Mood based music recommendation system

Conference Paper · June 2021

CITATIONS
2

READS
2,249

5 authors, including:

Ankita Mahadik
Don Bosco Institute of Technology
1 PUBLICATION **2** CITATIONS
SEE PROFILE

Vijaya Bharathi Jagan
Pillai College of Engineering
4 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Mood based music recommendation system

Ankita Mahadik
Department of Information Technology
Don Bosco Institute of Technology
Mumbai, India
ankita010899@gmail.com

Shambhavi Milgir
Department of Information Technology
Don Bosco Institute of Technology
Mumbai, India
smilgir0210@gmail.com

Janvi Patel
Department of Information Technology
Don Bosco Institute of Technology
Mumbai, India
pateljanvi411@gmail.com

Prof. Vijaya Bharathi Jagan
Department of Information Technology
Don Bosco Institute of Technology
Mumbai, India
vijayabharathi.prakasam@gmail.com

Prof. Vaishali Kavathekar
Department of Information Technology
Don Bosco Institute of Technology
Mumbai, India
vaishalik.dbit@dbclmumbai.org

Abstract— **A user's emotion or mood can be detected by his/her facial expressions. These expressions can be derived from the live feed via the system's camera. A lot of research is being conducted in the field of Computer Vision and Machine Learning (ML), where machines are trained to identify various human emotions or moods. Machine Learning provides various techniques through which human emotions can be detected. One such technique is to use MobileNet model with Keras, which generates a small size trained model and makes Android-ML integration easier.**

Music is a great connector. It unites us across markets, ages, backgrounds, languages, preferences, political leanings and income levels. Music players and other streaming apps have a high demand as these apps can be used anytime, anywhere and can be combined with daily activities, travelling, sports, etc. With the rapid development of mobile networks and digital multimedia technologies, digital music has become the mainstream consumer content sought by many young people.

People often use music as a means of mood regulation, specifically to change a bad mood, increase energy level or reduce tension. Also, listening to the right kind of music at the right time may improve mental health. Thus, human emotions have a strong relationship with music.

In our proposed system, a mood-based music player is created which performs real time mood detection and suggests songs as per detected mood. This becomes an additional feature to the traditional music player apps that come pre-installed in our mobile phones. An important benefit of incorporating mood detection is customer satisfaction. The objective of this system is to analyse the user's image, predict the expression of the user and suggest songs suitable to the detected mood.

Keywords—Face Recognition, Image Processing, Computer Vision, Emotion Detection, Music, Mood detection

I. INTRODUCTION

Human emotions can be broadly classified as: fear, disgust, anger, surprise, sad, happy and neutral. A large number of other emotions such as cheerful (which is a variation of happy) and contempt (which is a variation of disgust) can be categorized under this umbrella of emotions. These emotions are very subtle. Facial muscle contortions are very minimal, and detecting these differences can be very challenging as even a small difference results in different expressions. Also, expressions of different or even the same people might vary for the same emotion, as emotions are hugely context dependent. While the focus can on only those areas of the face which display a maximum of emotions like around the mouth and eyes, how these gestures are extracted and categorized is still an important question. Neural networks and machine learning have been used for these tasks and have obtained good results. Machine learning algorithms have proven to be very useful in pattern recognition and classification, and hence can be used for mood detection as well.

With the development of digital music technology, the development of a personalized music recommendation system which recommends music for users is essential. It is a big challenge to provide recommendations from the large data available on the internet. E-commerce giants like Amazon, EBay provide personalized recommendations to users based on their taste and history while companies like Spotify, Pandora use Machine Learning and Deep Learning techniques for providing appropriate recommendations. There has been some work done on personalized music recommendation to recommend songs based on the user's preference. There exist two major approaches for the personalized music recommendation. One is the content-based filtering approach which analyses the content of music that users liked in the past and recommends the music with relevant content. The main drawback of this approach is that the model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests. The other approach is the collaborative filtering approach which recommends music that a peer group of similar preference liked. Both recommendation approaches are based on the user's preferences observed from the listening behaviour. The major drawback of this approach is the popularity bias problem: popular (i.e., frequently rated) items get a lot of exposure while less popular ones are under-represented in the recommendations. Generally, a hybrid approach is implemented in which both content and collaborative techniques are combined to extract maximum accuracy and to overcome drawbacks of both types. [1]

In this work, the aim is to create a music recommendation system/music player which will detect the user's face, identify the current mood and then recommend a playlist based on the detected mood.

II. Related Work

A. Literature Survey

In a particular system [8], Anaconda and Python 3.5 softwares were used to test the functionality and Viola-Jones and haar cascade algorithms were used for face detection. Similarly, KDEF (Karolinska Directed Emotional Faces) dataset and VGG (Visual Geometry Group) 16 were used with CNN (Convolution Neural Network) model which was designed with an accuracy of 88%, for face recognition and classification that validated the performance measures. However, the results proved that the network architecture designed had better advancements than existing algorithms. Another system [9] used Python 2.7, Open-Source Computer Vision Library (OpenCV) & CK (Cohn-Kanade) and CK+ (Extended Cohn-Kanade) database which gave approximately 83% accuracy. Certain researchers have described the Extended Cohn-Kanade (CK+) database for those wanting to prototype and benchmark systems for automatic facial expression detection. The popularity and ease of access for the original Cohn-Kanade dataset this is seen as a very valuable addition to the already existing corpora. It was also stated that for a fully automatic system to be robust for all expressions in a myriad of realistic scenarios, more data is required. For this to occur very large reliably coded datasets across a wide array of visual variabilities are required (at least 5 to 10k examples for each action) which would require a collaborative research effort from various institutions.

It was observed in a cross-database experiment [1] that raw features worked best with Logistic Regression for testing RaFD (Radboud Faces Database) database and Mobile images dataset. The accuracy achieved was 66% and 36% respectively for both using CK+ dataset as a training set. The additional features (distance and area) reduced the accuracy of the experiment for SVM (Support Vector Machine) from 89%. The algorithm that had been implemented generalized the results from the training set to the testing set better than SVM and several other algorithms. An average accuracy of 86% was seen for RaFD database and 87% for CK+ database for cross-validation=5.The main focus was feature extraction and analysis of the machine algorithm on the dataset. But accurate face-detection algorithms become very important if there are multiple people in the image. One of the works [10] was tested by deriving expression from the live feed via the system's camera or any pre-existing image available in the memory. It has been implemented using Python 2.7, OpenCV and NumPy. The objective was to develop a system that can analyse the image and predict the expression of the person. The study proved that this procedure is workable and produces valid results.

There has also been research done on the Music Recommendation System. According to one such research [11], a preliminary approach to Hindi music mood classification has been described, that exploits simple features extracted from the audio. MIREX (Music Information Retrieval Evaluation eXchange) mood taxonomy gave an average accuracy of 51.56% using the 10-fold cross validation. In addition to this, there is an article [10] that states that the current music recommendation research results from the perspective of music resources description. It is suggested that there is a lack of systematic research on user behaviour and needs, low level of feature extraction, and a single evaluation index in current research. Situation was identified to be an important factor in the music personalized recommendation system. Finally, it was concluded that when the weights given to all the contextual factors were the same, greatly reduced the accuracy of the recommendation results.

Another research [12] states that their hybrid recommendation system approach concept will work once their model is trained enough to recognize the labels. The mechanism for the automatic management of the user preferences in the personalized music recommendation service automatically extracts the user preference data from the user's brain waves and audio features from music. In their study, a very short feature vector, obtained from low dimensional projection and already developed audio features, is used for music genre classification problems. A distance metric learning algorithm was applied in order to reduce the dimensionality of the feature vector with a little performance degradation. Proposed user's preference classifier achieved an overall accuracy of 81.07% in the binary preference classification for the KETI AFA2000 music corpus. The user satisfaction was recognizable when brainwaves were used.

B. Existing Systems

- **EMO Player**: Emo player (an emotion-based music player) is a novel approach that helps the user to automatically play songs based on the emotions of the user. [2]
- **SoundTree:** Sound Tree is a music recommendation system which can be integrated to an external web application and deployed as a web service. It uses people-to-people correlation based on the user's past behaviour such as previously listened, downloaded songs. [3]
- **lucyd**: lucyd is a music recommendation tool developed by four graduate students in UC Berkeley's Master of Information and Data Science (MIDS) program. lucyd lets the user ask for music recommendations using whichever terms they want. [4]
- **Reel Time.AI**: This system works by having the user subscribe to them. The user can then upload images of large gatherings such as shopping malls, movie theatres and restaurants. The system then identifies the moods happy and sad. It recognizes which faces portray happy emotion and which faces portray sad emotion, and gives the verdict of the situation from the faces of the people present.
- **Music.AI**: It uses the list of moods as input for mood of the user and suggests songs based on the selected mood. It is a combination of Collaborative filtering based and Content based filtering models. Emotion, time, ambience and learning history are the features taken into account for music recommendation. [5]

C. Existing Algorithms/Tools

- **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 backend 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. [14]
- **Hybrid approach of Music Recommendation:** There are several drawbacks to relying solely on collaborative filtering to recommend music. The biggest problem is the "Cold Start." Music tracks are only tagged as often as listeners are discovering or listening to them. In other words, there are little or no available 'tags' to describe new music or music that has not been discovered yet. Additionally, listeners are more willing to supply tags for songs they enjoy most than for songs they mildly enjoy or do not enjoy at all. [13]
- **Viola–Jones object detection framework:** The Viola-Jones algorithm is a widely used mechanism for object detection. The main property of this algorithm is that training is slow, but detection is fast. This algorithm uses Haar basis feature filters, so it does not use multiplications. The efficiency of the Viola-Jones algorithm can be significantly increased by first generating the integral image. [15]

III. Methodology

The mood-based music recommendation system is an application that focuses on implementing real time mood detection. It is a prototype of a new product that comprises two main modules: Facial expression recognition/mood detection and Music recommendation.

A. Mood Detection Module

This Module is divided into two parts:

- 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, initially the python library OpenCV was considered. But integrating it with an android app was a complex task so the FaceDetector class available in Java was considered. This library identifies the faces of people in a Bitmap graphic object and returns the number of faces present in a given image.
- Mood 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 was used but, in the survey, it was found that this approach takes a lot of time to train and validate and also works slowly when integrated with android apps. So, MobileNet which is a CNN architecture model for Image Classification and Mobile Vision was used. There are other models as well but what makes MobileNet special is that it has very less computation power to run or apply transfer learning to. This makes it a perfect fit for Mobile devices, embedded systems and computers without GPU or 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 [6] and MMA Facial Expression Recognition dataset [7] from Kaggle. The FER 2013 dataset contained grayscale images of size 48x48 pixels. The MMA Facial Expression Recognition dataset had images of different specifications. Thus, all these images were converted as per the images in FER 2013 dataset and combined to obtain an even larger dataset with 40,045 training images and 11,924 testing images. MobileNet was used with Keras to train and test our model for seven classes - happy, angry, neutral, sad, surprise, fear and disgust. We trained it for 25 epochs and achieved an accuracy of approximately 75%.

B. Music Recommendation Module

The dataset of songs classified as per mood was found on Kaggle for two different languages - Hindi and English. Research for a good cloud storage platform to store, retrieve and query this song data as per user's request was conducted. Options like AWS, Google Cloud, etc. were found but these were rejected as they were costly and provided very limited storage for free. Then research for open-source streaming services like Restream.io, Ampache, etc. was conducted, but again, these services were web based/used for live streaming on YouTube/available only for personal use. After a lot of research (and time constraints), Firebase was considered a backend server. It can be integrated with an android app just by one click and its free plan provides storage of 5GB. But functions like user queries, server updates, etc. are a part of a paid plan so it was decided to limit the scope of the project.

The mp3 versions of the songs were manually uploaded on Firebase storage and were linked in the Real Time database as per mood and language (for filters).

C. Integration

For the integration of these two modules in an Android application, the trained MobileNet model was saved as an .h5 file, and this .h5 file was then converted to a .tflite file using TensorFlow Lite Converter. It takes a TensorFlow

model as input and generates a TensorFlow Lite model as output with .tflite extension. Since the MobileNet model is used, the size of the tflite file is expected be around 20- 25 Megabyte (MB) which was the desired size. In Android Studio, an assets folder was created to store the .tflite file and labels.txt file. The labels.txt file contains the class labels of the model. All the appropriate methods were created for loading the model, running the interpreter and obtaining the results.

A project on Firebase was created and mp3 songs were uploaded in the storage section. These songs as per mood and language in the real time database section. After this, the Firebase database was linked to Android studio. An appropriate UI for the android application was created and the tflite model methods were linked with the songs on Firebase. Finally, the application was tested to fix the bugs if any.

Fig.1. displays the System Architecture Diagram and Fig.2. displays the Data Flow Diagram of the system.

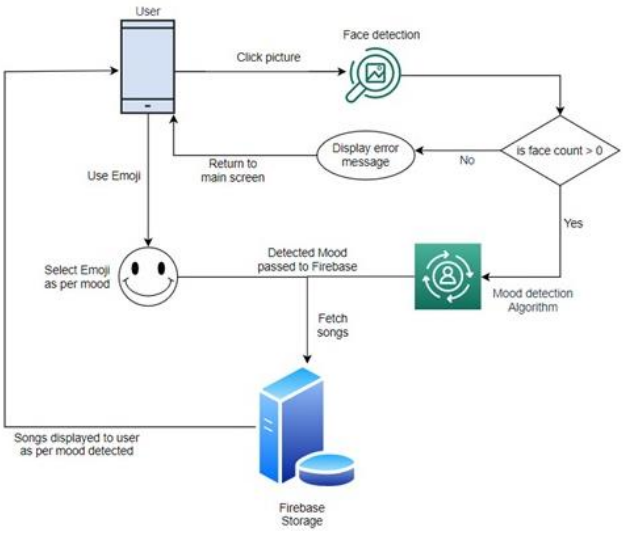

Fig.1. System Architecture Diagram.

The system architecture diagram depicts the overall outline of the software system and the relationships, constraints, and boundaries between components. When the user opens the android app, the main screen will be displayed which contains three buttons – take snap, use emoji, play songs. If the user clicks on “take snap” button, the camera opens, user clicks picture. This picture is given as input to face detection program. If no face is detected or multiple faces are detected, then an appropriate error message is displayed to the user. Upon successful single face detection, the picture is given as input to the mood detection module. The detected mood is displayed to the user, after which the “play songs” button gets enabled. The suitable playlist for the detected mood is displayed on the playlist screen as shown in Fig.9 where the user can select and play the song. If the user presses the “use emoji” button, then a screen of five emojis will be displayed as shown in Fig.10. User can click on any emoji to obtain the respective playlist. To exit the app, the user has to just press the back button.

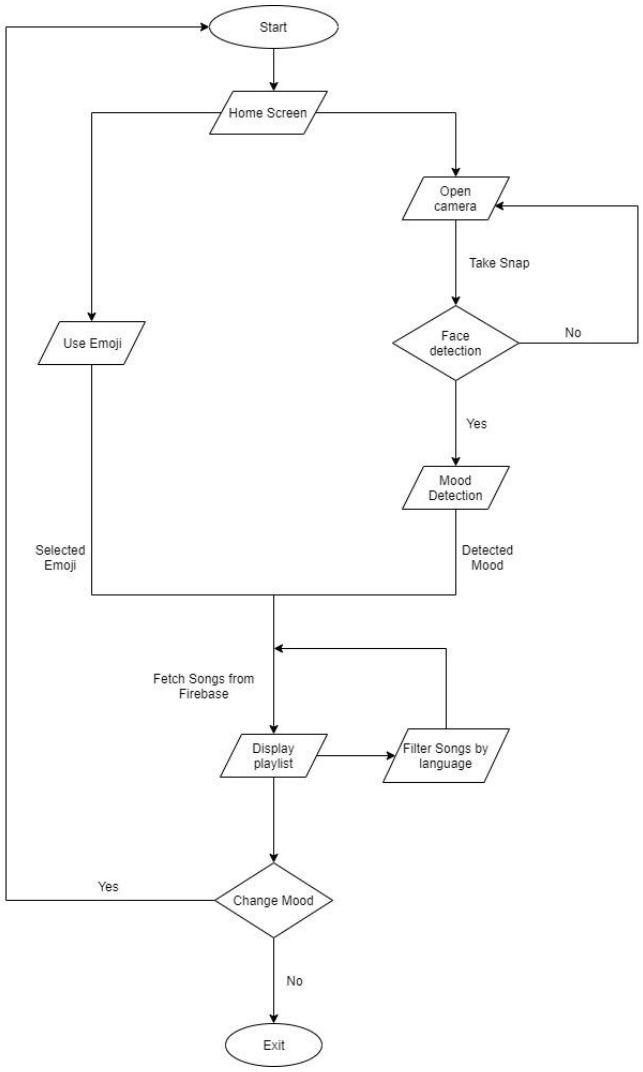

Fig.2. Data Flow Diagram of the system.

IV. Hardware and Software Requirements

A. *Hardware Requirements*

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The hardware requirements required for this project are:

- Minimum 4 Gigabyte (GB) RAM (used for processing)
- Webcam (for testing on laptop/desktop)
- Minimum 16 Megapixel (MP) Resolution camera (for testing on android device)
- 30 MB Memory space (approximate value)

B. *Software Requirements*

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is

installed. The software requirements that are required for this project are:

- Python 3.6
- OpenCV 3.1
- PyCharm IDE
- Android Studio

V. Results and Discussion

As every person has unique facial features, it is difficult to detect accurate human emotion or mood. But with proper facial expressions, it can be detected up to a certain extent. The camera of the device should have a higher resolution. The android application that we have developed runs successfully and following are some of the screenshots captured while using it. Fig.3. displays "sad" mood being detected, Fig.4. displays "angry" mood being detected, Fig.5. displays "happy" mood being detected, Fig.6. displays "neutral" mood being detected and Fig.7. displays "surprise" mood being detected. and Fig.8. displays "fear" mood being detected.

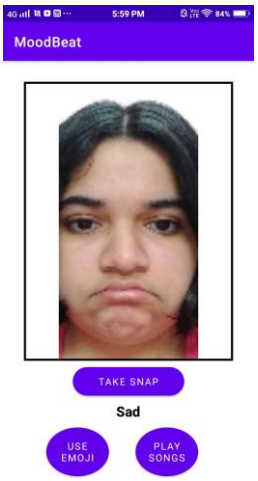

Fig.3. "Sad" mood detected successfully by the application.

Fig.4. "Angry" mood detected successfully by the application.

Fig.5. "Happy" mood detected successfully by the application.

Fig.6 "Neutral" mood detected successfully by the application.

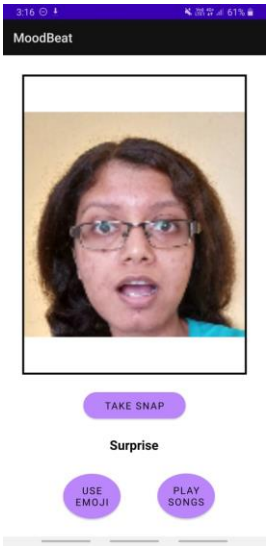

Fig.7. "Surprise" mood detected successfully by the application.

Fig.8. “Fear” mood detected successfully by the application.

The system detects the mood in real time and a playlist is displayed for that mood accurately. The playlist for “happy” mood can be seen in Fig.9. It is able to fetch and play the songs from the recommended playlist in the presence of a stable internet connection. The playlist displays the songs to the user in list view, and when the user selects a song to be played, the song is played in a media player, that comes with all the necessary buttons such as play, pause, shuffle, next and previous, along with a seek bar. For “angry”, “Fear”, “disgust” and “surprise” moods, devotional, motivational and patriotic songs are suggested to the user. Hence, the user is also provided with mood improvement.

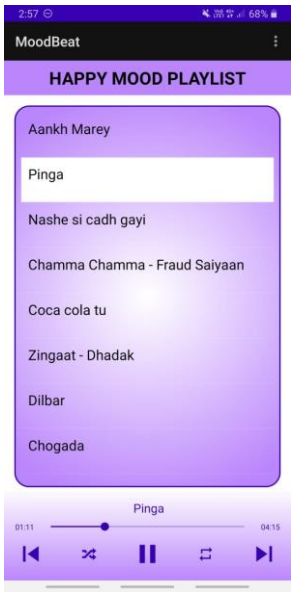

Fig.9. Playlist for “Happy” mood where a song is being played.

We have provided our users with the additional option of using emojis to generate the playlist. Whenever the user does not wish to, or is unable to take a snapshot of their mood due to various reasons such as extremely high or low lighting, their camera not working properly, they have a lower resolution camera which is unable to take a clear picture of their face, which in turn is unable to detect the proper mood, or any other reason, the user can click on the “Use Emoji” button and select the emoji which represents the mood which they are in, or the mood that they want their playlist to be generated of. Fig.10. is the screenshot of the screen that is displayed to the user when they click the “Use Emoji” button. The first is the emoji for the mood “happy”, the second for the mood “angry”, the third for the mood “surprise”, the fourth for the mood “sad”, and the fifth for the mood “neutral”.

Fig.10. The screen for the user to generate playlist via emojis.

In Fig.11(a), the graph displays the accuracy of our model, where the x-axis specifies the number of epochs and the y-axis specifies the accuracy. As it can be seen in the figure, our model has achieved approximately 75% accuracy. Since it is a fully computer-based system, it understands emotions in the way it has been trained.

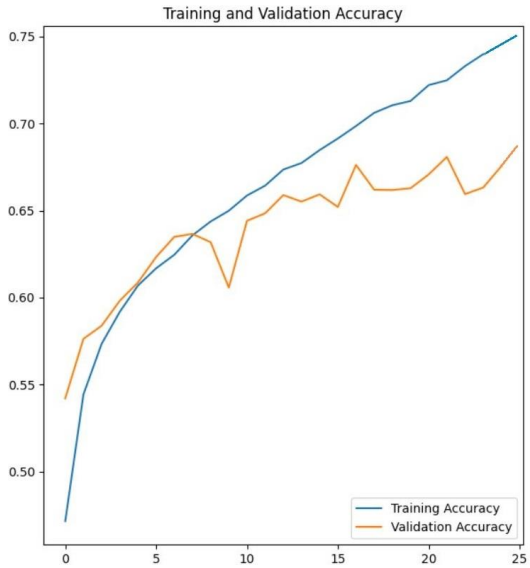

Fig.11(a). Training and Validation Accuracy.

In Fig.11(b), the graph displays the training and validation loss of our model, where the x-axis specifies the number of epochs and the y-axis specifies the loss.

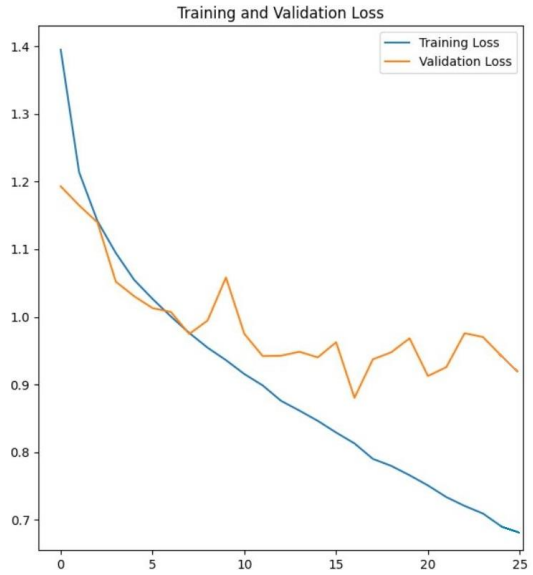

Fig.11(b). Training and Validation Loss.

The following table provides a comparison with respect to accuracy between the different existing systems and tools and our proposed system.

TABLE I. ACCURACY COMPARISON

Sr. No.	Existing Systems and Tools	Accuracy (%)
1.	Deep learning-based face recognition using keras	88
2.	Hybrid approach for music recommendation	80.7
3.	Viola Jones object detection framework	86
4.	Music.AI	78.84
5.	Mood based music recommendation system	75

VI. CONCLUSION

Even though human emotions are complex and subtle, it is possible for a machine learning model to be trained to accurately detect a set of emotions which can be differentiated from each other with certain facial expressions. The expression on a person's face can be used to detect their mood, and once a certain mood has been detected, music suitable for the person's detected mood can be suggested. Our model, having the accuracy of approximately 75%, is able to detect seven moods accurately: anger, disgust, fear, happy, sad, surprise and neutral; and our android application is able to play the music that would be suitable for the detected mood.

For accurate detection of fear and disgust moods, additional parameters such as heart rate or body temperature must also be considered rather than solely depending on facial expressions. In addition to that, finding suitable music to be played on detection of fear or disgust mood is also a challenge. As a result, it can be considered as a future scope for our project. Our trained model is an overfit model, which can sometimes lead to fluctuations in accurate detection. For example, the "disgust" mood is mostly classified as "angry" mood since the facial features (eyebrows, cheeks) are similar for both.

Thus, for more accurate results it needs to be trained for more images, and for a greater number of epochs. Recommendation of movies and TV series on the basis of mood detection can also be considered as a future scope for our project.

ACKNOWLEDGMENT

We would like to thank our project guides Prof. Vijaya J. and Prof. Vaishali K. for the valuable support and guidance they gave us on every step of the project execution. We would also like to thank the project review committee members Prof. Amiya T., Prof. Phiroj S. and Prof. Sunantha K. We would also like to express our gratitude to Prof. Janhavi B., Head of Department, Department of Information Technology, Don Bosco Institute of Technology, Mumbai who helped us accomplish this work.

REFERENCES

[1] Raut, Nitisha, "Facial Emotion Recognition Using Machine Learning" (2018). Master's Projects. 632. https://doi.org/10.31979/etd.w5fs-s8wd

[2] Hemanth P,Adarsh ,Aswani C.B, Ajith P, Veena A Kumar, "EMO PLAYER: Emotion Based Music Player", International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 4, April 2018, pp. 4822-87.

[3] Music Recommendation System: *"Sound Tree"*, Dcengo Unchained: Sıla KAYA, BSc.; Duygu KABAKCI, BSc.; Işınsu KATIRCIOĞLU, BSc. and Koray KOCAKAYA BSc. Assistant : Dilek Önal Supervisors: Prof. Dr. İsmail Hakkı Toroslu, Prof. Dr. Veysi İşler Sponsor Company: ARGEDOR

[4] Tim Spittle, lucyd, GitHub, , April 16, 2020. Accessed on: [Online], Available at: https://github.com/timspit/lucyd

[5] A. Abdul, J. Chen, H.-Y. Liao, and S.-H. Chang, "An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach," *Applied Sciences*, vol. 8, no. 7, p. 1103, Jul. 2018.

[6] Manas Sambare, FER2013 Dataset, Kaggle, July 19, 2020. Accessed on: September 9, 2020. [Online], Avalilable at: https://www.kaggle.com/msambare/fer2013

[7] MahmoudiMA, MMA Facial Expression Dataset, Kaggle, June 6, 2020. Accessed on: September 15, 2020. [Online], Available at: https://www.kaggle.com/mahmoudima/mma-facial-expression

[8] Dr. Shaik Asif Hussain and Ahlam Salim Abdallah Al Balushi, "A real time face emotion classification and recognition using deep learning model", 2020 Journal. of Phys.: Conf. Ser. 1432 012087

[9] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, San Francisco, CA, USA, 2010, pp. 94-101, doi: 10.1109/CVPRW.2010.5543262.

[10] Puri, Raghav & Gupta, Archit & Sikri, Manas & Tiwari, Mohit & Pathak, Nitish & Goel, Shivendra. (2020). Emotion Detection using Image Processing in Python.

[11] Patra, Braja & Das, Dipankar & Bandyopadhyay, Sivaji. (2013). Automatic Music Mood Classification of Hindi Songs.

[12] Lee, J., Yoon, K., Jang, D., Jang, S., Shin, S., & Kim, J. (2018). MUSIC RECOMMENDATION SYSTEM BASED ON GENRE DISTANCE AND USER PREFERENCE CLASSIFICATION.

[13] Kaufman Jaime C., University of North Florida, “A Hybrid Approach to Music Recommendation: Exploiting Collaborative Music Tags and Acoustic Features”, UNF Digital Commons, 2014.

[14] D Priya, *Face Detection, Recognition and Emotion Detection in 8 lines of code!*, towards data science, April 3, 2019. Accessed on: July 12, 2020 [Online], Available at: https://towardsdatascience.com/face-detection-recognition-and-emotion-detection-in-8-lines-of-code-b2ce32d4d5de

[15] bluepi, *“\Classifying Different Types of Recommender Systems*, November 14, 2015. Accessed on: July 7, 2020. [Online], Available on:https://www.bluepiit.com/blog/classifying-recommender-systems/#:~:text=There%20are%20majorly%20six%20types,system%20and%20Hybrid%20recommender%20system