A Group-Project Report on

EMOTIONAL ANALYSIS BASED CONTENT RECOMMENDATION SYSTEM

Submitted to the Dept. of Information Technology, SNIST
in the partial fulfillment of the academic requirements for the award of

B.Tech (Information Technology)

under JNTUH

by
B. DEEKSHA (17311A12C5)
P. PAWAN (17311A12D2)
K. AKHIL (17311A12E1)

under the guidance of

Dr. B. Indira Reddy
Professor, IT Dept.

Department of Information Technology
School of Computer Science and Informatics
SreeNidhi Institute of Science and Technology (An Autonomous Institution)
Yamnampet, Ghatkesar Mandal, R. R. Dist., Hyderabad – 501301

affiliated to
Jawaharlal Nehru Technological University Hyderabad
Hyderabad – 500 085

2020–21

Department of Information Technology

School of Computer Science and Informatics
SreeNidhi Institute of Science and Technology

Certificate

This is to certify that the Group-Project report on "**EMOTIONAL ANALYSIS BASED CONTENT RECOMMENDATION SYSTEM**" is a bonafide work carried out by B. DEEKSHA (17311A12C5), P. PAWAN (17311A12D2), K. AKHIL (17311A12E1) in the partial fulfillment for the award of B.Tech degree in Information Technology, SreeNidhi Institute of Science and Technology, Hyderabad, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad under our guidance and supervision.

The results embodied in the Group-Project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal guide
Dr. B. Indira Reddy

Project Coordinator
(Dr. K. Kranthi Kumar)

(Dr. V. V. S. S. S. Balaram)

Head of the Department

External Examiner
Date:

DECLARATION

We, B. DEEKSHA (17311A12C5), P. PAWAN (17311A12D2), K. AKHIL (17311A12E1)**,** students of **SreeNidhi Institute of Science and Technology, Yamnampet, Ghatkesar,** studying 4th year 1st semester, **Information Technology** solemnly declare that the Group project work, titled **“EMOTIONAL ANALYSIS BASED CONTENT RECOMMENDATION SYSTEM”** is submitted to **SreeNidhi Institute of Science and Technology** for partial fulfillment for the award of degree of Bachelor of technology in **Information Technology**.

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

Acknowledgements

We would like to express our immense gratitude and sincere thanks to **Dr. B. Indira Reddy; Professor, IT Dept.** for her guidance, valuable suggestions and encouragement in completing the Group-Project work within the stipulated time.

We would like to express our sincere thanks to **Dr. P. Narasimha Reddy**, Executive Director, **Dr. T. Ch. Siva Reddy**, Principal, **Dr. V. V. S. S. S. Balaram,** Professor & Head of the Department of Information Technology, **Dr. K. Kranthi Kumar**, Associate Professor & Group-Project Work Coordinator of the Department of Information Technology, SreeNidhi Institute of Science and Technology (An Autonomous Institution), Hyderabad for permitting us to do our Group-Project work.

Finally, we would also like to thank the people who have directly or indirectly helped us and parents and friends for their cooperation in completing the Group-Project work.

B. DEEKSHA (17311A12C5)
P. PAWAN (17311A12D2)
K. AKHIL (17311A12E1)

Abstract

As the saying goes, “We are what we see”; the content we see may have an adverse effect on our behavior sometimes. Especially in a country like India, where numerous films and TV series are highly prominent, there are great chances of watching explicit or disturbing content randomly. This may have adverse effects on behavior of people, especially children. And we also know “Prevention is better than cure”. Preventing inappropriate content from going online can be more effective than banning them after release.

To achieve this, we aim to create a content filtering and recommendation system that either recommends a film or TV series or alerts a user with a warning message saying it’s not recommended to watch. Netflix or any other Over-the-top (OTT) platforms perform a filtering process before they buy digital rights for any content. This is where our tool comes handy. It detects absurd or hard emotion inducing content with the help of human emotions. Through this project we aim to create a content detector based on human emotion recognition. We will project scenes to test audience and capture their live emotions. Then we use “Facebook Deep Face”, a pre-defined CNN based face recognition and facial emotion analysis model to identify faces and analyze their emotions.

We use “Deep Learning” methods to recognize facial expressions and then make use of Circumplex Model proposed by James Russell to classify emotions based on arousal and valence values. Based on majority emotion that is projected by audience we would either recommend or not recommend the content for going on-air. This system prevents inappropriate content from going on-air.

Team Members:
Name: K. Akhil
Roll Number: 17311A12E1
Contact: 9182870494
Email:kanukulaakhil2017@gmail.com

Name: P. Pawan
Roll Number: 17311A12D2
Contact: 9515872537

Name: B. Deeksha
Roll Number: 17311A12C5
Contact: 8712788690

Guide: Dr. B. Indira Reddy
Designation: Professor, IT Dept.
Contact: 9848698052
Email: bindira@sreenidhi.edu.in
Signature:

INDEX

1. INTRODUCTION

1.1 Scope

The main scope of this project is to create an awareness among people regarding how watching content available in various sources can impact them and their lives and also implement an apt emotional analysis-based content recommendation system which filters and refines the huge amount of content, ensures safe, healthy and appropriate content to watch and prevents filthy and inappropriate content from going on air in the OTT platforms.

1.2 Existing System:

The Process of Censoring a feature film In India

The general process for films releasing in theatres is as follows:

1) Reel wise length of the film is witnessed by the Film lab inspector. The lab inspector’s letter statement about the film length and his/her confirmation that the whole work of the film is completed and it is ready for the preview plays a crucial role in censoring.

2) The assigned supervision inspector, analyzes the application and passes the report to the Examination Committee. Again, the application will be previewed by the examination committee for verifying the primary report. That preview process will take around 8-15 days. Then, the examination committee forwards the report to the chairman.

3) The chairman or his representative reports the status of application to the producer and they also ask the producer to submit the film in front of CBFC within 14 days

This process is different for OTT platforms. There is no censorship for OTT content as of now. Though few objectionable scenes are being removed after the content is going on air , it is too late for that. This is increasing inappropriate content in the market. This can lead to a havoc in future.

1.3 Proposed Censor System

There are a large number of drawbacks in the current censor system where there might be few fraudulent activities that lead to undeserved censor certificates for movies, also it’s a time taking process. This drawback can be easily handled by the automated censorship using emotions.

We are proposing an automated censorship application that can prevent fraudulent censorship and recommends only apt content to go online. In this proposed system the activities done by the censor review committee are performed by the Deep Learning algorithms. In this procedure the live emotions of the viewer are captured by the camera and emotions like happiness, sadness, fear etc are identified by the trained algorithm. Based on the emotions captured it gives the review whether the movie is to be watched by the audience or not. Based on this system the movies can be categorized into comedy, horror, sad fields.

It also tries to give appropriate warnings to the user like “Face Not Detected”, “Movie is Recommended” ,”Movie is not Recommended” etc.

2. SYSTEM ANALYSIS

2.1 Software Requirements:

Python IDE:

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include: An editor designed to handle code and understand your code much better than a text editor. It usually provides features such as build automation, code linting, testing and debugging. This can significantly speed up your work. The downside is that IDEs can be complicated to use.

Visual Studio:

The term Visual denotes a brand-name relationship with other Microsoft programming languages such as Visual Basic, Visual FoxPro, Visual J# and Visual C++. All of these products are packaged with a graphical IDE and support rapid application development of Windows-based applications. Visual Studio is an integrated development environment that is used to develop computer programs for Windows. Visual studio can also be used for developing web sites, web applications, and web services.

2.2 Hardware Requirements:

GPU (Graphics Processing Unit):

A programmable processor specialized for rendering all images on the computer's screen. A GPU provides the fastest graphics processing, and for gamers, the GPU is a stand-alone card plugged into the PCI Express (PCIe) bus. GPU circuitry can also be part of the motherboard chipset or on the CPU chip itself.
A GPU performs parallel operations. Although it is used for 2D data as well as for zooming and panning the screen, a GPU is essential for smooth decoding and rendering of 3D animations and video. The more sophisticated the GPU, the higher the resolution and the faster and smoother the motion. GPUs on stand-alone cards include their own memory, while GPUs built into the chipset or CPU chip share main memory with the CPU. Since GPUs perform parallel operations on multiple sets of data, they are increasingly used for scientific and AI applications that require repetitive computations. For example, in 2010, a Chinese supercomputer achieved the record for top speed using more than seven thousand GPUs in addition to its CPUs.

Webcam:

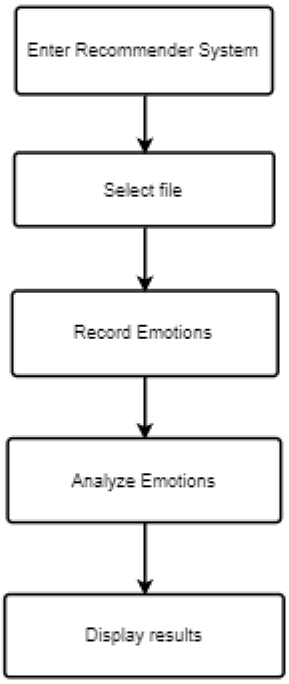
A webcam is a video camera that feeds or streams an image or video in real time to or through a computer to a computer network, such as the Internet. Webcams are typically small cameras that sit on a desk, attach to a user's monitor, or are built into the hardware. Webcams can be used during a video chat session involving two or more people, with conversations that include live audio and video. Webcam software enables users to record a video or stream the video on the Internet. As video streaming over the Internet requires much bandwidth, such streams usually use compressed formats. The maximum resolution of a webcam is also lower than most handheld video cameras, as higher resolutions would be reduced during transmission. The lower resolution enables webcams to be relatively inexpensive compared to most video cameras, but the effect is adequate for video chat sessions.

The term "webcam" (a clipped compound) may also be used in its original sense of a video camera connected to the Web continuously for an indefinite time, rather than for a particular session, generally supplying a view for anyone who visits its web page over the Internet. Some of them, for example, those used as online traffic cameras, are expensive, rugged professional video cameras.

3. SYSTEM DESIGN

3.1 Architecture Design

The first step in this system is entering recommender system interface created by Kivy framework. The second step is to select the video file from file system which will automatically invoke web camera. This web camera takes key facial emotions of user and send them to DeepFace Model. The model then evaluates the emotions by recognizing and analyzes them. Then we plot these emotions using matlplotlib and the recommender system either recommends or not recommends the film.



3.2 Modules:

OpenCV (Open-Source Computer Vision Library)

It 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. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many start-ups such as Applied Minds, Video Surf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick-up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a template interface that works seamlessly with STL containers.

To install OpenCV, the command is: **pip install Opencv-contrib-python print**

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. Itis used for both research and production at Google, TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). Tensor Flow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems, not least RankBrain in Google search and the fun DeepDream project. It can run on single CPU systems, GPUs as well as mobile devices and large-scale distributed systems of hundreds of machines.

To install tensorflow, the command is: **pip install tensorflow**

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. Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. Keras is a minimalist Python library for deep learning that can run on top of Theano or Tensor Flow. It was developed to make implementing deep learning models as fast and easy as possible for research and development. It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.

To install keras, the command is: **pip install keras**

Numpy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object

- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

To install numpy, the command is: **pip install numpy**

Scipy

SciPy in Python is an open-source library used for solving mathematical, scientific, engineering, and technical problems. It allows users to manipulate the data and visualize the data using a wide range of high-level Python commands. SciPy is built on the Python NumPy extension. SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. Like NumPy, SciPy is open source so we can use it freely.

To install scipy, the command is: **pip install scipy**

Kivy

Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps. Kivy runs on Linux, Windows, OS X, Android, iOS, and Raspberry Pi. You can run the same code on all supported platforms .It can natively use most inputs, protocols and devices including WM_Touch, WM_Pen, Mac OS X Trackpad and Magic Mouse, Mtdev, Linux Kernel HID, TUIO. A multi-touch mouse simulator is included.

To install Kivy, the command is:

```
python -m pip install "kivy[base] @ https://github.com/kivy/kivy/archive/master.zip"
```

PySimpleGui

PySimpleGUI is a Python package that enables Python programmers of all levels to create GUIs. You specify your GUI window using a "layout" which contains widgets (they're called "Elements" in PySimpleGUI). Your layout is used to create a window using one of the 4 supported frameworks to display and interact with your window. Supported frameworks include tkinter, Qt, WxPython, or Remi. The term "wrapper" is sometimes used for these kinds of packages

To install PySimpleGUI, the command is: **pip install PySimpleGUI**

Os

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system.

To install Os, the command is:**pip install os**

Deepface: A robust module from FaceBook Team

Deepface is a lightweight facerecognition and facial attribute analysis (age, gender, emotion and race) framework for python. It is a hybrid face recognition framework wrapping **state-of-the-art** models: VGG-Face, Google FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace and Dlib. The library is mainly based on Keras and TensorFlow.

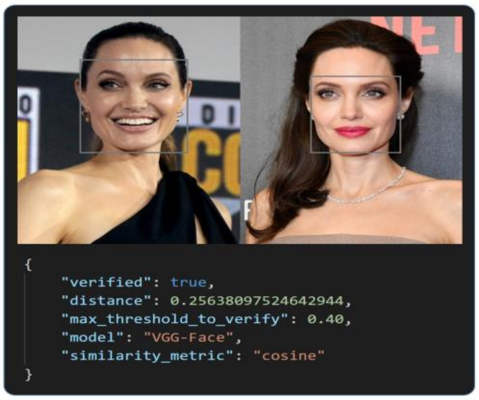
Face Recognition

A modern **face recognition pipeline** consists of 4 common stages: detect, align, represent and verify. Deepface handles all these common stages in the background. You can just call its verification, find or analysis function in its interface with a single line of code.

A) Face Verification

Verification function under the deepface interface offers to verify face pairs as same person or different persons. You should pass face pairs as array instead of calling verify function in a for loop for the best practice. This will speed the function up dramatically and reduce the allocated memory.

```
from deepface import DeepFace
result  = DeepFace.verify("img1.jpg", "img2.jpg")
#results = DeepFace.verify([['img1.jpg', 'img2.jpg'], ['img1.jpg', 'img3.jpg']])
print("Is verified: ", result["verified"])
```



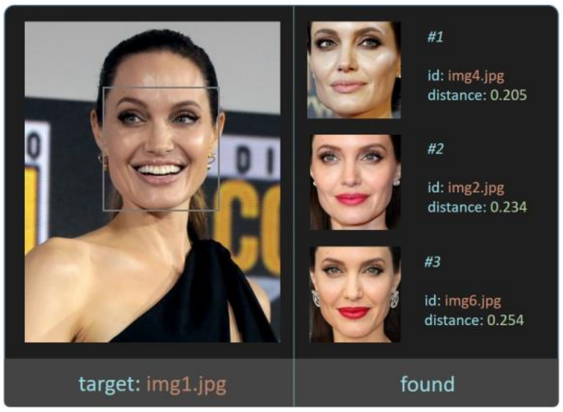
```
{
    "verified": true,
    "distance": 0.25638097524642944,
    "max_threshold_to_verify": 0.40,
    "model": "VGG-Face",
    "similarity_metric": "cosine"
}
```

deepFace

B) Face recognition

Face recognition requires to apply face verification several times. Herein, deepface offers an out-of-the-box find function to handle this action. It stores the representations of your facial database and you don't have to find it again and again. In this way, you can apply face recognition data set as well. The find function returns pandas data frame if a single image path is passed, and it returns list of pandas data frames if list of image paths are passed.

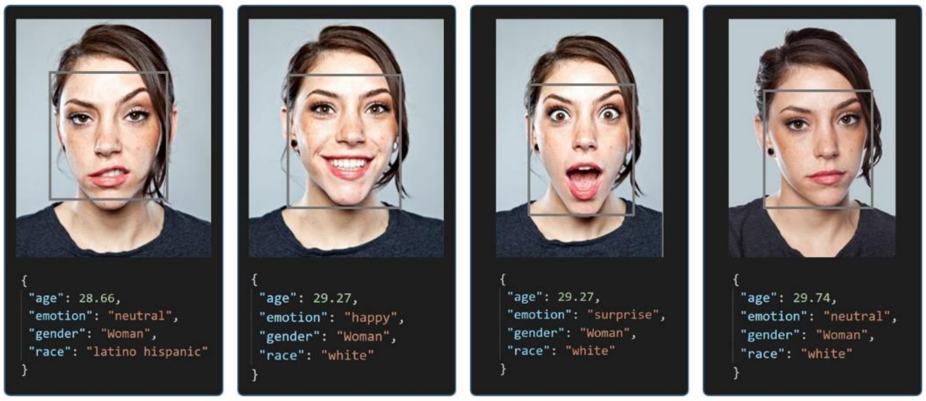
```
from deepface import DeepFace
import pandas as pd
df = DeepFace.find(img_path = "img1.jpg", db_path = "C:/workspace/my_db")
#dfs = DeepFace.find(img_path = ["img1.jpg", "img2.jpg"], db_path = "C:/workspace/my_db")
```



Facial Attribute Analysis

Deepface also offers facial attribute analysis including age, gender, facial expression (including angry, fear, neutral, sad, disgust, happy and surprise) and race (including asian, white, middle eastern, indian, latino and black) predictions. Analysis function under the DeepFace interface is used to find demography of a face.

```
from deepface import DeepFace
obj = DeepFace.analyze(img_path = "img4.jpg", actions = ['age', 'gender', 'race', 'emotion'])
#objs = DeepFace.analyze(["img1.jpg", "img2.jpg", "img3.jpg"]) #analyzing multiple faces same time
print(obj["age"]," years old ",obj["dominant_race"]," ",obj["dominant_emotion"]," ", obj["gender"])
```



```
{
 "age": 28.66,
 "emotion": "neutral",
 "gender": "Woman",
 "race": "latino hispanic"
}
```

```
{
 "age": 29.27,
 "emotion": "happy",
 "gender": "Woman",
 "race": "white"
}
```

```
{
 "age": 29.27,
 "emotion": "surprise",
 "gender": "Woman",
 "race": "white"
}
```

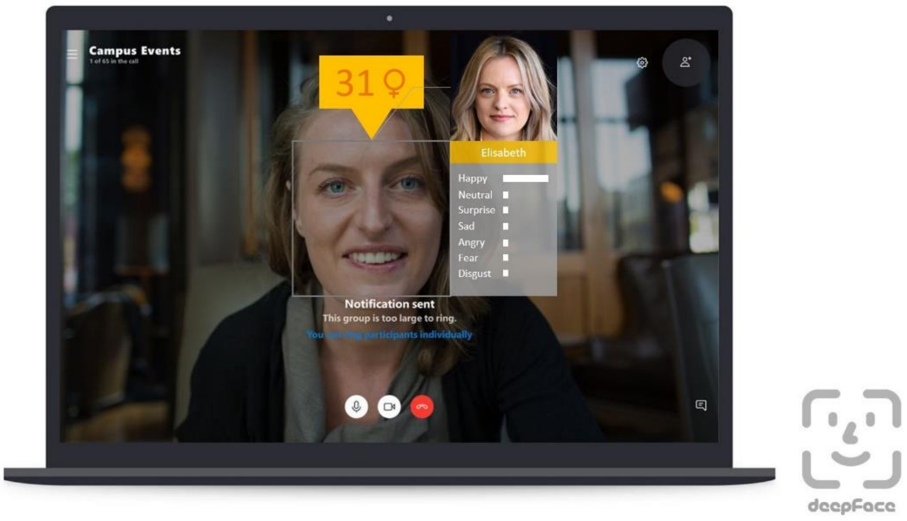
```
{
 "age": 29.74,
 "emotion": "neutral",
 "gender": "Woman",
 "race": "white"
}
```

Streaming and Real Time Analysis

You can run deepface for real time videos as well.

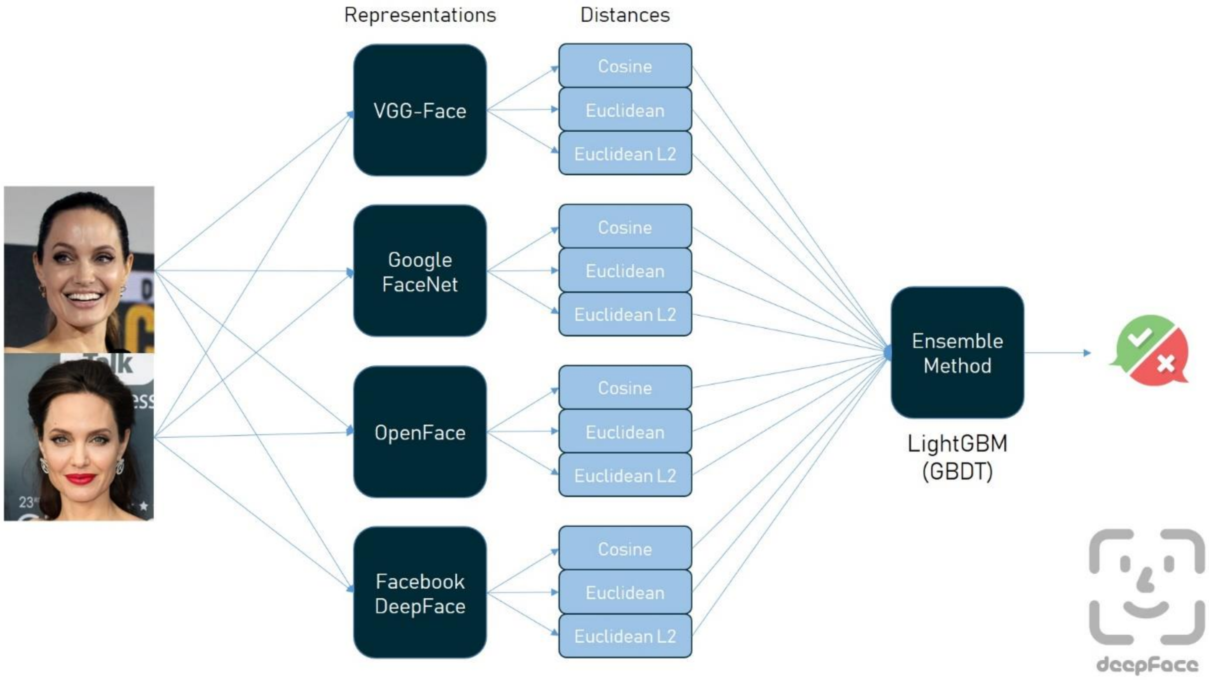
Calling stream function under the DeepFace interface will access your webcam and apply both face recognition and facial attribute analysis. Stream function expects a database folder including face images. VGG-Face is the default face recognition model and cosine similarity is the default distance metric similar to verify function. The function starts to analyze if it can focus a face sequantially 5 frames. Then, it shows results 5 seconds.

```
from deepface import DeepFace
DeepFace.stream("C:/User/Sefik/Desktop/database")
```



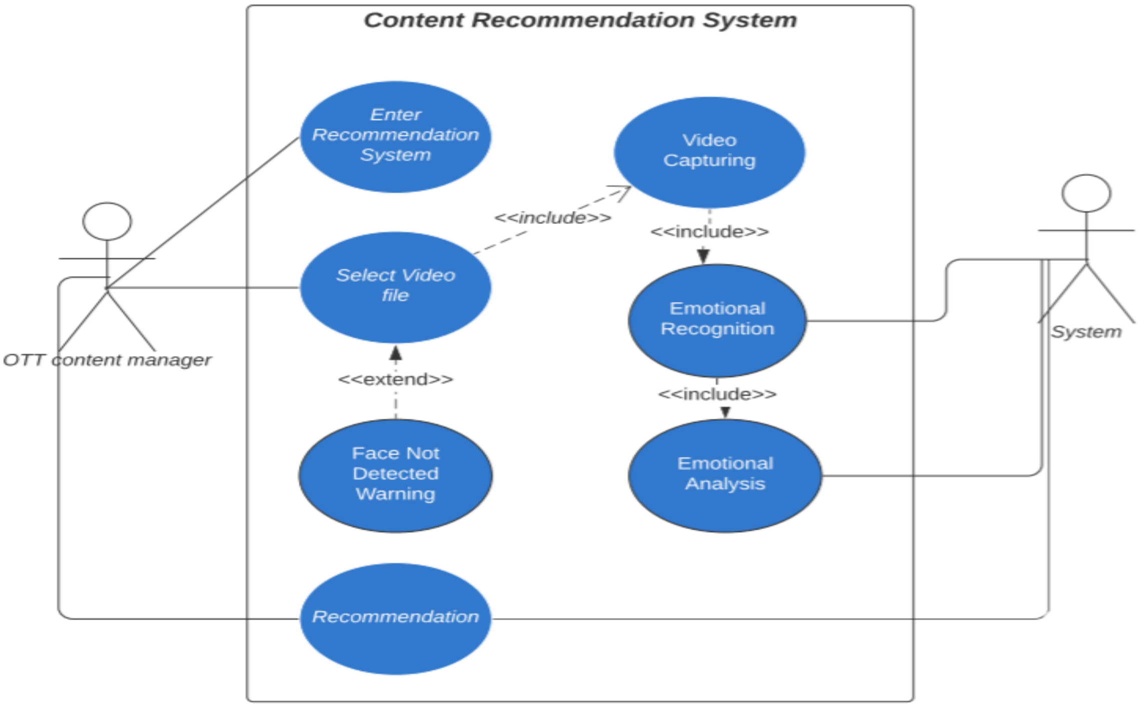
Ensemble learning for face recognition

A face recognition task can be handled by several models and similarity metrics. Herein, deepface offers a special boosting and combination solution to improve the accuracy of a face recognition task. This provides a huge improvement on accuracy metrics. Human beings could have 97.53% score for face recognition tasks whereas this ensemble method passes the human level accuracy and gets 98.57% accuracy. On the other hand, this runs much slower than single models.

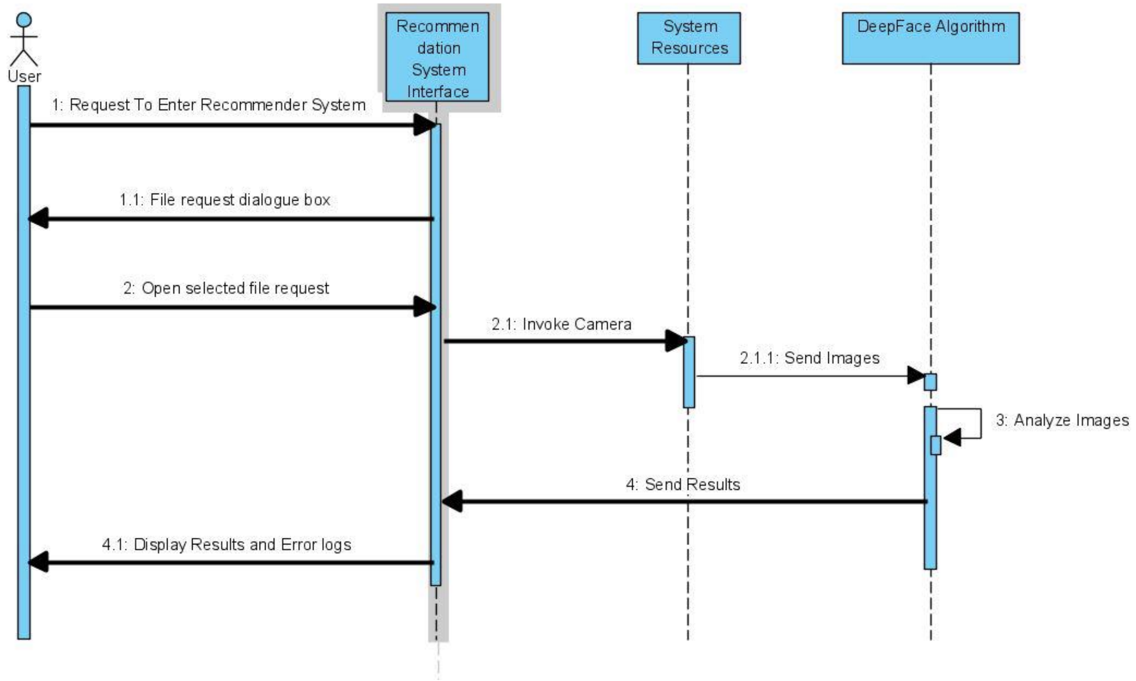


3.3 UML diagrams

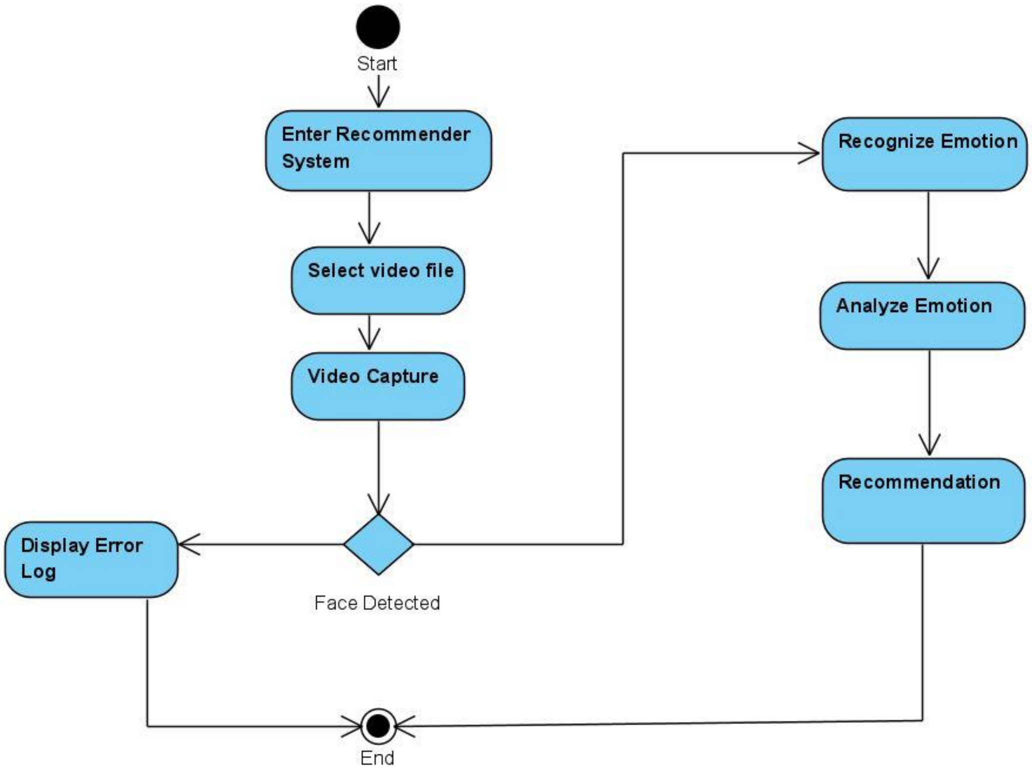
3.3.1 Use Case Diagram



3.3.2 Sequence Diagram


User
Recommendation System Interface
System Resources
DeepFace Algorithm
1: Request To Enter Recommender System
1.1: File request dialogue box
2: Open selected file request
2.1: Invoke Camera
2.1.1: Send Images
3: Analyze Images
4: Send Results
4.1: Display Results and Error logs

3.3.3 Activity Diagram



4. SYSTEM IMPLEMENTATION

The Kivy framework is used to create a simple yet effective interface for our emotion based content recommender system. The Deepface module created by FaceBook research team provides an enhanced highly accurate emotion recognizer that would help us analyze the content present in the specified video. After evaluation it either says the film has apt content or not.

The source code in Python 3.8 is as follows:

Code:

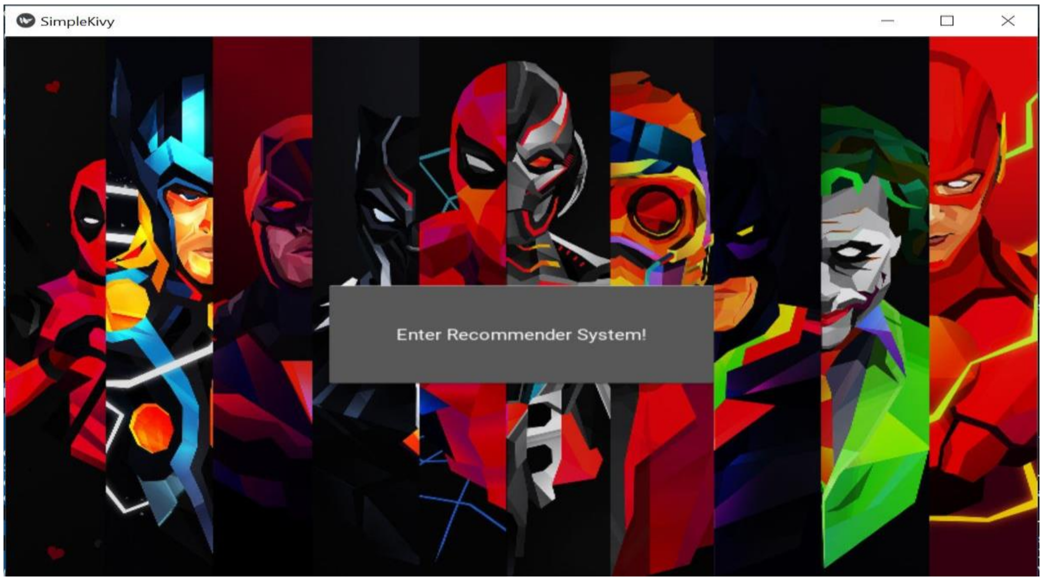
```
from kivy.app import App

from kivy.core.window import Window

from kivy.uix.boxlayout import BoxLayout

from kivy.uix.button import Button

from kivy.uix.label import Label

from kivy.uix.anchorlayout import AnchorLayout

from kivy.logger import Logger

from kivy.uix.widget import Widget

import PySimpleGUI as sg

from os import startfile

from deepface import DeepFace

import cv2 as cv

import matplotlib.pyplot as plt

import os



# From graphics module we are importing
# Rectangle and Color as they are
# basic building of canvas.
from kivy.graphics import Rectangle, Color


class myapp(BoxLayout):
    def __init__(self,**kwargs):
        super(myapp,self).__init__(**kwargs)
        self.padding = 250

        # Arranging Canvas
        with self.canvas:
            # Seting the size and position of image
            # image must be in same folder
            self.rect = Rectangle(source ='D:/bg.jpg',
                                  pos = self.pos, size = self.size)

            # Update the canvas as the screen size change
            # if not use this next 5 line the
            # code will run but not cover the full screen
            self.bind(pos = self.update_rect,
                  size = self.update_rect)
            layout = AnchorLayout(anchor_x ='center', anchor_y ='center')
            txt=Label(text="[color=ff3333]Welcome to Recommender System",markup=True)
        layout.add_widget(txt)


    # update function which makes the canvas adjustable.

    def update_rect(self, *args):

        self.rect.pos = self.pos

        self.rect.size = self.size



        btn = Button(text='Enter Recommender System!')

        btn.bind(on_press=self.proj_func)

        self.add_widget(btn)



    def clkfunc(self , obj):

        App.get_running_app().stop()

        Window.close()

    def proj_func(self,obj):

        #import all the packages



        #Gui for choosing a file

        sg.theme('DarkAmber')

        fname = sg.popup_get_file('Select a video file')

        startfile(fname)



        #Deepface Emotional Analysis
```

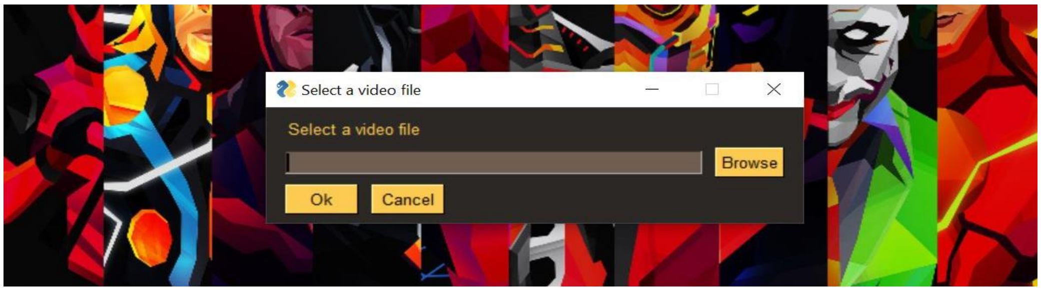
```
cap=cv.VideoCapture(0)

result_new=[]

try:

    while(True):

        _, img =cap.read()

        result_new.append(DeepFace.analyze(img,actions=['emotion']))

        cv.imshow('img',img)

        if(cv.waitKey(1) & 0xFF==ord('q')):

            cap.release()

            break

except:

    layout = [  [sg.Text("Face not detected \U0001F614",auto_size_text=True,size=(35, 1), font=("Calibri", 24),
text_color='orange')]]

    sg.theme('DarkAmber')

    window=sg.Window("Recommendations",layout)

    event, values = window.read()

    window.close()



#determining captured emotions

emotion_set=[i['dominant_emotion'] for i in result_new]

dom_set=[]

for i in result_new:

    dom=str(i['dominant_emotion'])
    dom_set.append(i['emotion'][dom])


dominant_emotions=dict()

fear=0

neutral=0

sad=0

happy=0

angry=0

surprise=0


#existing emotions list

emotion_li=['fear','neutral','sad','happy','angry','surprise']


#finding out emotion occurences

for i in emotion_li:

    dominant_emotions[i]=0

for i in emotion_set:

    if(i in dominant_emotions):

        dominant_emotions[i]+=1


#Matplotlib to plot emotions

fig=plt.figure(1)

ax = fig.add_axes([0,0,1,1])
```

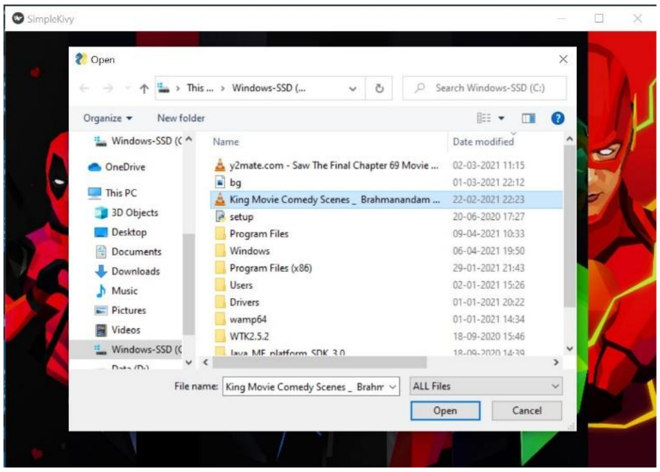
```
x=emotion_li
y=[dominant_emotions[i] for i in dominant_emotions]
ax.bar(x,y,color='r')
for i in ax.patches:
    # get_x pulls left or right; get_height pushes up or down
    ax.text(i.get_x()+.04, i.get_height()+0.4, \
    str(round((i.get_height()), 2)), fontsize=11, color='black',
        rotation=0)


#emotion values based on arousal and valence of emotions
emotion_matrix={'fear': -5, 'neutral': 1, 'sad': -3, 'happy': 5, 'angry': -4, 'surprise': 2}
#finding out emotion's cummulative value
arr=[]
analysis=[]
for j in dominant_emotions:
    arr.append(dominant_emotions[j]*emotion_matrix[j])
value=sum(arr)
#if emotion is positive
if(value>10):
    layout = [  [sg.Text("Yay! This film can be recommended! \U0001f600",auto_size_text=True,size=(35, 1), font=("Calibri", 24), text_color='green')],
      [sg.Text("This would make a great watch!",auto_size_text=True,size=(35, 1), font=("Calibri", 18), text_color='cyan')]]
    sg.theme('DarkAmber')
```

```
            window=sg.Window("Recommendations",layout)

            event, values = window.read()

            window.close()



#if emotion is negative

        else:

            layout = [  [sg.Text("Nay! This can't be recommended! \U0001F614",auto_size_text=True,size=(35, 1), font=("Calibri", 24), text_color='red')],

             [sg.Text("We would suggest few trims!",auto_size_text=True,size=(35, 1), font=("Calibri", 18), text_color='yellow')]]

            sg.theme('DarkAmber')

            window=sg.Window("Recommendations",layout)

            event, values = window.read()

            window.close()



class SimpleKivy(App):

    def build(self):

        return myapp()

if __name__ == '__main__':

    SimpleKivy().run()
```

5. Output:

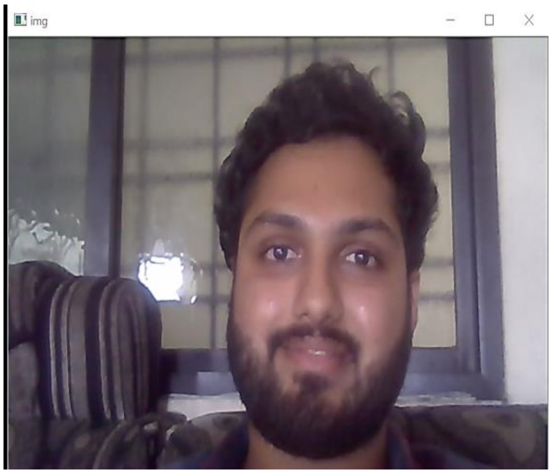




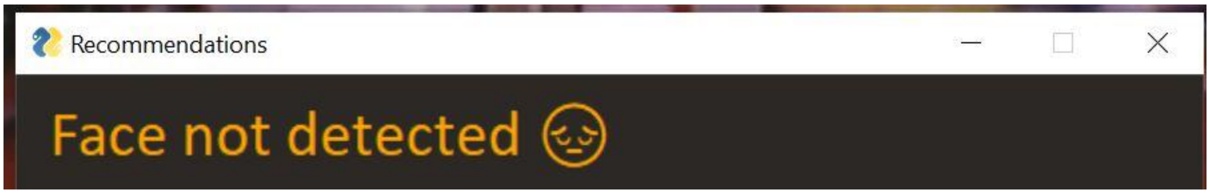


Name	Date modified
y2mate.com - Saw The Final Chapter 69 Movie ...	02-03-2021 11:15
bg	01-03-2021 22:12
King Movie Comedy Scenes _ Brahmanandam ...	22-02-2021 22:23
setup	20-06-2020 17:27
Program Files	09-04-2021 10:33
Windows	06-04-2021 19:50
Program Files (x86)	29-01-2021 21:43
Users	02-01-2021 15:26
Drivers	01-01-2021 20:22
wamp64	01-01-2021 14:34
WTK2.5.2	18-09-2020 15:46
Java ME platform SDK 3.0	18-09-2020 14:39

File name: King Movie Comedy Scenes _ Brahm

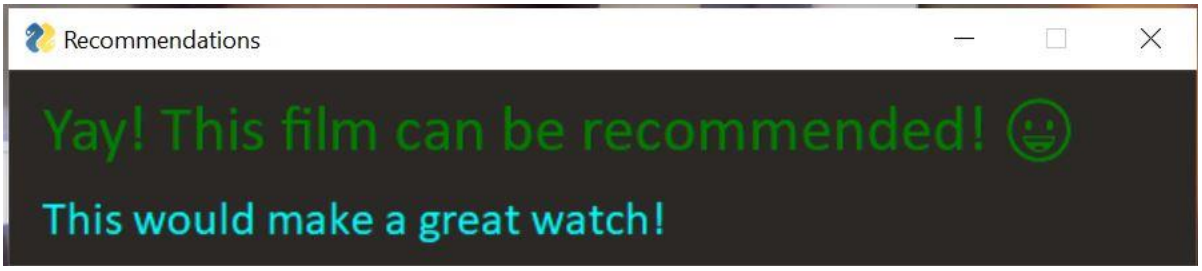


1.If face not detected



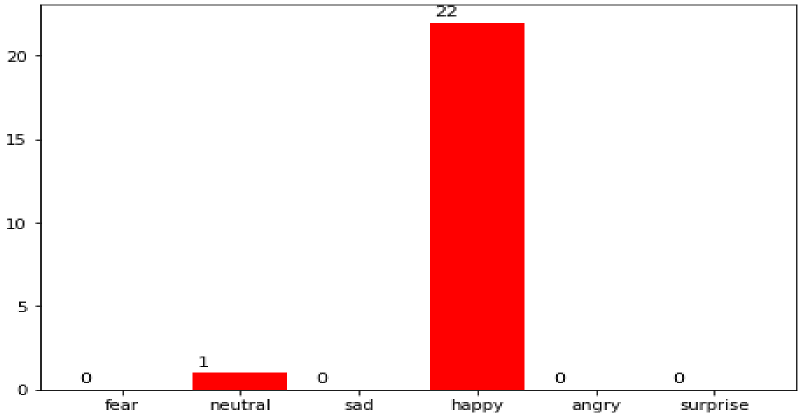
Face not detected 😔

2.Movie with Positive emotions:

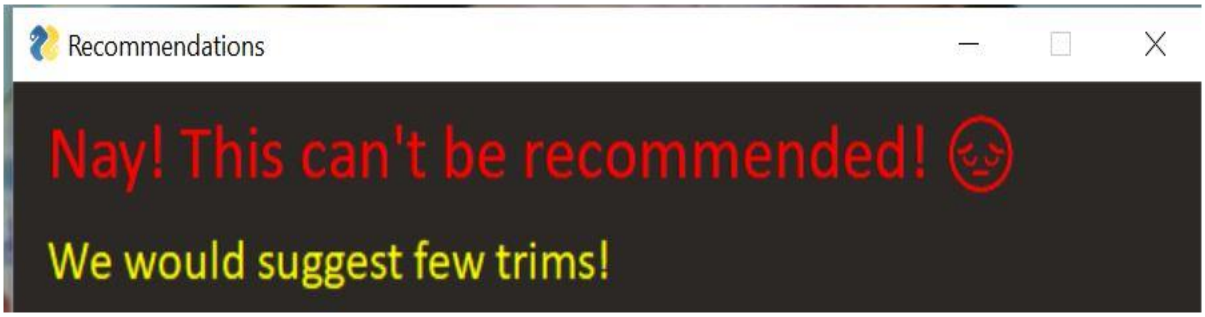


Yay! This film can be recommended! 😀

This would make a great watch!

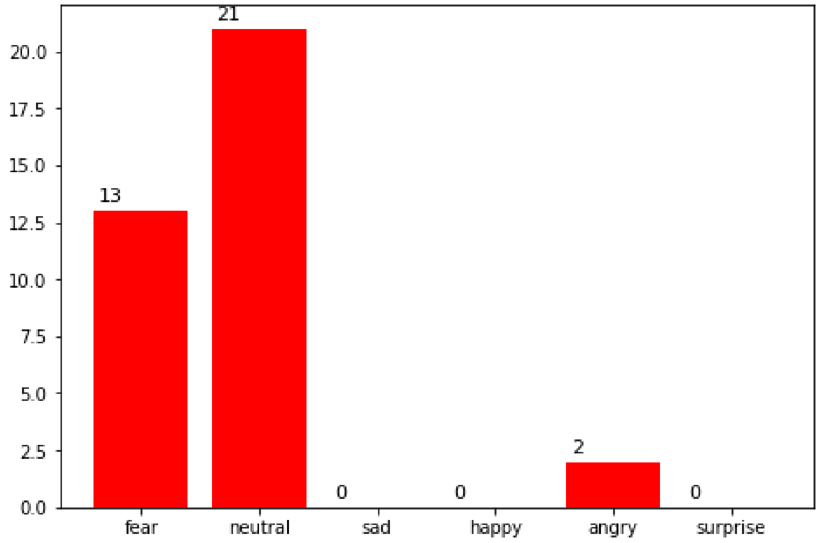


3.Movie with negative emotions



Nay! This can't be recommended! 😔

We would suggest few trims!



6. APPLICATIONS

1) The emotional analysis-based content recommendation system is widely used to recommend movies, music, web series and all other types of content to the users as per their interests by basing upon their emotions displayed regardless of the OTT platforms they choose.
2) This system is used to brilliantly enhance the value and functioning of print and electronic media domains.
3) Recommendation systems can also be used and modified to provide few other services such as:
   e-commerce
   e-library
   e-banking
   e-business
   e-government
4) This can be used by various business firms and organizations to facilitate effective market research.
5) This recommendation system is used to precisely understand users' interests, priorities and what they exactly want.
6) It is also used to increase the reach of: websites, nearby tourism spots, sports news and other updates among a wide range of audience.

7. ADVANTAGES

1) Effective content filtering:

   The emotional analysis-based content recommendation system efficiently and effectively filters and refines the content available
2) Relevant content:

   The recommendation system ensures only relevant data is delivered to the users

3) User-friendly:

The emotional analysis-based content recommendation system

4) Easy:

This is very easy to implement and understand

5) Report/analysis:

The emotional analysis-based content recommendation system also provides a detailed report and analysis with respect to the emotions and interests of users

6) Robust:

The process of filtering and refining of the content is very robust

DISADVANTAGES

1) Requires highly skilled set of people to solve technical issues.
2) Indistinct and appropriate movements and gestures are likely to be possible while capturing of users' emotions which may lead to few errors.
3) If any user fakes emotions, the system cannot distinguish between deliberate faking and genuine emotions.
4) Varying face angles can cause problems while face detection.

8. CONCLUSION

Through our project, we hope that any OTT platform can implement this kind of recommendation system and ensure users that only healthy and appropriate media content is recommended and brought forward. Though the recommendation systems do their job in filtering and refining the content, it finally depends on the users regarding what their interests are and what they want to they watch. As nothing in this world is perfect, there are few limitations even in content recommendation systems. But as the needs of people keep changing daily; and as there are many rapid advancements and developments in the fields of science, technology, R&D; we can be definitely sure of achieving best recommendation systems in future; with no or very few limitations and great accuracy.

9.FUTURE SCOPE

More than half of the population all over the world is already familiar with these kinds of recommendation systems as they provide a wide range of applications and services to the users. There is a dire need to refine and filter the media content before it is made available on OTT platforms and before it reaches the public. As the type of content what we watch has a great impact on us, we should be very careful with respect to what we watch. We are planning to improve the features and get rid of limitations using audio-based recognition (sound signals) and release it as an open-source project. Any OTT platform can use our project to filter the content, make sure that only healthy content is available to the users and inappropriate content is prevented from going on air.

10. REFERENCES

[1] Yaniv Taigman; Ming Yang; Marc'Aurelio Ranzato; Lior Wolf (June 24, 2014), "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", Conference on Computer Vision and Pattern Recognition (CVPR), Facebook Research Group

[2] John Bohannon (5 February 2015), "Facebook will soon be able to ID you in any photo", Science (website), American Association for the Advancement of Science, doi:10.1126/science.aaa7804

[3] CHANEL, Guillaume, ANSARI ASL, Karim, PUN, Thierry. Valence-arousal evaluation using physiological signals in an emotion recall paradigm. In: 2007 IEEE International Conference on Systems, Man and Cybernetics, SMC 2007 : proceedings. Institute of Electrical and Electronics Engineers (IEEE), 2007.

[4] Mohammad Soleymani, Guillaume Chanel, Joep Kierkels, Thierry Pun,Thierry . Valence-arousal evaluation using physiological signals in an emotion recall paradigm

[5] . A. Hanjalic and L. Q. Xu, "Affective video content representation and modeling", IEEE Transactions on Multimedia, vol. 7, no. 1, pp. 143--154 (2005)

[6] Posner, Jonathan; Russell, J.A.; Peterson, B. S. (2005). "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology". Development and Psychopathology. **17** (3): 715–734. doi:10.1017/s0954579405050340. PMC 2367156. PMID 16262989.

[7] "Kivy 2.0.0". github.com. 2020-12-10. Retrieved 2021-03-10.

[8]https://pypi.org/project/PySimpleGUI/#:~:text=PySimpleGUI%20is%20a%20Python%20package,%22Elements%22%20in%20PySimpleGUI).

[9] https://pypi.org/project/deepface/