

Recommendations based on your mood

REVIEW 3



TRAKREX

A Complete Biometric-based entertainment and productivity system

An application of Blue Eyes technology to
make every day life better

(USE OF HUMAN SENSES IN A.I. TECHNOLOGY)

GUIDE

Slides:

- Project title
- Our team
- Introduction to the subject
- Existing work and challenges
- Our proposal
- Novelty of our project
- Real-life applications
- Technical requirements
- Architecture diagram
- Literature review
- Module description
- Module work flow explanation
- Implementation and coding
- Demo video
- Implementation screenshots
- Testing
- Result and discussion
- Conclusion



TEAM MEMBERS



MOHAJIT PAUL:
20BCE10630



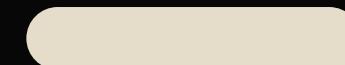
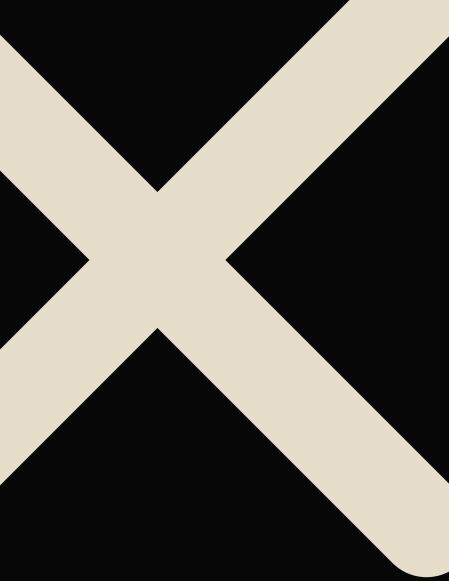
PARITOSH NIMDEO:
20BCE10490

YASH MANIKONDA:
20BCE10535

MADHAV GUPTA:
20BCE10020

AASHAY KULKARNI:
20BCE10402

INTRODUCTION



'Blue Eyes' technology makes use of technical approaches to achieve a balance of cognitive sciences, psychology, and technology.

We have aimed to work on and produce an example that can demonstrate the use of this technology, in the real-world in a way that is practical and creative in its approach.

In this presentation, we will demonstrate the final result we have achieved, and elaborate on what we expect the future of our project to be like.

EXISTING WORK WITH LIMITATIONS

- The applications of Blue Eyes technology that exist are quite primitive, and the potential largely remains untapped due to technical and creative limitations.
- The mood recommendations aspect also comes with its own set of challenges, which make them merely ideal solutions but difficult to put into practice.

Challenge 1: Cold start problem

Challenge 2: Automatic playlist continuation

Challenge 3: Evaluating music recommender systems

Why hasn't this been done before?

PROPOSED WORK AND METHODOLOGY

To build a software that maps a human face and based on tracking, detects the emotions and identifies the person's mood.

To connect the software with entertainment websites and recommend suitable suggestions as per the person's mood.

To integrate the software into a website that can be used to demonstrate the designed product with ease of access.

To show a real-time application based on Blue Eyes technology, and implement it with the help of crowd testing.

NOVELTY OF THE PROJECT

Projects and designs so far that made use of Blue Eyes technology were held back by their own vision and ability to explore creative solutions.

Advancements in this field up until now have been mainly theoretical, with hardly any real practical applications seeing the light of day.

We build on existing work, and take it a step further and integrate it into real-life in a way that is easy to access and simple enough to test out.

What are we doing different?

“

This system works well because it can

- Map your face and apply location co-ordinates to it, to be able to track your facial movements
- Compare this input to the existing database of co-ordinates, and detects your mood based on that
- Recommend you suggestions based on what you wish to do - listen to music, watch movies - and takes you directly to the associated platforms

REAL TIME USAGE

Where can this be used?

Where it can be used:

- Personal use, for when the user wants to just use the recommendations system
- Workplaces, so that employees can use their breaks efficiently so as to continue with their day productively

HARDWARE & SOFTWARE REQUIREMENTS

Hardware requirements

	MINIMUM	RECOMMENDED	HIGH
CPU	Intel Core i5 8th gen AMD Ryzen 5 1600x	Intel Core i5 9th gen AMD Ryzen 5 2600X	Intel Core i7 9th gen AMD Ryzen 7 4800H
RAM	8 GB	12 GB	16GB
GPU	Intel integrated graphics Ryzen integrated graphics	Nvidia GTX 1650 AMD Radeon RX 570	_____
STORAGE	5 GB HDD (SSD recommended)	7 GB SSD	10 GB SSD
WEBCAM	Integrated Webcam (3 MP/720p/30fps)	External Webcam (5MP/1080p/30fps)	_____

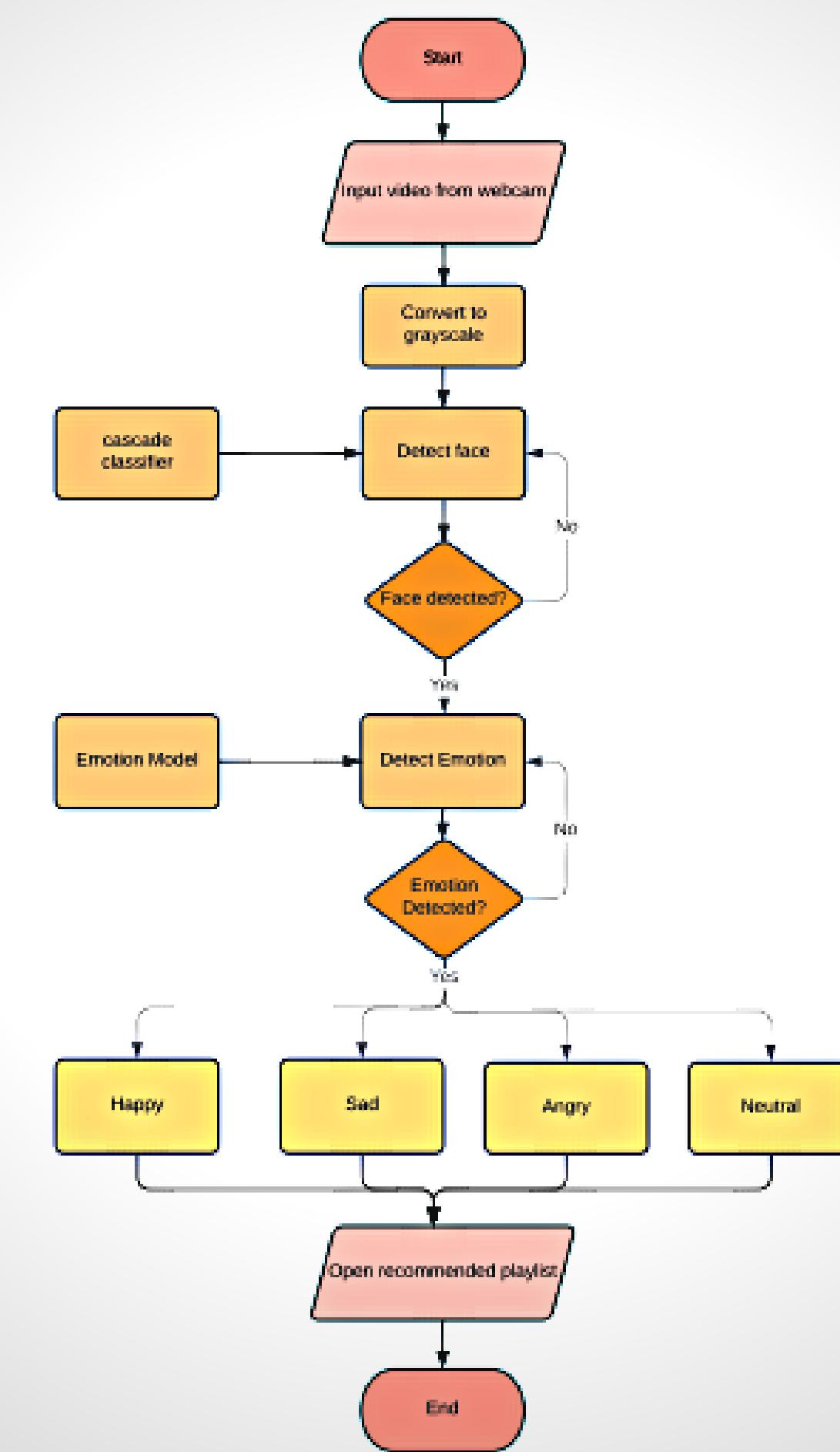
Software requirements

- OS - Windows 10 / 10 pro
- Python 3.8.5
- Pycharm IDE
- Anaconda distribution
- Visual studio code IDE
- Django
- Open CV

OVERALL ARCHITECTURE DIAGRAM AND FLOW DIAGRAM

How does this work?

Flow Diagram



LITERATURE REVIEW

This idea has not been implemented by any mainstream industries, others have tried building emotion detection systems but there are not a lot of projects which give out recommendations to the user.

But there are few research papers based on the core topics of our program. We have studied them, which were mainly based on emotion recognition, DeepFace and OpenCV. It has helped us to build the program from scratch in a less amount of time. The research papers we used are mentioned below.

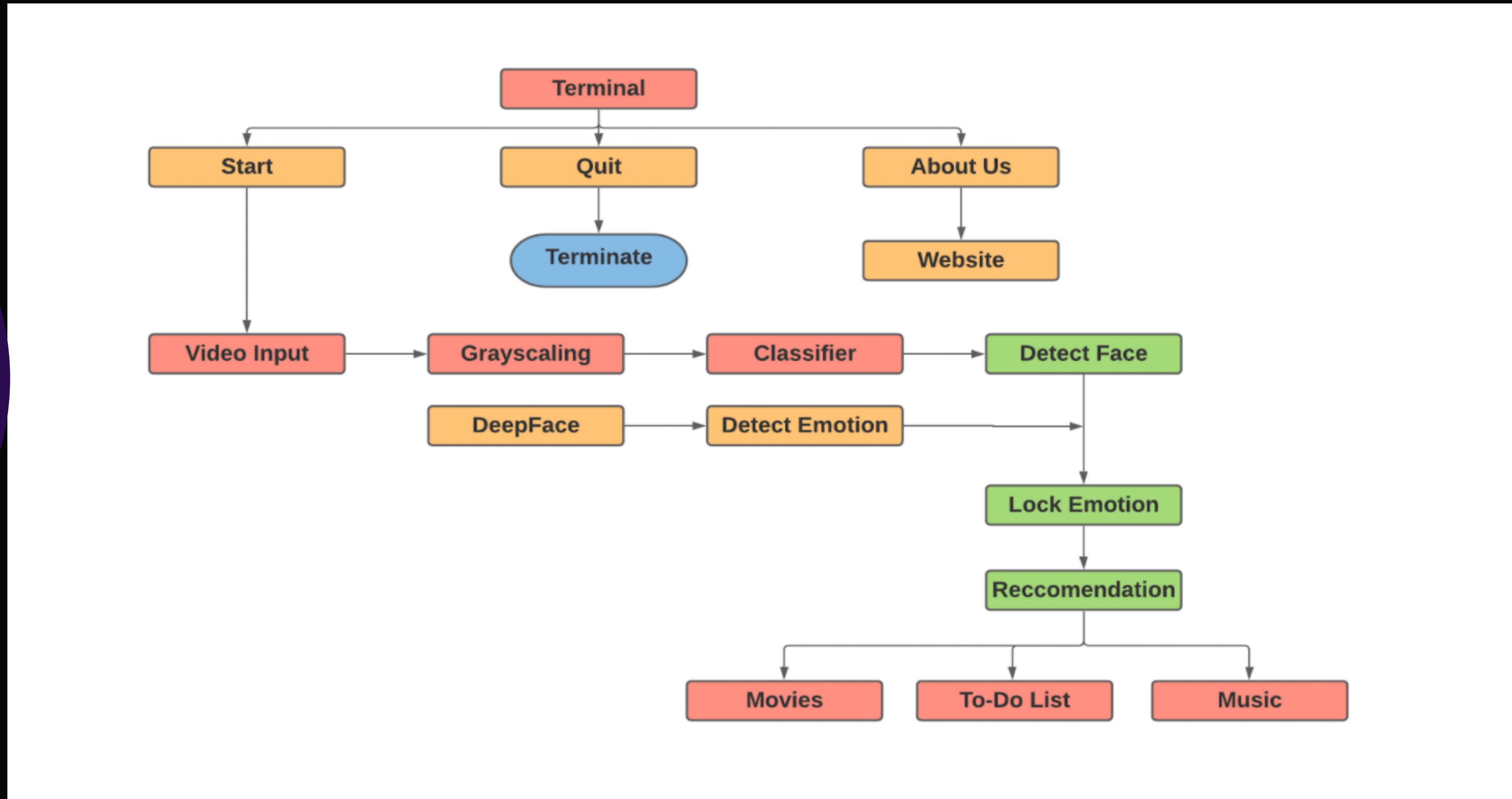
1. M. R. Mizna, M. Bachani and S. Memon, "Blue eyes technology," Eighth International Conference on Digital Information Management (ICDIM 2013), 2013, pp. 294-298, doi: 10.1109/ICDIM.2013.6693995.
2. A. A. Varghese, J. P. Cherian and J. J. Kizhakkethottam, "Overview on emotion recognition system," 2015 International Conference on Soft-Computing and Networks Security (ICSNS), 2015, pp. 1-5, doi: 10.1109/ICSNS.2015.7292443.
3. Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

MODULE DESCRIPTION

REQUIREMENTS FOR THE SOFTWARE

- OPEN-CV (PIP INSTALL OPENCV-PYTHON)
- DEEPFACE (PIP INSTALL DEEPFACE)
- HAAR CASCADE_FRONTALFACE_DEFAULT.XML
(FOUND IN GOOGLE)

MODULE WORK FLOW EXPLANATION



IMPLEMENTATION AND CODING

```
import cv2
from deepface import DeepFace
from tkinter import *
from tkinter import ttk
import webbrowser

root = Tk()
style = ttk.Style()
root.geometry('600x400')
root.configure(background="black")

myLabel1 = Label(root, text="Confused about what to do? Get started with",
font=("Courier", 15), fg='white', bg='black')
myLabel1.pack(padx=20, pady=1)
myLabel2 = Label(root, text="TRAKREX", font=("Arial", 40), fg='#5c007d',
bg='black')
myLabel2.pack(padx=20, pady=10)
myLabel3 = Label(root, text=' "The cure for boredom is curiosity." - Dorothy
Parker', font=("Monotype Corsiva", 20),fg='white', bg='black')

def OpenUrl():
    webbrowser.open('https://trakrexx.wixsite.com/home/about-us')

def QuitBtn():
    root.destroy()
```

```
def run():
    # faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
    faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

    cap = cv2.VideoCapture(1)
    # Check if the webcam is opened correctly
    if not cap.isOpened():
        cap = cv2.VideoCapture(0)

    if not cap.isOpened():
        raise IOError("Cannot open webcam")

    while True:
        ret, frame = cap.read() ## read one image from a video

        result = DeepFace.analyze(frame, actions=['emotion'],
enforce_detection=False)
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # print (faceCascade.empty())
        faces = faceCascade.detectMultiScale(gray, 1.1, 4)
```

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

        font = cv2.FONT_HERSHEY_SIMPLEX

        # Use putText() method for
        # inserting text on video
        cv2.putText(frame,
                    result['dominant_emotion'],
                    (50, 50),
                    font, 3,
                    (0, 0, 255),
                    2,
                    cv2.LINE_4)

        cv2.imshow('Original video', frame)

        dominant_emotion = result['dominant_emotion']

        if cv2.waitKey(2) & 0xFF == ord('q'):
            if dominant_emotion == 'neutral':
                webbrowser.open("https://trakrexx.wixsite.com/neutral")
            if dominant_emotion == 'happy':
                webbrowser.open("https://trakrexx.wixsite.com/happy")
            if dominant_emotion == 'sad':
                webbrowser.open("https://trakrexx.wixsite.com/sadness")
            if dominant_emotion == 'angry':
                webbrowser.open("https://trakrexx.wixsite.com/angry")
            break

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

```
Getstarted = Button(root, text="I'm feeling lucky!", borderwidth=0,  
height=2, width=18, bg='white', fg='#7F14B1', font='Arial', command=run)  
Quit = Button(root, text="Quit", borderwidth=0, bg='white', height=2,  
width=10, fg='#000AFF', font='Arial', command=QuitBtn)  
AboutUs = Button(root, text="About Us", borderwidth=0, bg='#00FFD1',  
width=15, height=2, font='Arial', command=OpenUrl)  
  
Getstarted.pack(padx=20, pady=15)  
Quit.pack(pady=15)  
myLabel3.pack(padx=20, pady=10)  
AboutUs.pack(padx=20, pady=10)  
  
root.mainloop() # making loop to make it run continuously
```

Not secure | trakrex.tech

Apps YouTube Maps Gmail News Translate Reading list

This site was designed with the WIX.com website builder. Create your website today. Start Now



TRAKREX

a mood-based recommendation system

Contact Us About Us

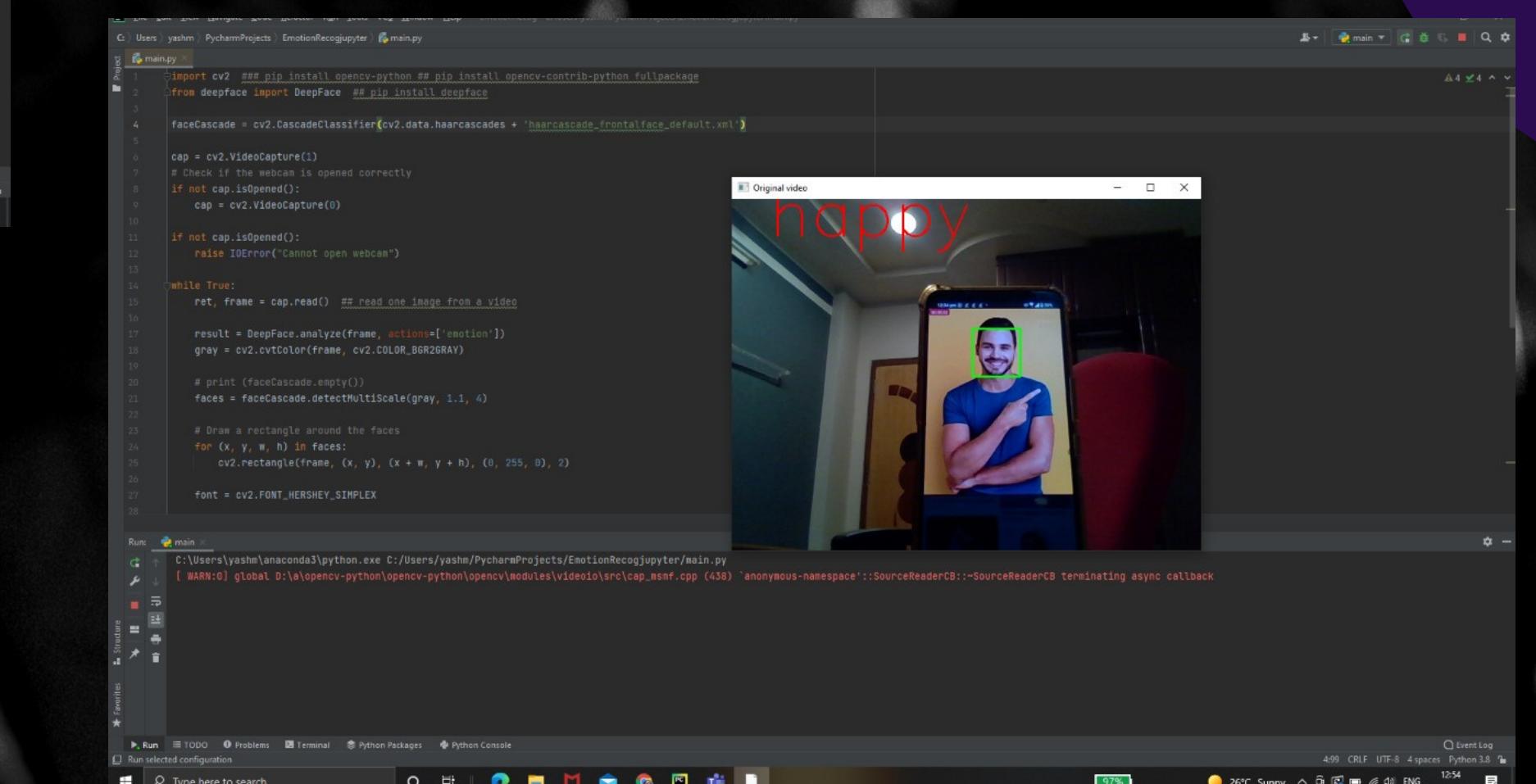
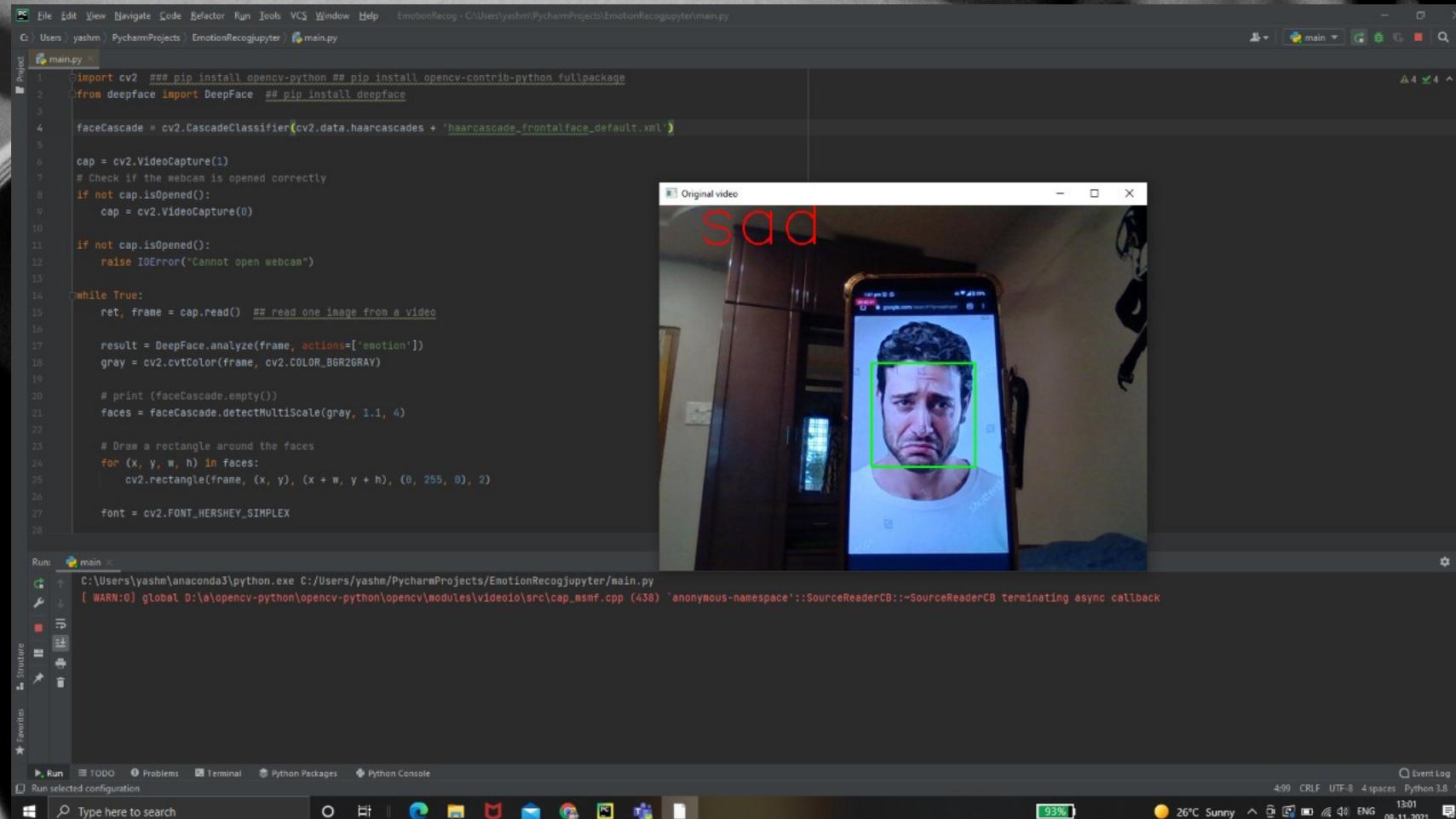
WELCOME TO THE OFFICIAL TRAKREX WEBSITE

DISCOVER

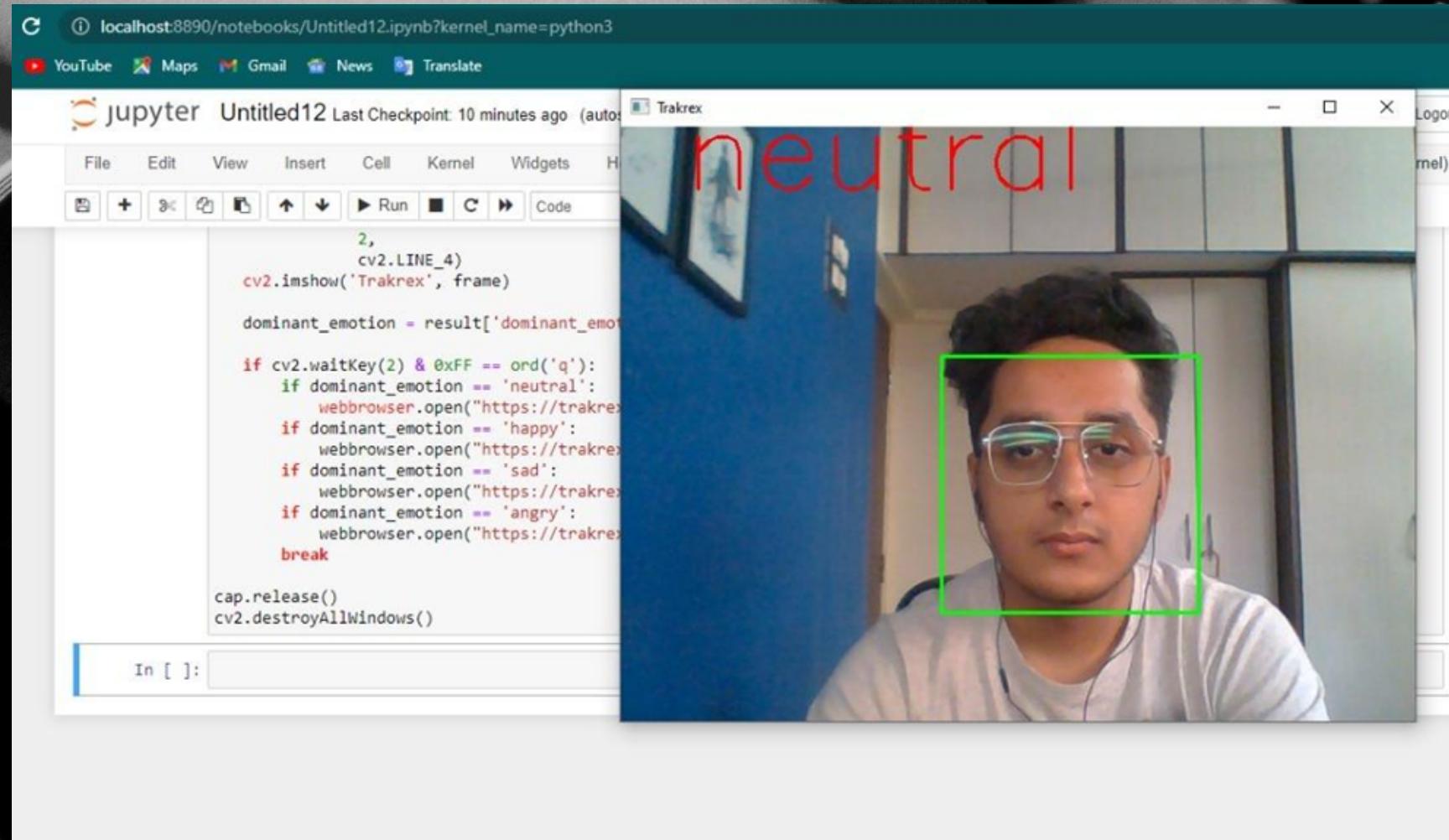


DEMO VIDEO

SNAPSHOT OF PROJECT



SNAPSHOT OF PROJECT



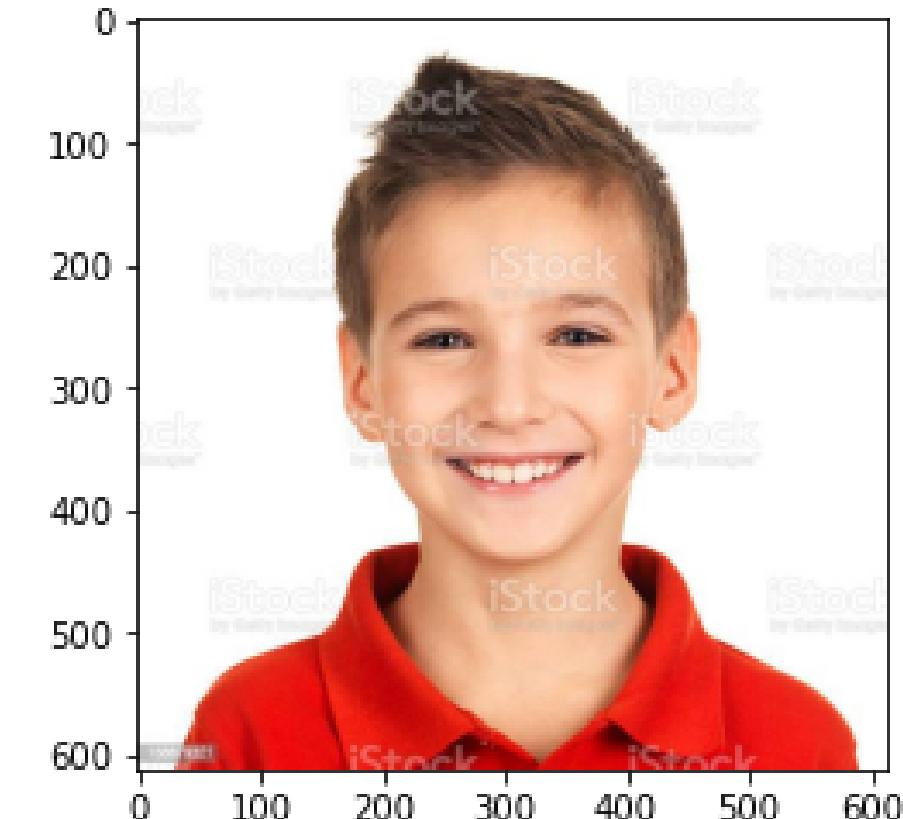
TESTING

```
In [9]: predictions
```

```
Out[9]: {'emotion': {'angry': 3.6894442385404625e-15,
'disgust': 7.147908799450446e-20,
'fear': 3.1471684914570084e-13,
'happy': 99.99997615814209,
'sad': 6.14294381963594e-12,
'surprise': 3.4986905950509795e-11,
'neutral': 2.7220590936849476e-05},
'dominant_emotion': 'happy',
'region': {'x': 150, 'y': 137, 'w': 312, 'h': 312},
'age': 23,
'gender': 'Man',
'race': {'asian': 0.041313385904661014,
'indian': 0.14181481147902472,
'black': 0.004924989620550842,
'white': 77.04637920303593,
'middle eastern': 10.723678234377118,
'latino hispanic': 12.041888440622277},
'dominant_race': 'white'}
```

```
In [6]: plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
```

```
Out[6]: <matplotlib.image.AxesImage at 0x1e9d483aa90>
```



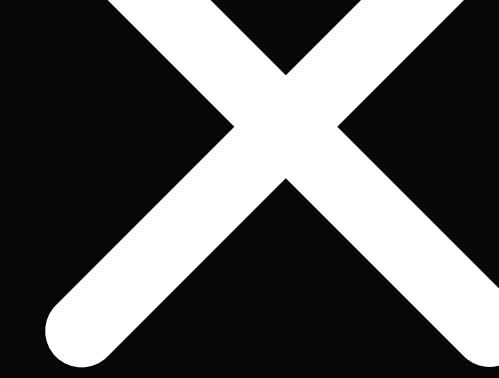
The program we have developed uses the data from deepface to analyze the faces and detect the emotions. As you can see in the above testing, the dominant emotion is 'happy' with an accuracy of 99.99 points

RESULT AND DISCUSSION



- **SIGNIFICANT PROJECT OUTCOMES:**
 - The program achieved the original objective of designing and implementing a system that would recommend entertainment and productivity options for users, and also help them save time in coming up with such solutions by doing it for them. Operating with the input of the user's facial expressions would simplify the process, and take it one step further towards being efficient.
- **PROJECT APPLICABILITY ON REAL-WORLD APPLICATIONS:**
 - The project can be used for personal use in homes as an accessibility device, when the user simply wants to use the recommendation systems. It can also be used in offices and workplaces, in order to ensure that employees can use their daybreaks efficiently.

CONCLUSION



1. LIMITATION/CONSTRAINTS OF THE SYSTEM

- Due to the unavailability of proper funding, some of the features of the program cannot be worked on and this makes it somewhat less accessible to the general public. This makes it a program mainly for testing by programmers and the more technically inclined. In order to also make it more accessible for other users, some features have to be made simpler and therefore less practical.

2. FUTURE ENHANCEMENTS

- In the near future, if our little idea gains enough steam, we would like to expand our recommendations catalog beyond just movies, music and hobbies to pass your time with. In fact, we would like to add things like TV shows, books, and other similar palatable art forms through our primary website.



THANK YOU

T R A K R E
EST. 2021 X

