CONTENTS

1.INTRODUCTION

Music plays an important role in our daily life. Music has always known to alter the mood of a person. It is often confusing for a person to decide which songs to listen from a massive collection of songs on the Internet. While music genre plays a huge role in building and displaying social identity, the emotion expression of a song and even more importantly its emotional impression on the listener is often under-estimated in the domain of music preferences. Capturing and recognizing the emotion shown by a person and recommending suitable songs matching one's mood and increasingly calm the mind of a user. People often face a tough time in creating playlists manually when they have a lot of songs. It is also difficult to keep track of all songs. People tend to express their emotions, mainly by their facial expressions. Music has always been known to alter the mood of an individual. Capturing and recognizing the emotion being voiced by a person and displaying appropriate songs matching the one's mood and can increasingly calm the mind of a user and overall end up giving a pleasing effect. The project aims to capture the emotion expressed by a person through facial expressions. Two of the most important functions of music are it is ability is participants rated to help them achieve a good mood and become more self-aware. A music player is designed to capture mortal emotion through the web camera interface available on computing systems. The software captures the image of the user and then with the help of image segmentation and image processing techniques extracts features from the face of a target human being and tries to detect the emotion that the person is trying to express. The project aims to lighten the mood of the user, by playing songs that match the requirements of the user by capturing the image of the user. Since ancient times the best form of expression analysis known to humankind is facial expression recognition. The best possible way in which people tend to dissect or conclude the emotion or the feeling or the studies that another person is trying to express is by facial expression.

2. PROJECT REQUIREMENT AND SPECIFICATIONS

2.1. REQUIREMENT ANALYSIS

2.1.1. Proposed System:

Our model requires the use of a webcam and MATLAB to perform facial recognition. The recent model of the face based on the deep learning detection models, compared to the existing model it not only shortens our time but the accuracy is effectively improved comparing to the existing model.

2.1.2. Scope:

- Music Streaming Platforms: Emotion-based music recommendation system can be integrated into music streaming platforms like Spotify.
- Mental Health: Music has been shown to have significant impact on mental health.
- Advertising: Emotion-based music recommendation systems can be used in advertising to create more effective campaigns.
- Education: Emotion-based music recommendation systems can be used in education settings to enhance learning. The system can recommend music that can help students focus, relax, or stimulate creativity, depending on the learning objective.
- Healthcare: Music is also used in healthcare settings to help patients manage pain, reduce stress, and improve overall well-being.

An emotion-based music recommendation system has wide scope and can be used in various domains to personalize the user experience, enhance learning, improve mental health, create more effective advertising campaigns, and support healthcare goals.

2.2. FEASIBILITY STUDY

2.2.1. Technical Feasibility:

The technical feasibility of an emotion-based music recommendation system depends on the ability to accurately identify and categorize emotions in music. This requires a combination of advanced machine learning algorithms, natural language processing

techniques, and data analysis capabilities. It may also require the integration of various technologies such as natural language processing and computer vision.

2.2.2. Economic Feasibility:

The economic feasibility of the system would involve assessing the costs and benefits of developing and implementing the system. This would include factors such as the cost of acquiring and analyzing data, the cost of developing and maintaining the system, and the potential revenue generated from the system.

2.2.3. Operational Feasibility:

The operational feasibility of the system would involve assessing the practicality of implementing the system. This would include factors such as the availability of the necessary resources, the training required for system operators, and the ease of use of the system for users. Firstly, emotion recognition technology has improved significantly over the years and there are several approaches to detect emotions including facial expression analysis, voice analysis, and physiological signal analysis. Secondly, the accuracy of the recommendation algorithms depends on the size and quality of the music database. The larger and more diverse the music collection, the more accurate the recommendations are likely to be. However, assembling a comprehensive music database can be challenging and time-consuming. Thirdly, the effectiveness of the recommendation algorithms depends on how well they are designed and implemented. Recommendation algorithms can be simple, rule-based systems or more sophisticated machine learning-based models. The choice of algorithm will depend on the complexity of the task and the resources available.

It may also face challenges in terms of data availability, accuracy, and user adoption. However, if successfully implemented, such a system could have a significant impact on the music industry and improve the listening experience for users.

2.3. HARDWARE AND SOFTWARE REQUIREMENTS

2.3.1. Hardware Requirements:

- Processor: Intel core

2.3.2. Software Requirements:

- Operating system: Windows

2.3.3. Front end and Back end:

- Front end: HTML, CSS
- Back end: PHP, SQL, Python

3.SYSTEM ANALYSIS

3.1. Screen Design

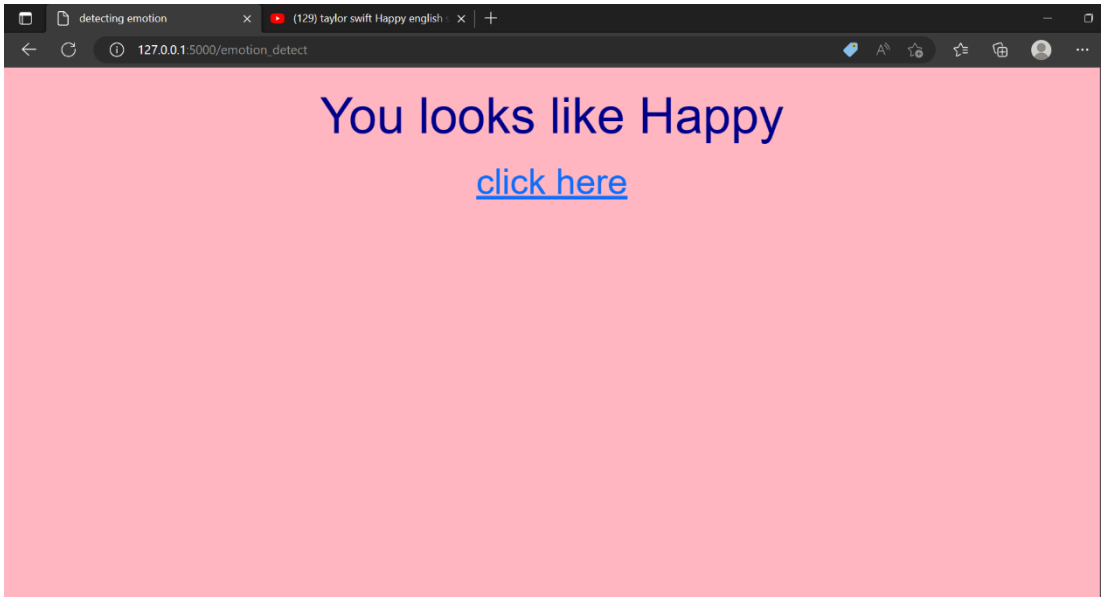


Fig.3.1. Emotion Detection

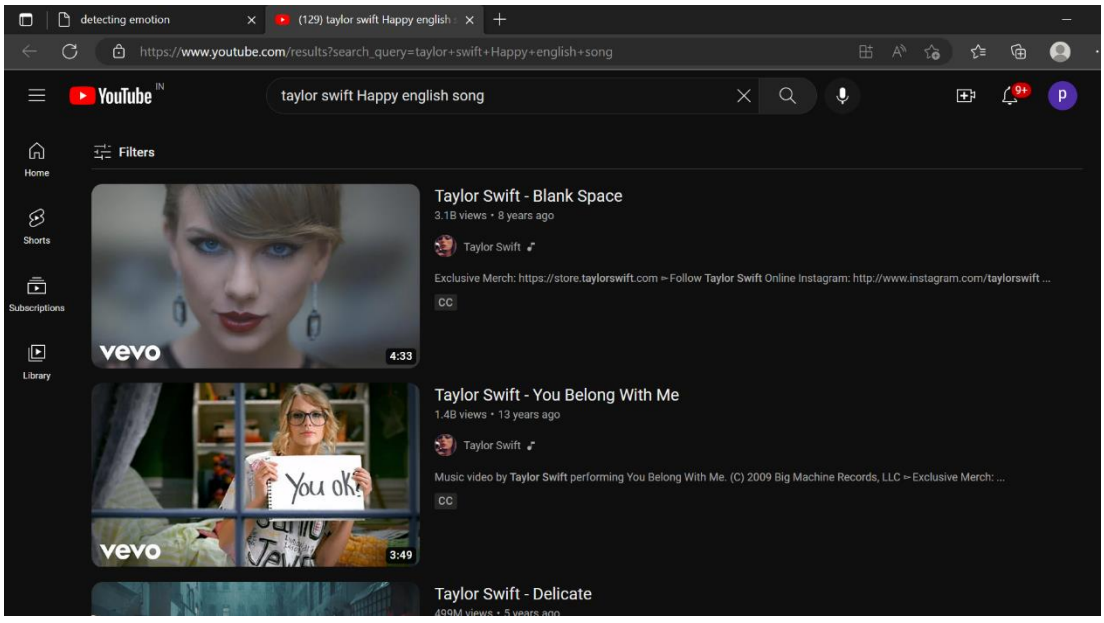
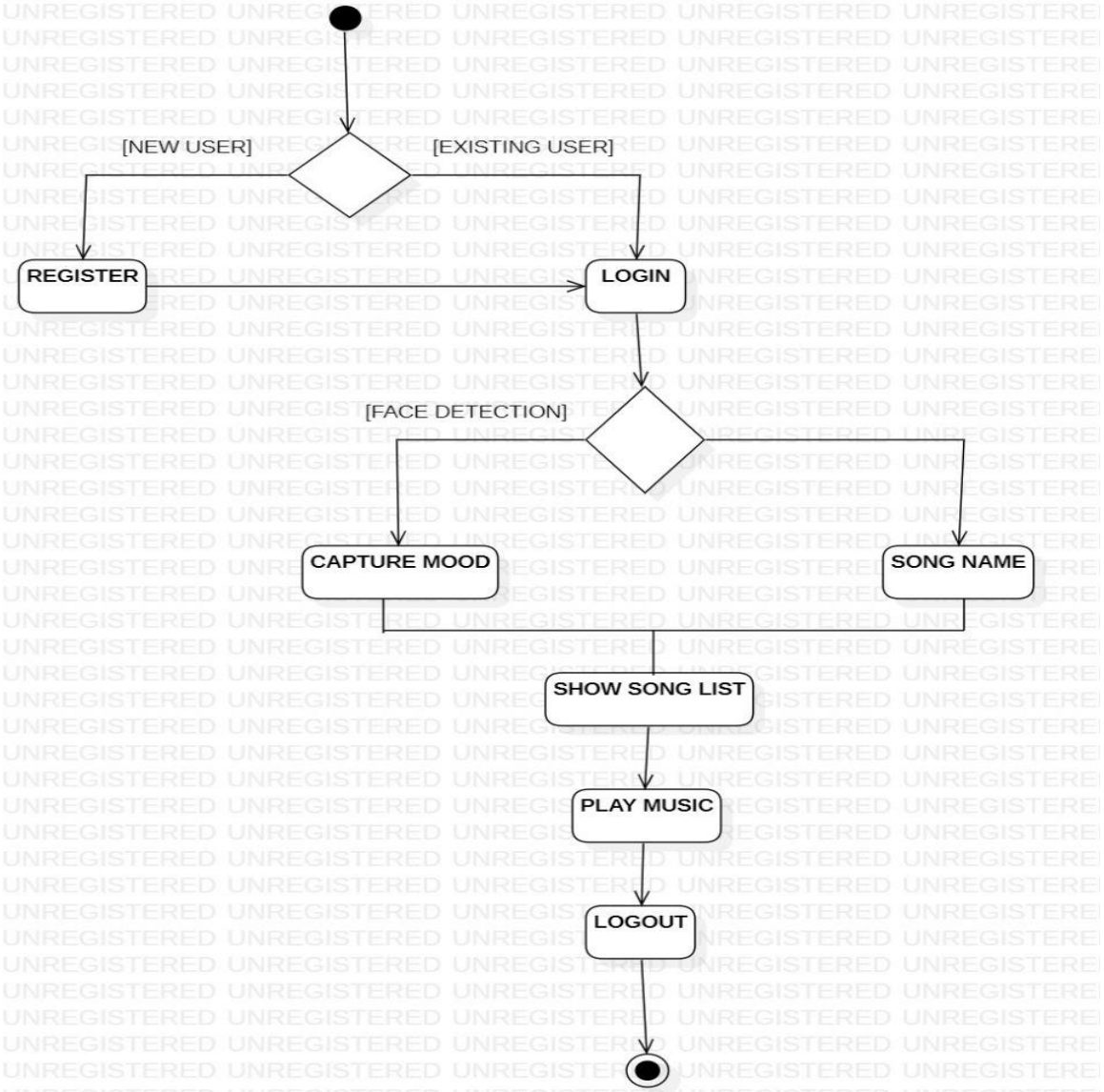


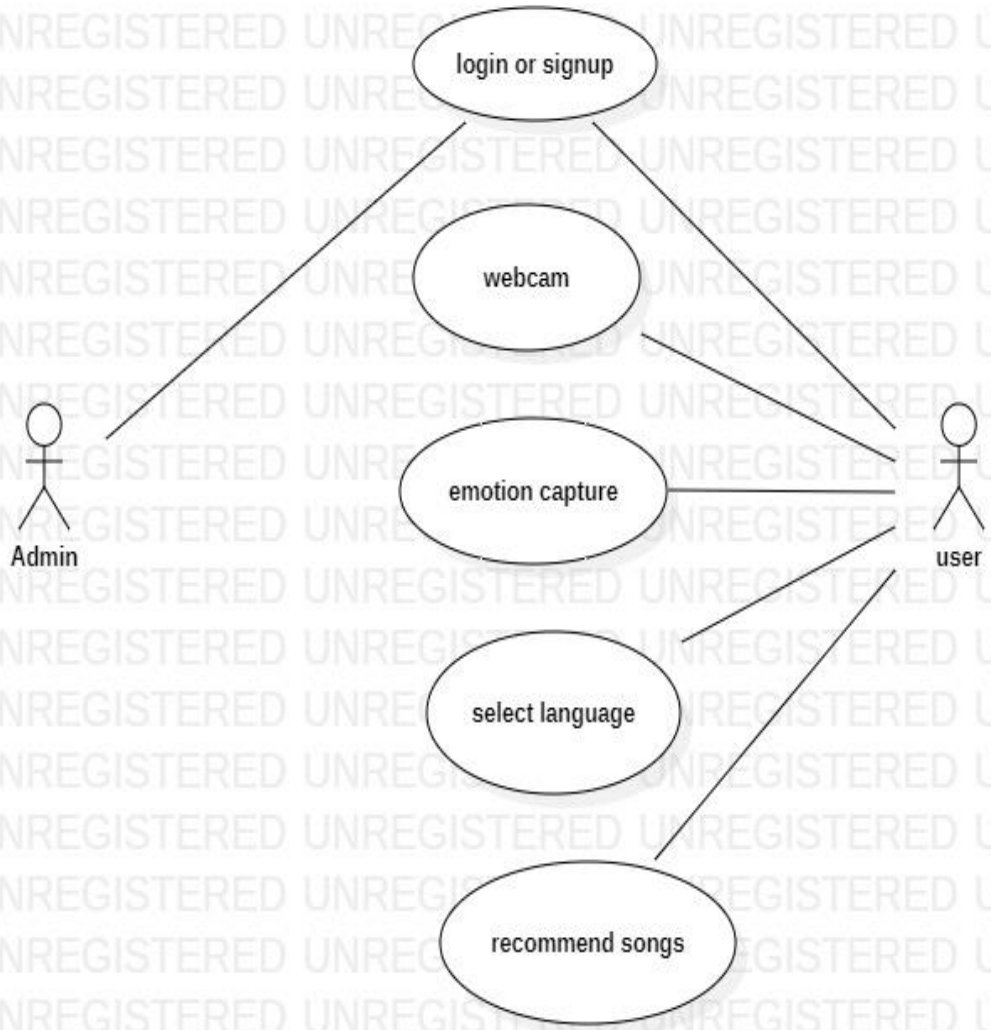
Fig.3.1. Predicted Songs

3.2. UML DIAGRAMS

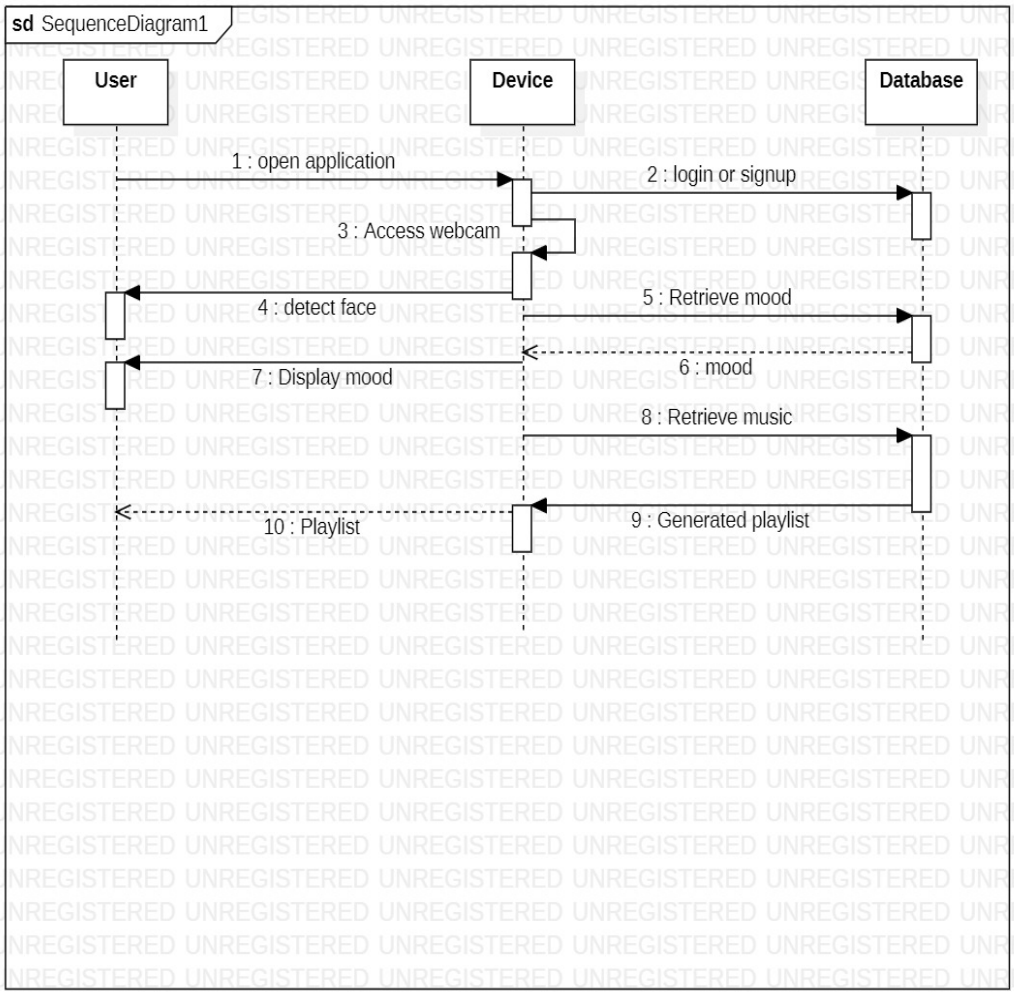
3.2.1. Activity Diagram



3.2.2. Use case Diagram



3.1.3. Sequence Diagram



4. CODING

PYTHON:

```
from flask import Flask, render_template, request
import numpy as np
import cv2
from keras.models import load_model
import webbrowser


app = Flask(__name__)

app.config['SEND_FILE_MAX_AGE_DEFAULT'] = 1

info = {}

haarcascade = "haarcascade_frontalface_default.xml"
label_map = ['Anger', 'Neutral', 'Fear', 'Happy', 'Sad', 'Surprise']
print("+"*50, "loadin gmmodel")
model = load_model('model.h5')
cascade = cv2.CascadeClassifier(haarcascade)



@app.route('/')
def index():
        return render_template('index.html')

@app.route('/choose_singer', methods = ["POST"])
def choose_singer():
        info['language'] = request.form['language']
        print(info)
        return render_template('choose_singer.html', data = info['language'])


@app.route('/emotion_detect', methods=["POST"])
def emotion_detect():
        info['singer'] = request.form['singer']

        found = False

        cap = cv2.VideoCapture(0)
        while not(found):
                _, frm = cap.read()
                gray = cv2.cvtColor(frm,cv2.COLOR_BGR2GRAY)

                faces = cascade.detectMultiScale(gray, 1.4, 1)

                for x,y,w,h in faces:
```

```
                    found = True
                    roi = gray[y:y+h, x:x+w]
                    cv2.imwrite("static/face.jpg", roi)

        roi = cv2.resize(roi, (48,48))

        roi = roi/255.0

        roi = np.reshape(roi, (1,48,48,1))

        prediction = model.predict(roi)

        print(prediction)

        prediction = np.argmax(prediction)
        prediction = label_map[prediction]

        cap.release()

        link  = f
"https://www.youtube.com/results?search_query={info['singer']}+{prediction}+{inf
o['language']}+song"
        webbrowser.open(link)

        return render_template("emotion_detect.html", data=prediction, link=link)

if __name__ == "__main__":
        app.run(debug=True)
```

HTML:

```
<!DOCTYPE html>
<html>
<head>
   <title>music recommend</title>
   <link rel="stylesheet" type="text/css" href="{{url_for('static',
filename='style.css')}}">
   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">
<style>

   body {
      background-image: url("bg.jpg");
      background-repeat: no-repeat;
      background-size: cover;
    }
```

```
     * Style the logo */
    .logo {
      position: absolute;
      top: 0;
      right: 0;
      width: 100px;
      height: 100px;
      margin: 20px;
    }
</style>
</head>
<body style="background-color: #DDA0DD;">
    <img src="_LOGO.png" alt="Logo" class="logo">
    <div class="title">
        <h1 > <p style="color: DARKRED;">WELCOME TO MUSIC MYSTI
!</p></h1>
        <h1><p style="color: DARKRED;">We hope you'll get a music of ur
emotion</p> </h1>
    </div>

    <form action="{{url_for('choose_singer')}}" method="POST">

        <ul>
        <li>
        <label><input type="radio" name="language" value="tamil"
checked="true"><div class="bullet">
            <p style="color: FireBrick;"><B>Tamil </B> </p>
        </div>
        </li>
        </label>
        <li>
        <label><input type="radio" name="language" value="english"><div
class="bullet">
            <p style="color: FireBrick;"> <B>English </B></p> </label>
        </li>
        <br>
        <li>
        <label><input type="radio" name="language" value="hindi"><div
class="bullet">
            <p style="color: FireBrick;"><b>  Hindi</b></div>
        </li>
        </label>
        <li>
        <label><input type="radio" name="language" value="malayalam"><div
class="bullet">
            <p style="color: FireBrick;"><b>  Malayalam </b></div>
        </li>
        </label>
        <br>
```

```
      <input type="submit" name="btn" value="next" class="btn btn-outline-success
btn-lg">
         <table>
            <div class="title">
               <h1 class="display-4">you looks like {{data}}</h1>
               <h1><a href="{{link}}" title="">go to here</a></h1>
            </div>
         </table>
         </form>
            <img src="{{url_for('static', filename='face.jpg')}}">
</form>
   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf"
crossorigin="anonymous"></script>

body{
      text-align: center;
   }

   h1{
      font-family: sans-serif;
      color: #289672;
   }
   .title{
      margin-top: 20px;
   }
   input[type="radio"]{
      display: none;
   }
   .box{
      background-color: #ddffbc;
      height: 330px;
      width: 450px;
      padding: 30px;
      text-align:center;
      font-size: 40px;
      font-family: sans-serif;
      line-height: 220px;
      margin: 10px 10px;
      border: 2px solid #52734d;
      border-radius: 10%;

   }
   .box:hover{
      background-color: #91c788;
      transition-duration: 1s;
   }

   input[type=radio]:checked + .box{
```

```
        background-color: #52734d;
        color: white;
        transition-duration: 1s;
    }
    .btn{
        width: 300px;
        height: 60px;
        }

    img{
        margin-top: 50px;
        height: 400px;
        border : 2px solid green;
        border-radius: 30px;
    }
</style>
</head>
<body style="background-color: #edffec;">
<form>
<table>
    <div class="title">
        <h1 class="display-4">you looks like {{data}}</h1>
        <h1><a href="{{link}}" title="">go to here</a></h1>
    </div>
</table>
</form>
    <img src="{{url_for('static', filename='face.jpg')}}">

form action="{{url_for('emotion_detect')}}" method="POST">
        {%if data == "tamil"%}

        <div class="title">
        <h1 class="display-4">Tamil Singer</h1>

        </div>
        <label><input type="radio" name="singer" value="ar rahman"
checked="true"><div class="box">A R Rahman</div></label>
        <br>
        <label><input type="radio" name="singer" value="anirudh"><div
class="box">Anirudh</div></label>
        <br>
        <label><input type="radio" name="singer" value="yuvan shankar raja"><div
class="box">Yuvan</div></label>
        <br>
        <label><input type="radio" name="singer" value="gvprakas"><div
class="box">GV Prakash</div></label>
        <br>
        <label><input type="radio" name="singer" value="pradeepkumar"><div
class="box">Pradeep Kumar</div></label>
        <br>
```

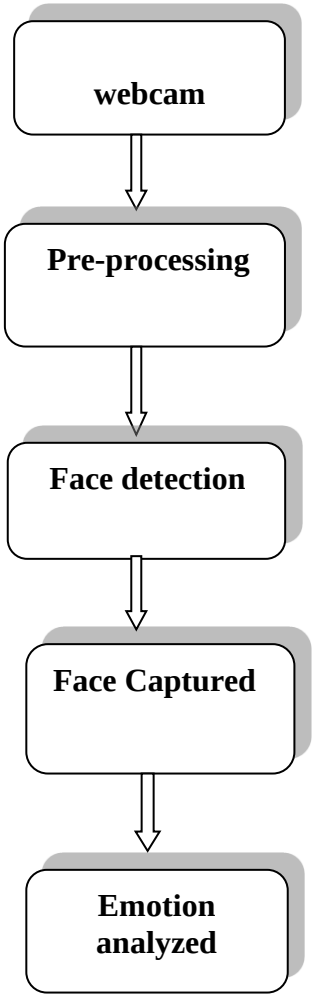
```
        <label><input type="radio" name="singer" value="shreyagoshal"><div
class="box">Shreya Goshal</div></label>
        <br>
        <label><input type="radio" name="singer" value="k.s.janaki"><div
class="box">K.S. Janaki</div></label>
        <br>
        <label><input type="radio" name="singer" value="SP
BalaSubramaniam"><div class="box">SP BalaSubramaniam</div></label>
        <br>
        <label><input type="radio" name="singer" value="ilayaraja"><div
class="box">Ilayaraja</div></label>
        <br>
        <label><input type="radio" name="singer" value="nivas"><div
class="box">Nivas</div></label>

        {%elif data == "english"%}

        <div class="title">
        <h1 class="display-4">English singer</h1>
        </div>
        <label><input type="radio" name="singer" value="taylor swift"
checked="true"><div class="box">Taylor Swift</div></label>
        <br>
        <label><input type="radio" name="singer" value="harry styles"><div
class="box">Harry Styles</div></label>
        <br>
        <label><input type="radio" name="singer" value="selena gomez"><div
class="box">Selena Gomez</div></label>
        <br>
        <label><input type="radio" name="singer" value="justin beiber"><div
class="box">Justin Beiber</div></label>

        {%else%}
        <div class="title">
        <h1 class="display-4">Choose your own singer</h1>

        </div>
        <input type="text" name="singer">
        {%endif%}

        <br>
        <input type="submit" name="btn" value="next" class="btn btn-outline-success
btn-lg">
    </form>

</body>
</html>
```

5.METHODOLOGY

We built the convolutional Neural Network model using the Kaggle dataset. The database is FER2013 which is split into two parts training and testing dataset. The training dataset consists of 24176 and the testing dataset contains 6043 images. There are 48x48 pixel grayscale images of faces in the dataset. Each image in FER-2013 is labeled as one of five emotions: happy, sad, angry, surprise, and neutral. The faces are automatically registered so that they are more or less centered in each image and take up about the same amount of space. The images in FER-2013 contain both posed and unposed headshots, which are in grayscale and 48x48 pixels. The FER-2013 dataset was created by gathering the results of a Google image search of every emotion and synonyms of the emotions. FER systems being trained on an imbalanced dataset may perform well on dominant emotions such as happy, ad, angry, neutral, and surprised but they perform poorly on the under-represented ones like disgust and fear.

5.1 FACE DETECTION

Face detection is one of the applications which is considered under computer vision technology. This is the process in which algorithms are developed and trained to properly locate faces or objects in object detection or related system in images. This detection can be real-time from a video frame or images. Face detection uses such classifiers, which are algorithms that detect images. Face detection uses such classifiers, which are algorithms that detect what's either a face or not a face in an image. Classifiers are trained to detect faces using numbers of images to get more accuracy. OpenCV using one sort of classifier, HaarCascades. A Haar classifier is used for face 11 detection where the classifier is trained with pre-defined varying face data which enables it to detect different faces pre-defined varying face data which enables it to detect different face accurately. The main aim of face detection is to spot the face within the frame by reducing external noises and other factors. It is a machine learning-based approach where the cascade function is trained with a group of input files. It is supported the Haar Wavelet technique to research pixels inside the image into squares by function. This uses machine learning techniques to urge a high degree of accuracy from what is called "training data".



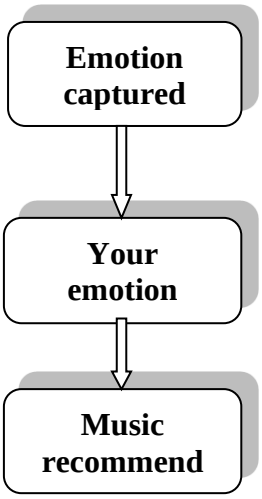
5.2 FEATURE EXTRACTION

While performing feature extraction, we treat the pre-trained network that is sequential model as an arbitrary feature extractor. Allowing the input image to pass on it forward, stopping at the pre-specifies layer, and taking the outputs of the layer as our features. Starting layers of a convolutional network extract high-level features from the taken image, so use only a few filters. Load the input image for which we want to view the Feature map to know which features were prominent to classify the image. Feature maps are obtained by applying Filters or Feature detectors to the input image or the feature map output of the prior layers. Feature map visualization will provide insight into the interior representations for specific input for each of the Convolutional layers within the model.

5.3 EMOTION DETECTION

Convolution neural network Architecture. Convolution neural network architecture applies filters or feature detectors to the input image to get the feature maps or activation maps using the Relu activation function. Feature detectors or filters help in identifying various features present in the image such as edges, vertical lines, horizontal lines, bends, etc. After that pooling is applied over the feature maps for invariance to translation. Pooling is predicted on the concept that once we change the input by a touch amount, the pooled outputs don't change. We can use any of the pooling from min, average, or max. But max-pooling provides better performance than min or average pooling. Flatten all the input and giving these flattened inputs to a deep neural network which are outputs to the class of the object. The class of the image will be binary, or it will be a multi-class classification for identifying digits or separating various apparel items. Neural networks are as a black box, and learned features in a Neural Network are not interpretable. So basically, we give an input image then the CNN model returns the results. Emotion detection is performed by loading the model which is trained by weights using CNN. When we take the real-time image by a user then that image was sent to the pre-trained CNN model, then predict the emotion and adds the label to the image.

.



5.4 MUSIC RECOMMENDATION

The input is acquired in real-time so the camera is used to capture the video and then the framing are done. The value of each landmark in the face is calculated and is stored for future use. The efficiency of classifier is about 90-95%. so that even when there are any changes in the face due to environmental conditions the system can still identify the face and the emotion being expressed. The values are transferred to the web service. The song is played from the emotion detected. The emotions are assigned for every song. There are six emotions that can be used and the emotions are happy, anger, sad, surprise, fear, disgust and neutral. When the happy emotion is recognized the songs that are assigned for that particular emotion are played and the same happens with the other emotions as well that is it the songs are played for the emotions detected respectively.

6.SYSTEM IMPLEMENTATION

Our project is based on emotion-based music recommendation system. Here, in our first page there will be a welcome page and also we will be displaying multi languages to the users so that they can select. By selecting any one of the languages it will take to the third page which will display the artist’s name. According to the user's favorite artist they can select the artist’s name. When you will select the artist’s name it will take to the fourth page which will make the camera flash and predict your emotion and that predicted result will Pass the data and it will recommend the YouTube links. The user can also change the songs according to their perspective.

7. FUTURE ENHANCEMENTS

The emotion-based music system will be of great advantage to users looking for music based on their mood and emotional behavior. It will help reduce the searching time for music thereby unnecessary time and hence increasing the overall accuracy and efficiency of the system. The methodology of enhancement in the automatic play of songs are done by detection of the facial expression. An alternative method, based on additional emotions which is excluded in our system as disgust and fear. On this emotion included to support the playing of music automatically. Interactivity: the system could allow users to provide feedback on recommended songs, so that it can adjust its recommendations accordingly. The system could be adapted to account for cultural differences in musical preferences and emotions.

This work can be extended by the following measures.

- Making this as a real time application so that actual users would be able to use it.
- Giving an option to the user about what language songs he/she wants to listen.
- Extracting songs from third party API in real time.
- Track users' mood and listening history to enhance the recommended playlist.
- Deploying this application in any cloud platform such as Azure, Google app engine etc.

8. CONCLUSION

Even though human emotions are complex and subtle, it is possible for a machine learning model to be trained to accurately detect a set of emotions which can be differentiated from each other with certain facial expressions. The expression on a person's face can be used to detect their mood, and once a certain mood has been detected, music suitable for the person's detected mood can be suggested. Our model, having the accuracy of approximately 75%, is able to detect seven moods accurately: anger, disgust, fear, happy, sad, surprise and neutral; and our android application is able to play the music that would be suitable for the detected mood. This project is designed for the purpose of making better interaction between the music system and the user because Music is helpful in changing the mood of the user and for some people it is a stress reliever. Recent development it shows a wide prospective in the developing the emotion-based music recommendation system. Thus, the present system presents Face (expressions) based recognition system so that it could detect the emotions and music will be played accordingly.

9. BIBLIOGRAPHY

1.https://www.linkedin.com/pulse/mood-based-music-recommendation-python-spotify-api-burak-%C3%B6zt%C3%BCrk?trk=public_profile_article_view

2.https://github.com/Dhruba59/Music-recommendation-based-on-facial-emotion-recognition

3. https://pemagrg.medium.com/build-a-web-app-using-pythons-flask-for-beginners-f28315256893

4.https://levelup.gitconnected.com/how-to-build-a-real-time-emotion-detection-web-app-ce7e3ed7b7de

5. https://github.com/Pawandeep-prog/emotion-based-music-ai

6.https://www.w3schools.in/php/phpheaders#:~:text=Redirecting%20Browser,as%20the%20command%20is%20executed.