REVIEW 2

Recommendations based on your mood

TRAKREX

EST. 2021

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
- Real-life applications
- Project objective
- Technical requirements
- Architecture diagram
- Module split-up and explanation
- 50% Module Implementation with Demo
- Implementation Screenshots
- References

TEAM MEMBERS

MOHAJIT PAUL:
20BCE10630

PARITOSH NIMDEO:
20BCE10490

AASHAY KULKARNI:
20BCE10402

YASH MANIKONDA:
20BCE10535

MADHAV GUPTA:
20BCE10020

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 progress we have made so far and elaborate on what we hope to achieve in the further stages of this project.

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
- Worplaces, so that employees can use their breaks efficiently so as to continue with their day productively

OBJECTIVE / PROBLEM STATEMENT

- To demonstrate the real-life applications of Blue Eyes technology through facial tracking and mood-recognition based software.
- To create suggestions for tasks and activities based off the user's currrent mood and help them have a productive day.

HARDWARE & SOFTWARE REQUIREMENTS

Hardware requirements

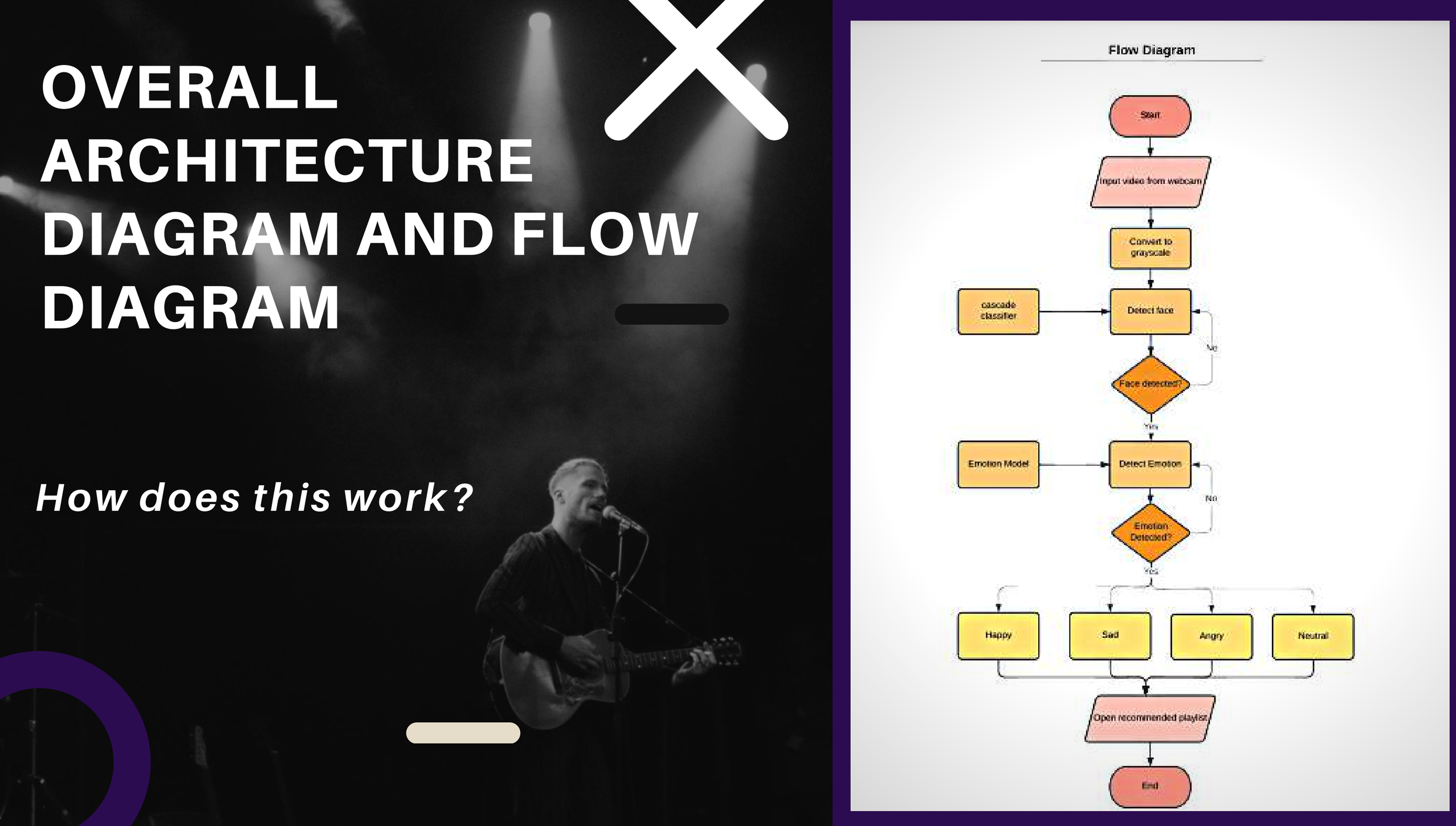
	MINIMUM	RECOMMENDED	HIGH
CPU	Intel Core i5 8th gen AMD Ryzen 5 1600x	Intel Core i5 9th gen AMD Ryzen 5 2600X	Intel Core i7 9th gen AMD Ryzen 7 4800H
RAM	8 GB	12 GB	16GB
GPU	Intel integrated graphics Ryzen integrated graphics	Nvidia GTX 1650 AMD Radeon RX 570	_____
STORAGE	5 GB HDD (SSD recommended)	7 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?



COMPLETE MODULE SPLIT-UP AND EXPLANATION

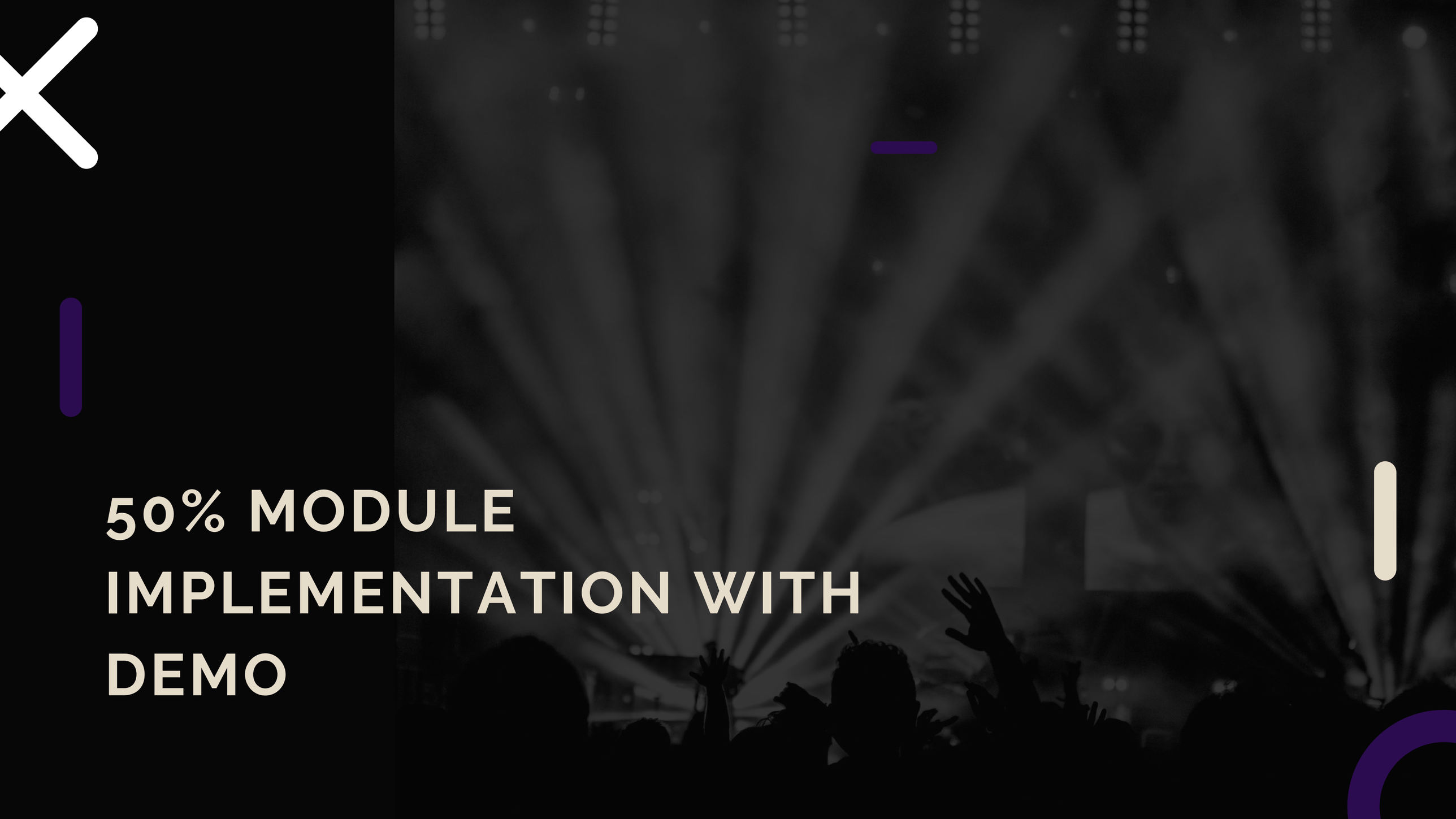
REQUIREMENTS FOR THE SOFTWARE

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

“THE FOLLOWING CODE ONLY DETECTS EMOTION, SONG RECOMMENDATION IS YET TO BE ADDED”

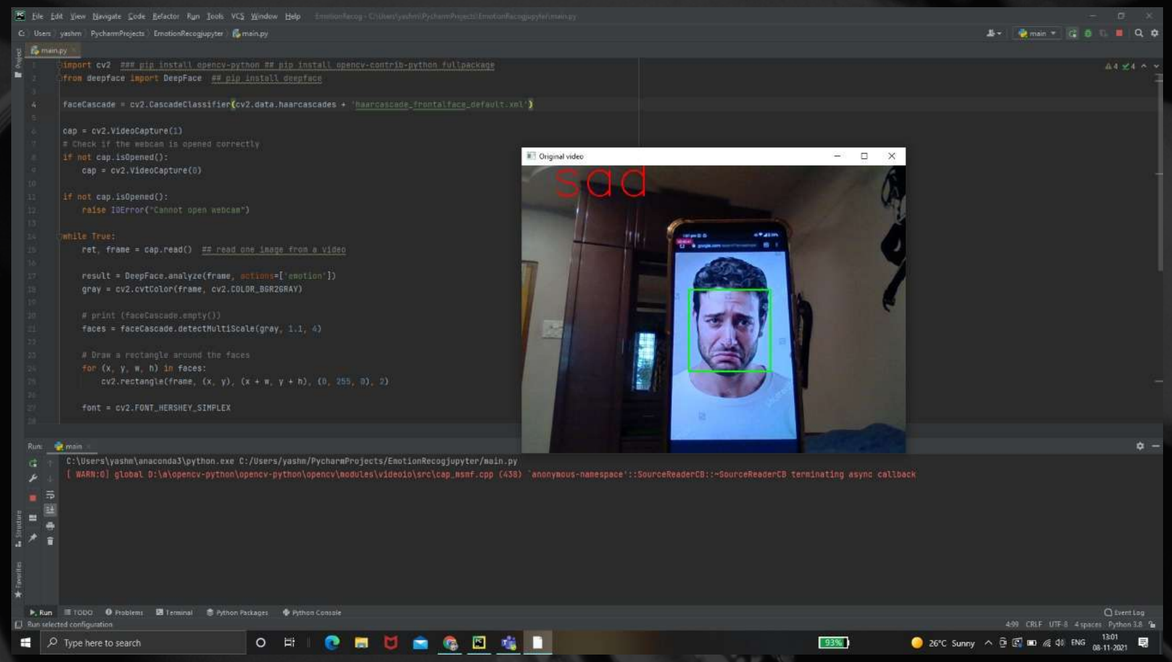
```python
import cv2  ### pip install opencv-python ## pip install opencv-contrib-python fullpackage
from deepface import DeepFace  ## pip install deepface

faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + '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
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

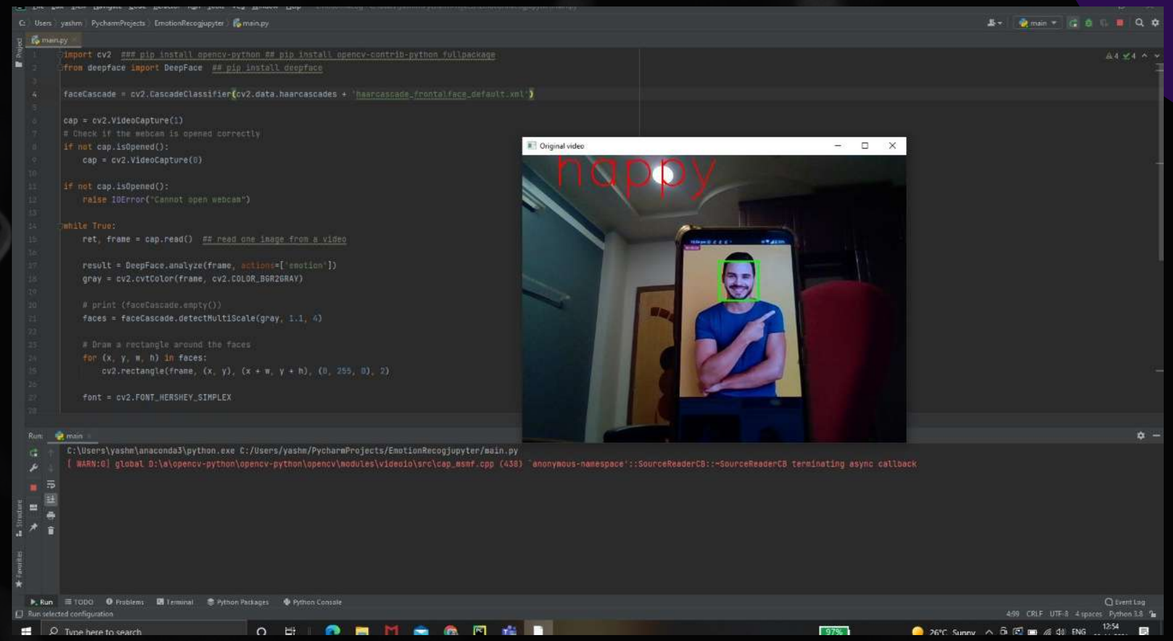
```
    # print (faceCascade.empty())
    faces = faceCascade.detectMultiScale(gray, 1.1, 4)

    # 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)

    if cv2.waitKey(2) & 0xFF == ord('q'):
        break

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



50% MODULE IMPLEMENTATION WITH DEMO

IMPLEMENTATION SCREENSHOTS





References

- https://docs.opencv.org/4.x/d7/da8/tutorial_table_of_content_imgproc.html
- https://viso.ai/computer-vision/deepface/
- https://www.sciencedirect.com/science/article/pii/S1877050920318019
- https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08
- https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades/

THANK YOU

TRAKREX

EST. 2021