TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PASHCHIMANCHAL CAMPUS

(*MUSIC RECOMMENDATION BASED ON FACIAL EMOTION RECOGNIZATION*)

PROJECT SUPERVISOR:
Er. Ramesh Thapa

SUBMITTED BY:
Anish Sharma (PAS074BCT006)
Dipesh Adhikari (PAS074BCT014)
Gaurab Shrestha (PAS074BCT015)
Mohan Singh KC (PAS074BCT021)

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR'S DEGREE IN COMPUTER ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING
POKHARA, NEPAL
MAY, 2022

Acknowledgement

It is a great contentment to have the opportunity to extend our wholehearted gratitude to everyone who helped us throughout the course of this project. We are profoundly grateful to our supervisor **Mr. Ramesh Thapa**, Head of Department of Electronics and Computer Department, for his expert guidance, continuous encouragement and ever willingness to spare time from his busy schedule for the project's progress reviews. His continuous inspiration has made us complete this project in time and achieve its target.

We would also like to express our genuine appreciation to **Mr. Laxmi Prasad Bastola,** Deputy Head of Department of Electronics and Computer Department, for his constant motivation, support and for providing us with a suitable working environment. We would also like to extend our sincere regards to all the faculty members for their support and encouragement.

Our backbone was the support of the classmates who have directly or indirectly encouraged and assisted us in carrying out this work. Their constructive criticism and motivation are the key to our success towards our project. Moreover, sincere appreciation to our group members for giving their best to finish the project within the timeline as we have planned.

Anish Sharma(PAS074BCT006)

Dipesh Adhikari(PAS074BCT014)

Gaurab Shrestha(PAS074BCT015)

Mohan Singh KC(PAS074BCT021)

Abstract

In this digital age, where the internet is speeding up and changing people's lifestyles, to enlighten, music recommendation systems such as Spotify, Deezer, and others have replaced manually playing music, using wearable computing devices, or categorizing based on auditory attributes. We presented a novel way to improve this recommender system by recommending a playlist of tracks based on the user's facial emotion. It is difficult for a person to choose which music to listen to from a vast array of available selections. Depending on the user's mood, there have been numerous suggestion frameworks available for concerns such as music, dining, and shopping. Face recognition technology has gotten a lot of press because of its vast range of applications and market possibilities. It is used in a variety of fields, including security systems, digital video processing, and other technical advancements. Furthermore, music is a type of art that is seen to have a stronger emotional connection.

Our music recommendation's main goal is to give recommendations that are tailored to their facial emotions. The examination of the user's facial expression/emotion may lead to a better understanding of the user's present emotional or mental state. Humans are well-known for using facial expressions to indicate what they wish to say and the context in which they meant their words. More than 60% of users believe that at some point in the future, the number of songs in their music collection will be so large that they will be unable to find the song they need to play. By creating a suggestion playlist, it may be possible to aid a user in deciding which music to listen to based on the emotion. The user would not have to waste time looking for music since the best track matching the user's mood would be detected and a playlist would be provided to the user based on his or her mood. With the use of a webcam, the user's image is captured. The user's photo is taken, and then, based on the user's mood/emotion, an appropriate song from the playlist is displayed.

Keywords

Face Recognition, Feature extraction, Emotion detection, SVM, Random Forest, Logistic Regression, Local Binary Pattern, Grayscale image, Pygame, Tkinter, Music Player, Webcam

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
AAM	Active Appearance Model
AU	Action Units
LBP	Local Binary Pattern
FER	Face Emotion Recognition
GUI	Graphical User Interface

CHAPTER 1

1 INTRODUCTION

The mood of any human being can be detected by the facial emotion or expression expressed by him/her. Angry, disgust, fear, happy, neutral, sad, and surprise are all examples of human emotions. This umbrella of emotions can include a variety of additional emotions such as cheerful (which is a variation of joy) and contempt (which is a variation of disgust). These are very faint emotions. Facial muscle contortions are quite little, and recognizing them can be difficult because even minor variations result in distinct expressions. Also, because emotions are very context-dependent, different or even the same people's expressions of the same feeling may differ.

While the focus may be on simply those portions of the face that express the most emotions, such as the mouth and eyes, how these gestures are extracted and classified remains a key question. For these purposes, neural networks and machine learning have been applied with good results.

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

The goal of this project is to develop a music recommendation system/music player that can recognize the user's face, determine their current mood, and then offer a playlist depending on that mood.

1.1 Background

Music has become an inseparable part of our lives. Music is frequently used as a source of amusement and refreshment. Music recommender plays a vital part in making these refreshment sessions less tiresome and more effective. Capturing and identifying a person's emotion and playing appropriate music to match that emotion can help to soothe the user's mind and provide a pleasurable effect.

The goal of the project is to capture a person's feelings through their facial expressions. Through the web camera interface available on computing systems, a music player is meant to record human expression. The software captures the user's image and then extracts characteristics from the face of a target human being using image segmentation and image processing techniques in order to discern the emotion that the person is attempting to express. The project seeks to brighten the user's mood by playing tunes that match the user's criteria by detecting the user's photograph.

1.2 Problem Statement

People nowadays are more concerned with making their lives easy through entertainment. One method is to find music that appeals to your taste and emotions. Music has been ingrained in our culture. People are listening to music in greater numbers than ever before, and creation is increasing at an exponential rate. So, in a sea of tunes, finding songs that suit and match your emotion will be more difficult. Humans have the innate capacity to read someone's emotions by looking at their face. This ability, if taught by an electronic device, could be useful in the actual world. Music is significantly more powerful than language in terms of evoking emotions and feelings. Music has the ability to get deep into our emotional core as humans. As a result, listening to wonderful music can make us feel better.

To make this challenge easier to solve, we propose developing an application that uses facial expression recognition algorithms to capture the user's feelings. After capturing the emotion, a list of music based on the sentiment is provided.

1.3 Objective

1.3.1 Specific Objectives

- To develop a system that provides music playlist recommendations based on user facial emotion.

1.3.2 General Objectives

The general objectives can be listed as:

- To develop a GUI application for music player.
- To develop a system for real-time facial emotion detection.

1.3.3 Academic Objectives

In terms of academic level, this project helps us learn more on Data Mining and Machine Learning which are widely used topics. Because of this project we should be able to learn about scripting languages such as Python. Besides these, we will obtain a new level of academic step learning about supervised methods of machine learning and be familiar with the terms like accuracy, different classifiers, precision, recall, hyperparameter tunings, etc. In brief, following are the academic objectives of doing this project:

- To apply theoretical engineering knowledge into practical industrial approach
- To use proper object-oriented approach for building the project
- To learn and implement machine learning, data mining technology

1.4 Scope

The scopes of our project are:

- We can collaborate with big tech music recommendation companies like Spotify, Deezer, etc. for further enhancing the user requirements and boosting users' satisfaction.
- This will help the physically challenged people experience the music without any inconvenience.
- This can be used as a filter in trending social media like Instagram, snapchat, etc.

CHAPTER 2

2 LITERATURE REVIEW

Emotion-based music players are in high demand and will benefit industries such as Emotion Intelligence, Medical Science, and Psychology. Techniques like Neural Networks (NN) and Support Vector Machines (SVM) have been applied recently. We investigate these strategies in relation to our application.

[1]Anagha S. Dhavalikar and Dr. R. K. Kulkarni Proposed an Automatic Facial Expression recognition system. In this method of theirs, there were three phases or steps.

- Face detection
- Feature Extraction and
- Expression Recognition

The first phase of face detection consists of an RGB color model analysis that includes ISO illumination processing for obtaining a face and operations for maintaining the required face, including the eyes and lips. The Active Appearance Model (AAM) is also used in the suggested algorithm. This is a technique for extracting facial characteristics. Several places and features on the face (also known as Action Units, AU) such as eyes, eyebrows, mouth, and lips are identified in this technique, and a file is created that contains the properties of the action points observed. Face emotions are sent into the AAM Model, which alters depending on the expression.

Advantages:-

- Considers all detailed aspects of the face for analysis.
- This model is based on supervised learning.

Disadvantages:-

- Shows lower efficiency for blur images.
- Each AU is needed for the best results.

[2]Yong-Hwan Lee, Woori Han, and Young Kim proposed a system based on Bezier curve fitting. This algorithm uses a multiple-phase process for facial expression and emotions. The first one is the detection and processing of facial area from the input original image. The second step is the verification of emotion in the region of interest. The first phase of face detection uses color still image based on skin color pixel by initialized spatial filtering, based on the result of lighting compassion. Later it uses the Feature Map To measure eyes and lips face location and facial shape. After extracting regions of interest, this method extracts points from the feature map in order to apply the Bezier curve to the eye and mouth techniques to study the difference between Hausdorff distance and the Bezier curve between the image from the database and the face picture entered.

Advantages:-

- The size of the dataset isn't restricted. The algorithm shows high efficiency even on large datasets.
- The algorithm can also be used for 3D pictures. Thus, a 3D picture can also be exploited for information.

Disadvantages:-

- Each point of study in the data has a global influence. There are no outliers. This can cause overfitting. Hence the efficiency can decrease if the data set is distorted due to overfitting.

[3]A. Lehtiniemi and J. Holm suggested a system for music recommendation based on an animated mood picture. This program allows users to connect with a group of photographs to generate Pictorial music recommendations. This music suggestion program was produced by Nokia Research. To define the encoding of audio and genre signals, this application uses text meta tags.

Advantages:-

- This algorithm studies real-time data for analysis.
- The data is extracted from the reactions of the users, and hence, provides high accuracy.

Disadvantages:-

- The availability of data can be limited. Audio signals can have lower efficiency because of the noises and signal mixing.

[4]Yusuf Yaslan et al. proposed an emotion-based music recommendation system that learns the user's emotion from signals obtained through wearable computing devices that are integrated with galvanic skin response (GSR) and photoplethysmography (PPG) physiological sensors in their paper. Emotions are a basic part of human nature. They play a vital role throughout life. In this paper, the emotion recognition problem is taken into account as arousal and valence prediction from multi-channel physiological signals.

[5]In Ayush Guidel stated that human beings' state of mind and current emotional mood can be easily observed through their facial expressions. This system was developed by taking basic emotions (happy, sad, anger, excitement, surprise, disgust, fear, and neutral) into consideration. Face detection in this project was implemented by using a convolutional neural network. Music is usually told as a "language of emotions" throughout the planet.

[6]CH Radhika advised manual segregation of a playlist and annotation of songs, following the current emotional state of a user, as a labor-intensive and time-consuming task. Numerous algorithms had been proposed to automate this manner. However, the prevailing algorithms are slow, increase the overall cost of the system by using additional hardware (e.g., EEG structures and sensors), and feature much less accuracy. The paper presents an algorithm that automatically does the process of generating a playlist of audio, based on the facial expressions of a person, for rendering salvage of time as well as labor, invested in performing this process manually. The algorithm given in the paper directs at reducing the overall computational time and the cost of the designed system.

[7]A. Madhuri, m. Deepali, S. Upasana and G. Megha proposed Music Recommendation Based on Face Emotion Recognition where they have used FER-2013 dataset and CNN for emotion detection. They proposed a system that tends to reduce computational time involved in obtaining the results and overall cost of the designed system. But they have used the FER-2013 dataset which is a very unbalanced dataset, images are not aligned contrast variation, outliers and incorrectly labelled.

Improvement in our proposed system:

- We used our self-created dataset which gives the best results than other available public datasets.
- We have used two feature extraction methods like LBP as an image feature extractor and grayscale pixel values as a separate feature.

CHAPTER 3

3 REQUIREMENT ANALYSIS

3.1 Planning

In the planning phase, the study of reliable and effective algorithms is done and discussed which algorithm is to be used. On the other hand, how data were collected and preprocessed for more fine and accurate results was discussed. Since a huge amount of data was needed for better accuracy, we had collected the data surfing the internet. Since we are new to this project, we have decided to use grayscale pixel values as a separate feature and another local binary pattern algorithm for feature extraction and support vector machine, random forest, and logistic regression for training the dataset. We used three machine learning algorithms to select the best machine learning model. We have decided to implement these algorithms by using the OpenCV framework. For the music database, songs related to different emotion classes were downloaded and collected from the internet and asked from different people according to their preferences for the related emotion.

3.2 Software and Hardware Requirements

3.2.1 Hardware Requirements

- 4GB RAM
- 2GHz dual-core processor or better
- Minimum 25 GB HDD space
- Webcam

3.2.2 Software Requirements

- Python programming language
- Jupyter Notebook
- VSCode IDE(Selective)
- OpenCV framework
- Google Colaboratory

3.3 Feasibility Study

Before starting the project, a feasibility study is carried out to measure the viability of the system. Feasibility study is necessary to determine if creating a new or improved system is friendly with the cost, benefits, operation, technology and time. Following feasibility study is given as below:

3.3.1 Technical Feasibility

Technical feasibility is one of the first studies that must be conducted after the project has been identified. The technical feasibility study includes the hardware and software devices. The required technologies (Python, VScode IDE, Jupyter Notebook, and Google colaboratory) existed.

3.3.2 Operational Feasibility

Operational Feasibility is a metric that assesses how well a proposed system solves the problem and exploits the opportunities indicated during the scope definition. The following factors were taken into account when determining the project's technical feasibility:

- The system will recognize and capture a face image.
- The captured image is then (identified in which category).
- A music playlist is then provided based on the categorized emotion.

3.3.3 Economic Feasibility

The purpose of economic feasibility is to determine the positive economic benefits that include quantification and identification. The system is economically feasible due to the availability of all requirements such as a collection of data from:

- CK+48
- FER-2013
- Self-created dataset(Images downloaded from Google)

3.3.4 Schedule Feasibility

The timeliness of a project is measured by its schedule feasibility. Because the system is constructed in such a way that it will complete within the specified time, it was deemed to be schedule viable.

CHAPTER 4

4 METHODOLOGY

4.1 Proposed System

The proposed system allows us to present the user's involvement with the music player which is a GUI application. The system's goal is to adequately capture the face using the camera. Captured images are cropped in the face area, image features are extracted and fed into a machine learning algorithm that classifies and predicts the emotion. Then, using the emotion obtained from the acquired image, a playlist of songs is generated. Our proposed system's main goal is to automatically generate a music playlist based only on the user’s facial emotion that is detected by the model, not based on user preferences. This tedious task of manually segregating or grouping songs into different lists is reduced and helps in generating an appropriate playlist based on an individual's facial emotion. Music Recommendation Based on Facial Emotion Recognition aims at scanning and interpreting the facial emotion and accordingly creating a playlist based on the parameters provided. Thus, our proposed system focuses on detecting human emotions for developing an emotion-based music player.

4.2 Data collection

For our research project, we collected different public datasets available on the internet including our own self-created dataset to train our model for facial emotion detection. We collected public image datasets from Kaggle and we downloaded images from Google, Facebook, etc, and requested our friends and requested different facial emotion images databases for access to the databases.

4.2.1 Fer-2013 dataset

This dataset was prepared by Google in 2013 and consists of face images of 48x48 pixels. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The FER-2013 dataset consists of 28,709 labelled images in the training set, and 3,589 labelled images in the test set. Each image in FER-2013 is labelled as one of seven emotions: happy, sad, angry, afraid, surprise, disgust, and neutral with happy being the most prevalent emotion, providing a baseline for random guessing of 24.4%. The images in FER-2013 consist of both posed and unposed headshots, which are in grayscale.

Some images of the FER-2013 dataset are presented below:

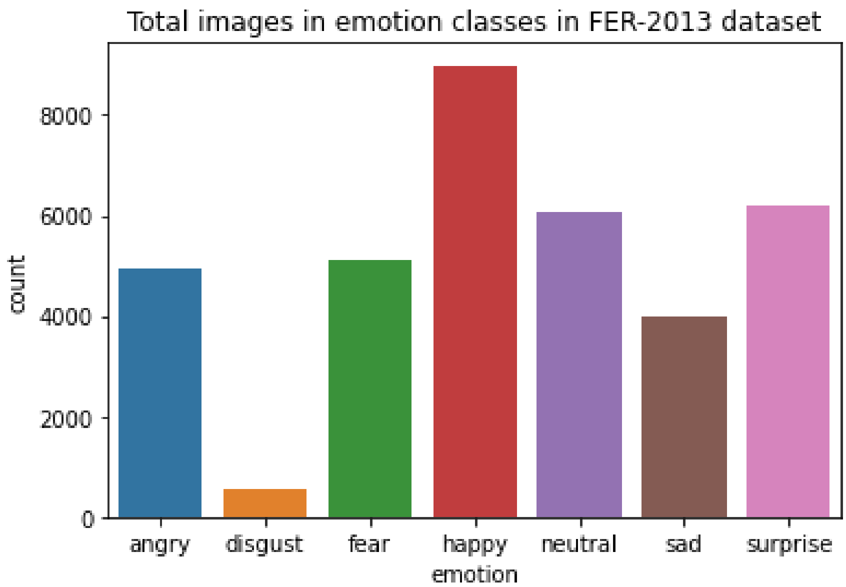
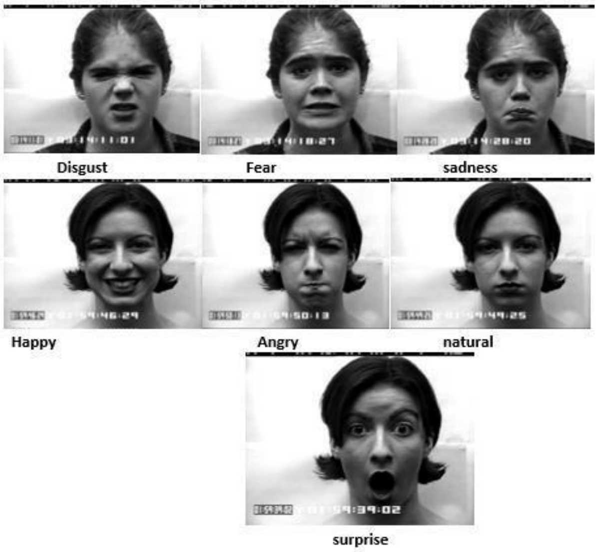


Figure 1 Images Distribution in Seven Classes in FER-2013 Dataset

4.2.2 CK+48 (Cohn Kanade) dataset

This dataset consists of eight expressions (seven primary expressions plus contempt) that were posed by more than 200 adults ranging from 18 to 30 years of age. It generally consisted of Euro-American, Afro-American, and Asian individuals. The images were taken in time frames where the initial frame was neutral, which was then transitioned into an expression that was desired at the end frame. These images were saved some in grayscale and some in colour, in 48x48 pixels. The contempt expression images are excluded in this work.

Some of the images of the CK+48 dataset are shown below:



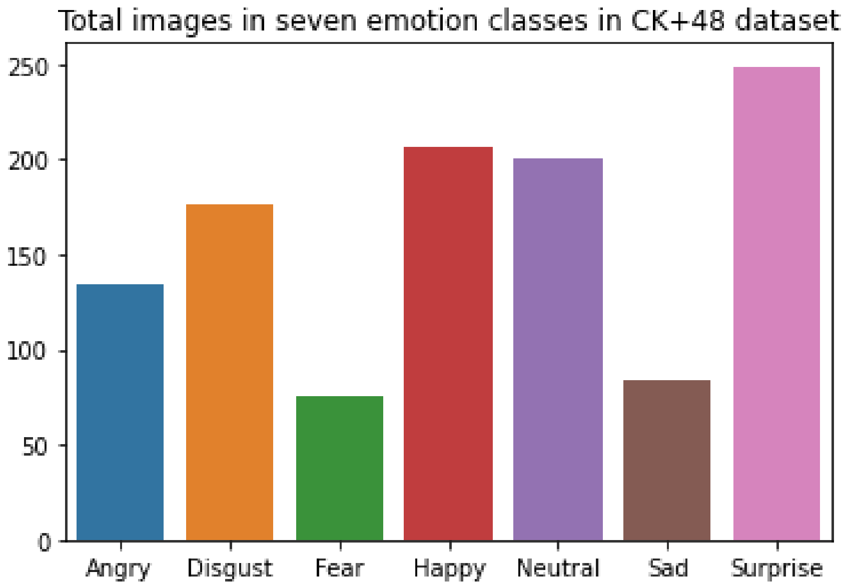
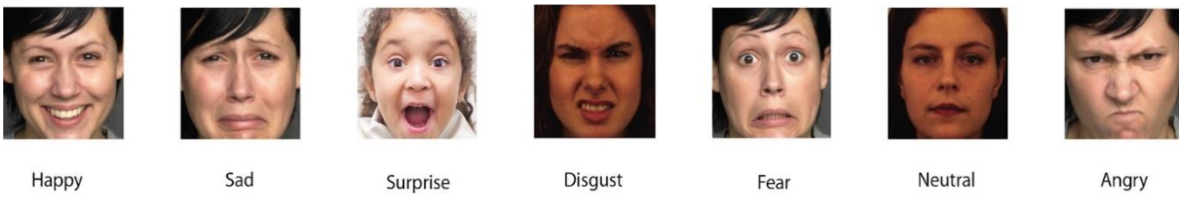


Figure 2 Images Distribution in Seven Classes in CK+48 Dataset

4.2.3 Self created dataset

For the project, we created our own dataset by collecting images from our friends, downloading images from google, by requesting some databases like KDEF databases, etc. We have collected about 4336 images including all 7 emotions where all emotions contain about an equal number of images. The collected images are cropped in the face area using some algorithm and removed most of the background part which is noise or outliers for the feature extraction. In our self-created dataset, there are RGB images that will be converted to grayscale for feature extraction and model training.

Some of the images of the self-created dataset are shown below:



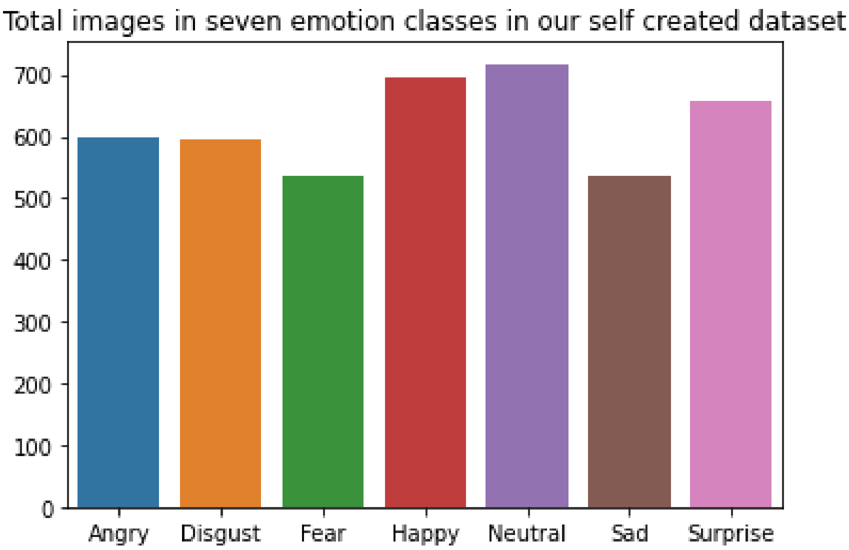


Figure 3 Images Distribution in Seven Classes in Self Created Dataset

Table 1 Data Collection

Datasets	Sample Details	Available Description
FER-2013	• Training set and Test set having about 28,709 and 3,589 images in each set • 48x48 pixels images • Contains 7 emotions images • Imbalanced dataset	"This dataset was made by Google in 2013 by gathering the results of a Google image search of every emotion and synonyms of the emotions."
CK+48	• Total 1127 images • 48x48 pixels images • Age: 18 to 30 years • Gender: 65% female • Ethnicity: 15% African-Americans and 3% of Asians and Latinos	"Annotation of FACS Action Units and emotion specified expressions."
Self-Created Dataset	• Total of 4336 images • Contains only the face area of humans	"The dataset is created by ourselves that contains images from our friends, images from google and different facial databases."

4.3 Data preprocessing

The proposed system was trained and tested using three datasets names- FER-2013, CK+48 (COHN-KANADE) dataset, and our own self-created dataset. Our model is trained on 80% of train images which are about 7240 images in total and tested on 20% of test images which are about 1810 images.

The FER-2013 dataset is very imbalanced wherein the training set, happy emotion class has about 7215 images but disgust emotion class has only 436 images. It also has intra-class variation problems, occlusion, contrast variation, eye-glasses images, outliers, etc. Some samples do not contain faces and some images are not aligned and some of them are incorrectly labelled. We copied the same dataset and doubled them and added the testing set images to the training set and a same number of images of other emotion classes were randomly selected and made the train image set.

For our self-created dataset and the other two datasets that are FER-2013 and CK+48, we normalised the faces to 48x48 pixels. Based on the structure of the face, facial images of different pixels were cropped by detecting the frontal face using the face detector algorithm which is Haar Cascade Classifier (haar_cascade_frontalface_default.xml). The results of face detection including face location, face width, and face height were automatically created. Finally, images were cropped in accordance with the result given by the face detector, and further cropped images were used for training and testing.

Figures below show the original image and cropped image:

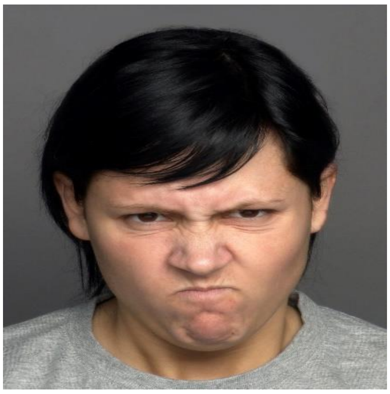


Figure 4 Original Image

Figure 5 Cropped Image

4.4 System Design

System design shows the overall design of the system. In this section, we discuss in detail the design aspects of the system.

4.4.1 System Diagram

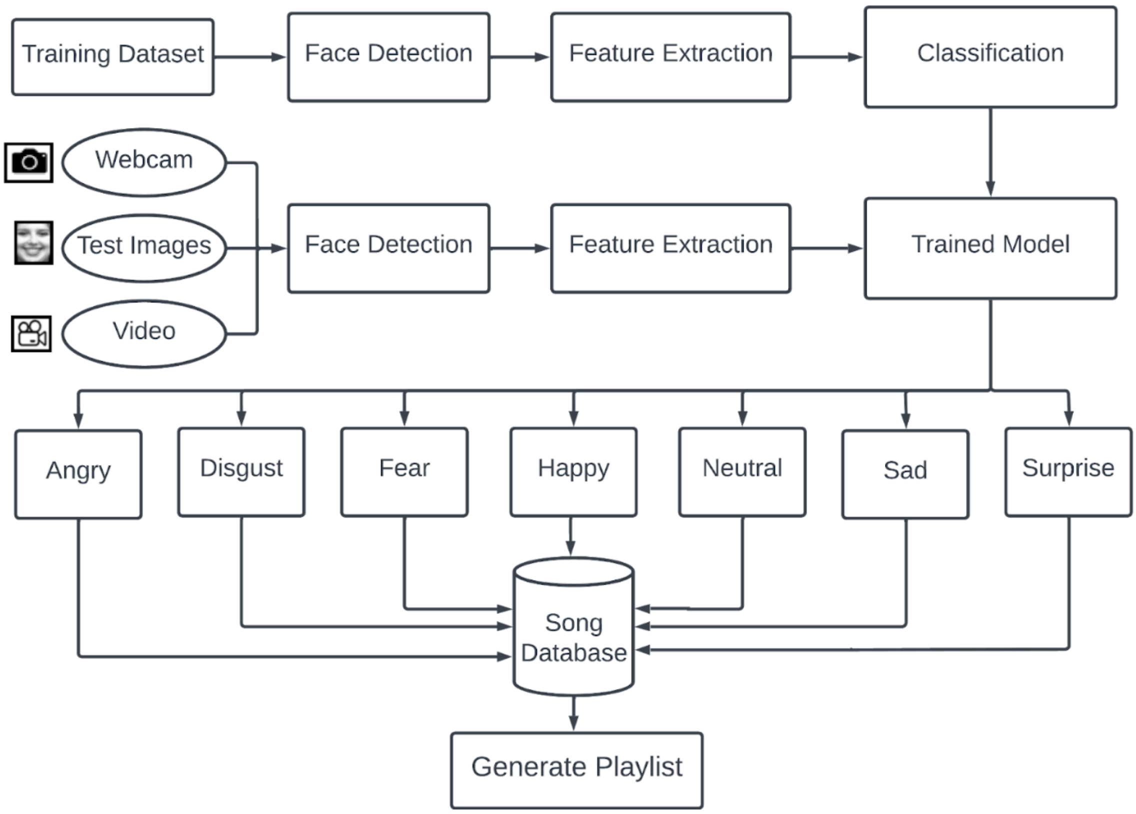


Figure 6 System Diagram

4.4.2 System Flowchart

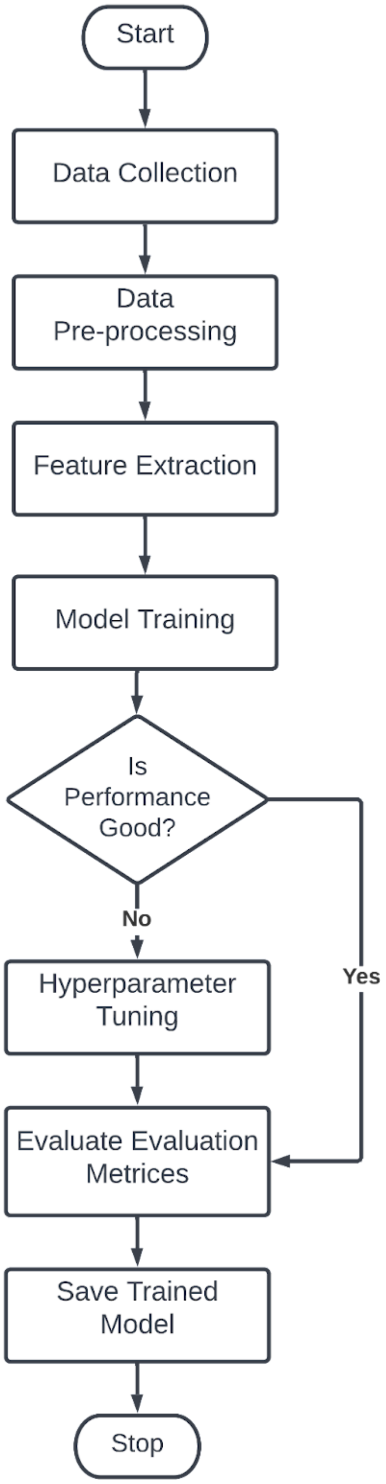


Figure 7 Model Training Flowchart

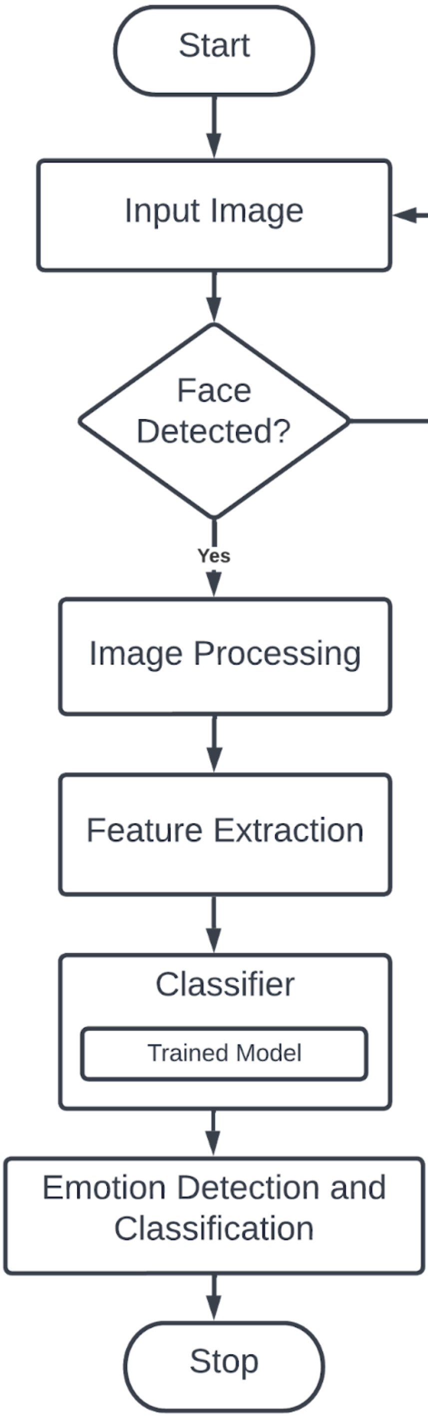


Figure 8 Model Testing Flowchart

4.4.3 User’s View Diagram

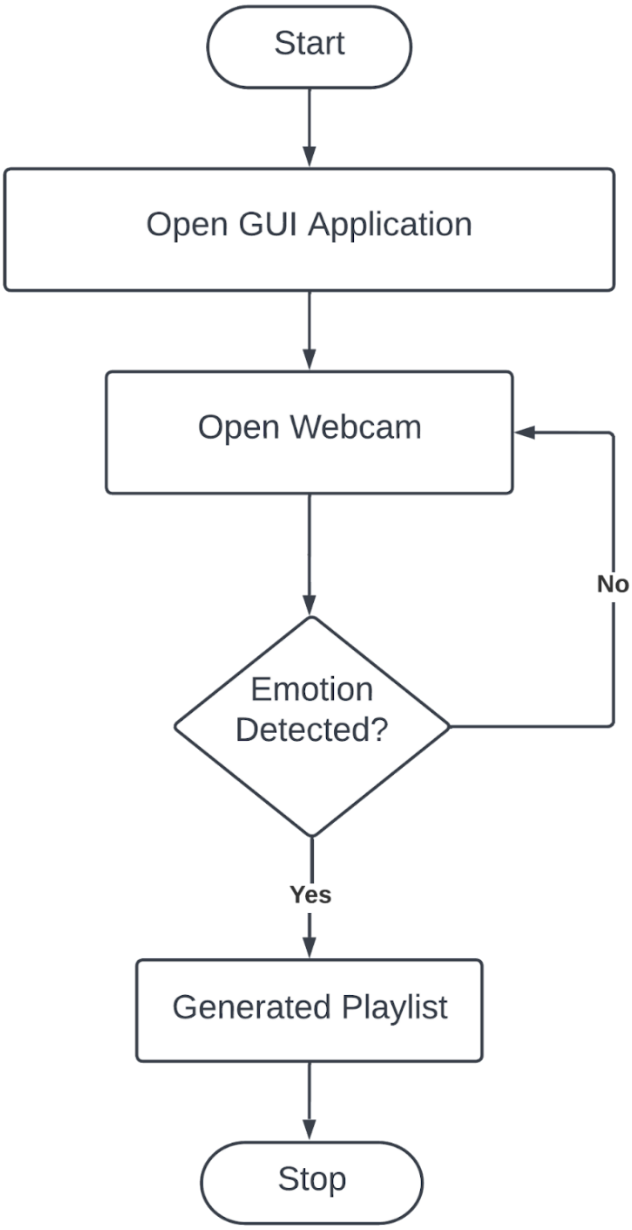


Figure 9 User's Diagram

4.5 Modules Used In Our Proposed System

As our proposed system is divided into two modules:

4.5.1 Emotion Detection Module

The facial emotion detection and recognition module is trained using a supervised learning approach in which it takes images of different facial expressions like anger, disgust, fear, happiness, neutral, sad, and surprise. The system includes the training and testing phase followed by image acquisition, face detection, image preprocessing, feature extraction, and emotion classification. Face detection and feature extraction are carried out from face images and classified into seven emotion classes.

4.5.1.1 Image Acquisition/Face Capturing

Images used for facial expression recognition are static images or image sequences. Images of faces can be captured using a camera or webcam.

4.5.1.2 Face Detection

It is one of the applications that fall under the purview of computer vision technology. This is the process of developing and training algorithms to correctly locate faces or objects in object detection or related systems in images. Real-time detection from a video frame or photos is possible. Face detection employs classifiers, which are algorithms that determine if an image contains a face or not. To improve accuracy, classifiers are trained to recognize faces using a large number of images. In our project, we use the Haar Cascade Classifier which is trained with pre-defined varying face data, allowing it to accurately detect different faces. Face detection’s major goal is to reduce external noise and other elements in order to spot the face within the frame. The cascade function is trained with a group of input files using a machine learning approach.

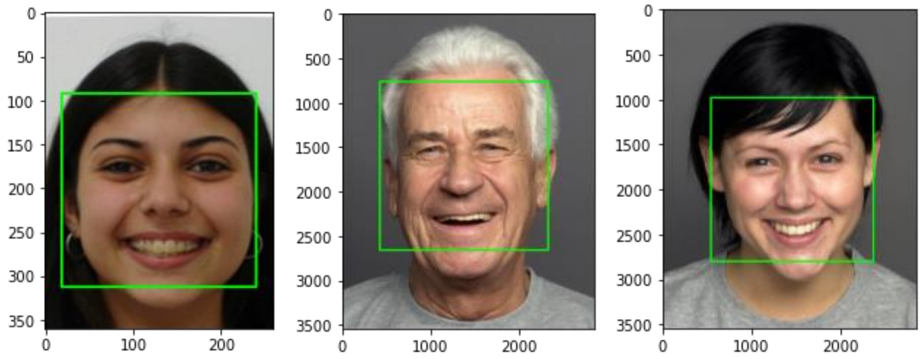


Figure 10 Face Detection

4.5.1.3 Image Pre-processing

Image pre-processing includes the removal of noise and normalisation against the variation of pixel position or brightness by colour normalisation, etc.

4.5.1.4 Feature Extraction

In a pattern classification problem, feature vector selection is the most important step. After preprocessing the image of the face, the relevant features are extracted. Scale, pose, translation, and fluctuations in illumination level are all fundamental challenges in image classification. In our project, we use two methods as feature extractors 1) grayscale pixel value as a feature and 2) LBP (Local Binary Pattern) as a feature extractor. We used both the features and used them separately to train the model.

A) Grayscale Pixel Value as Feature

We can use grayscale pixel values as a feature. An RGB or BGR image has 3 channels but a grayscale image has a single channel. An RGB or BGR image can be converted to grayscale by using OpenCV (gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)). That’s why it is the simplest way to create features from an image. The main focus is to use the raw pixel values as separate features. If the image is 48x48 pixels then it will have a total of 4608 features in grayscale.

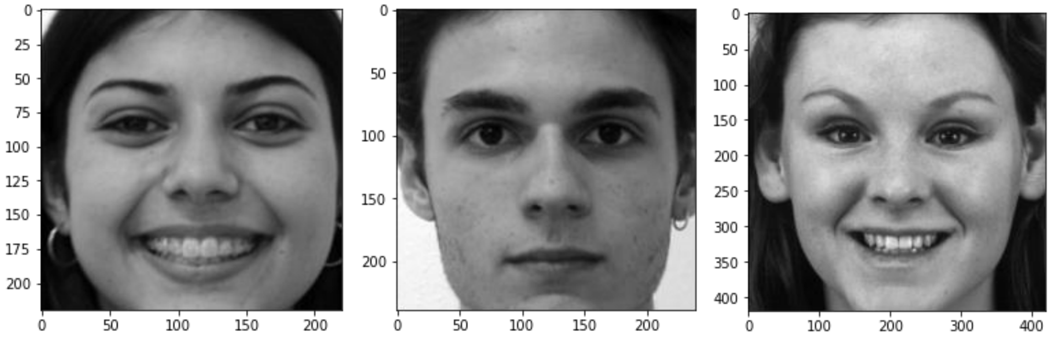


Figure 11 Grayscale Image

B) Local Binary Pattern

LBP is the feature extraction technique. The original LBP operator points the pixels of an image with decimal numbers, which are called LBPs or LBP codes that encode the local structure around each pixel. Each pixel is compared with its eight neighbours in a 3 X 3 neighbourhood by subtracting the centre pixel value. In the result, negative values are encoded with 0 and the others with 1. For each given pixel, a binary number is obtained by merging all these binary values in a clockwise direction, which starts from one of its top-left neighbours. The corresponding decimal value of the generated binary number is then used for labelling the given pixel. The derived binary numbers are referred to the LBPs or LBP codes.

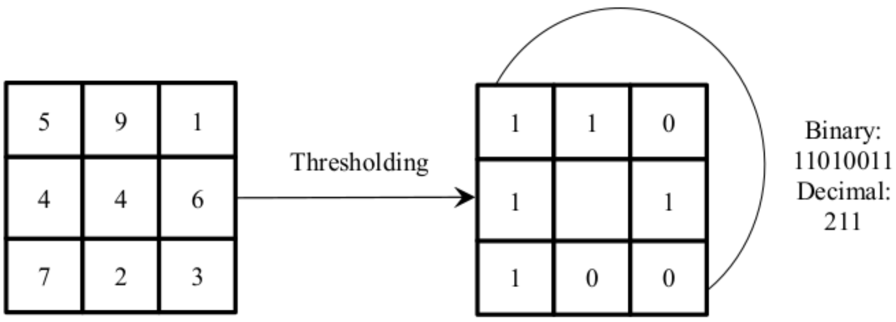


Figure 12 The Basic LBP Operator

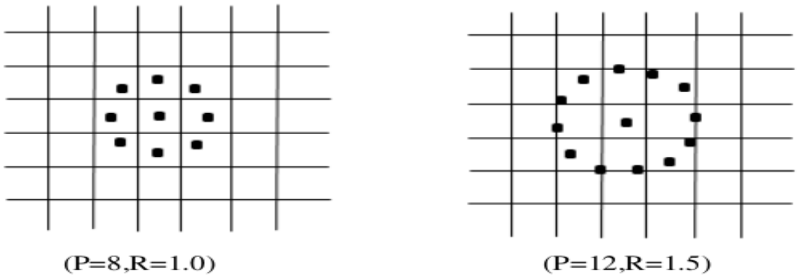


Figure 13 Two examples of extended LBP

The limitation of the basic LBP operator is that its small 3×3 neighbourhood cannot capture the dominant features of large-scale structures. As a result, to deal with the texture at different scales, the operator was later extended to use neighbourhoods of different sizes. Using circular neighbourhoods and bilinearly interpolating the pixel values allow any radius and number of pixels in the neighbourhood. Examples of the extended LBP are shown above (Figure 2), where (P, R) denotes sampling points on a circle of radius of R.

Further extension of LBP is to use uniform patterns. An LBP is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. E.g. 00000000, 001110000, and 11100001 are uniform patterns. A histogram of a labelled image f1(x, y) can be defined as

$$\mathrm{Hi} = \sum_{x,y} \mathrm{I}\ (f_l(x,y) = i), \quad i = 0, \ldots, n-1$$

Where n is the number of different labels produced by the LBP operator and

$$\mathrm{I}(A) = \begin{cases} 1 \text{ A is true} \\ 0 \text{ A is false} \end{cases}$$

This histogram contains information about the distribution of the local micro-patterns, such as edges, spots, and flat areas, over the whole image. For efficient face representation, the feature extracted should retain spatial information. Hence, the face image is divided into m small regions R0, R1,..., Rm, and a spatially enhanced histogram is defined as

$$\mathrm{Hi} = \sum_{x,y} \mathrm{I}\ (f_l(x,y) = i)\mathrm{I}((x,y) \epsilon R_j$$

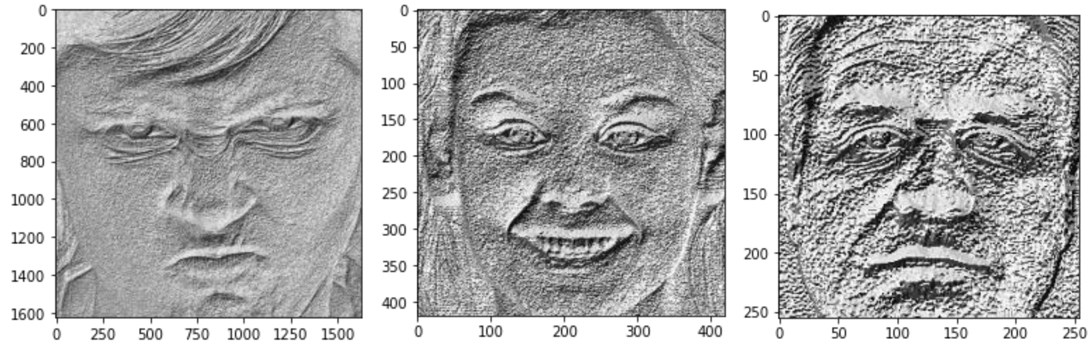


Figure 14 Images given by LBP

4.5.1.5 Emotion Classification

After the face is detected and the facial part is cropped and pre-processed then the facial feature is extracted. Then the extracted feature is fed to the machine learning algorithm to obtain the trained model to detect, recognize and classify the emotions that are present in the facial image. The classification is based on the feature that is given by grayscale pixel value as a feature and the image feature given by LBP. For emotion detection, recognition, and classification, we used three machine-learning algorithms: Support Vector Machine, Logistic Regression, and Random Forest Classifier. We trained all three algorithms and took the model that had the best accuracy.

1) SVM

SVM is a pattern recognition algorithm that is commonly used. SVM is a cutting-edge machine learning technique based on statistical learning theory. SVM is capable of achieving a near-optimal separation of classes. SVMs are trained to classify face expressions based on the features presented. SVMs are the maximal hyperplane classification method that relies on statistical learning theory results to provide strong generalisation performance.

Kernel functions are used to efficiently transfer non-linearly separable input data to a high-dimensional feature space where linear methods can be applied. SVMs are particularly well suited to a dynamic, interactive approach to expression recognition since they display good classification accuracy even when just a little quantity of training data is given.

When the hyperplane and the training data of any class are the largest, an optimal separation is attained. The decision surface is this dividing hyperplane. SVM has been used to solve a variety of classification problems, including text categorization, genetic analysis, and face detection. Given a training set of labelled samples:

$$D = \{(x_i , y_i \mid x_i \epsilon R^n, y_i \epsilon \{-1,1\}\}^p_{i=1}$$

A SVM tries to find a hyperplane to distinguish the samples with the smallest errors.

$$w . x - b = 0$$

For an input vector xi, the classification is achieved by computing the distance from the input vector to the hyperplane. The original SVM is a binary classifier.

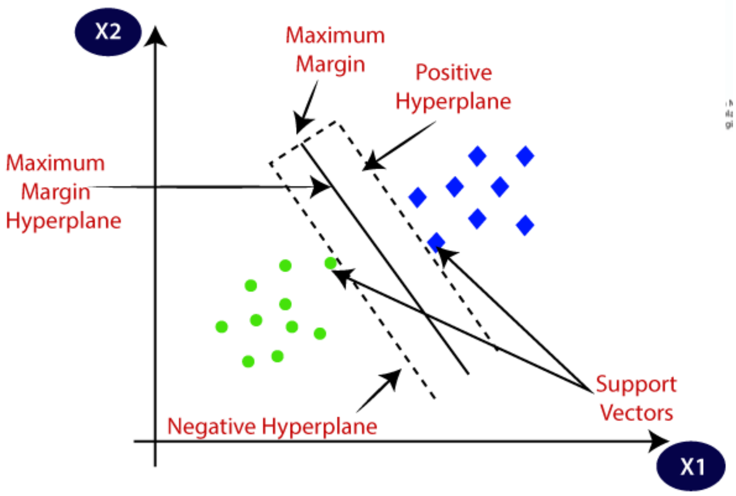


Figure 15 Graph for SVM

2) Logistic Regression

Logistic regression is another technique borrowed by machine learning from the field of machine learning. Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, or classify different classes, based on prior observations of a data set.

A logistic regression model analyzes the relationship between one or more existing independent variables to predict a dependent data variable. It's the method of choice for binary classification issues.

Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also called the sigmoid function is given by:

$$f(x) = 1/(1+e-x)$$

f(x) is the output of the logistic regression model and has the values in the range [0,1] where e is the base of the natural logarithms, x is the feature of the datasets. The graph for logistic regression is shown below:

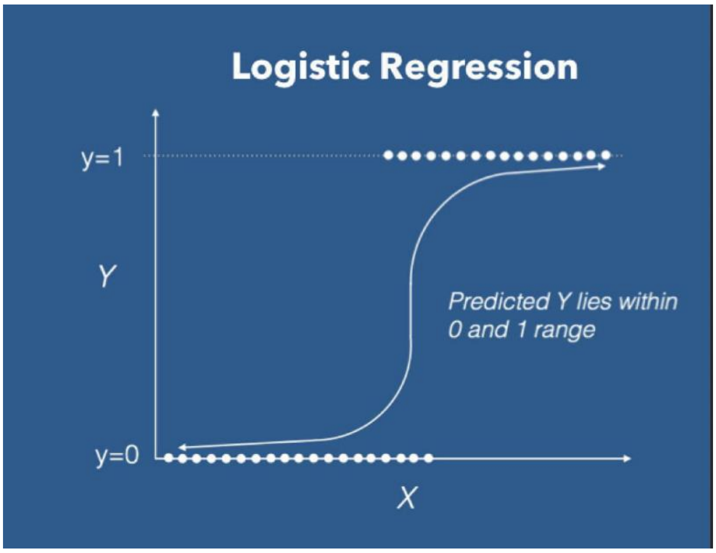


Figure 16 Graph for Logistic Regression

3) Random Forest Classifier

Random forest is a supervised learning algorithm. It produces a "forest" out of a collection of decision trees that are often trained using the "bagging" method. The main idea of the bagging method is that combining many learning models enhances total output. The random forest has the advantage of being able to address classification and regression problems, which are common in machine learning systems nowadays.

The bagging approach is used to train Random Forests. Bagging, also known as Bootstrap Aggregating, entails picking subsets of the training data at random, fitting a model to these smaller data sets, then aggregating the results. Random forest is trained to perform facial expression classification using the features proposed. The random forest adds more randomness to the model as it grows the trees. Instead of looking for the most important feature when dividing a node, it seeks the best feature from a random group of features. As a result, there is a great deal of variability, resulting in a better model.

Each new data point in the Random Forests algorithm goes through the same procedure as before, except this time it visits all of the trees in the ensemble, which was constructed using random samples of both training data and features. The aggregation functions utilized will vary depending on the task at hand. It employs the mode or most frequent class predicted by individual trees (also known as a majority vote) for classification issues, and the average prediction of each tree for regression tasks.

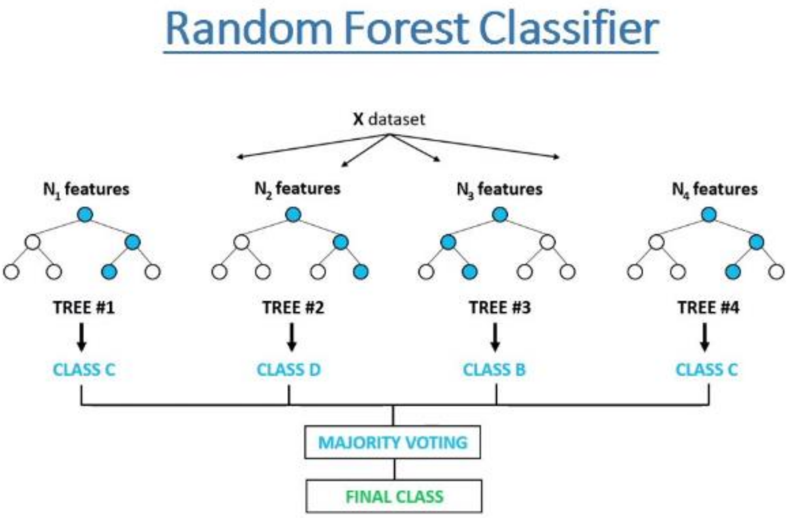


Figure 17 Graph for Random Forest Classifier

4.6 Music Recommendation Module

4.6.1 Song Database

We created a song database that includes Nepali, Hindi, and English songs. Each emotion is represented by 50 to 100 songs. Music, as we all know, plays an important role in elevating our mood. So, if a user is happy, the system will recommend a music playlist that is stored as happy music.

4.6.2 Music Playlist Recommendation

The emotion module detects the user's emotion in real-time. This will result in labels such as Happy, Sad, Angry, Surprise, Disgust, Fear, and Neutral. Using Python's os.listdir() method, we linked these labels to the folders of the songs database that we had created. The list of songs is shown in Table 2. This os.listdir() method returns a list of all files in the specified directories.

```
Self.playlist = []
if emotion == ‘Happy’:
     self.songlist = []
  directory            ='/home/gauri/Desktop/Project            Sample
Codes/Sample   1/Songs/Happy Song/'
     self.track['text'] = 'You seem {emotion}.'
     for root_, dirs, files in os.walk(directory):
           for file in files:
               if os.path.splitext(file)[1] == '.mp3':
                   path = (root_ + '/' + file).replace('\\', '/')
                   self.songlist.append(path)

       self.playlist = self.songlist
```

Table 2 Song Database

Emotion	Songs
Happy	Maya Gara Mayalule.mp3
	Mein tera.mp3
	Man Magan – Deepak Bajracharya.mp3
Sad	Euta Manche Ko Mayale Kati.mp3
	Parkhi Base Aaula Bhani - Narayan Gopal.mp3
	Narayan Gopal - JUN PHOOL MAILE 'जुन फूल मैले'.mp3
Angry	1.mp3
	2.mp3
Fear	1.mp3
	2.mp3
Disgust	1.mp3
	2.mp3
Surprise	1.mp3
	2.mp3
Neutral	HERDA RAMRO MACHHAPUCHHARE.mp3
	'Malashree Dhoon'.mp3

This will result in the user's recommended playlist being shown in the music player's GUI by showing captions based on the identified emotions. For playing the music, we used the Pygame library, which supports a variety of multimedia formats like audio, video, and so on. This library's functions for interfacing with the music player include playSong, pauseSong, nextSong, previousSong, and stopSong. The names of all songs, the state of presently active songs, and the main GUI window are stored in variables like playlist, songstatus, and root, respectively. Tkinter was used to create the user interface.

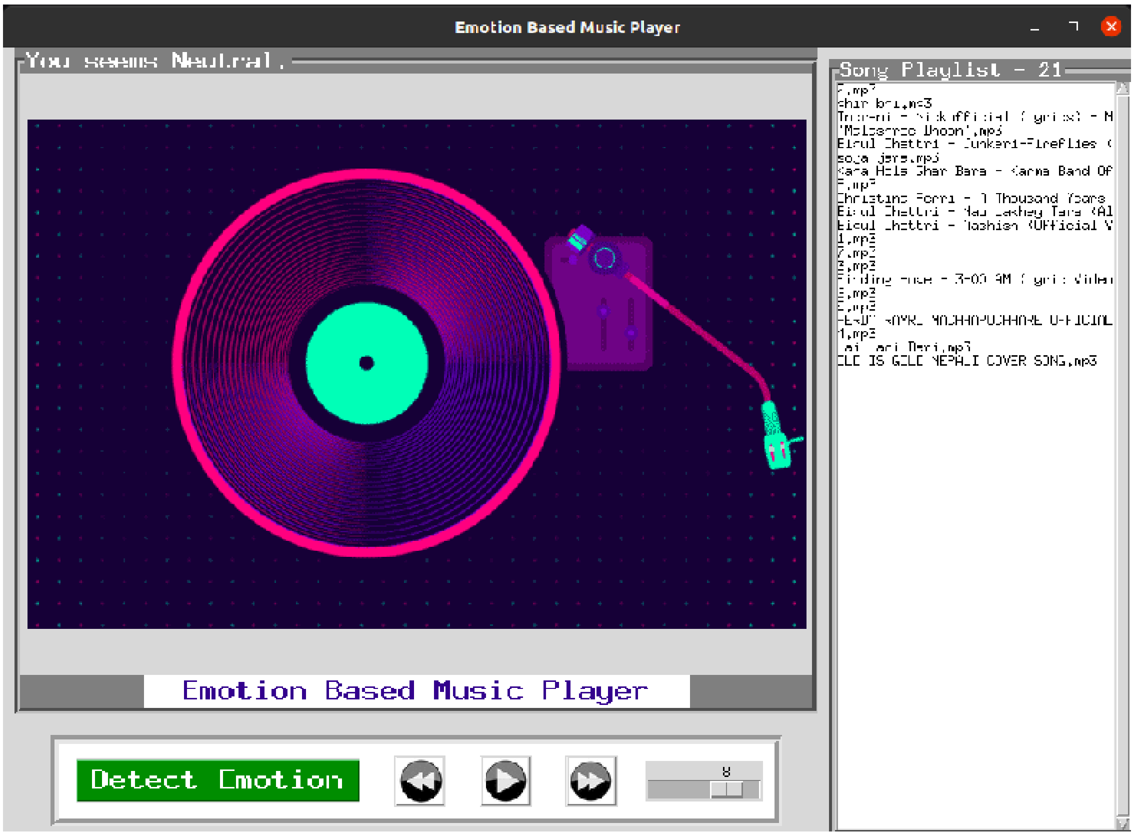


Figure 18 Music Playlist Recommendation

4.7 Hyperparameter Tuning

Hyperparameter tuning is a parameter that is supplied as an argument to the estimator classes constructor. Hyperparameters direct the model's parameter selection.

4.7.1 Hyperparameter for Logistic Regression

- C: inverse of regularization strength. Regularization is the process that constrains the size of the model coefficients. C is a floating point number; it's 1.0 by default and we increase the regularization by making the number smaller.
- Solver: Algorithm to use in the optimization problem. Default is 'lbfgs'. Other values are *{'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}*.

4.7.2 Hyperparameter for SVM

- C: inverse of regularization strength. Regularization is the process that constrains the size of the model coefficients. C is a floating point number; it's 1.0 by default and we increase the regularization by making the number smaller.
- Kernel: Specifies the kernel type to be used in the algorithm. If none is given, 'rbf' will be used. Other kernels are *{'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'}*.
- Gamma: Kernel coefficient for 'rbf', 'poly' and 'sigmoid'. Parameters are *{'scale', 'auto'}*.
- Probability: Whether to enable probability estimates. This must be enabled prior to calling fit, will slow down that method as it internally uses 5-fold cross-validation, and predict_proba may be inconsistent with predict.

4.7.3 Hyperparameter for Random Forest

- n_estimators: The number of trees in the forest. Default value is 100.

4.8 System Evaluation

Evaluation of the system can be done using the following methods:

a) Precision

Precision evaluates the algorithm's predictive capacity by estimating the predictive value of a label, which can be positive or negative depending on the class for which it is derived. The percentage of correctly assigned expressions in relation to the total number of aspects is known as precision.

precision = tp / (tp+fp)

Where,

tp = true positive

fp = false positive

b) Recall

Recall is a function of both successfully classified (true positives) and misclassified (false positives) samples (false negatives). The fraction of correctly assigned expressions to the total number of expressions is known as recall.

recall = tp / (tp+fn)

Where,

tp = true positive

fp = false negative

c) F-score:

The F-score is a composite metric that rewards algorithms with better sensitivity while challenging those with higher specificity. When $\beta = 1$, the F-score is evenly balanced. When $\beta > 1$, it favours precision; otherwise, it favours recall.

$$\text{F-measure} = ((2+1)*\text{precision}*\text{recall})/(2*\text{precision}*\text{recall})$$

All three measures distinguish between correct label classification within various classes. They concentrate on a single subject (positive examples). As a result, accuracy and recall assess different properties, and we require a combined quality measure to figure out which aspect of expression category mappings is the best match. The F-measure (fm) calculates the harmonic mean of precision and recall and allows both attributes to be considered at the same time. It's worth noting that the total recall, or recallov, is also referred to as accuracy.

CHAPTER 5

5 DEVELOPMENT AND TESTING

5.1 Implementation Tools

5.1.1 Programming Language and Libraries

1) Python

Python is a powerful scripting language and is very useful for solving statistical problems involving machine learning algorithms. It has various utility functions which help in pre-processing. Processing is fast and it is supported on almost all platforms. Integration with C++ and other image libraries is very easy, and it has in-built functions and libraries to store and manipulate data of all types. It provides the pandas and NumPy framework which helps in the manipulation of data as per our needs. A good feature set can be created using the NumPy arrays which can have n-dimensional data.

2) Scikit-learn

Scikit-learn is the machine learning library in python. It includes matplotlib, NumPy, and a variety of machine learning algorithms. The API is very easy to use and understand. It has many functions to analyse and plot the data. A good feature set can be formed using many of its feature reduction, feature importance, and feature selection functions. The algorithm it provides can be used for classification and regression problems and their subtypes.

3) Numpy

NumPy is a popular Python library for processing large multi-dimensional arrays and matrices using a large collection of high-level mathematical functions. It is extremely useful for basic scientific computations in Machine Learning.

4) Pandas

Pandas is a well-known Python data analysis toolkit. It has nothing to do with Machine Learning. The dataset must be prepared before the training, as we all know. Pandas come in helpful in his scenario because it was designed expressly for data extraction and preparation.

5) Matplotlib

Matplotlib is a well-known Python data visualisation package. It's very useful when a coder needs to see how data patterns are represented. It's a 2D charting package that allows you to make 2D graphs and charts.

6) OpenCV

It is the library we will be using for image transformation functions such as converting the image to grayscale. It is an open-source library that can be used for many image functions and has a wide variety of algorithm implementations. C++ and Python are the languages supported by OpenCV. It is a complete package that can be used with other libraries to form a pipeline for any image extraction or detection framework. The range of functions it supports is enormous, and it also includes algorithms to extract feature descriptors.

5.1.2 DEVELOPMENT ENVIRONMENT

1) Jupyter Notebook

Jupyter Notebook is the integrated development environment (IDE) for combining Python with all of the libraries we'll be using in our solution. Although certain complex computations take longer to execute, it is interactive. Plots and photos appear in real-time. It may be utilised as a one-stop-shop for all of our needs, and most libraries, such as OpenCV and Scikit-learn, can be simply incorporated.

2) Google Colab

Google Colab, also known as Collaboratory, is a Jupyter environment that works with CPUs, GPUs, and even TPUs, and is provided and supported by Google. It's just like any other Jupyter notebook, with the ability to code in Python and write explanations in Markdown, as well as all of the other Jupyter features. Google Colab is a product that emphasises collaboration. It's also hosted on Google's servers, so there's no need to download anything. The notebooks are also preserved in your Google Drive account.

3) Visual Studio Code

Microsoft's Visual Studio Code is a source code editor for Windows, Linux, and macOS. Debugging, embedded Git control, GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring are all supported.

5.2 System Testing

Different training and testing datasets were used to test the system. This test was carried out to see if the system could accurately predict the outcome. Throughout the development phase of the system, our system was thoroughly tested. The following is a list of the tests that were carried out:

1) Unit testing

We designed the entire system in a modularized form for unit testing, and each module was tested separately. We worked on the same module until we got accurate output from each separate module.

2) Integration Testing

Following the completion of separate modules, the modules were integrated to form a full system. The system was then put to the test to see if the prediction made by the training dataset for the testing set was right. We attempted to get the highest level of accuracy possible. Our system's average accuracy was enhanced after a couple of days of integration testing.

i) Alpha Testing

Alpha testing is the first stage of software engineering which is considered as a simulated or actual operational testing done by the individual member of the project. Alpha testing is conducted by the project developers, in the context of our project.

ii) Beta Testing

Beta testing comes continuously after alpha testing which is considered a form of external user acceptance testing. The beta version of the program is developed and provided to a limited audience. This is the final test process in the case of this project. In this system, the beta-testing is done by our colleagues and the project supervisor.

CHAPTER 6

6 EPILOGUE

6.1 Experiments, Results and Outputs

This section gives the details of the experiments performed, their results and the obtained output of the whole project. Our proposed system consists of two modules: Facial Emotion Detection Module and Music Recommendation Module. Emotion detection module focuses on training and testing the model and selecting the best model for real time emotion detection. Music recommendation module focuses on song database and music playlist generation and recommendation based on the emotion detected by the emotion detection module.

6.1.1 Experiment on Facial Emotion Detection Module

For facial emotion detection, we have used three datasets FER-2013, CK+48 and a self-created dataset. We used three machine learning algorithms SVM, Logistic Regression and Random Forest to train our model and select the best model for our project. The aim of this module is to develop a complete facial emotion recognition system for recommending music playlists. First of all, the system was trained using different random samples in each dataset by supervised learning. In each datasets, the data were partitioned into two parts for training and testing where 80% for training and 20% for test. For betterment and increasing the accuracy of the model, we have tuned hyperparameters for each algorithm.

6.1.1.1 Experiments Based on CK+48 Dataset

There are altogether 1127 images in this dataset. All the images in the dataset were resized to 48x48 pixels and the image features were grayscale pixel values and the feature extracted by LBP. Then the machine learning classifier algorithms such as SVM, Random Forest and Logistic Regression were trained. The parameters were defined through a grid search and 5-fold cross-validation was used to improve algorithm performance. The accuracy of the experiments with different features are shown in below tables.

1) Grayscale pixel values as features

It is the simplest way to create features from an image and use these raw pixel values as a separate feature. The model, accuracy and best hyperparameter using grayscale pixel values as features are presented below:

Table 3 Model, accuracy score and hyperparameters

	Model	best_score	best_params
0	SVM	0.954159	{‘C’:100, ‘kernel’ :’ rbf’}
1	Random_forest	0.866742	{‘n_estimators’ :10}
2	logistic_regression	0.941737	{‘C’: 5}

As from the above table, SVM has the best accuracy score. So, we have taken the SVM. Here, the best parameters for SVM are C=100, kernel = rbf.

Table 4 Accuracy of CK+48 using SVM

Evaluation Types	Result Percentages
Precision	98%
Recall	98%
F1-score	98%
Accuracy	98%

The above table shows that 98% of the expressions were predicted and 98% of the expressions were correctly assigned. The harmonic mean of precision and recall was 98%.

The confusion matrix is shown below:

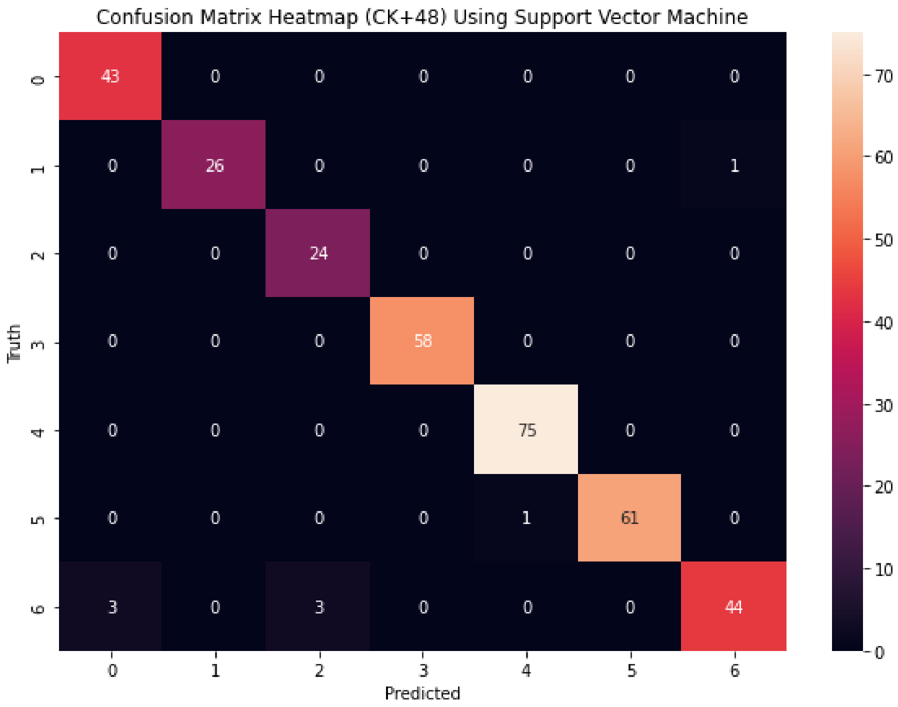


Figure 19 Confusion Matrix Heatmap using SVM

Here, 0 = Disgust, 1 = Fear, 2 = Sad, 3 = Happy, 4 = Surprise, 5 = Neutral, 6 = Angry

2) Image Feature extracted from LBP

The model, accuracy and best hyperparameter using image feature extracted using LBP are presented below:

Table 5 Model, accuracy score and hyperparameters

	Model	best_score	best_params
0	SVM	0.718681	{‘C’:10, ‘kernel’: ‘rbf’}
1	Random_forest	0.372511	{‘n_estimators’:10}
2	logistic_regression	0.619659	{‘C’:1}

As from the above table, SVM has the best accuracy score. So, we have taken the SVM. Here, the best parameters for SVM are C=10, kernel = rbf.

Table 6 Accuracy of CK+48 using SVM

Evaluation Types	Result Percentages
Precision	73%
Recall	71%
F1-score	72%
Accuracy	71%

The above table shows that 73% of the expressions were predicted and 71% of the expressions were correctly assigned. The harmonic mean of precision and recall was 72%.

The confusion matrix is shown below:

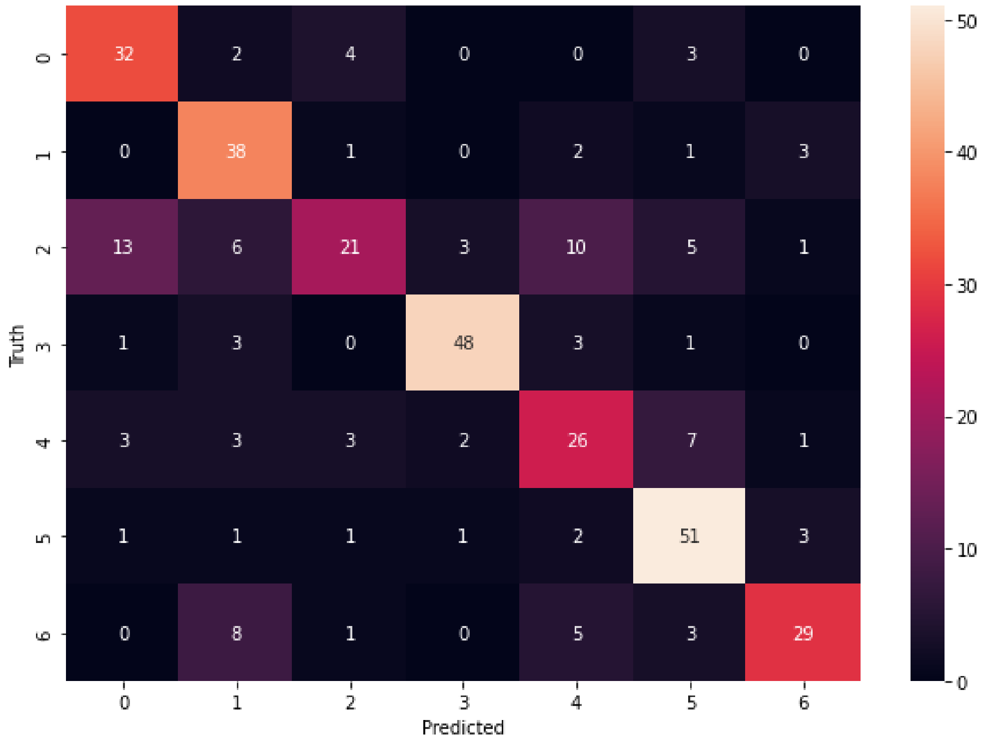


Figure 20 Confusion matrix (CK+48) using Linear Binary Pattern

Here, 0 = Disgust, 1 = Fear, 2 = Sad, 3 = Happy, 4 = Surprise, 5 = Neutral, 6 = Angry

6.1.1.2 Experiment Based on FER-2013 dataset

This dataset is very imbalanced so we randomly took images from each emotion to make it balanced. We randomly took about 500 images from each emotion class and 80% are used for training and 20% are used for testing. Each image is resized to 48x48 pixels and converted to grayscale.

1) Grayscale pixel values as features

The model, accuracy and best hyperparameter using grayscale pixel values as features are presented below:

Table 7 Model, accuracy score and hyperparameters

	Model	best_score	best_params
0	SVM	0.4281	{‘C’:100, ‘kernel’: ‘rbf’}
1	Random_forest	0.3260	{‘n_estimators’:10}
2	logistic_regression	0.3025	{‘C’:1}

As from the above table, SVM has the best accuracy score. So, we have taken the SVM. Here, the best parameters for SVM are C=100, kernel = rbf.

Table 8 Accuracy of FER-2013 using SVM

Evaluation Types	Result Percentages
Precision	42%
Recall	43%
F1-score	42%
Accuracy	43%

The above table shows that 42% of the expressions were predicted and 43% of the expressions were correctly assigned. The harmonic mean of precision and recall was 42%.

The confusion matrix is shown below:

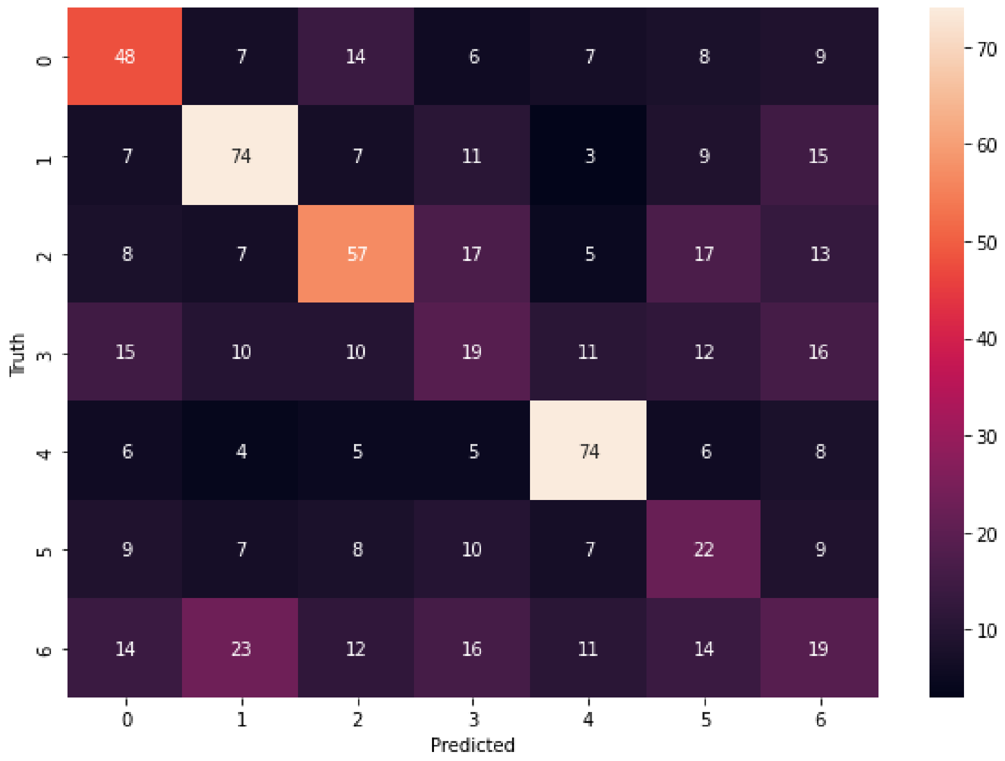


Figure 21 Confusion Matrix Heatmap using SVM

Here, 0 = Sad, 1 = Surprise, 2 = Happy, 3 = Angry, 4 = Disgust, 5 = Neutral, 6 = Fear

2) Image Feature extracted from LBP

The model, accuracy and best hyperparameter using image feature extracted using LBP are presented below:

Table 9 Model, accuracy score and hyperparameters

	Model	best_score	best_params
0	SVM	0.3611	{'C':100, 'kernel': 'rbf'}
1	Random_forest	0.2558	{'n_estimators':10}
2	logistic_regression	0.2804	{'C':1}

As from the above table, SVM has the best accuracy score. So, we have taken the SVM. Here, the best parameters for SVM are C=100, kernel = rbf.

Table 10 Accuracy of FER-2013 using SVM

Evaluation Types	Result Percentages
Precision	36%
Recall	37%
F1-score	36%
Accuracy	36%

The above table shows that 36% of the expressions were predicted and 37% of the expressions were correctly assigned. The harmonic mean of precision and recall was 36%.

The confusion matrix is shown below:

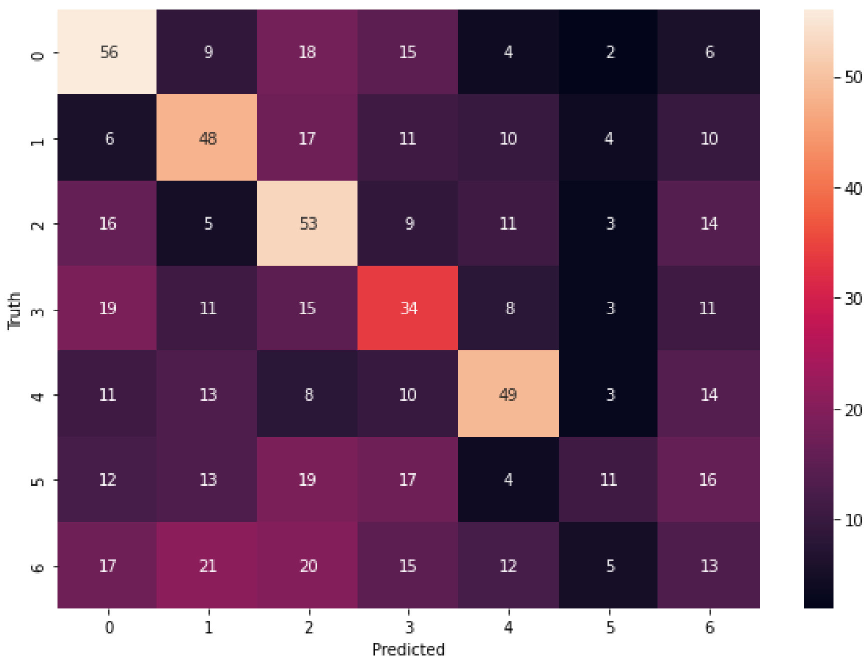


Figure 22 Confusion matrix using Linear Binary Pattern

Here, 0 = Sad, 1 = Surprise, 2 = Happy, 3 = Angry, 4 = Disgust, 5 = Neutral, 6 = Fear

6.1.1.3 Experiment Based on self-created dataset

This dataset is created by ourselves where we have downloaded images from google, facebook, etc., requested images from our friends and requested database access to many facial databases. The images are cropped in the facial part and resized, grayscaled and feature extracted. In this dataset also, we used 80% of the images for training and 20% of the images for testing.

1) Grayscale pixel values as features

The model, accuracy and best hyperparameter using grayscale pixel values as features are presented below:

Table 11 Model, accuracy score and hyperparameters

	Model	best_score	best_params
0	SVM	0.7769	{‘C’: 10, ‘kernel’: ‘rbf}
1	Random_forest	0.5965	{‘n_estimators’: 10}
2	logistic_regression	0.69557	{‘C’: 1}

As from the above table, SVM has the best accuracy score. So, we have taken the SVM. Here, the best parameters for SVM are C=10, kernel = rbf.

Table 12 Accuracy of self-created using SVM

Evaluation Types	Result Percentages
Precision	77%
Recall	77%
F1-score	77%
Accuracy	77%

The above table shows that 77% of the expressions were predicted and 77% of the expressions were correctly assigned. The harmonic mean of precision and recall was 77%.

The confusion matrix is shown below:

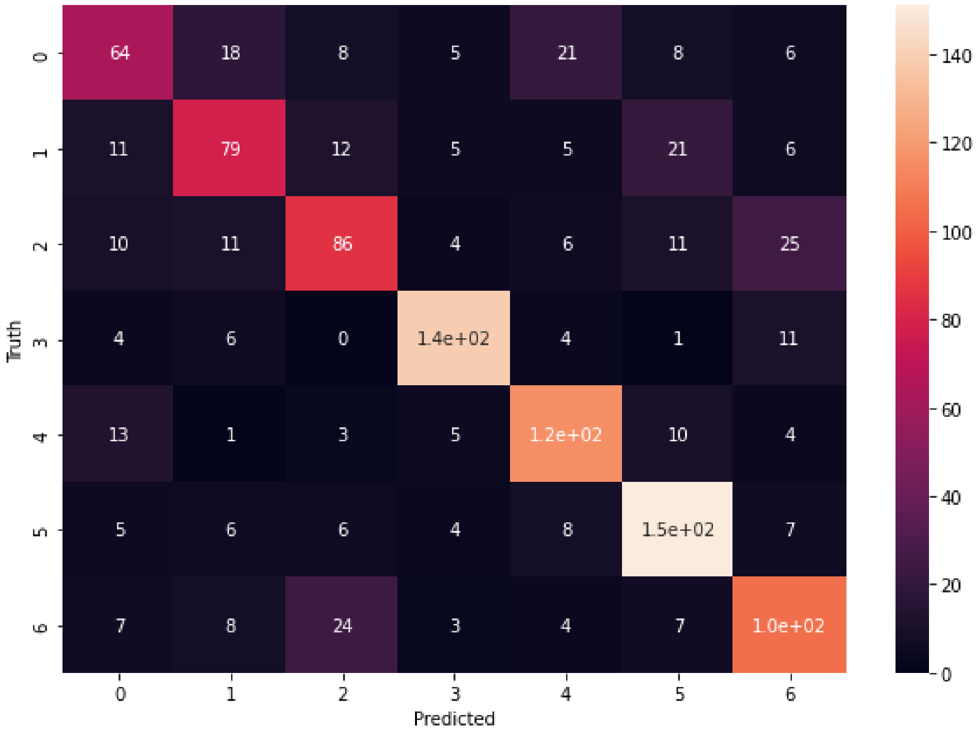


Figure 23 Confusion Matrix Heatmap using SVM

Here, 0 = Fear, 1 = Sad, 2 = Angry, 3 = Happy, 4 = Surprise, 5 = Neutral, 6 = Disgust

2) Image Feature extracted from LBP

The model, accuracy and best hyperparameter using image feature extracted using LBP are presented below:

Table 13 Model, accuracy score and hyperparameters

	Model	best_score	best_params
0	SVM	0.7269	{'C':10, 'kernel': 'rbf'}
1	Random_forest	0.6255	{'n_estimators':10}
2	logistic_regression	0.4965	{'C':1}

As from the above table, SVM has the best accuracy score. So, we have taken the SVM. Here, the best parameters for SVM are C=10, kernel = rbf.

Table 14 Accuracy of self-created using SVM

Evaluation Types	Result Percentages
Precision	73%
Recall	73%
F1-score	73%
Accuracy	73%

The above table shows that 73% of the expressions were predicted and 73% of the expressions were correctly assigned. The harmonic mean of precision and recall was 73%.

The confusion matrix is shown below:

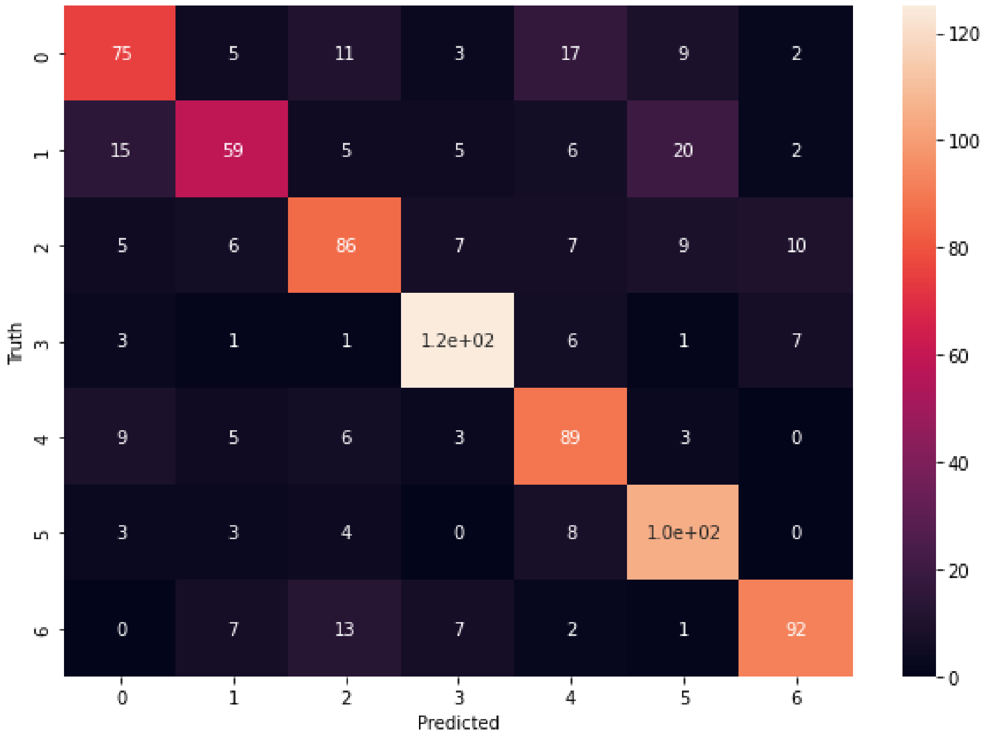


Figure 24 Confusion matrix using Linear Binary Pattern

Here, 0 = Fear, 1 = Sad, 2 = Angry, 3 = Happy, 4 = Surprise, 5 = Neutral, 6 = Disgust

6.1.2 Project Outputs

As our proposed system is Music Recommendation Based On Facial Emotion Recognition, we built a GUI application/interface for the music player. In the interface, there are three sections:

- Main Section where emotion name and music name are shown.
- Playlist Section where recommended music playlists are shown.
- Button Section where Detect Emotion and music control button are shown.

The GUI application of our project is shown below:

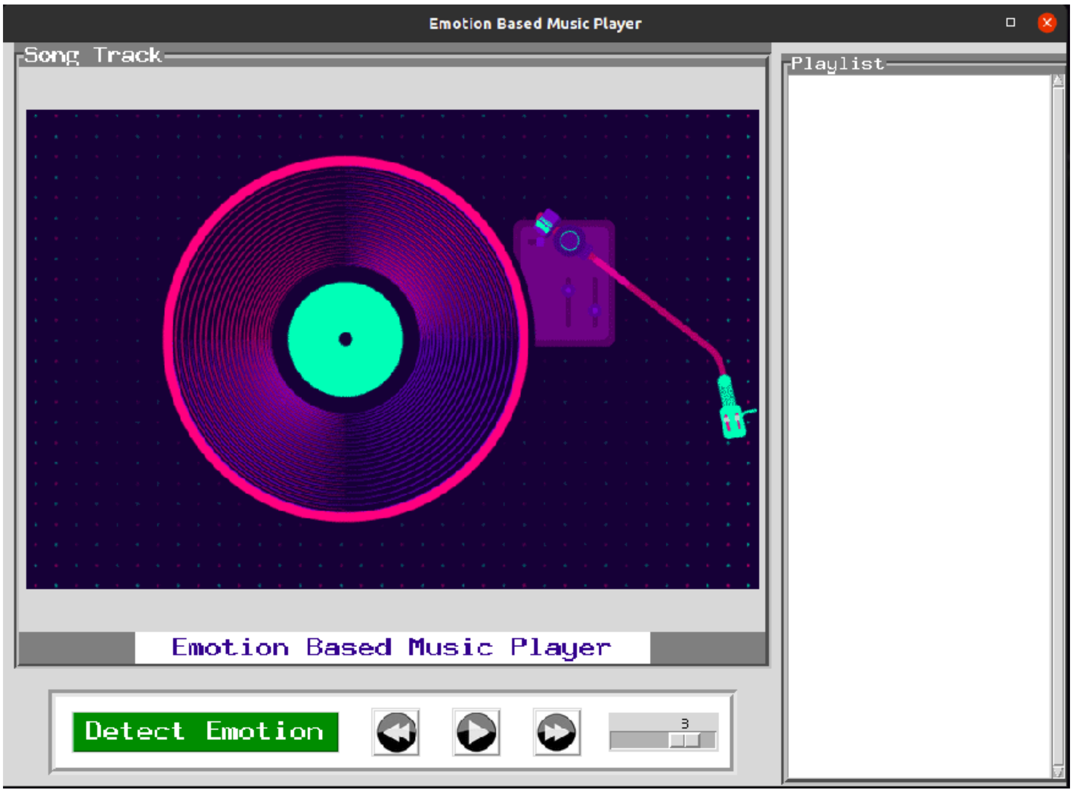


Figure 25 Music Player GUI

6.1.2.1 Emotion Detection Module Output

When we click the “Detect Emotion” button present in the Button Section, the webcam is opened entitled as “Face Emotion Recognition”. The output of the emotion detection module is shown below:

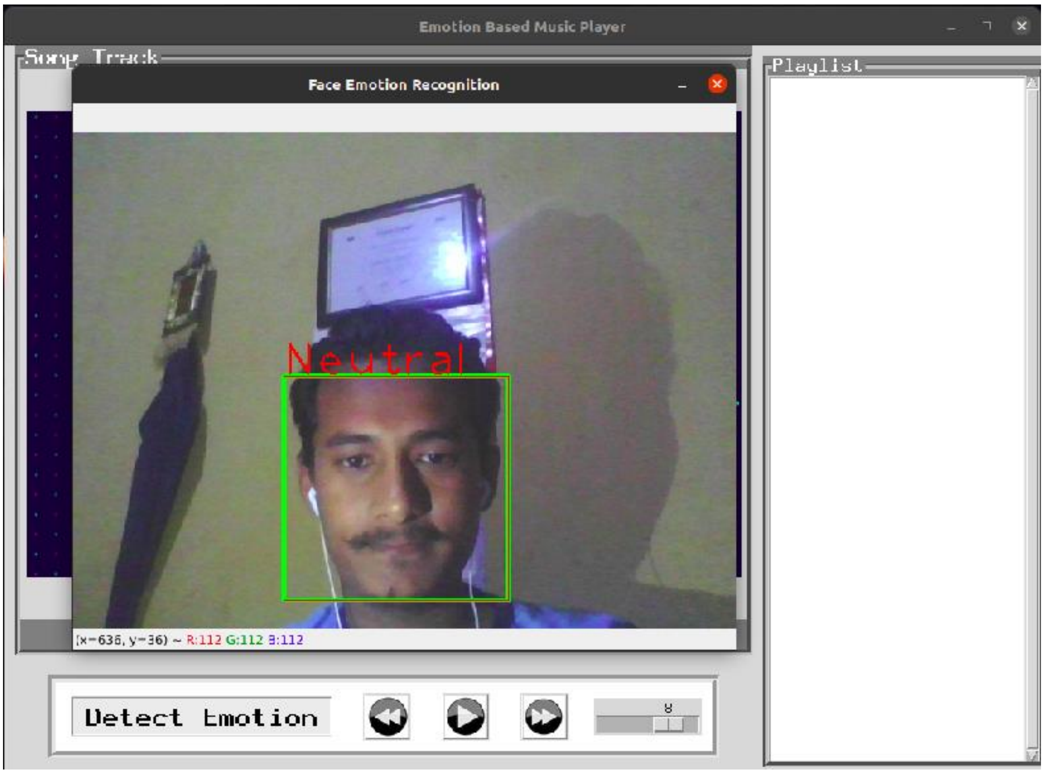


Figure 26 Emotion Recognition through Webcam

6.1.2.2 Music Recommendation Module Output

This module generates the music playlist according to the emotion detected by the emotion detection module. The output of this module is shown below:

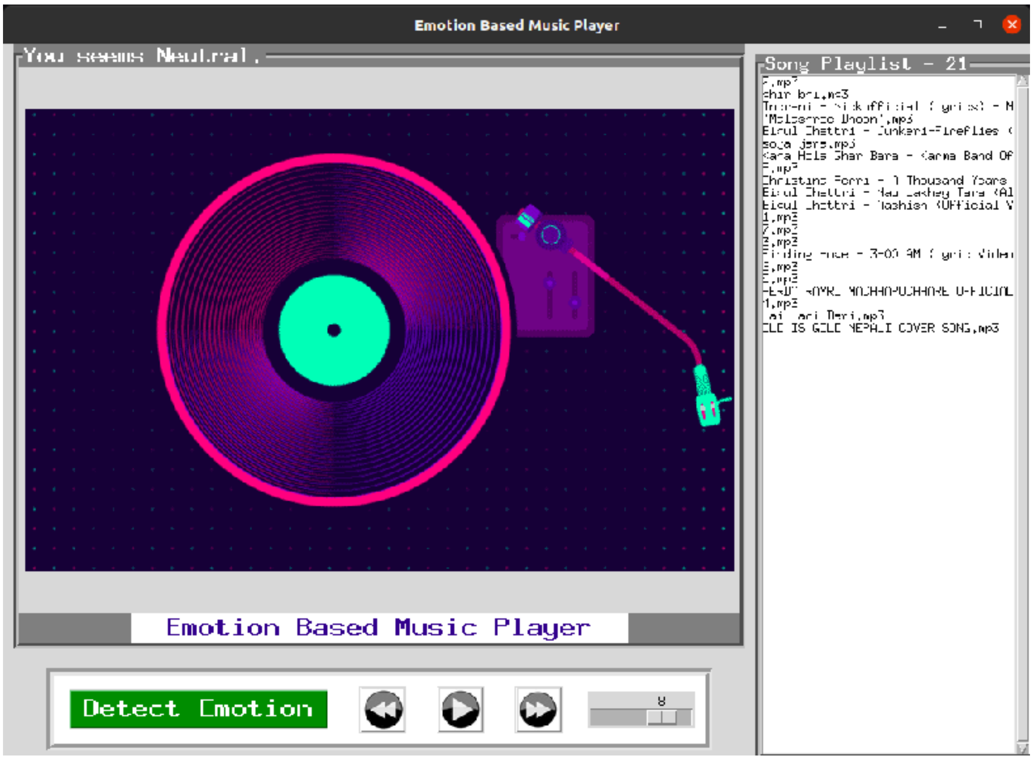


Figure 27 GUI application recommending music playlist

6.2 Discussion and Future Enhancement

As our proposed system is to built Music Recommendation Based on Facial Emotion Recognition, for this we built a GUI application that functions as a music player. The webcam can be accessed after clicking the button available there. The music playlist can be generated after the facial emotion is recognized. We have built the trained model using three machine learning algorithms SVM, Logistic Regression, Random Forest and used three datasets FER-2013, CK+48 and self-created dataset. As from the above experiments and results, the best accuracy is given by SVM in CK+48 but it doesn't give good accuracy in real time emotion detection. For real time emotion detection, the model trained by using SVM and self-created dataset gives the best accuracy and desired output. So, we used this model for emotion detection in our system.

6.2.1 Limitations of Our Project

It's very difficult to develop a project without any limitations. Since our project being developed under certain budget and time frame criteria, there are some limitations listed below:

1) There is no user login and registration functionality hence there is no user preferences.
2) User can't upload their songs to the playlist. The playlist is predefined and static.
3) There are not many songs in the song database and currently our song database is the local device file system.
4) Sometimes, some emotions like sad and fear can't be detected by our emotion detection module.
5) The system can't perform well in extremely bad light conditions and poor camera resolution.

6.2.2 Future Enhancement

1) We will endeavor to add a huge number of songs in the future, which will strengthen the recommendation and provide customers with even better playlists.
2) We'll use CNN for emotion recognition as well, in order to compare findings and improve them.
3) We will include login, registration and user rating functionality in the future, allowing users to set their preferences.
4) The current system does not perform well in extremely bad light conditions and poor camera resolution thereby provides an opportunity to add some functionality as a solution in the future.
5) For a better user experience, we intend to transform this GUI into a mobile application in the future.

6.3 Conclusion

In this project, we propose a music recommendation system based on user facial emotions. The human face is used as an input, and facial emotion is identified, with music being played automatically based on the emotions. We harnessed the capability of machine learning algorithms to determine the user's emotion. SVM, logistic regression and random forest algorithm were mainly used as a machine learning algorithm for detection of the facial expression of the user. Out of these three algorithms, SVM provided the best result with highest accuracy for all the training set. As a result, SVM was mostly considered.

Another component was the creation of a music playlist system that provided a playlist of songs based on the identified emotion. Songs are saved in a database based on emotion categories. A playlist of the song is retrieved from the database and presented to the user based on the identified facial emotion.

A GUI system was constructed using the Tkinter Python package to provide users with a basic and accessible interface. This graphical user interface (GUI) application allowed users to play, pause, and change the song according to their preferences. Pygame, a Python library, was used to provide the functionality provided by the built-in GUI application. In the following way, music recommendation based on facial emotion recognition system was realized.

7 REFERENCE

1. Anagha S.Dhavalikar and Dr. R. K. Kulkarni, “Face Detection and Facial Expression Recognition System” 2014 International Conference on Electronics and Communication System (ICECS -2014).

2. Yong-Hwan Lee, Woori Han and Youngseop Kim, “Emotional Recognition from Facial Expression Analysis using Bezier Curve Fitting” 2013 16th International Conference on Network-Based Information Systems.

3. A. Lehtiniemi and J. Holm, "Using Animated Mood Pictures in Music Recommendation," 2012 16th Conference on Information Visualisation, Montpellier, 2012, pp. 143-150.

4. Deger Ayata, Yusuf Yaslan, and Mustafa E. Kamasak, Emotion-based music recommendation system using wearable physiological sensors, IEEE transactions on consumer electronics, vol. 14, no. 8, May 2018.

5. AYUSH Guidel, Birat Sapkota, Krishna Sapkota, Music recommendation by facial analysis, February 17, 2020.

6. CH. Sadhvika, Gutta. Abigna, P. Srinivas Reddy, Emotion-based music recommendation system, Sreenidhi Institute of Science and Technology, Yamnampet, Hyderabad; International Journal of Emerging Technologies and Innovative Research (JETIR) Volume 7,Issue 4, April 2020.

7. M. Athavle, D.Mudale, U. Shrivastav and M. Gupta (2021) "Music Recommendation Based on Face Emotion Recognition", *Journal of Informatics Electrical and Electronics Engineering (JIEEE)*, vol. 2, no. 2, pp. 1-11, 2021.

8. D. Parikh, "Emotion Based Music Recommendation System", *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 10, pp. 1674-1680, 2021.

9. D. Samuvel, B. Perumal and M. Elangovan, "Music recommendation system based on facial emotion recognition", *3C Tecnología Glosas de innovación aplicadas a la pyme*, pp. 261-271, 2020.

10. P. Michel and R. Kaliouby, "Facial Expression Recognition Using Support Vector Machines", *Computer Laboratory*, vol. 2, no. 1, 2018.

11. G. CHIDAMBARAM, A. DHANUSH RAM, G. KIRAN, P. KARTHIC and A. KAIYUM, "MUSIC RECOMMENDATION SYSTEM USING EMOTION RECOGNITION", *International Research Journal of Engineering and Technology (IRJET)*, vol. 08, no. 07, 2016.

12. Chen, J., Chen, Z., Chi, Z., & Fu, H. (2014, August). Facial expression recognition based on facial components detection and hog features. International Workshops on Electrical and Computer Engineering Subfields (pp.884-888).

13. James, H. Immanuel, J. James Anto Arnold, J. Maria MasillaRuban, M. Tamilarasan, and R. Saranya. "EMOTION BASED MUSIC RECOMMENDATION SYSTEM." EMOTION 6, no. 03 (2019).