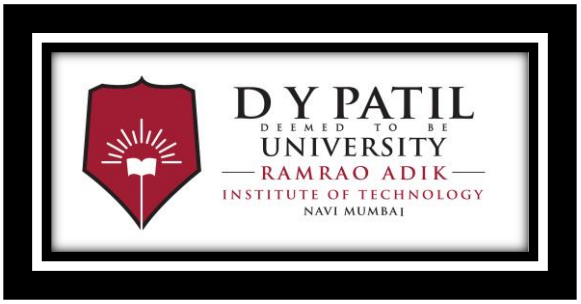


MINI-PROJECT REPORT

ON

EMOTION BASED MUSIC PLAYER

Submitted by

Roll No	Name of the student
19IT1100	**Hritik Matte**
19IT2006	**Anurag Lende**
18IT2001	**Mayank Jalwal**
19IT2027	**Rushikesh Bhadane**

Supervisor

Ms. Varsha Sanap

Department of Information Technology

Dr. D. Y. Patil Group's

Ramrao Adik Institute of Technology

Nerul, Navi Mumbai 400706.

(Affiliated to University of Mumbai)

(2021)

ABSTRACT:

The objective of this project is to detect emotion and select music to be played based on the detected emotion. Music or songs can be a powerful tool to describe human emotion here in this project is a trial to build a powerful tool that can help the user to play music based on stated emotion or detected one. The project builds an app (Desktop, and Mobile “iOS and Android”) that is extremely easy to use and also easy in a technical way of usage. The second objective is to use a Convolutional Neural Network with high accuracy throughout train as much as possible of data, and also test the results to check for any errors because if so, the data should be trained once again. This dissertation presents related work from previous researchers and details of it in the work methodology that helped in developing this project. Finally, results from testing the app using live captured images and to detect the emotion and select music accordingly are presented.

The methodology of solving this problem is to build a fully functional app (Front End and Back End) that solves this problem, starting from the front end there an easy and understandable interface anyone can use, this interface is fully connected to the back end. On the back end the main algorithm in this project is to build a Convolutional Neural Network to help in the goal of achieving high accuracy rate, because the Convolutional Neural Network is the best in the science of building any network that works with images, and also there are plenty of similar research papers that achieved very well seen success in this field of research. A fully functional app that built to solve this problem (Desktop Only) and also trained almost 28000 images with different states of emotions (Happy, Sad, Angry, and Normal) with a very high accuracy rate which is “85%” for training and “83%” for testing rate, the application is successfully suggesting music by suggesting single songs that fits any user’s emotion.

.

Table of Contents:

System Components:

1.Frontend: Tkinter

1.1 Tkinter:

The tkinter package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at Active State.)

Running **python -m tkinter** from the command line should open a window demonstrating a simple Tk interface, letting you know that tkinter is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

2.Backend: Python, NumPy, OpenCV, Keras, Webbrowser

2.1 Python:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. The fast edit-test-debug cycle makes this simple approach very effective.

2.2 Pygame:

Pygame is a cross-platform set of Python modules designed for writing video games. It includes computer graphics and sound libraries designed to be used with the Python programming language.

2.3 NumPy:

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.

2.4 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. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

2.5 Keras:

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

2.6 Webbrowser:

The `webbrowser` module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the `open (). Function` from this module will do the right thing.

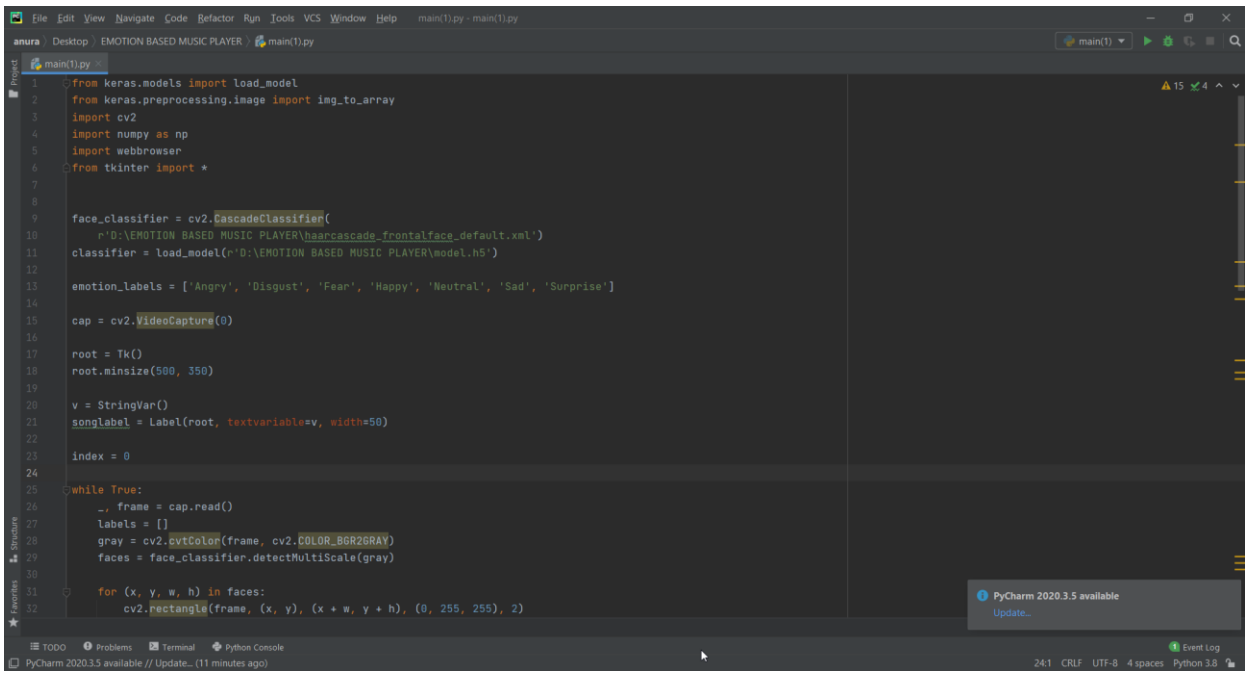
Introduction:

This project develops a Convolutional Neural Network. The CNN was selected because this project is images based solution, there are many ways to solve this problems through them, like body language, voices techniques. Facial Expressions is a way to detect emotions and that's the methodology implemented through the Convolutional Neural Network and the science itself. This project is a trial to implement and solve one of the most important problems that no one around

notice, music is taking a very huge part of our everyday routine and we choose what to listen to base on many things on them is Emotions. Emotions and music are nearly close in their structure and the way they built with are almost near. In this project is a simple implementation that applies playing music based on any user's Emotion, maybe he/she is happy, sad, nervous, neutral, etc. This is one of the hardest problem that not many researchers around solved, so this project is a real hard work to solve this problem in a very effective way. Also in this project we are studying Emotions in most of the parts, what is Emotions? How it can be constructed? And also how to detect them? Because knowing these things will me it easy develop the app and also help the algorithms know what to learn and why from the beginning. Among the history the constructions of the emotions are very different from history to another and from group of people to another, that's what makes it not easy to solve, and at the beginning it will start be studying very small group of people and expanding to the others by the time.

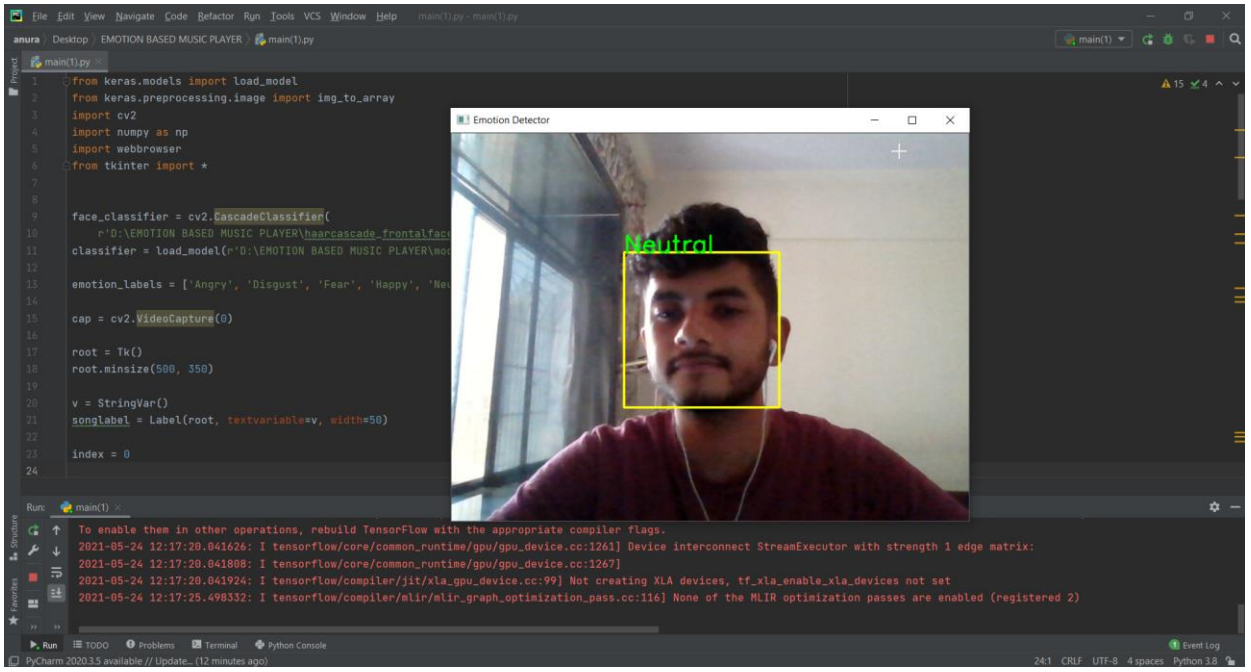
WORKING OF SYSTEM

1. **Run the MusicPlayer.py File.**

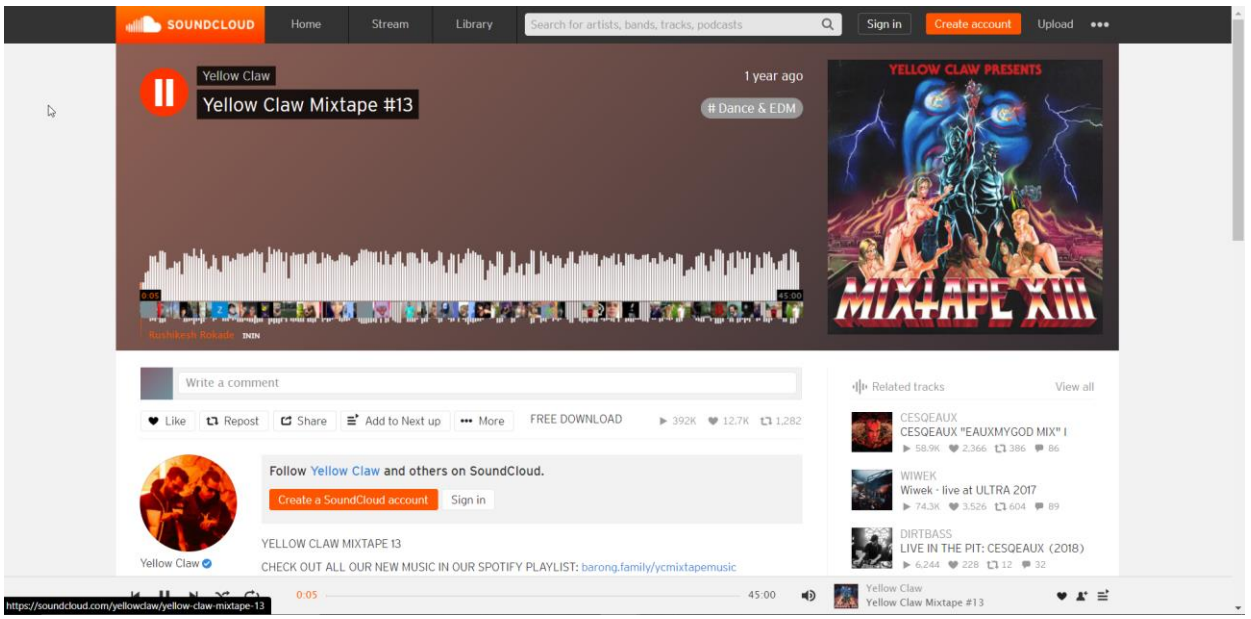


```
from keras.models import load_model
from keras.preprocessing.image import img_to_array
import cv2
import numpy as np
import webbrowser
from tkinter import *


face_classifier = cv2.CascadeClassifier(
    r'D:\EMOTION BASED MUSIC PLAYER\haarcascade_frontalface_default.xml')
classifier = load_model(r'D:\EMOTION BASED MUSIC PLAYER\model.h5')

emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad', 'Surprise']

cap = cv2.VideoCapture(0)

root = Tk()
root.minsize(500, 350)

v = StringVar()
songlabel = Label(root, textvariable=v, width=50)

index = 0

while True:
    _, frame = cap.read()
    labels = []
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray)

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255), 2)
```

1. **Camera will turn on & it will display the user's emotion detected by the software in a rectangular square around the user's face.**



3.The program will play SoundCloud playlists procured to suit the users' emotions detected.


SOUNDCLOUD
Home
Stream
Library
Search for artists, bands, tracks, podcasts
Sign in
Create account
Upload
Yellow Claw
Yellow Claw Mixtape #13
1 year ago
Dance & EDM
YELLOW CLAW PRESENTS
MIXTAPE XIII
Write a comment
Like
Repost
Share
Add to Next up
More
FREE DOWNLOAD
Follow Yellow Claw and others on SoundCloud.
Create a SoundCloud account
Sign in
YELLOW CLAW MIXTAPE 13
CHECK OUT ALL OUR NEW MUSIC IN OUR SPOTIFY PLAYLIST: barong.family/ycmixtapemusic
Yellow Claw
Related tracks
View all
CESQEAUX
CESQEAUX "EAUXMYGOD MIX" I
WIWEK
Wiwek - live at ULTRA 2017
DIRTBASS
LIVE IN THE PIT: CESQEAUX (2018)
0:05
45:00
Yellow Claw
Yellow Claw Mixtape #13
https://soundcloud.com/yellowclaw/yellow-claw-mixtape-13

Source code:

```
from keras.models import load_model
from keras.preprocessing.image import img_to_array
import cv2
import numpy as np
import webbrowser
from tkinter import *


face_classifier = cv2.CascadeClassifier(
    r'D:\EMOTION BASED MUSIC PLAYER\haarcascade_frontalface_default.xml')
classifier = load_model(r'D:\EMOTION BASED MUSIC PLAYER\model.h5')

emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Neutral', 'Sad',
'Surprise']

cap = cv2.VideoCapture(0)

root = Tk()
root.minsize(500, 350)

v = StringVar()
songlabel = Label(root, textvariable=v, width=50)

index = 0

while True:
    _, frame = cap.read()
    labels = []
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_classifier.detectMultiScale(gray)

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

        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)

            prediction = classifier.predict(roi)[0]
            label = emotion_labels[prediction.argmax()]
            label_position = (x, y)
            cv2.putText(frame, label, label_position,
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        else:
            cv2.putText(frame, 'No Faces', (30, 80),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
    cv2.imshow('Emotion Detector', frame)


    def directorychooser():
```

```
        if label == 'Happy':
            webbrowser.open('https://soundcloud.com/taylor-sidzyik/sets/hype-
rap')

        if label == 'Sad':
            webbrowser.open('https://soundcloud.com/liam-twyford/sets/oof')

        if label == 'Neutral':
            webbrowser.open('https://soundcloud.com/yellowclaw/yellow-claw-
mixtape-13')

        if label == 'Angry':

webbrowser.open('https://www.youtube.com/watch?v=YF4lYEyBXb0&list=WL&index=11
')

        if label == 'Surprise':
            webbrowser.open('https://soundcloud.com/babalos/hi-tech-babalos-
snow-crystal-185')

        if label == 'Disgust':
            webbrowser.open('https://www.youtube.com/watch?v=cI1ngpbxo7k')

        if label == 'Fear':
            webbrowser.open('https://soundcloud.com/se7enflo/sets/chill-
beats')

        else:
            print('\n')


    if cv2.waitKey(1) & 0xFF == ord('q'):
        print("you are feeling", label)
        cv2.destroyWindow('Emotion Detector')
        directorychooser()
```

Result:

In the experiment, the images dataset was very huge with almost 27000 images or even more and there are 7 main songs with different varieties genres and tempos. The accuracy rate for the app was 85% which somehow good but it needs more training to be more accurate. The app is working properly with almost 90% of the running trials, which is also a good result, but when it comes to detecting the face in real time working also properly and very well, but when it comes to detecting the app at the beginning did the recognition so many times wrong but by more training and more editing through the Convolutional Neural Network scripts the detection is improving every time the app do more training. To say that the app is running is a perfect thing but it has to make sure that the CNN is implemented right, with the layer needed to build a complete fully functional Convolutional Neural Network starting from the input layer which always work properly, to convolutional layer which also important to the basic feature selection process and then the pool layer which turns every negative value within the matrix to zero which means detecting the black nodes within the images or the matrix, all of the above layers are implemented right, the last layer which is the most critical layer, it is also implemented right and it is used to extract the very advanced features from the images.

Conclusion:

The system thus aims at providing the Windows operating system users with a cheaper, additional hardware free and accurate emotion-based music system. 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 reducing the unnecessary computational time and thereby increasing the overall accuracy and efficiency of the system. The system will not only reduce physical stress but will also act as a boon for the music therapy systems and may also assist the music therapist to therapize a patient. Also, with its additional features mentioned above, it will be a complete system for music lovers and listeners.

References

- YouTube
- GitHub
- Python Documentation