data Handling:

Some cases can occur with network requirement, they are:

a. The sender did not get any acceptance from the receiver: This type of error can be handled by letting the sender continue his song, but a notification will be sent to the receiver containing the name of the previously sent song and the name of sender.
b. Two users want to send songs to each other at the same time: they cannot! The program will have a synchronization process to choose which user sent first, then the program suggests the other user (with the song sent secondly) if he/she wants to close his/her song and listen to the sent one.
c. A user sends a song while another user is listening to a song (not available): A notification displays on the sender screen ("Your friend is unavailable") and the sender continues to listen to the song played. Besides, a notification displays on the receiver screen ("Your friend wanted to share a song with you, reply to him with a song"), and display the sender's name.

3.1.2 Nonfunctional Requirements

3.1.2.1 **Performance**

- Response time will be less than 4 seconds
- Application's landing page must support 6000 users per hour
- User input in the application must be provided without any obstacle

3.1.2.2 **Reliability**

- To succeed high reliability, all bugs must be eliminated in the code

- Before user performing a significant action on app, user must be asked for acknowledgement

3.1.2.3 **Usability**

- Application must be understandable for users
- User actions must be done responsively by the app
- Implementation of the project must be easy to understand for developers to work on it efficiently
- User’s guide manual will be updated when the changes occur in the system

3.1.2.4 **Security**

- To be protected from cyber-attacks, all the data must be encrypted with most reliable encryption methods
- Users' authentication processes must be saved on the local device to compare and it is needed to gain access after permission
- When the SQL injection occurs, Sqlmap tool will be used
- Connection will be encrypted with SSH, when the connection problems occur

3.1.2.5 **Maintainability**

- To be able to reach a high rate of bug repair, maintainability rate must be %77 for debugging in 24 hours.

3.1.2.6 **Portability**

- The system works within a python environment, where it can be moved to mobile devices using the framework React Native within a stable internet connection.
- Python is cross-platform; it is possible to evolve iOS and Android applications; consequently, the portability of the system increases.

3.1.2.7 **Scalability**

- Application must be capable of handling increasing user data without any delay as time progresses.

3.2 System Design

3.2.2 UML Use case Diagram(s) for the main use cases

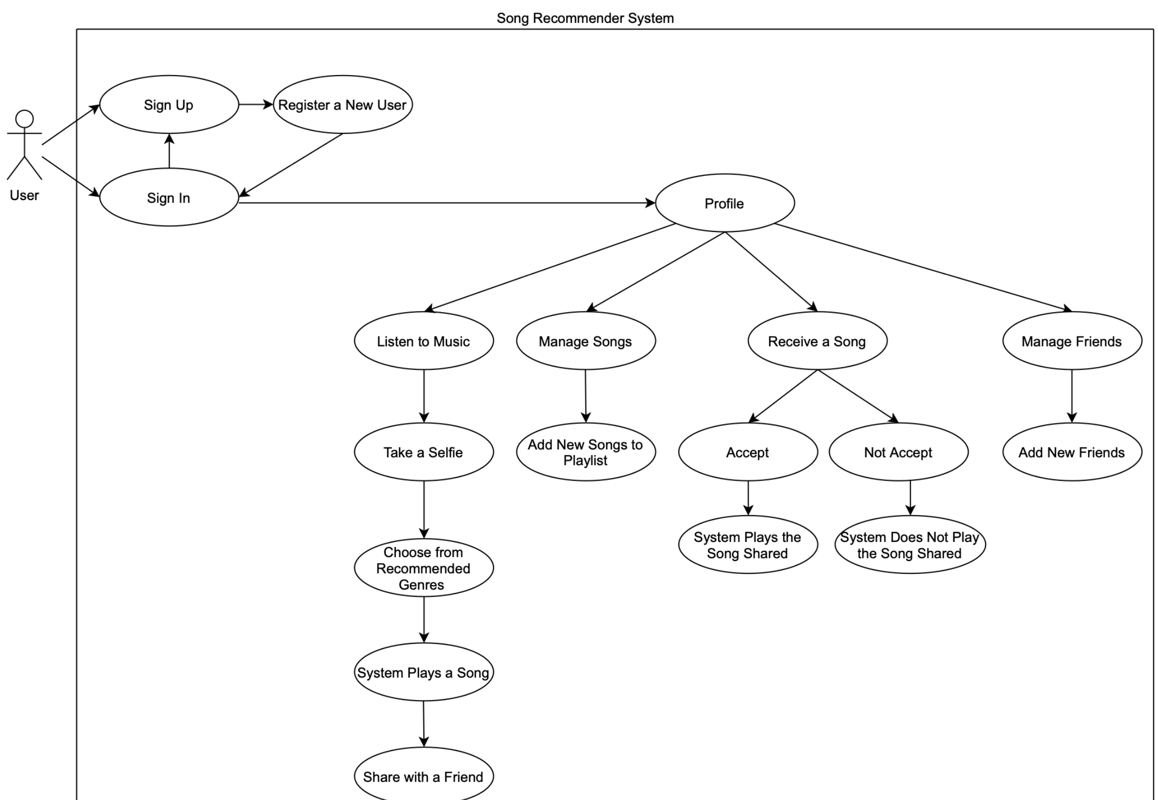

Figure 1: Use case Diagram of Song Recommender System

Figure 1 shows the main case diagram of our system. The user should sign up for an account. Then each user will have their own profiles where they can listen to music, manage their songs, receive a song from other users or manage their friends. In order to listen to music, users should take a selfie and according to the user's mood the program will recommend genres; users should choose a genre and according to the genre the user chosen program will play songs and while listening music users can share the listening music with friends. In order to manage songs, users will add songs they like to their playlists. Users will have an option about receiving songs: they can accept the songs or not accept them. According to the choice of user, either system will play shared songs or not. In order to manage friends, users can add their friends.

3.2.3 UML Class and/or Database ER diagram(s)

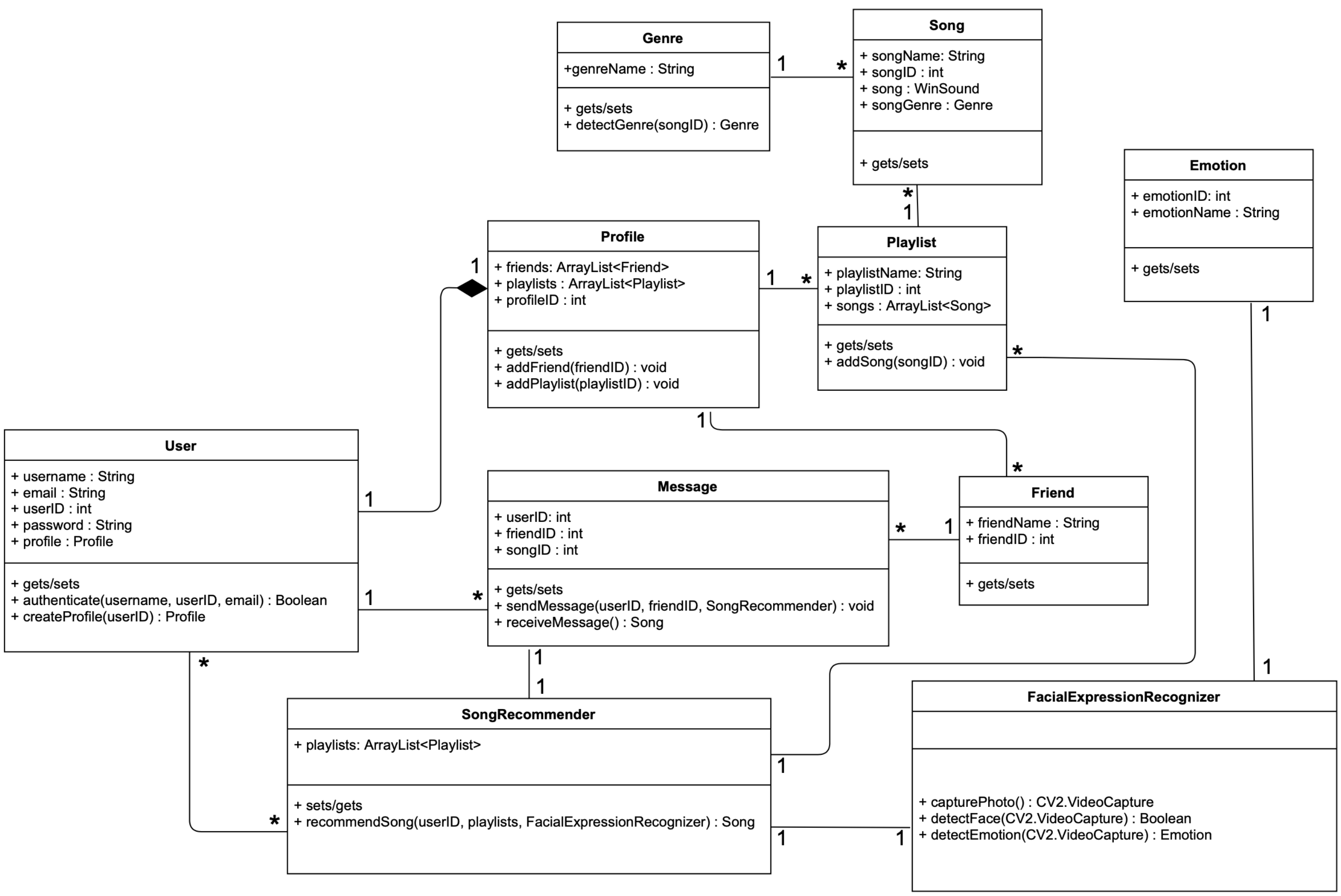

Figure 2: UML Class Diagram

Figure 2 shows the UML class diagram of our system. There are ten classes, which have multiple relations between each other.

3.2.4 User Interface

In Figure 3, the first screen is shown which is the main page.

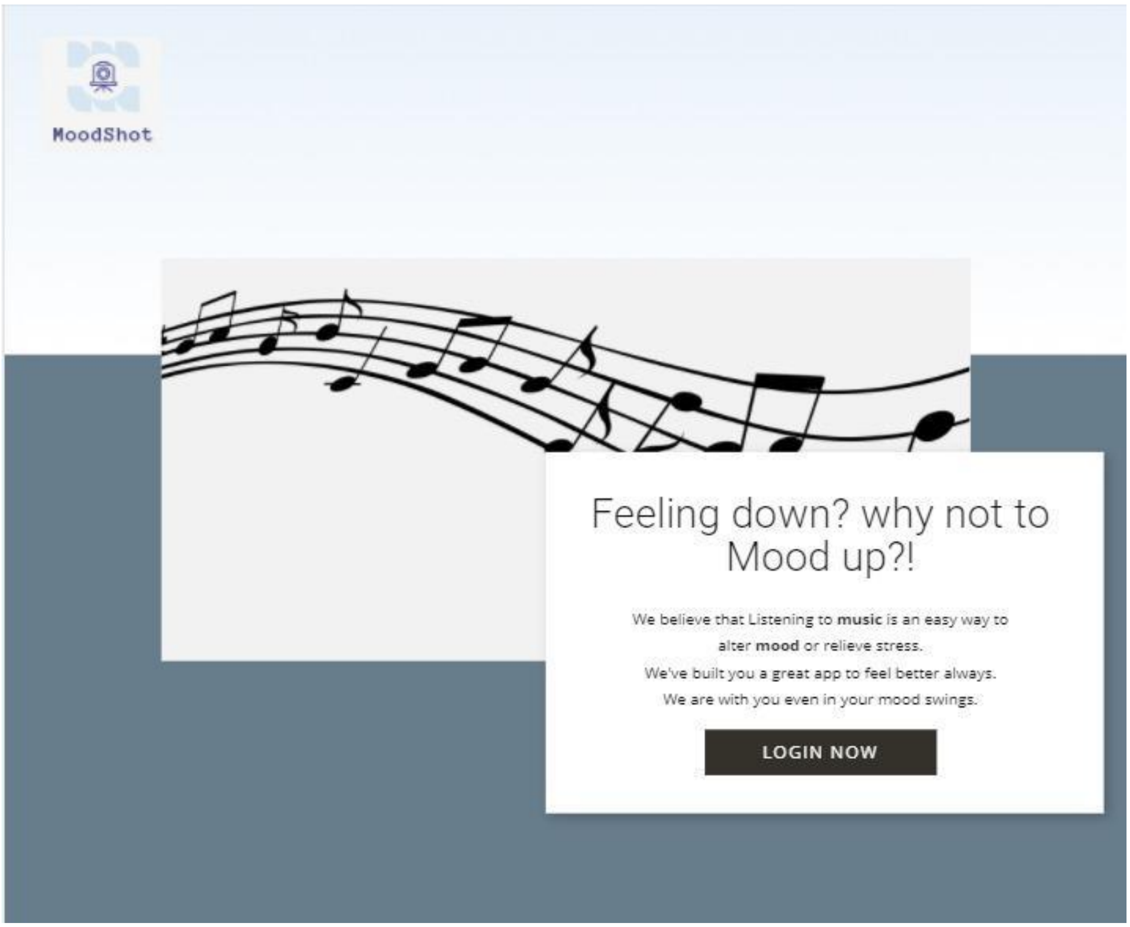

Figure 3: Main Screen

Second screen, which is shown in Figure 4, shows the signing in and signing up page.

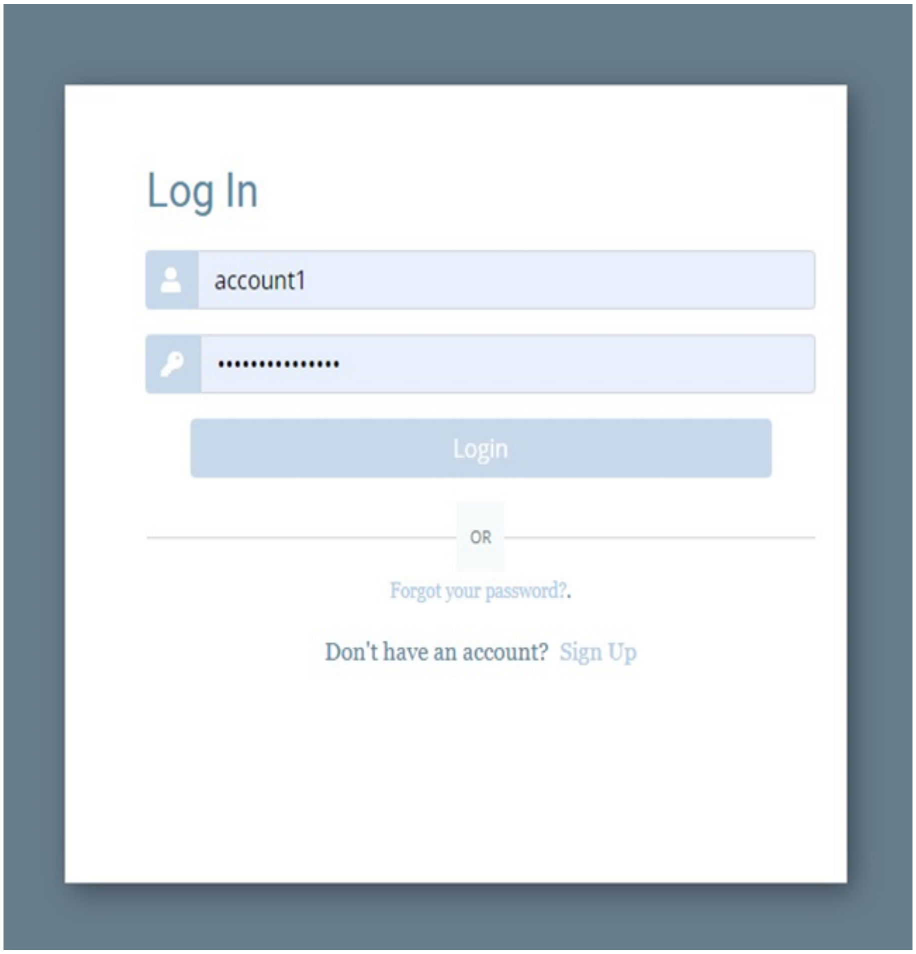

Figure 4: Sign Up / In Page

Third screen, which is shown in Figure 5, shows the profile of each user; displaying his/her own playlists as well as friends.

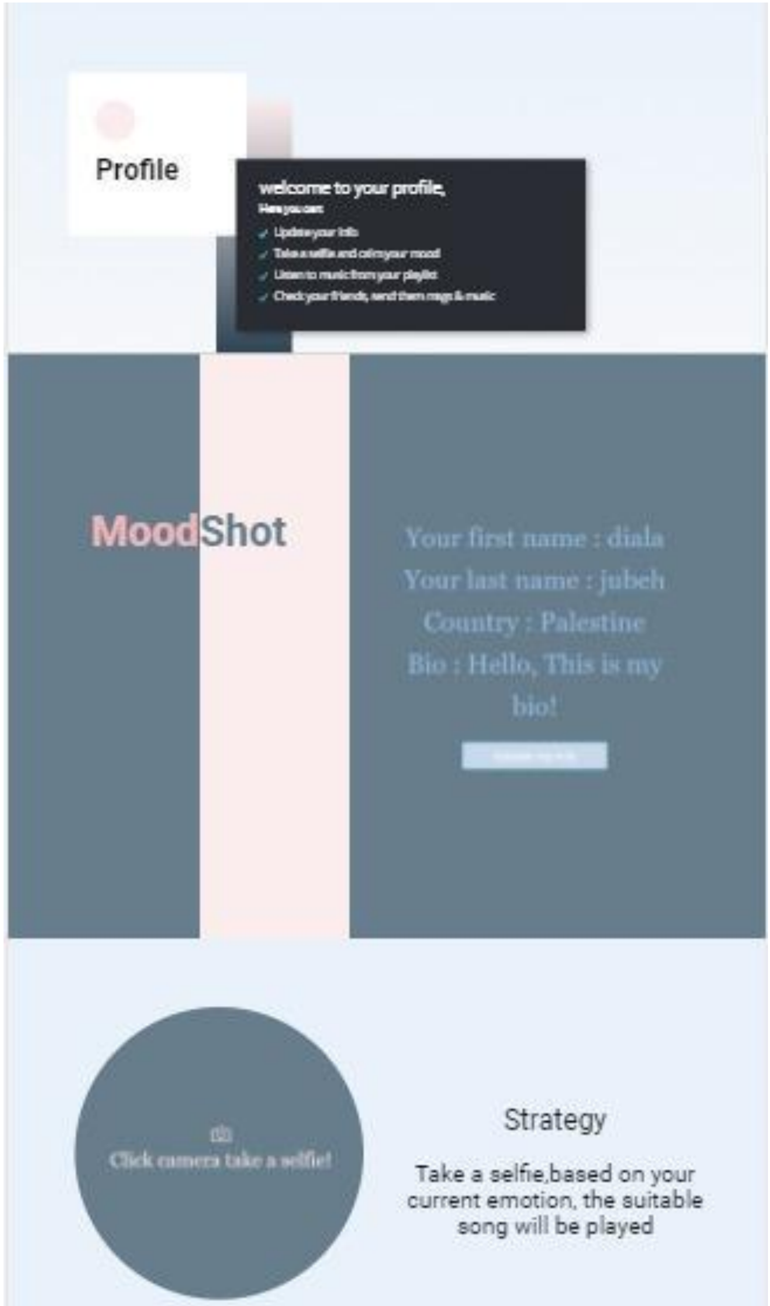

Figure 5: Profile Screen

Fourth screen, which is shown in Figure 6, shows the music player page where the user sees the song description, besides there is a choice for selecting one of the user's friends and sending the playing song.

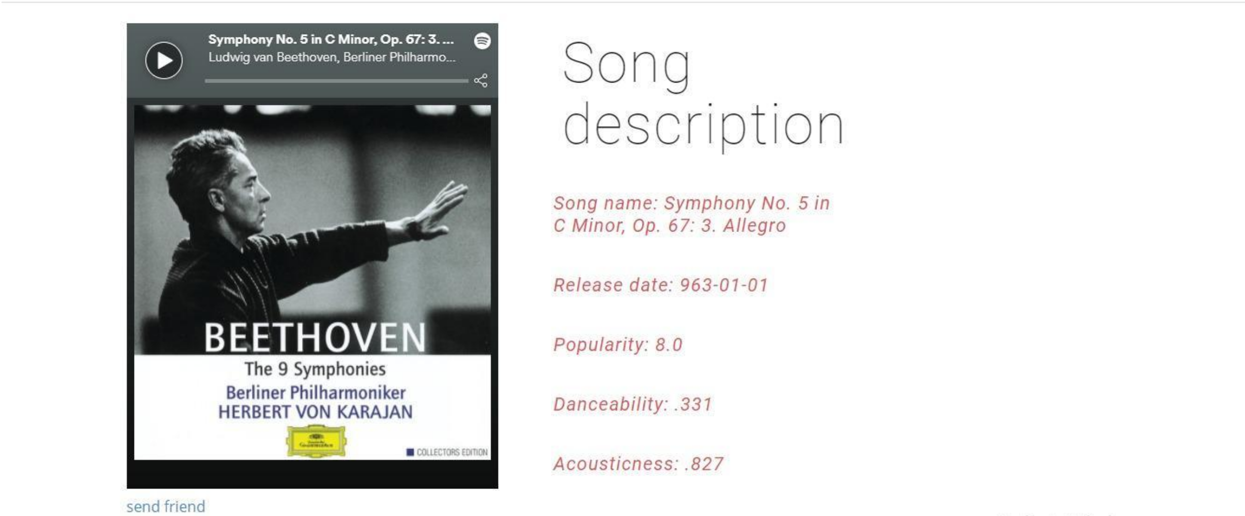

Figure 6: Music Player Screen

Fifth screen, which is shown in Figure 7, shows the sharing page where users share their played songs with their friends.

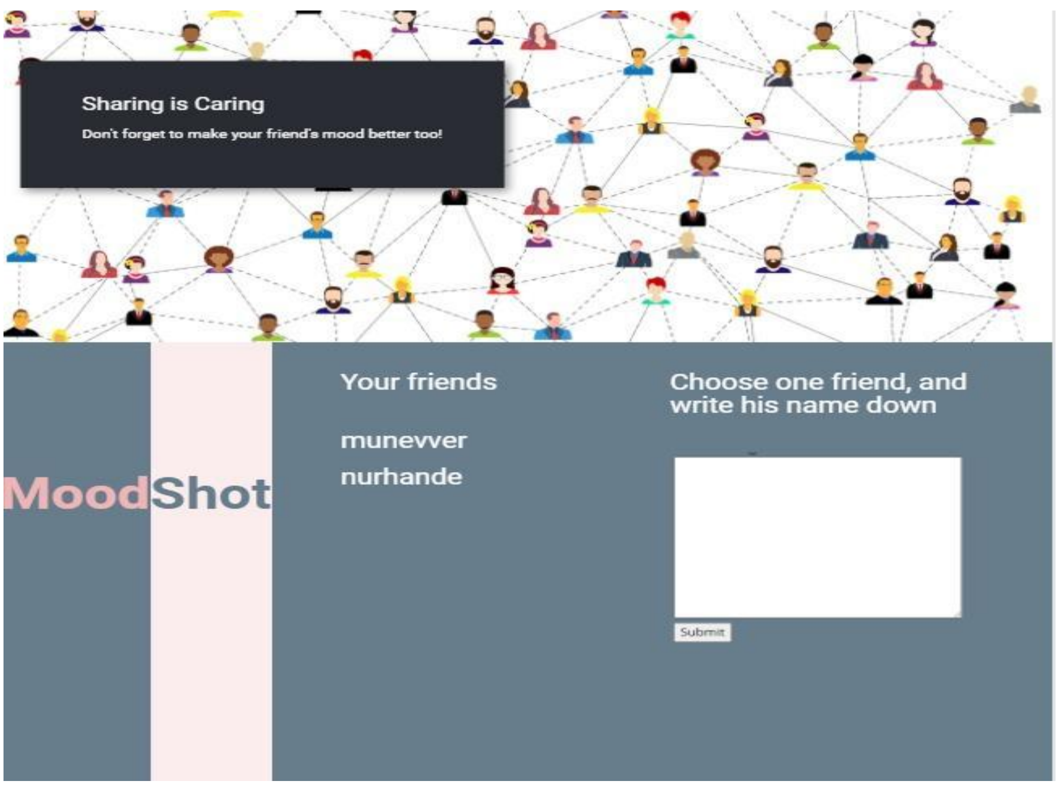

Figure 7: Sharing Screen

3.2.5 Test Plan

Testing is a key for a project to succeed; we plan to divide the project into smaller tasks, wherein each task we will develop a piece of code and a testing strategy for this specific task; if the test passes, we move on to the next task, and so on...

Testing the three classifiers

The data we collected contains both training and testing datasets, soon after a model is trained, a test will be applied to verify its operation. The testing data will implement the already trained model, as a result, a confusion matrix is used to conceive the performance of the model (using confusion_matrix imported from sklearn.metrics). In the matrix that cells illustrate images predicted correctly (TP or TN) or wrongly (FP or FN). Given the confusion matrix, accuracy can be tested using the following equation.

Accuracy (Emotion detection) = Correctly classified images upon emotions / The total number of test images.

We assume accuracy would be more than or equal to 70%.

The models applying the testing:

1. **Testing face detection model's accuracy:**

   The testing goes as follows; we will add up the number of test images that were classified correctly (same result from the classifier and the test data), if it was correct we will give it 1; if not, 0. Over the total number of test images, if the result was more than 80% the test passes, otherwise it fails.If it fails, we plan to reconcile the classifier via re-training it, we will assign each photo a probability of how much it includes features from the OpenCV (by using a function for OpenCV called setInput() and forward()), this number should be more than or equal a defined threshold (used to filter weak face detections). As a result, the model will be trained again but with higher limitations.

2. **Testing emotion detection model's accuracy:**

Using Keras API to classify each test image to its corresponding emotion, then comparing the emotion resulted with the data result.

3. **Testing the music's genre classifier:**

This classifier is different from other classifiers, where it does not include a testing dataset; thus we will apply train and test split on the dataset using the cross-validation method, then classify them using the model KNN; as a result, we get the accuracy on test data.

Network testing:

It is accomplished with the help of SQLite database.

Overall testing:

This comes in the last parts of the project, it includes checking each aim of our project if it is met or not. Also, it includes checking the user interface. All the app actions must be shown on the screen properly. The GANTT chart for the test plan is given in Figure 8.

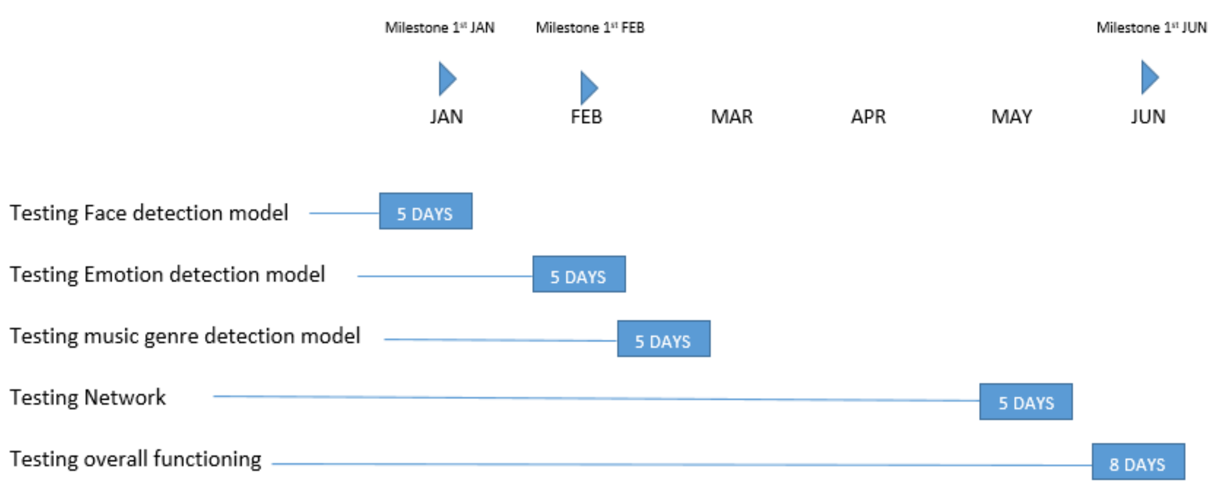

Figure 8: GANTT chart for Test plan.

3.3 Software Architecture

3.3.1 Data Flow

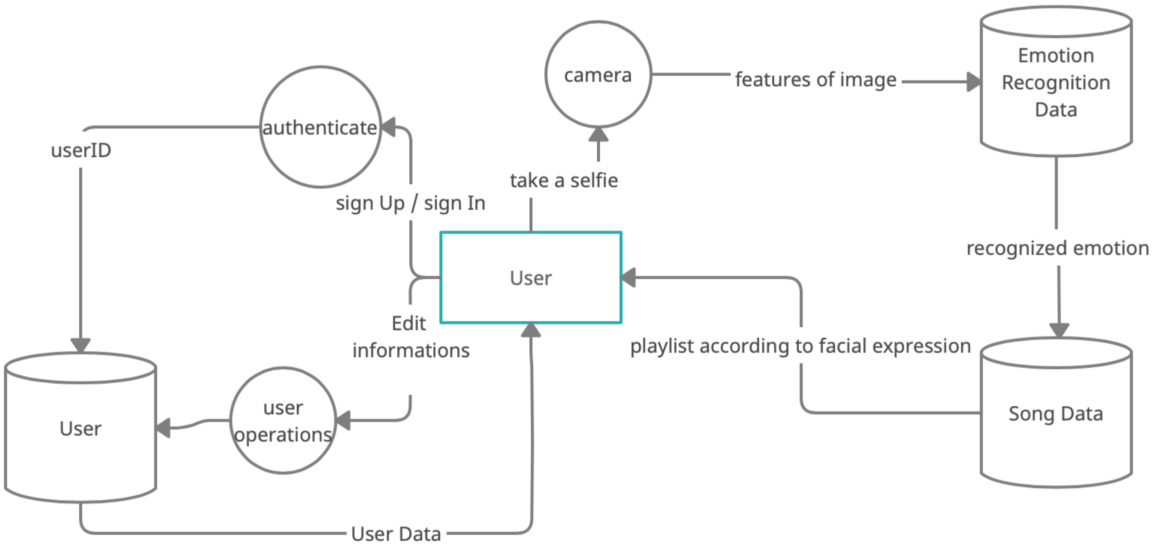

Figure 9: Data flow of the main system

There are three databases shown in Figure 9: *User*, *Emotion Recognition Data* and *Song Data*. *User data* contains all the information about users such as songs they listen to, friends they follow and personal information of users. *Song data* is the data taken from Spotify API in order to classify the songs according to emotions. *Emotion recognition data* is the data taken from Kaggle to extract facial emotion from the picture taken by the user.

3.3.2 Control Flow

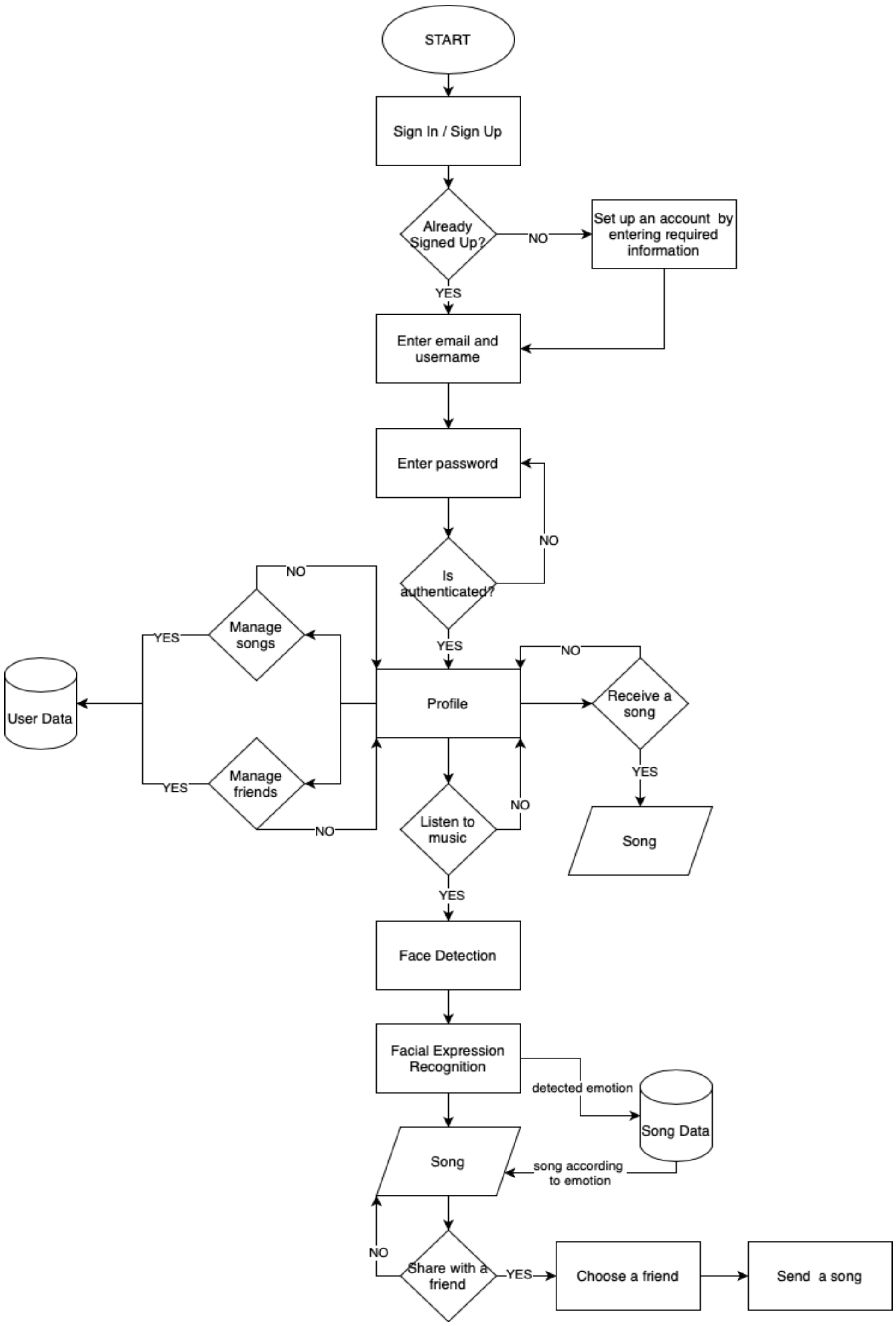

Figure 10: Flow Diagram of the Main System

In Figure 10, the flow diagram shows how the whole system works.

Figure 11 shows the flow of the face detection phase. First of all, the image is captured, preprocessed and converted from RGB to Grey model. Then, the features cascading and calculation process is handled by the haar *cascade algorithm,* and according to comparison of feature values with threshold face is detected.

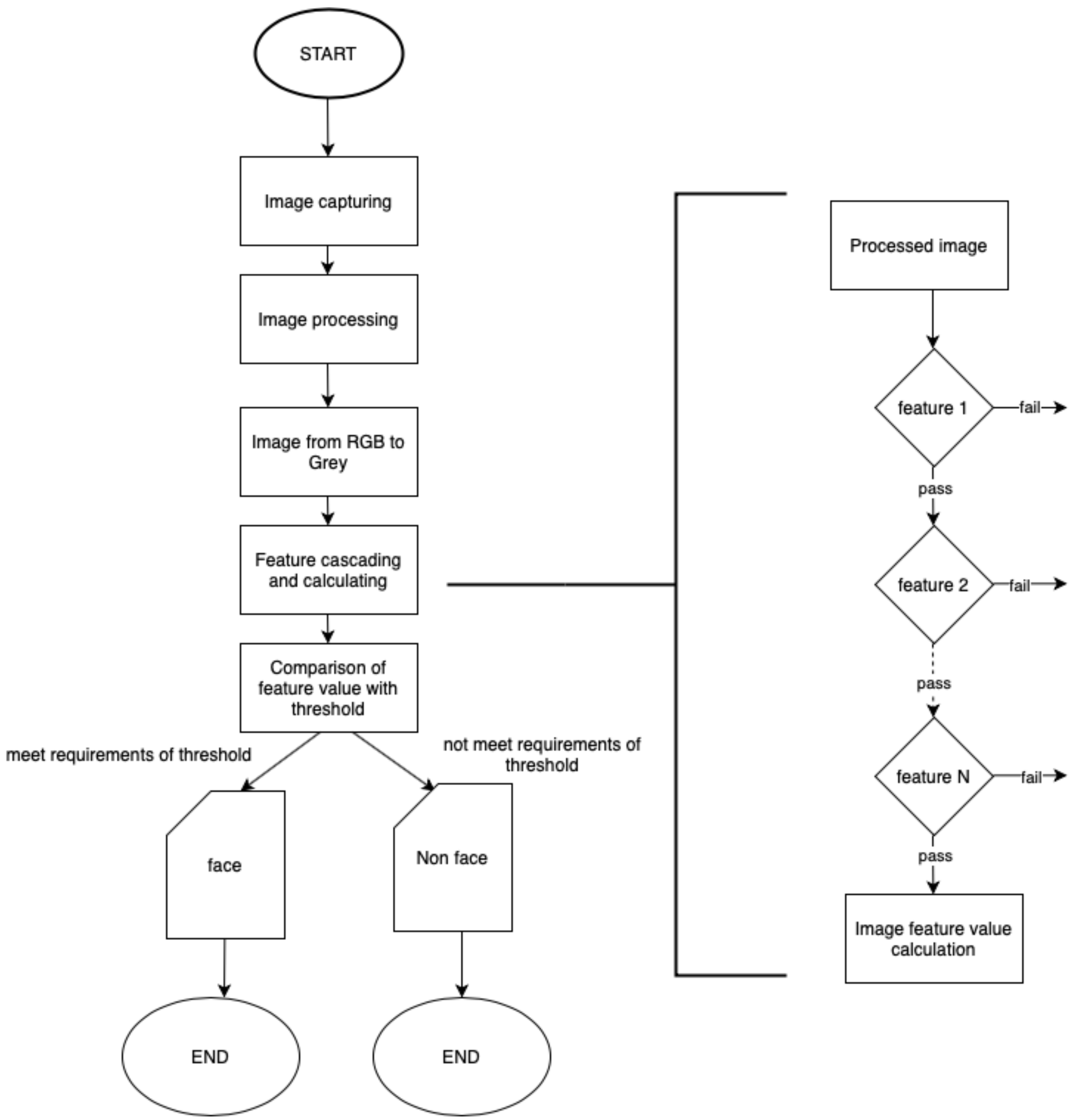

Figure 11: Flow Diagram of Face Detection

Figure 12 shows the flow diagram of facial expression recognition. The image with face is taken as input, then it is processed by removing noises and converted from RGB to Grey color model. Then, according to the data taken from Kaggle, the emotion of the face in the image is recognized and will be used for recommending songs.

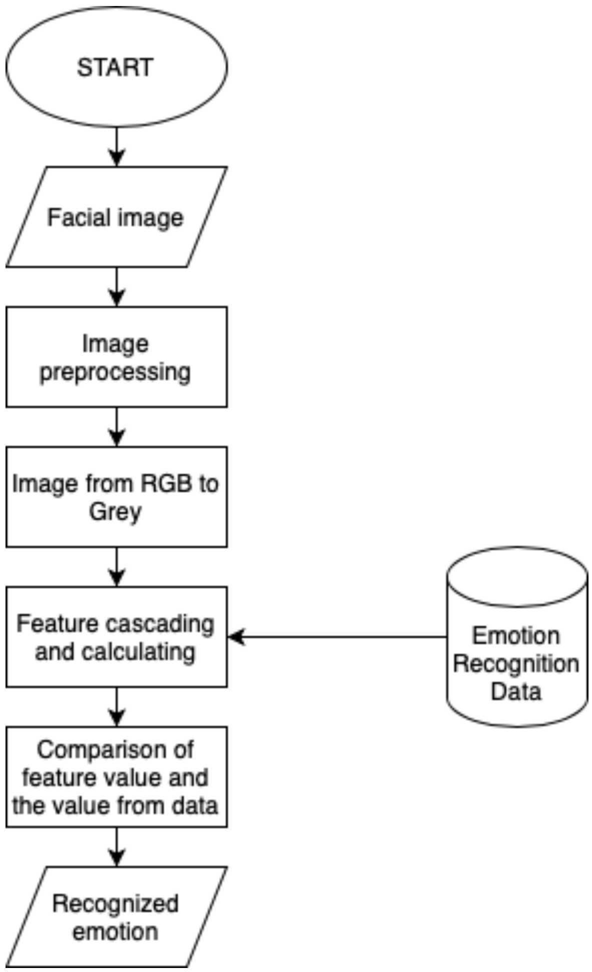

Figure 12: Flow Diagram of Emotion Recognition

Figure 13 shows the flow diagram of song classification. Songs will be classified in two ways (text-based and audio-based). First of all, the audio file of the song is given as input, text of the song and audio of the song is taken and uniquely words and signals are extracted. Then, genre is found with corresponding emotions by using *KNN algorithm*.

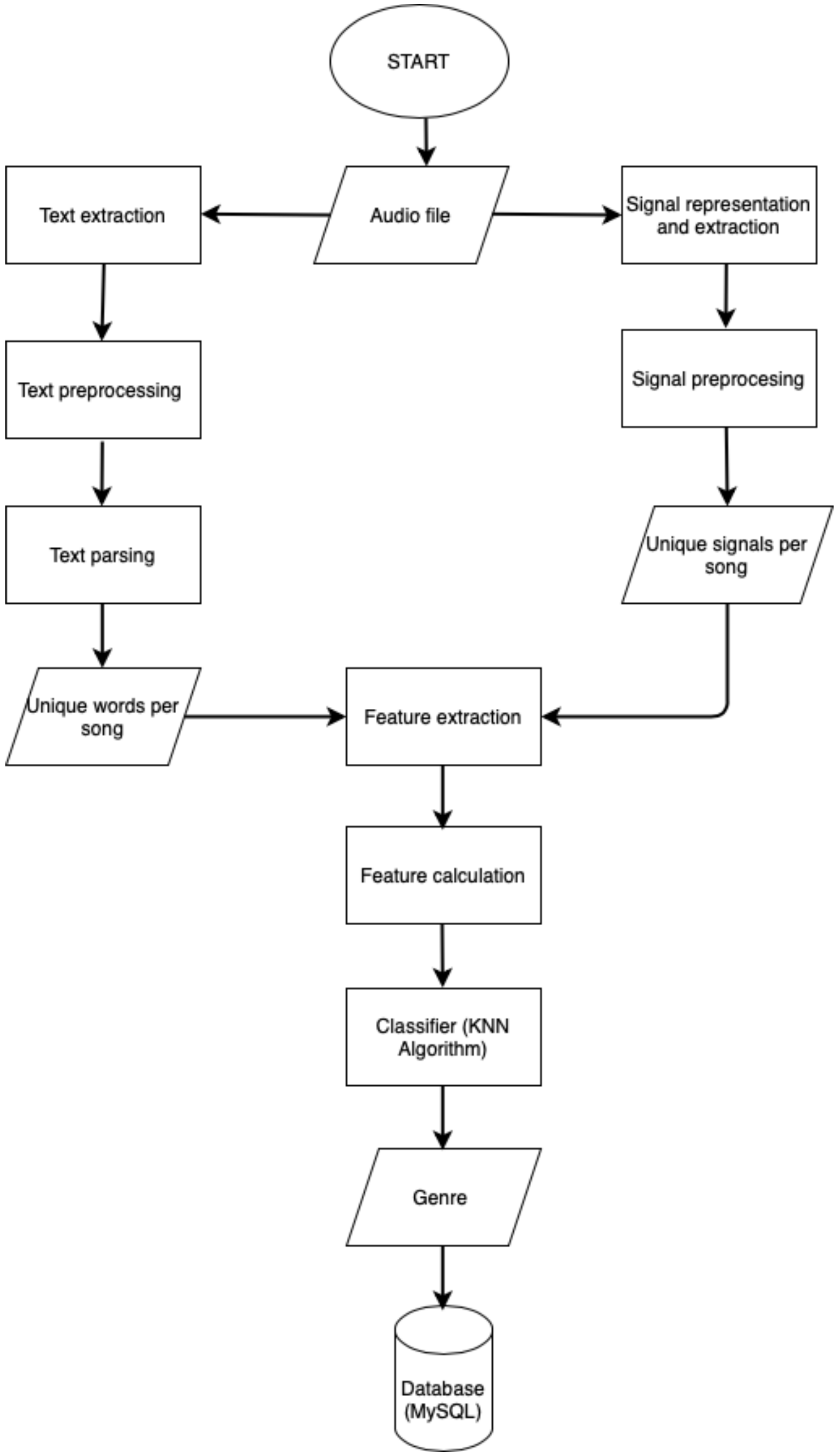

Figure 13: Flow Diagram of Song Classification

4 TECHNICAL APPROACH AND IMPLEMENTATION DETAILS

4.1 Technical overview

Our project consists of five phases:

- face detection
- facial expression recognition
- categorization of songs by moods
- song recommendation according to facial expression
- share of songs

In Figure 14, the process flow diagram of the system is shown.

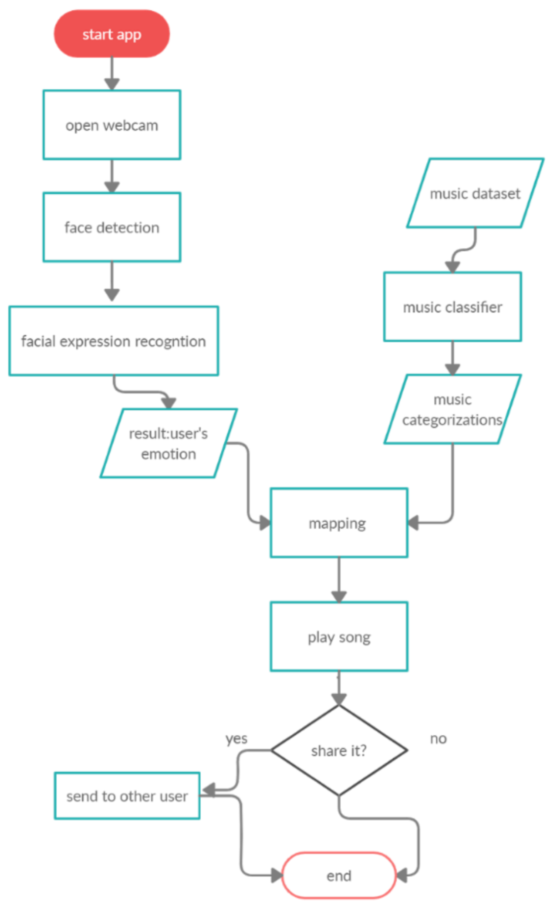

Figure 14: Process Flow Diagram of the System

4.2 Main software requirements

- Dataset is originally from kaggle.
- Preprocessing data was done with Jupyter as well as google colab.
- Face detection uses the algorithm Haar cascade with the help of Viola-Jones formula, besides Adaboost classifier. OpenCV library is used.
- Emotion detection uses the CNN (Convolution neural networks), applying a multilayered network with the usage of keras libraries.
- Song's classifier based on mood status we used Spotify API which provides us information about songs about acousticness, loudness, mood, energy, tempo, valence etc. for acoustic analysis [20].
- Webapp is built from python as a backend, and HTML5, CSS and Javascript for frontend.
- Database used is SQLite.

4.3 Tasks methodology

4.3.1 Data preprocessing

The datasets we manipulate in the project are 3 datasets, as explained before, one that includes images with and without faces (from kaggle), the face images with their corresponding emotion (from kaggle), and the Spotify song dataset (from Spotify API).

The images related datasets are manipulated in GOOGLE COLAB; the reason for choosing this program is the ability to offer GPU and 12 GB of memory free. We upload the datasets, clean them from empty data and/or redundant data, besides, we did data augmentation to have richer data info. The data was firstly in CSV form, later we got the model file as an MD5 file (a checksum file used to verify the integrity of a disc).

The song dataset is manipulated in Jupyter, we chose this program due to its ability to deal with data, some cleaning for data is done, later we got 4 output files each having songs for a particular category. The data was categorized according to multiple criteria, from our own knowledge plus some research [18].

4.3.2 Face detection

Face detection is a major study under the computer vision topic with a wide range of applications.The Haar Cascade method is utilized, which is an Advanced Tool for detecting faces in images or real-time images. The algorithm implies lines and edges detection features proposed by Viola and Jones in their research paper "Rapid Object Detection using a Boosted Cascade of Simple Features" published in 2001 [19].

How is the face detection classifier trained with the algorithm?

The model is fed with many images with and without faces, each image is converted into grayscale. A collection of Haar features are collected from the dataset by the Viola Jones algorithm, first it searches for edges or the lines in the image, second it pick areas where there is a sudden change in the intensities of the pixels, finally the data of all of those areas put together, helps the algorithm determine where the face is.[20]

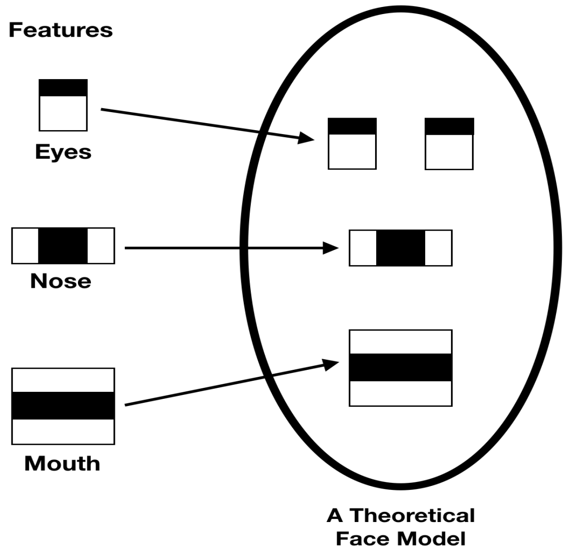

Figure 15: Representation of a theoretical face

Sooner Adaboost classifier is implied, the collected features mentioned above were applied to the images separately to create Weak Learners. In order to have an accurate one learner, the weak learners are ensembles using the Adaboost technique.

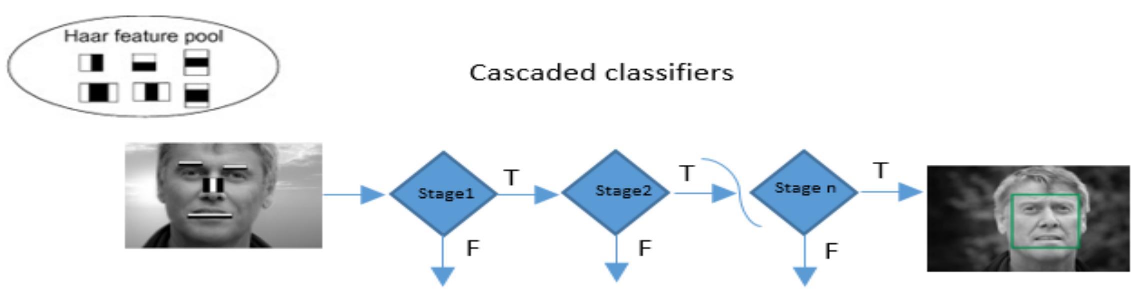

Figure 16: Representation of cascaded classifiers by Adaboost

4.3.3 Emotion detection

In this section, we used convolution neural networks. Neural networks are robust due to their properties in storing information on the entire network as well as their training huge amount of data.[21]

- Preprocessing data:

The data includes 48x48 pixel grayscale pictures. almost all faces are equally centered. The data is split into 2 columns: the data pixel's representation and the corresponding emotion(seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral)). The training dataset has 28,709 images and the test dataset has 3,589 images.

- The CNN model architecture:

The model is set to be a Sequential model which designates that all the layers within the network will be one after the other in sequential order.

The architecture:

Input: The model has the image represented in pixels as the input.

The input image passes through a series of four blocks with partial differences as listed below:

Blocks convolution layers: In each block 2 convolution layers are used following each other. The layer creates a convolutional network for the network. It applies filters, which adjust the image's appearance by stacking layers of convolutions on top of each other; to let the model be able to detect abstract concepts, for instance increasing the sharpness can detect the boundaries of a face's.

The number of the filters increases in each block, the blocks use 64 .64, 128, and 256 filters correspondingly; the reason behind the number changing is that as you go deeper in the CNN model as the learning rate increases.

Blocks BatchNormalization layers: Standardize the inputs to a layer for each mini-batch; allowing for considerably greater learning rates, permitting networks to train at a faster pace.

Blocks Maxpooling2D layers: The pooling layer is used to reduce the dimensions of the feature maps; in order to have less computation expensive training.

Blocks dropout layers: A way where randomly selected neurons are ignored during training; avoiding overfitting issues.

Flatten layer:This layer is used for converting the pooled feature map to a single column to create the fully connected layer.

Dense layer: It applies a technique to set all the neurons in a layer to be connected to those in the next layer.

The output: The percentage of all possible emotions of this specific input image.

4.3.4 Emotion-song mapping

After getting the emotion detected, we map it to a specific song (song recommendation). This approach is done in python and at the running time. Each emotion has a particular song's category that is mapped to, the program will get the song from this category file songs by KNN (K nearest algorithm).

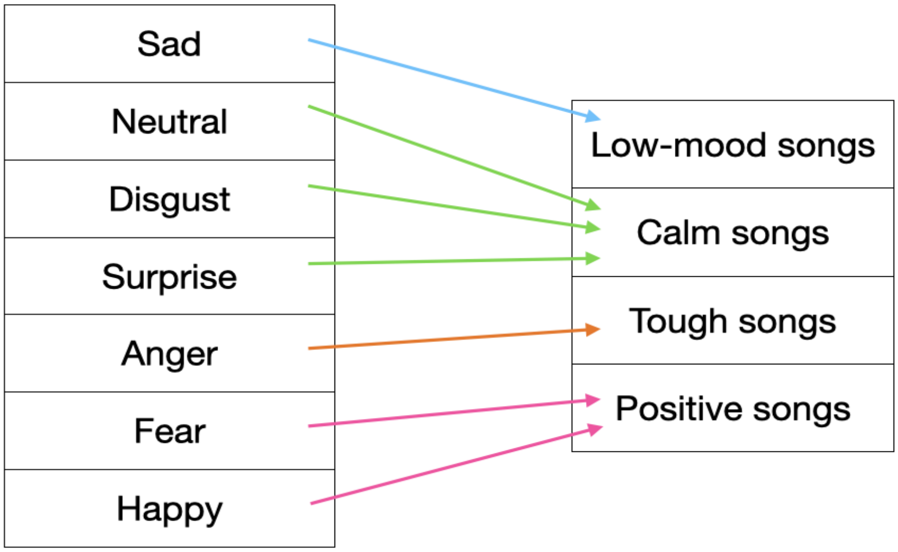

Table 1: Representation of the mapping emotions with songs

4.3.5 Backend building

The backend for the web application is built within the python programming language, and we chose Django as our framework; due to multiple reasons:

- Fast processing: Because of its clean, readable, and maintainable code organization.
- Security: which helped us to keep privacy for our users.
- The offered SQLite database: Django automatically creates a SQLite database; which helped us not to have another database program.

The backend includes the systematic approach of the project. for illustrating, signing in, making a profile for each person, adding friends, adding playlists, taking a picture, recommending a song, playing the song, sending the song, getting the analysis of the user's recent activity.

4.3.6 Frontend building

The main software requirements are HTML5, CSS, and Javascript. With the collection of these programs, we could build a user-friendly frontend.

5 SOFTWARE TESTING

5.1 Test methods

- ❖ Methods we have followed for the testing model functioning:

  Test dataset evaluation: We used a test data set, as we measured the performance of the model by passing a test dataset and checked the accuracy.
- ❖ Methods we have followed for the web app overall functioning :
  a. Unit testing: we used this technique where a specific task or part of source code is tested individually to determine whether they are fit for use; this technique was used in almost every part of our project, including backend building (from signing up to profile editing and song playing tasks).
  b. System testing: we used this method when our app was ready; where it validates the complete and fully integrated software.
  c. A simple way we tested in, which is real by testing videos, pictures, or recordings of us interpreting a specific emotion.

5.2 Experimental results

5.2.1 Results of CNN model

We applied some testing as mentioned in subchapter 5.1 on the emotion categorizer. Firstly, we are specifying the hyper parameter of our CNN, the number of epochs; where each epoch means training the neural network with all the training data for one cycle.

We got different accuracy percentages while trying to find the optimal number of epochs, as you can see in Table 2 and Figure 17.

Number of epochs	*Number of Training Data*	*Number of Testing Data*	*Accuracy before data augmentation*	*Accuracy after data augmentation*
50	28,709	7178	50%	55.3%
150	28,709	7178	52%	60.7%
300	28,709	7178	60.5%	69.8%

Table 2: Model accuracy of the facial expression recognition with the different number of epochs

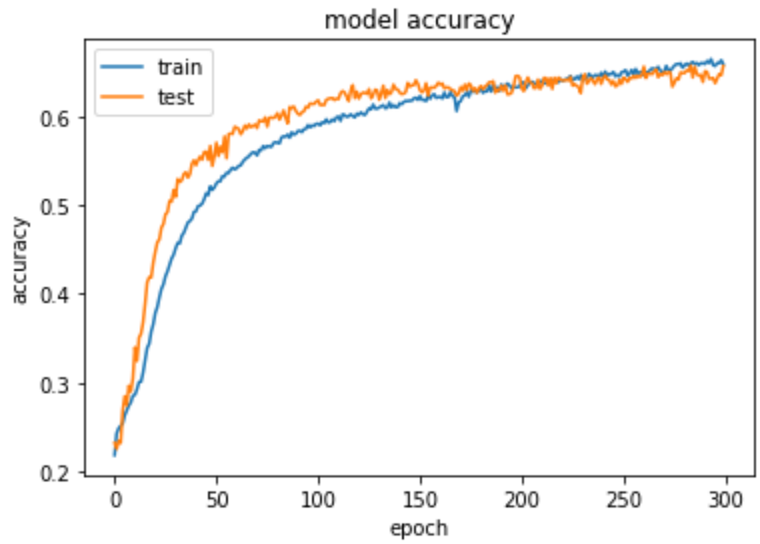

Figure 17: Model accuracy of the facial expression recognition with the different number of epochs

Therefore, we chose the 300 number of epochs as it achieved better accuracy. Later we were not satisfied with our model results; as a result, we applied data augmentation. To illustrate, without actually gathering new data, the data augmentation allows practitioners to greatly enhance the diversity of data available for training models.

6. CONCLUSION AND FUTURE WORK

In this project, we created a new way of playing music upon emotions where people are able to discover new music according to their moods by taking selfies, and are able to share the songs they are listening to.

For future works, we are thinking of improving our accuracy for facial expression recognition. In addition, there will be options for users to choose if they want higher-mood or lower-mood songs after emotion recognition of users are captured.

REFERENCES

[1] Schäfer, T., Sedlmeier, P., Städtler, C., & Huron, D. (2013). The psychological functions of music listening. Frontiers in Psychology, 4 (511), 1–7. https://doi.org/10.3389/fpsyg.2013.00511

[2] II, G. C. B. (1990). Music, Mood, and Marketing. *Journal of Marketing*, *54*(4), 94. https://doi.org/10.2307/1251762

[3] Atkinson, M., Barrios-Choplin, B., McCraty, R., & Tomasino, D. (1998). The effects of different types of music on mood, tension, and mental clarity. *Alternative Therapies in Health and Medicine*, *4*(1), 75–84. https://pubmed.ncbi.nlm.nih.gov/9439023/

[4] Moris, D. N., & Linos, D. (2012). Music meets surgery: two sides to the art of “healing.” *Surgical Endoscopy*,*27*(3), 719–723. https://doi.org/10.1007/s00464-012-2525-8

[5] Wegrzyn M, Vogt M, Kiratlioglu B, Schneider J, Kissler J (2017) Mapping the emotional face. How individual face parts contribute to successful emotion recognition. PLoS ONE 12(5): e0177239. https://doi.org/10.1371/journal.pone.0177239

[6] Web API Reference | Spotify for Developers. (n.d.). SpotifyForDevelopers. Retrieved February 19, 2021, from https://developer.spotify.com/documentation/web-api/reference/#endpoint-get-audio-features

[7] GeeksforGeeks. (2019, July 29). K-Nearest Neighbours. https://www.geeksforgeeks.org/k-nearest-neighbours/

[8] OpenCV: Cascade Classifier. (n.d.). OpenCV. Retrieved February 25, 2021, from https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[9] Cast, J., Schulze, C., & Fauci, A. (autumn 2013-2014). Music Genre Classification. Music Genre Classification. doi:http://cs229.stanford.edu/proj2013/FauciCastSchulze-MusicGenreClassification.pdf

[10] Gabriel, C., & Crickmore, L. (1977). Emotion and Music. Psychology of Music, 5(1), 28–31.

[11] J. Kemper, K., & C. Danhauer, S. (december 13, 2004). Music as therapy. *Music as Therapy,* 282-288.

[12] Goupil, L., & Aucouturier, J. (feb 2019). Musical pleasure and musical emotions [Abstract]. Proceedings of the National Academy of Sciences, 3364-3366. doi:10.1073/pnas.1900369116

[13] Dhavalikar, A. S., & Kulkarni, R. K. (2014, February). Face detection and facial expression recognition system. *2014 International Conference on Electronics and Communication Systems (ICECS)*. https://doi.org/10.1109/ecs.2014.6892834

[14] Boyd, C. (2019, November 12). *How Spotify Recommends Your New Favorite Artist - Towards Data Science*.Medium. https://towardsdatascience.com/how-spotify-recommends-your-new-favorite-artist-8c1850512af0

[15] Dureha, A. (2014). An Accurate Algorithm for Generating a Music Playlist based on Facial Expressions. *International Journal of Computer Applications*, *100*(9), 33–39. https://doi.org/10.5120/17557-8163

[16] Bansal, S., & Nagar, P. (2015). Emotion Recognition from Facial Expression Based on Bezier Curve. *International Journal of Advanced Information Technology*,*5*(3/4/5/6), 1–7. https://doi.org/10.5121/ijait.2015.5601

[17] Rabashette, D., Tale, R., Hinge, A., Padale, K., Chavan, R., & Deshmukh, N. (2016). Emotion Based Music System. *International Journal of Emerging Trends in Science and Technology*, *3*(5), 3923–3927. https://doi.org/10.18535/ijetst/v3i05.12

[18] *Organize Your Music*. (n.d.). Organize Your Music. Retrieved July 2, 2021, from http://organizeyourmusic.playlistmachinery.com

[19] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Published. https://doi.org/10.1109/cvpr.2001.990517

[20] Lienhart, R., & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. *International Conference on Image Processing*, I–900. https://doi.org/10.1109/ICIP.2002.1038171

[21] Imran, M. (2021, April 5). *Advantages of Neural Networks – Benefits of AI and Deep Learning*. Folio3AI Blog. https://www.folio3.ai/blog/advantages-of-neural-networks/