ISBN 978-1-6654-3986-2

Mind Frame

Music and Movie Recommendations to uplift the current mood using Deep Learning

Manu Gupta
Department of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India
manugupta@sreenidhi.edu.in

Sai Vivek Amirishetty
Department of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India
saivivekamirishetty99@gmail.com

Bommerla Nithin
Department of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India
nithinbommerla99@gmail.com

Harish Kurakula
Department of Electronics and Computer Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, India
harishkurakula18@gmail.com

2021 International Conference on System, Computation, Automation and Networking (ICSCAN) | 978-1-6654-3986-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICSCAN53069.2021.9526435

***Abstract*—Mind Frame is a web-application designed in proposed work that gives recommendations of music and movies to users based on their current mood. The proposed system identifies moods of the user such as happy, sad, peaceful and gives music and movie recommendations to the user based on mood identified. The OpenCV library of Python is used to extract the features from the facial expressions. Neural network is trained with labelled images of different emotional moods of the user, this trained model is then used to predict the mood of the user. Finally, the user is recommended the playlists of music or movies from the library created containing over hundred thousand music and movies. For storing and viewing the data and user details and also for processing requests from each user, Django is used as the web framework along with Python for front end and back-end.**

***Keywords*—Computer Vision; Image Processing; Deep Learning; Computer Networks; Web Development.**

I. INTRODUCTION

People keep an eye out to communicate their emotions, generally by their facial expressions. This is a very simple and straight-forward method to express themselves. It is found that not only by their facial expressions, one's likings can help to find their emotions and Music has reliably been known to change the outlook of an individual. This project work intends to identify the emotion of a user conveyed through his/her facial expressions. The web cam interface of the proposed model captures the image of the user and subsequently through image segmentation and image processing techniques extracts features from the face of a target individual. Next, the emotions of the individual are identified from extracted features. A recommender system seeks to estimate and predict user content preference regarding games, stories or videos. These systems extract data from the history of users and give recommendations based on their interests.

After considering the advancements in the above fields of technologies, we found that by using computer vision and image processing the emotion of the user can be captured accurately using artificial intelligence. We believe that this system will solve the problem of changing the current mood of the user, as the mood changes every time, whenever a user is having a long day and wants to chill out, he/she needs recommendations to boost his/her mood, not matter what kind of activity the User did in the previous day. It should not be based on the previous activity but based on the current situation of the user.

In this paper we have proposed mind frame recommendation system, which aims at uplifting the mood of the user, by playing tunes/songs/music that match the necessities of the user by getting the image of the user. The best possible way in which people tend to analyze or conclude the emotion or the feeling or the thoughts that another user is trying to express is by facial expression. In proposed work face recognition is performed using the Haar Cascade face classifier and emotions are detected with deep learning methods. Based on the identified emotion, movies and music are recommended by filtering the genres. Our proposed model will help in mood alteration and it is found that mood upliftment supports in overcoming situations like depression and sadness.

II. LITERATURE SURVEY

Movies and Music are the sole source of entertainment, people search for songs and movies which they want to listen to or watch. Many recommendation systems have been proposed in literature for music recommendation. Some of them are discussed as follows:

A recommendation system based on the movie lens data set is presented by Snehal et al. [1] They used Deep Neural Networks (DNNs) to discover user-item relationships. The computations are done in a hidden layer and the user is recommended the most predicted item. The recommendation system presented by Moscato et al. [2] and Champika et al. [3] used behavior of users in various social media networks such as Facebook for giving music suggestions. Social media content such as posts, comments, stories, interactions and profiles are utilized to identify artistic likings. Iyer et al. [4] presented an Android application called 'EmoPlayer', which helps users save time by proposing a playlist of music depending on their current mood. This model utilizes the Viola Jones method to recognize faces and the Fisherfaces

ISBN 978-1-6654-3986-2

classifier to classify emotions. It only works if you have access to the internet. The Movie Recommendation system developed by Shreya et al. [5] utilizes a hybrid technique that combines content-based and collaborative filtering. Classification is performed using the Support Vector Machine classifier and a genetic algorithm is utilized to increase accuracy, quality, and scalability. This model used the Movie Lens Data set and filtered it based on user ratings of 1 to 5, with each user rating at least 20 films. The model uses the user's input to determine the type and quantity of movies to recommend. Finally, K-means clustering techniques are applied to classify the related information. The work proposed by Shin et al. [6] employs EEG (Electro Encephalo Graphy) impulses to assess the user's mood in real time and suggest music based on the user's emotional state.

Drawbacks of Existing System

Most of the existing Recommendation Systems recommend music or movies based on most watched content and hence cannot help in uplifting the current mood of the user. Social Networks and Social Media Content doesn't tell the exact mood of a user as the people posting the content can pretend and show fake mood/behavior. Even though having some similarities in viewership it doesn't mean the user will be entertained with the collaborative filtering recommendations. Determining mood using Brain-wave Signals requires complex equipment and can't be afforded by many users. Using multiple models increases the processing time and affects the time to generate results. Some systems use ratings which can be deceiving since interests are subjective, content disliked by some users can be liked by some other users. To overcome these issues, the proposed model uses facial expression of the person for mood identification and suggests music and movies to users based on identified mood.

III. DESIGN

This project aims to implement a web application with a deep learning model for music and movie recommendation to uplift the mood. Recommendation systems currently in use are not accurate in recommending content which suits the user's mood. Hence, there is a requirement for proper content for user relaxation for which the most superior solution is entertainment in the form of music and movies. The proposed recommendation system will overcome these drawbacks by performing face detection and emotion recognition for music and movie recommendation, and recommend the content most suitable to uplift the user mood.

A. *Architecture of Proposed System*

The architecture of the proposed system is demonstrated in Fig. 1. Firstly, the user image is captured, followed by his/her face recognition, then the emotion is detected based on facial expression and finally the data is filtered to recommend music and movies to users. So, with the aim of simplifying our product we decided to also give faster responses to the end user. Our project has a simple frontend where the user will have to capture his image and our backend will send it to an API server, which is also designed mainly to reduce the load of processing of deep learning neural networks and images. Whenever an image is captured by the user the backend will send it to the API server, in the form of a stream and in the API server we first detect the face of the User and that detected face will be sent into our trained convolutional neural network model and an emotion is predicted by it. This emotion is sent back into our web application server and it will be used by a filter manager to further give appropriate recommendations of music and movies. SQL Lite is used to store the user data. JSON is used in the proposed system to store music and movies data.

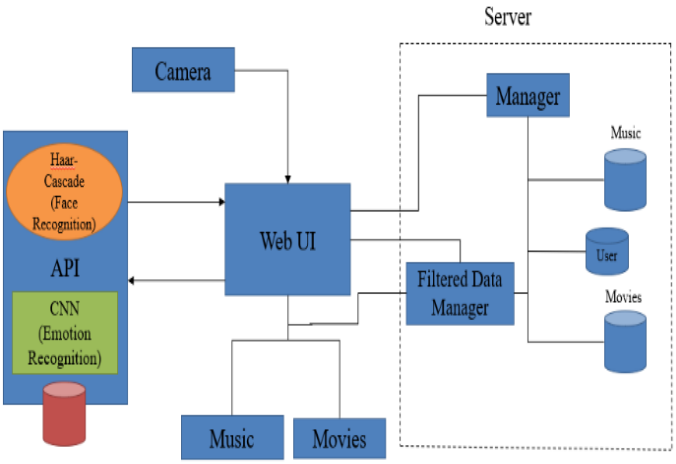
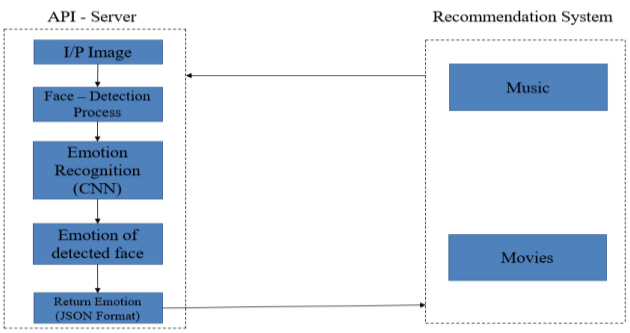


Fig 1. Architecture of the proposed system

B. *Back-end*

Django is used as the web framework to run both the web application server and the API server. The processes occurring in the backend are: On the main server, first the images are captured and stored, then the API server is connected and finally images are sent to the server by POST method. In the first stage, API server grabs the image and converts it into the OpenCV format of image for better processing, then Haar Cascade Face Classifier compatible with OpenCV is used to detect the face on any image efficiently, after detecting, image is sent to the trained neural network for prediction, where it has different categories of emotions, then the emotion will be sent back to main server where, the emotion is used to filter the data available to recommend music and movies, as shown in Fig.2.



ISBN 978-1-6654-3986-2

Fig 2. Back-end of the proposed system

C. Face – Recognition Process

In the API server, as shown in Fig.3, Haar Cascade face classifier is used which is compatible with OpenCV and Python. The images are resized to 48x48 pixels and converted to black and white to normalize the process. The Haar Cascade algorithm is applied to detect the face and set the boundaries or bounding box around the face. The bounding box coordinates help in cropping the image. The cropped image is the output of the face detection process which is used in the next step for detecting the emotion.

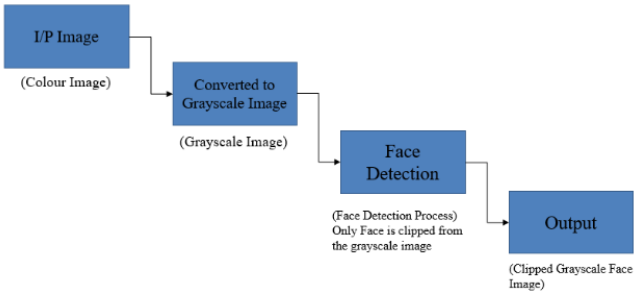


Fig 3. Face – Recognition Process

D. Emotion Recognition

In the API server, shown in Fig.4., trained Neural network model is used for predicting the emotion, the input to the network will be the output of the face detection process. The cropped image of the face will be sent to the first layer of the neural network in form of one dimensional arra. Next, we have used three convolutional layers with 128 neurons each for extracting more features from each image and each convolutional layer is supported with a max pooling layer for compressing the data. ReLu also called as activation function which stands for rectified linear, used for cutting out negative weights on the neurons, then in the end we are using the softmax activation function to better categorize the data, for predicting the emotion It categorizes into four different emotions those are Happy, Frustrated, Sad and Peaceful. Then output of the model is dumped into the JSON format for sending it into the main server.

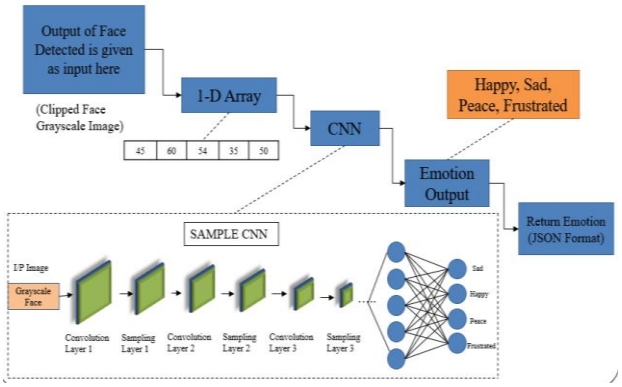


Fig 4. Emotion Recognition Process

E. Recommendation System

On the Main Server that hosts the Web Application, Output from API server is captured and then decode the data from JSON and send to the main server's backend for further processing. The detected emotion will be used by filter manager to filter out the data for predicting better music and movies to uplift the mood of the user using our product. The filter manager for music will use the emotion and based on that it will filter the JSON files to recommend music as shown in Fig.5. This data will be converted to a python object and sent to the front end to display it to the user, in the same way movie filter manager will use the emotion variable to filter out the movie, convert them to object format and send them to recommend it to the user.

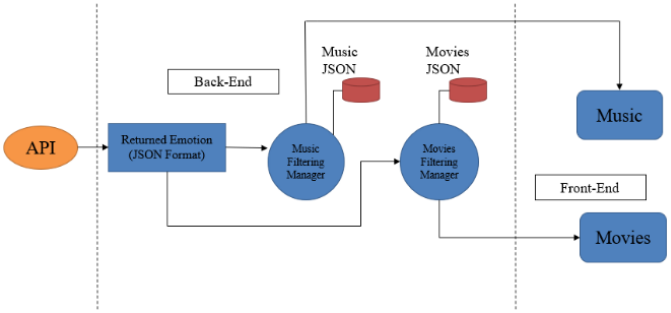


Fig 5. Recommendation System Process

IV. IMPLEMENTATION

This section gives you insights about the methods used in the execution of the proposed system.

A. Haar Cascade (FACE-RECOGNITION)

A Haar classifier, also known as a Haar cascade classifier [7], is a machine learning object recognition algorithm that can recognize objects in images and videos. It consists of four stages:

Computing Haar Features - A Haar feature is a set of computations performed on consecutive rectangular sections of a detection window at a specified position. The calculation is adding up the pixel intensities in each region and then subtracting the total.

Making Integral Images - The computation of these Haar characteristics is sped considerably by using integral pictures. Rather than computing at each pixel, it divides the screen into sub-rectangles and produces array references for each of them. The Haar features are then computed using them.

Adaboost Training - Adaboost basically picks the best highlights and prepares the classifiers to utilize them. It creates a "strong classifier" by combining "weak classifiers" that the algorithm may use to detect items.

Executing Cascading Classifiers - The cascade classifier consists of a sequence of stages, each of which contains a group of weak learners. Weak learners are taught via boosting, which produces a highly accurate classifier based on the average prediction of all weak learners.

ISBN 978-1-6654-3986-2

B. Web Application Server Side

In this phase we are showing the product to the consumer or user, here is where the user interacts with the product. We are using Django as the web framework because it is made with Python, which is a highly compatible language when using machine learning libraries. It has features and libraries which are for image processing, networking, JSON parsing and dumping which are used in our project. In the front end, where the user interacts, we are letting them register for an account and then with the Django's ability to authentic users easily, we have that functionality and for capturing images we are using OpenCV library of Python to capture an image and store it and later using this for detection which will be in API server phase, then connecting to the API server to get the emotion of the image captured. After receiving the image's emotion from the API server, we are using the emotion to filter the data from both the movies dataset and music dataset. Separate filter manager is used to filter music and movies data and storing the data in the JSON file. JSON objects are easily converted to Python objects, and we have done an analysis on different JSON parsers performance and finalized ujson library which is ultra-fast json parser, better for loading data. Finally, after filtering the data, it will be displayed to the user.

C. API Server Side

On this site we are hosting the API commands, for detecting faces and predicting emotions, API stands for Application Programming Interface. The main purpose of an API is to reduce the load on one server, mainly the server on which the user can interact; it should have faster loading speeds and fewer amounts to processes. REST APIs are always faster and lightweight, so we have decided to use them in our project, for communicating between the two servers. As our API server receives the image, it converts them into OpenCV version for faster processing. The image is first preprocessed to be set to a specific length and width, then it is converted to black and white. Next, face detection is performed using Haar Cascade classifier and image is cropped to the length of the face so that the neural network does not have to deal with other objects in the image. The output of this phase is sent in to the neural network for facial emotion prediction, the input layer of the network converts the image matrix in form of a one dimensional array, for easier processing of data and the three convolutional hidden layers will extract the features, and proceed based on the trained weights of the neurons and point them to last layer.

D. CNN MODEL (EMOTION-RECOGNITION)

A Convolutional Neural Network (CNN) is a type of neural network that specializes in processing data with a grid-like architecture, such as an image. There are typically three layers in a CNN: a convolutional layer, a pooling layer, and a fully connected layer. CNN's main building block is the convolution layer. It is responsible for the majority of the network's computational load. The pooling layer uses a summary statistic of neighboring outputs to replace the network's output at specific spots. This reduces the representation's spatial size, which reduces the amount of computation and weights necessary. The Fully Connected layer aids in the mapping of representations between input and output. Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

V. ANALYSIS

In proposed models 1600 images from FER-2013 are used, it contains 48x48 pixel grayscale images and are divided into 4 different emotions. These are further categorized into train and test data, divided in the ratio of 4:1 or 80% of data is used as training and 20% as validation The system is trained using three convolutional neural network layers for better feature extractions and max pooling layer is used to compress the data by almost 30-40%, Rectified linear (ReLu) is used as activation function in the hidden layers. Categorical Cross entropy is considered as loss function and Adam is used as the optimization function. The batch size is 32 for 50 epochs. Batch size means the number of data elements it takes before updating the model, in proposed work it refers to the number of images and epochs means covering all of the training dataset or images. Better results are achieved for larger batch size and with one layer having a smaller number of neurons. The accuracy of the proposed model is obtained as 81% and for a higher number of epochs, the model is overfitting. Figure 6 shows the training accuracy and testing accuracy of the model for 50 epochs and Fig. 7 displays the training loss and testing loss of the model for 50 epochs.

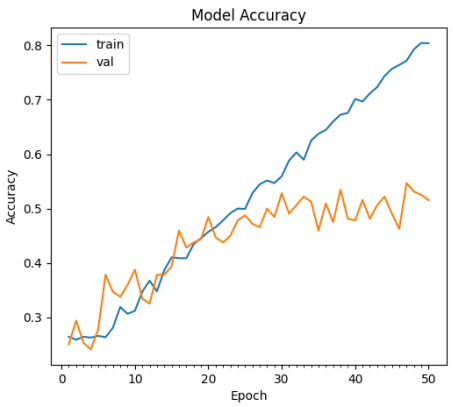


Fig 6. Accuracy of Model Used in this Proposed System

JSON Parsing

The different json models are described below:

- -rapidjson: It is a fast JSON parser written in C++, for XML style APIs.
- -ujson: ultrajson or ultra-fast json, it streams data in the file and as it does that, it interprets the strings in the file as objects.

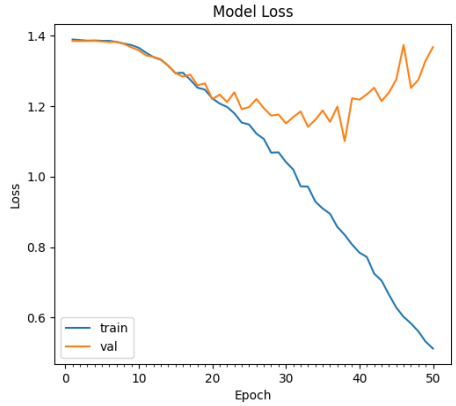


Fig 7. Loss of Model Used in this Proposed System

- -orjson: It is similar to ujson but it serializes and deserializes the data.
- -json: It is the stock library included with python, which is used to load and store data.

We have observed different loading speeds tested 10 different times for each method. The results obtained show that ujson and orjson are better for loading data. As ujson is generating better results most of the time, ultra-fast json (ujson) is used in a proposed method for loading data into the web application. Overall, for movie json data we have seen approximately 20-25% improvement in loading speed by using rapidjson and 40-50% improvement using ujson and orjson. For music data we have seen approximately 25-35% improvement in loading speed using rapidjson and 35-45% with orjson and 45-55% from ujson. As ujson provides better performance comparative to other json models, it is used in proposed study. Table 1 shows the comparative analysis of various JSON Parsers.

Table 1: Comparison of various JSON Parsers

Types of JSON	Music (JSON)	Movies (JSON)
ujson	0.2-0.25 ms	0.25-0.3 ms
orjson	0.25-0.3 ms	0.25-0.3 ms
Rapidjson	0.3-0.35 ms	0.35-0.4 ms
json(stock)	0.4-0.45 ms	0.45-0.5 ms

Fig 8 shows the percentage improved using ujson when compared with the stock json library that comes with Python and indicates an increase in loading speed by 1.5 times.

Web Application

The results of the web application designed for proposed model are shown in Fig. 9 to Fig. 14.

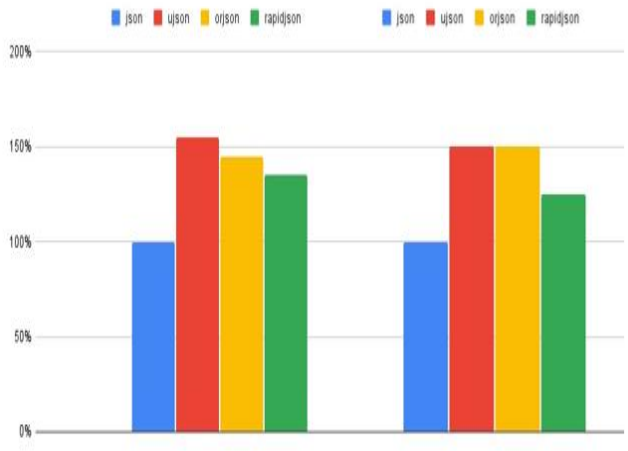


Fig 8. Performance of JSON parsers in Music data (Left) and Movies Data (Right)

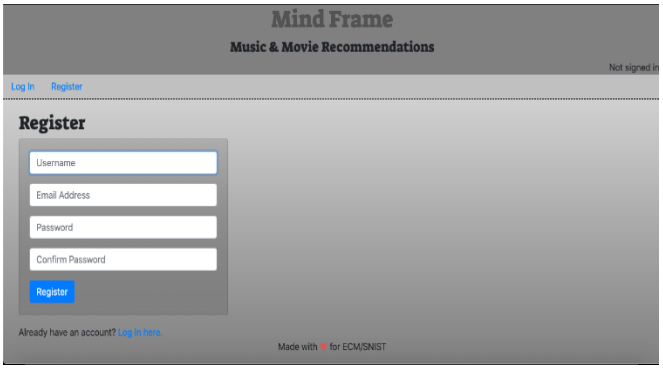


Fig 9. Registration Page of Web-Application

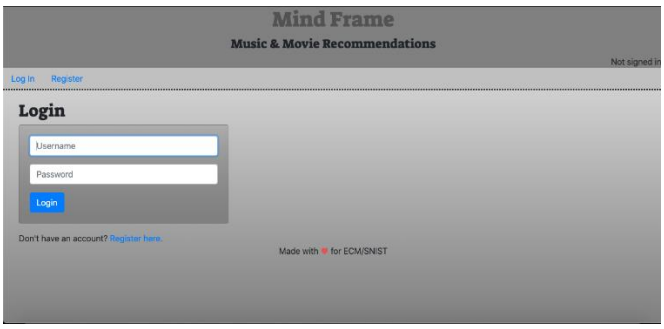


Fig 10. Login Page of Web-Application

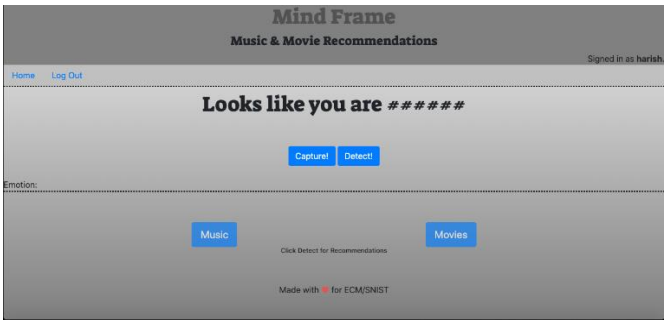


Fig 11. After Logging in

ISBN 978-1-6654-3986-2

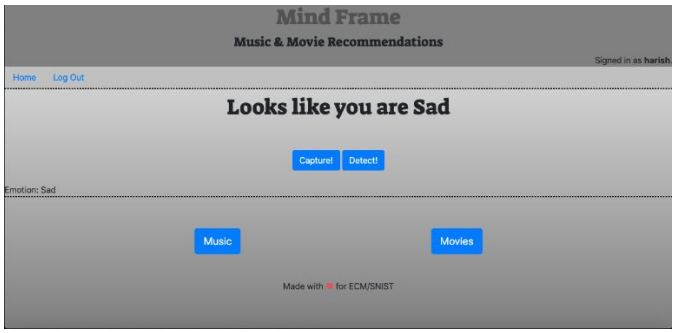


Fig 12. Emotion detected Result

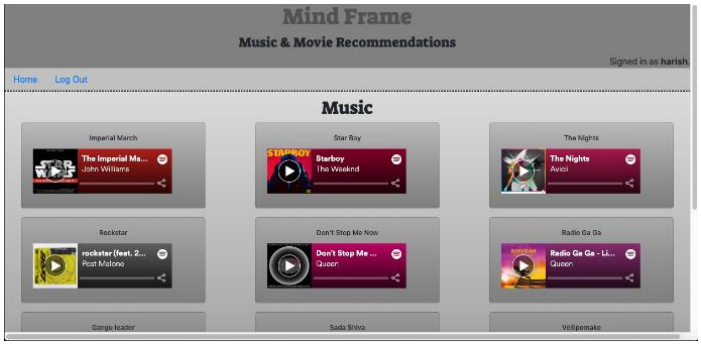


Fig 13. Music Recommendations based on emotion detected

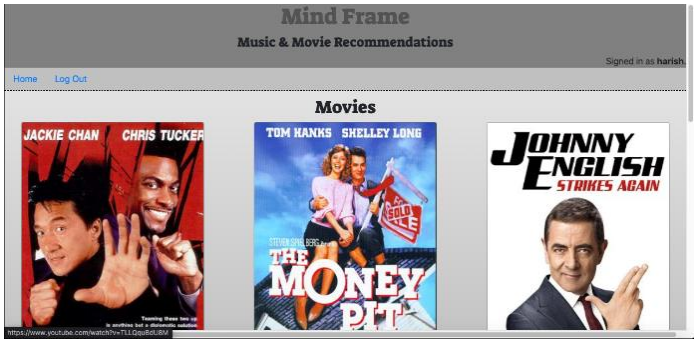


Fig 14. Movie Recommendations based on emotion detected

VI. CONCLUSION AND FUTURE SCOPE

Generally, recommendations are made by knowing the previous activity of the user but this does not recommend the right music or movies because people's liking towards things change frequently and the mood of the user is not the same all the time. The proposed system is a web application which recommends Music and Movies depending on the current mood of the user by detecting their emotion using the CNN Model. The Haar Cascade algorithm is applied for facial feature extraction. It is found that the proposed recommendation system based on the user's facial expressions provide music and movie recommendations with 81% accuracy. Ujson parser is used for faster recommendations reducing latency in processing the data and increasing the overall performance by 45-55% compared to other json parsers.

The proposed recommendation system will not only reduce latency in processing large amount data on the go and also in creating patterns based on the previous activity of the user, by instantly determining the mood of the user and making recommendations according to it and also this can be blended with previously used recommendations methods to make it more useful for each user.

In the future work, an emotion pattern recognition system based on the activity of streaming movies or music can be designed. This may help to understand what kind of content a user likes to watch while he/ she is in various moods and estimate the emotion pattern and recommend the proper content.

REFERENCES

[1] Snehal R.Chavare, Chetan J.Awati, Suresh K.Shirgave. "Smart Recommender System Using Deep Learning." 6th International Conference on Inventive Computation Technologies (ICICT), (2021), pp. 231-235

[2] Vincenzo Moscato, Antonio Picariello, Giancarlo Sperlí, "An Emotional Recommender System for Music." IEEE Intelligent Systems, (2020), pp. 1-10

[3] Champika H.P.D. Wishwanath, Supunmali Ahangama, "A Useralized Music Recommendation System based on User mood." 19th International Conference on Advances in ICT for Emerging Regions (ICTer), (2019)

[4] Aurobind V. Iyer, Viral Pasad, Karan Prajapati, "Emotion Based Mood Enhancing Music Recommendation." 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), (2017), pp 1573-1577

[5] Shreya Agarwal, Pooja Jain, "An Improved Approach for Movie Recommendation System." International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), (2017), pp. 336-342

[6] Saim Shin, Dalwon Jang, Jongseol J. Lee, Sei-Jin Jang and Ji-Hwan Kim, "MyMusicShuffler: Mood-Based Music Recommendation with the Practical Usage of Brainwave Signals." IEEE International Conference on Consumer Electronics (ICCE), (2014), pp. 355-356

[7] Padilla, Rafael, C. F. F. Costa Filho, and M. G. F. Costa. "Evaluation of haar cascade classifiers designed for face detection." World Academy of Science, Engineering and Technology 64 (2012), pp. 362-365.

[8] Meng, Zibo, Ping Liu, Jie Cai, Shizhong Han, and Yan Tong. "Identity-aware convolutional neural network for facial expression recognition." In 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), pp. 558-565.

[9] Peng, Dunlu, Lidong Cao, and Wenjie Xu. "Using JSON for data exchanging in web service applications." Journal of Computational Information Systems 7, no. 16 (2011), pp. 5883-5890.

[10] Seungjae Lee, Jung Hyun Kim, Sung Min Kim and Won Young Yoo , "SMOODI: MOOD-BASED MUSIC RECOMMENDATION PLAYER." IEEE International Conference on Multimedia and Expo, (2011).