Recommending Songs Based On Facial Expression

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR'S OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

Akshay Ganjiwar(111915009)

K Rohit Reddy(111915059)

K Karthik Kumar(111915061)

Semester: III

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Indian Institute of Information And Technology Pune

Near Bopdev Ghat, Kondhwa Annexe, Yewalewadi, Pune, Maharashtra
411048

JUNE 2020

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **“Recommending Songs Based On Facial Expression”** submitted by **Akshay Ganjiwar,K Rohit Reddy,K Karthik Kumar** bearing the **MIS No: 111915009,111915059,111915061**, in completion of his project work under the guidance of **Dr.Sanjeev Sharma** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering in the Department of Computer Science and Engineering, Indian Institute of Information Technology,Pune, during the academic year 2020-21.

Dr.Sanjeev Sharma
Project Guide
Professor
Department of CSE
IIIT Pune

Dr.Tanmoy Hazra
Head of the Department
Assistant Professor
Department of CSE
IIIT Pune

Project Viva-voce held on 23-09-2020

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. I would like to thank the people who helped me directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our beloved director, **Dr.Anupam Shukla**, for providing his kind support in various aspects.

I would like to express my gratitude to my project guide **Dr.Sanjeev Sharma**, Professor, Department of CSE, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this M.Tech project.

I would like to express my gratitude to the head of department **Dr.Tanmoy Hazra**, Professor, Department of CSE, for providing his kind support in various aspects.

I would also like to thank all the faculty members in the Dept. of CSE and my classmates for their steadfast and strong support and engagement with this project.

Abstract

The human face plays an important role in knowing an individual's mood. The required input are extracted from the human face directly using a camera. One of the applications of this input can be for extracting the information to deduce the mood of an individual.

This data can then be used to get a list of songs that comply with the "mood" derived from the input provided earlier. This eliminates the time-consuming and tedious task of manually Segregating or grouping songs into different lists and helps in generating an appropriate playlist based on an individual's emotional features.

Recommend Songs Based On Facial Expression aims at scanning and interpreting the data and accordingly creating a playlist based the parameters provided. Thus our proposed system focus on detecting human emotions for developing emotion based music player, which are the approaches used by available music players to detect emotions, which approach our music player follows to detect human emotions and how it is better to use our system for emotion detection. A brief idea about our systems working, playlist generation and emotion classification is given.

Keywords : Facial Expression Detection, Facial Landmark Extraction, Cnn Classification,Keras

TABLE OF CONTENTS

Abstract **i**

List of Figures **iv**

List of Tables **v**

1 Introduction **1**

2 Literature Review And Research Gap **3**

3 Objectives **4**

4 Methodologies **5**

4.1 opencv 5

4.2 Face Detection 5

4.3 Facial Feature Extraction 5

4.4 Emotion Classification 6

4.5 Music Recommendation 6

5 Work Done So Far **8**

5.1 Libraries Imported 8

5.2 Algorithm used:- 8

5.3 Program 9

5.4 Program explation 13

6 Conclusion **15**

7 Results **16**

8 References **18**

List of Figures

4.1 facial extract 6

4.2 flowchart 7

5.1 flowchart 14

7.1 resulthappy 16

7.2 resultsad 17

List of Tables

Listings

5.1 training program 9

5.2 testing program 12

Chapter 1

Introduction

Chapter Introduction

Human emotion detection is implemented in many areas requiring additional security or information about the person. It can be seen as a second step to face detection where we may be required to set up a second layer of security, where along with the face, the emotion is also detected. This can be useful to verify that the person standing in front of the camera is not just a 2-dimensional representation.

Generally people have a large number of songs in their database or playlists. Thus to avoid trouble of selecting a song, most people will just randomly select a song from their playlist and some of the songs may not be appropriate for the current mood of the user and it may disappoint the user. As a result, some of the songs are not matching to the users current emotion. Moreover, there is no commonly used application which is able to play songs based on the current emotions of the user. Music plays a very important role in enhancing an individuals life as it is an important medium of entertainment for music lovers and listeners and sometimes even imparts a therapeutic approach. In todays world, with ever increasing advancements in the field of multimedia and technology, various music players have been developed with features like fast forward, reverse, variable playback Although these features satisfy the users basic requirements, yet the user has to face the task of manually browsing through the playlist of songs and select songs based on his current mood and behaviour.

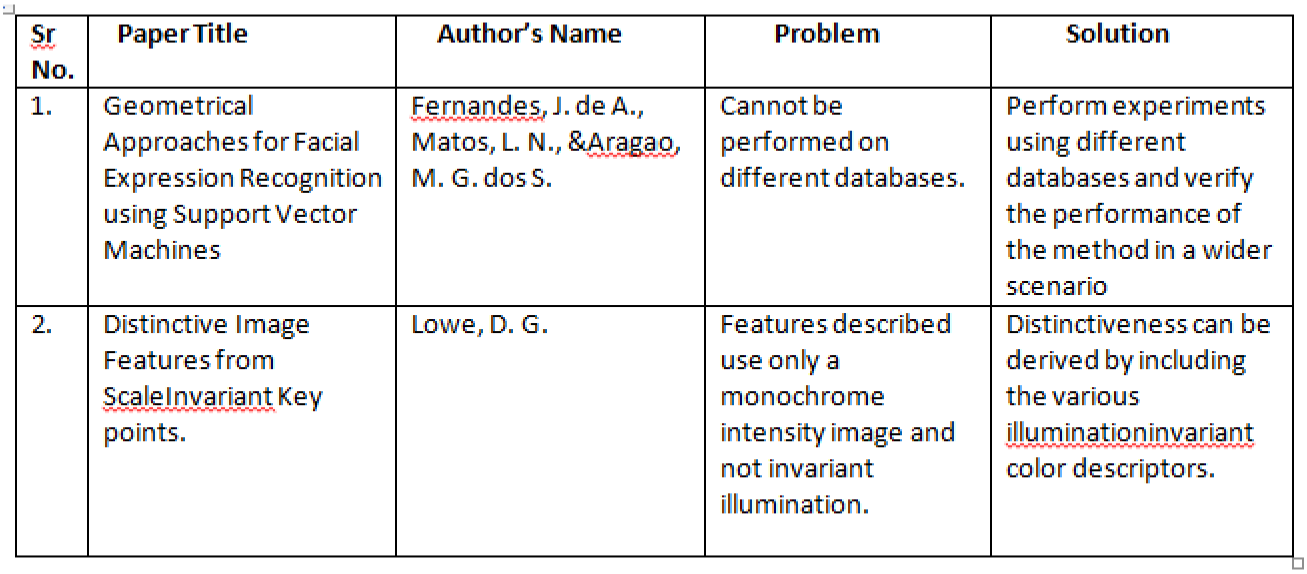
The main objective of this project is to recommend songs based on current emotional state and behavior of the user. Face detection and facial feature extraction from image is the first step in Recommend songs based on facial expression. For the face detection to work effectively, we need to provide an input image which should not be blur and tilted. We have used algorithm that is used for face detection and facial feature extraction. We have generated landmarks points for facial features. The next step is the classification of emotion for which we have used multiclass CNN classification. The emotion classified by CNN is then passed to some third party music app and accordingly music will be played.

Chapter 2

Literature Review And Research Gap

Introduction of chapter 2.

Various techniques and approaches have been proposed and developed to classify human emotional state of behavior.



Sr No.	Paper Title	Author's Name	Problem	Solution
1.	Geometrical Approaches for Facial Expression Recognition using Support Vector Machines	Fernandes, J. de A., Matos, L. N., &Aragao, M. G. dos S.	Cannot be performed on different databases.	Perform experiments using different databases and verify the performance of the method in a wider scenario
2.	Distinctive Image Features from ScaleInvariant Key points.	Lowe, D. G.	Features described use only a monochrome intensity image and not invariant illumination.	Distinctiveness can be derived by including the various illuminationinvariant color descriptors.

Chapter 3

Objectives

- To provide an interface between the music system.
- To provide a very good entertainment for the users.
- implement the ideas of machine learning.
- To provide a new age platform for music lovers.
- To bridge gap between growing technologies and music techniques.

Chapter 4

Methodologies

main chapter 1

4.1 opencv

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. in these we use opencv to detect face. Face detection is important as it will classify only if face is present.

4.2 Face Detection

A facial recognition system uses biometrics to map facial features from a photograph or video. It compares the information with a database of known faces to find a match.The main objective of face detection technique is to identify the face in the frame by reducing the external noises and other factors.

4.3 Facial Feature Extraction

Facial Features extraction is an important step in face recognition and is defined as the process of locating specific regions, points, landmarks, or curves/contours in a given 2-D image or a 3D range image. In this feature extraction step, a numerical feature vector is generated from the resulting registered image. Common features that can be extracted area. a.Lips

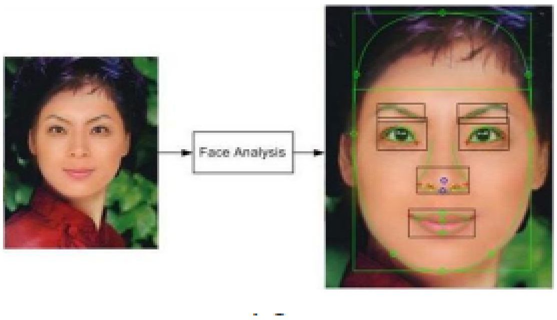


Figure 4.1: facial extract

b. Eyes

c. Eyebrows

d. Nose tip

figure reference fig : 4.1 in the page : 6.

4.4 Emotion Classification

In this classification, the algorithm attempts to classify the given faces portraying one of the seven basic emotions.Different approaches which are followed for Facial emotions Recognition.we use neural network approach.That is the neural network contained a hidden layer with neurons. The approach is based on the assumption that a neutral face image corresponding to each image is available to the system. Each neural network is trained independently with the use of on-line back propagation. Neural Network will be discussed later.

4.5 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 cnn model classification are used for processing the framed images. The frames that are obtained are considered in all frames and all pixel formats for the purpose of emotion classification..The emotions are then identified using the values that are obtained that are being set and from the value of the pixel that is received is being compared to that of the values that is present as threshold in the code. The values is transferred to the web service. The song are played from the emotion deteced. The

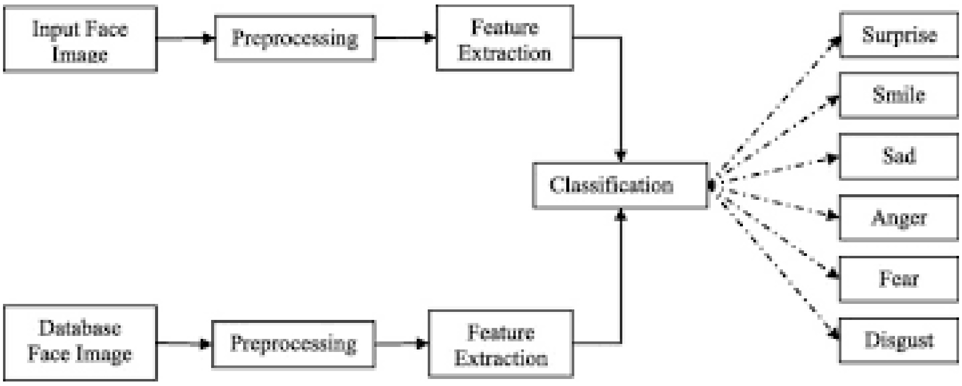


Figure 4.2: flowchart

emotions are assigned for every song. When the emotion is transferred the respective song will be plays.

figure reference fig : 4.2 in the page : 7.

Chapter 5

Work Done So Far

5.1 Libraries Imported

Here we imported three libraries in our program.

They are:1)opencv 2)numpy 3)keras 1)opencv is an open source computer vision and

machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

2)NumPy is a python library used for working with n-dimensional arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.Numpy stands for numerical python.

3)Keras is an open source neural-network library written in python for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code

5.2 Algorithm used:-

CNN is an efficient recognition algorithm which is widely used in pattern recognition and image processing. ... Generally, the structure of CNN includes two layers one is feature extraction layer, the input of each neuron is connected to the local receptive fields of the previous layer, and extracts the local feature.

figure reference fig : 5.1 in the page : 14.

5.3 Program

```python
from __future__ import print_function
import keras
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Dense,Dropout,Activation,Flatten,BatchNormalization
from keras.layers import Conv2D,MaxPooling2D
import os

num_classes = 5
img_rows,img_cols = 48,48
batch_size = 8

train_data_dir = r'C:\Users\Dell\Desktop\expression detection\images\train'
validation_data_dir = r'C:\Users\Dell\Desktop\expression detection\images\validation'

train_datagen = ImageDataGenerator(
          rescale=1./255,
          rotation_range=30,
          shear_range=0.3,
          zoom_range=0.3,
          width_shift_range=0.4,
          height_shift_range=0.4,
          horizontal_flip=True,
          fill_mode='nearest')

validation_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
          train_data_dir,
          color_mode='grayscale',
          target_size=(img_rows,img_cols),
          batch_size=batch_size,
          class_mode='categorical',
          shuffle=True)

validation_generator = validation_datagen.flow_from_directory(
              validation_data_dir,
              color_mode='grayscale',
              target_size=(img_rows,img_cols),
              batch_size=batch_size,
              class_mode='categorical',
              shuffle=True)


model = Sequential()

# Block-1

model.add(Conv2D(32,(3,3),padding='same',kernel_initializer='he_normal',input_shape=(img_rows,img_cols,1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(32,(3,3),padding='same',kernel_initializer='he_normal'
,input_shape=(img_rows,img_cols,1)))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-2

model.add(Conv2D(64,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(64,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-3

model.add(Conv2D(128,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(128,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-4

model.add(Conv2D(256,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Conv2D(256,(3,3),padding='same',kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.2))

# Block-5

model.add(Flatten())
model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-6

model.add(Dense(64,kernel_initializer='he_normal'))
model.add(Activation('elu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))

# Block-7

model.add(Dense(num_classes,kernel_initializer='he_normal'))
model.add(Activation('softmax'))

print(model.summary())

from keras.optimizers import RMSprop,SGD,Adam
from keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau

checkpoint = ModelCheckpoint('Emotion_little_vgg.h5',
                             monitor='val_loss',
                             mode='min',
                             save_best_only=True,
                             verbose=1)

earlystop = EarlyStopping(monitor='val_loss',
                          min_delta=0,
                          patience=3,
                          verbose=1,
                          restore_best_weights=True
                          )

reduce_lr = ReduceLROnPlateau(monitor='val_loss',
                              factor=0.2,
                              patience=3,
                              verbose=1,
                              min_delta=0.0001)

callbacks = [earlystop,checkpoint,reduce_lr]

model.compile(loss='categorical_crossentropy',
              optimizer = Adam(lr=0.001),
              metrics=['accuracy'])

nb_train_samples = 24176
nb_validation_samples = 3006
epochs=25

history=model.fit_generator(
                train_generator,
                steps_per_epoch=nb_train_samples//batch_size,
                epochs=epochs,
                callbacks=callbacks,
                validation_data=validation_generator,
                validation_steps=nb_validation_samples//batch_size)
```

Listing 5.1: training program

```
from keras.models import load_model
from time import sleep
from keras.preprocessing.image import img_to_array
from keras.preprocessing import image
import cv2
import numpy as np

face_classifier = cv2.CascadeClassifier(r’C:\Users\Dell\Desktop\expression detection\Facial-Expressions-Recognition-master\haarcascade_frontalface_default.xml’)
classifier =load_model(r’C:\Users\Dell\Desktop\expression detection\Emotion_little_vgg.h5’)

class_labels = [’Angry’,’Happy’,’Neutral’,’Sad’,’Surprise’]

cap = cv2.VideoCapture(0)



while True:
    # Grab a single frame of video
    ret, frame = cap.read()
    labels = []
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        roi_gray = gray[y:y+h,x:x+w]
        roi_gray = cv2.resize(roi_gray,(48,48),interpolation=cv2.INTER_AREA)
    # rect,face,image = face_detector(frame)


        if np.sum([roi_gray])!=0:
            roi = roi_gray.astype(’float’)/255.0
            roi = img_to_array(roi)
            roi = np.expand_dims(roi,axis=0)

        # make a prediction on the ROI, then lookup the class

            preds = classifier.predict(roi)[0]
            label=class_labels[preds.argmax()]
            label_position = (x,y)
            cv2.putText(frame,label,label_position,cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
        else:
            cv2.putText(frame,’No Face Found’,(20,60),cv2.FONT_HERSHEY_SIMPLEX,2,(0,255,0),3)
    cv2.imshow(’Emotion Detector’,frame)
    if cv2.waitKey(1) & 0xFF == ord(’q’):
        break

cap.release()
cv2.destroyAllWindows()
```

Listing 5.2: testing program

5.4 Program explation

In these we have two parts first is training part and other is testing part. First we imported all the libraries that are required such as opencv,numpy,keras.

Next we read the train dataset and validation dataset and next generate more data by train data generator then validation data genrator

There after we have to build our keras model.we have to buit neural network blocks by using cnn classification

Then we do optimisation and callbacks using keras library.after that we star training of that dataset.

In the testing part first import all required libraries.

Next we read the face classifier using opencv library from haarcascade file.And load classifier model

Then defining class labels and using opencv library read videocapture for webcam use argument 0.

Now grab a single frame of video and make a region of intrest.

There after make a prediction on region of intrest,then lookup class

After getting the class(or)emotion based on that their would be recommended playlist of songs.

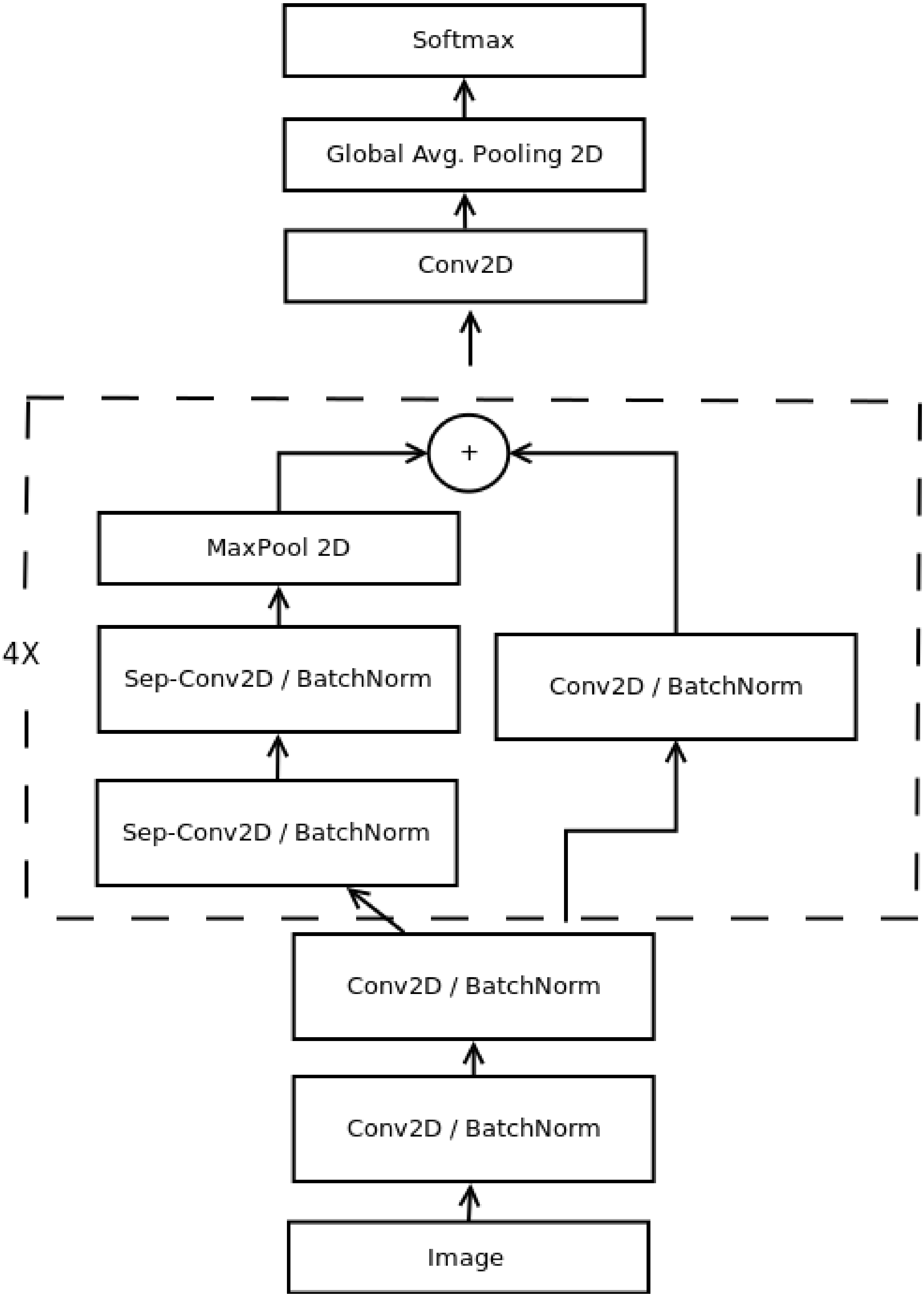


Figure 5.1: flowchart

Chapter 6

Conclusion

In this project, we presented a model to recommend a songs based on the emotion based detected from the facial expression.

This project proposed designed and developed an emotion based music recommendation system using face recognition System. Music are the one that has the power to heal any stress or any kind of emotions. Recent development promises a wide scope in developing emotion based music recommendation system.

Thus the proposed system presents Face based emotion recognition system to detect the emotions and play music from the emotion detected.

Chapter 7

Results

This chapter provides the experimental results of the proposed scheme.

figure reference fig : 7.1 in the page : 16.

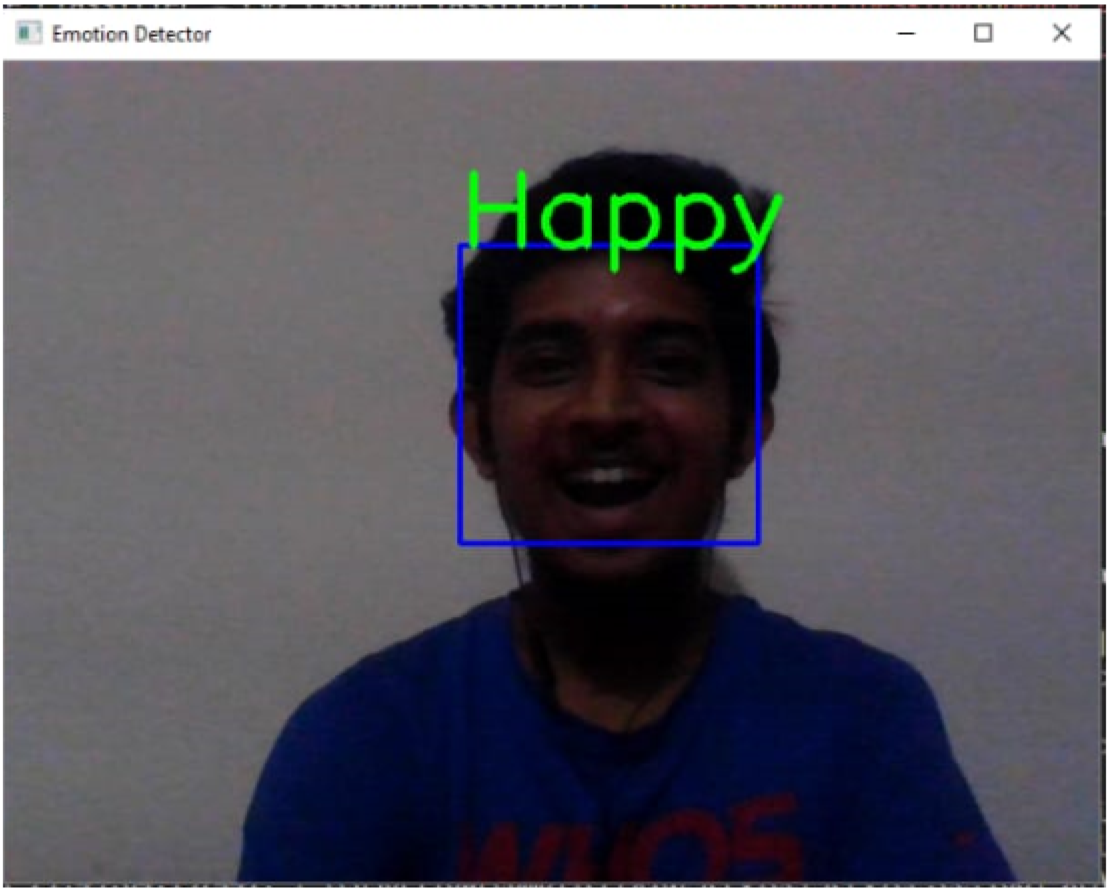


Figure 7.1: resulthappy

figure reference fig : 7.2 in the page : 17

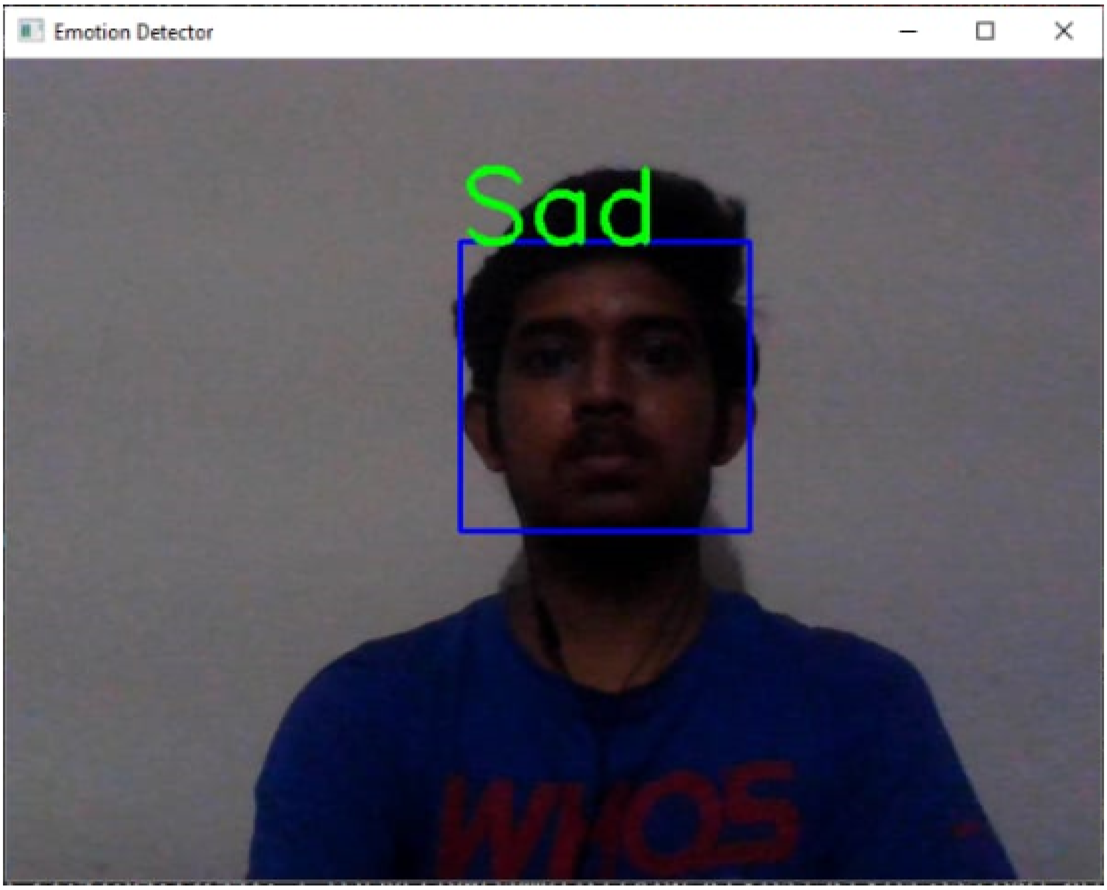


Figure 7.2: resultsad

Chapter 8

References

1)The research by Lowe, D. G. on Distinctive Image Features from ScaleInvariant Key points.

2)Anukriti Dureha "An Accurate Algorithm for Generating a Music Playlist based on Facial Expressions" : IJCA 2014.

3)Sachin R, Sowmya V, Govind D, et al. Dependency of various color andintensity planes on CNN based image classification. 2017.