**SOCIAL ROBOT**

## AN END-SEMESTER PROJECT REPORT ON THE SUBJECT OF INTRODUCTION TO AI ROBOTICS

***Submitted to***

## Amrita Vishwa Vidyapeetham

***in partial fulfillment for the award of the degree of***

## BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE ENGINEERING (AIE)

***By***

**GURUPRASATH M R**

# CH.EN.U4AIE22015

***Submitted to***

**DR. GOLAK BIHARI MAHANTA**

****

# AMRITA VISHWA VIDYAPEETHAM AMRITA SCHOOL OF COMPUTING

# CHENNAI – 601103

## April 2024

****

**BONAFIDE CERTIFICATE**

Certified that this project report **SOCIAL ROBOT”** is the bonafide work of **“Guruprasath M R”** who carried out the project work under my supervision towards his completion of the end semester project for the subject “INTRODUCTION TO AI ROBOTICS (22AIE214)”.

|  |
| --- |
| **SIGNATURE** |
| **Dr. Golak Bihari Mahanta**  **Course Instructor**  Assistant Professor (Sr. Gr)  Dept. of Mechanical Engineering Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Chennai |

Project Overview:

The robotics project is a comprehensive system designed to interact with humans through various modalities including face recognition, speech-to-text conversion, emotion detection, and natural language processing. The project is divided into three main parts, each handled by a separate team, given below

Team-A(My self): Face Recognition

**Objective**: Implement a face recognition system capable of identifying known individual

**Methodology**: Utilize computer vision techniques and machine learning algorithms to train a model for face recognition.

**Output:** Upon detecting a face, the system will identify the individual if they are registered in the database.

Team-B: Speech-to-Text and Emotion Detection

**Objective:** Develop a system to capture speech, convert it to text, and detect emotions from the speech input.

**Methodology:** Employ speech recognition algorithms to transcribe spoken words into text. Additionally, utilize sentiment analysis techniques to detect emotions such as happiness, sadness, excitement, etc.

Output: The system will provide both the transcribed text and the detected emotions from the speech input.

Team-C: UI

**Objective:** Design a user interface to display the emotions detected by Team-B's system.

**Methodology:** Create an interactive web interface using HTML, CSS, and JavaScript, Integrated with Flask for backend functionality. The interface will dynamically showcase detected emotions through text and emojis.

**Output:** The interface will provide users with a visual representation of emotions, along with options for manual input of emotions and speech recording.

MY PART(TEAM A – FACE RECOGNITION)

My responsibility involves creating the preliminary code for extracting the video feed from the frontend, extracting text, emotions, and queries, and feeding them into the NLP part. Shyam has developed the NLP component and received the output, which is then passed to the frontend. Next, Pradeep optimized my code to seamlessly integrate it into the frontend. all codes and my trails are uploaded in this GitHub Repo “https://github.com/gru13/Robotics-project”

**Dataset Creation**

To create a dataset for face recognition using images captured from a webcam, you can utilize the OpenCV library in Python. OpenCV provides a comprehensive set of tools for image processing and computer vision tasks. First, set up your Python environment with OpenCV installed. Then, write a script to capture images from the webcam and save them along with labels corresponding to the identities of the individuals in the images.

Steps involved

* Import Libraries: Import OpenCV and OS libraries.
* Initialize Webcam: Use OpenCV to initialize the webcam and start capturing frames.
* Capture Images: Capture successive image frames from the webcam with a 5-millisecond interval between each frame.
* Save Images: Once a face is detected, save the cropped face region along with its corresponding label (e.g., person's name or ID) into a dataset directory.
* Dataset Organization: Organize the dataset directory structure with subdirectories for each individual, containing their respective images.

**Face Detection**

Initially, I and Deepak attempted to utilize OpenCV's pre-trained Haar cascades with increments in image quantity: starting from 30 images, then 50, 100, 200, and finally 500. However, we encountered issues during testing, achieving only a 50% accuracy rate. Upon receiving a suggestion from my friend Neelraj, we decided to discontinue its use and instead transitioned to employing YOLO v8, achieving a 98% accuracy rate.

**Emotion Detection**

For emotion detection, I utilized the deepface module in Python, which includes built-in modules for pre-trained emotional detection models.

**Voice Recognition**

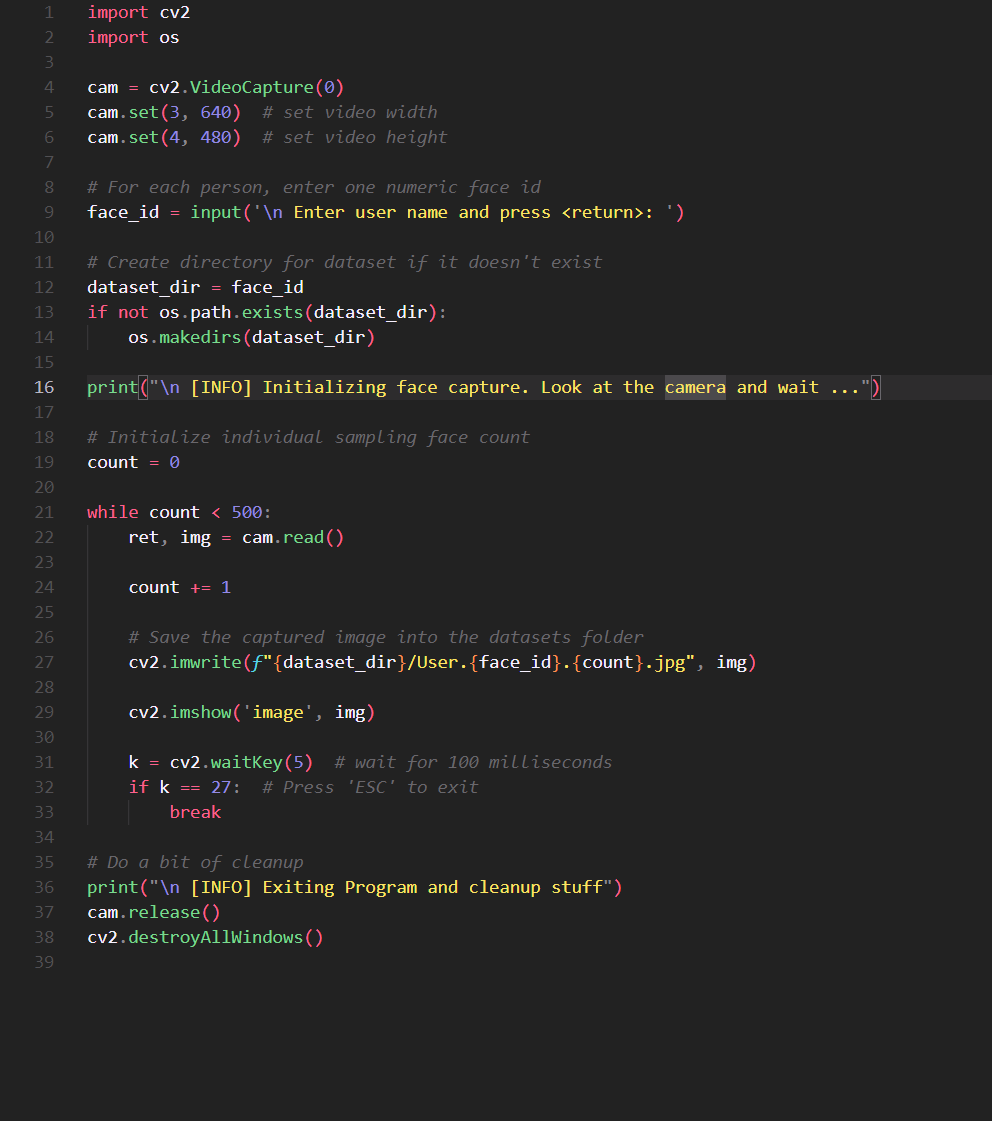
Here, we utilized OpenAI’s Whisper model for voice recognition.

**Integration Process**

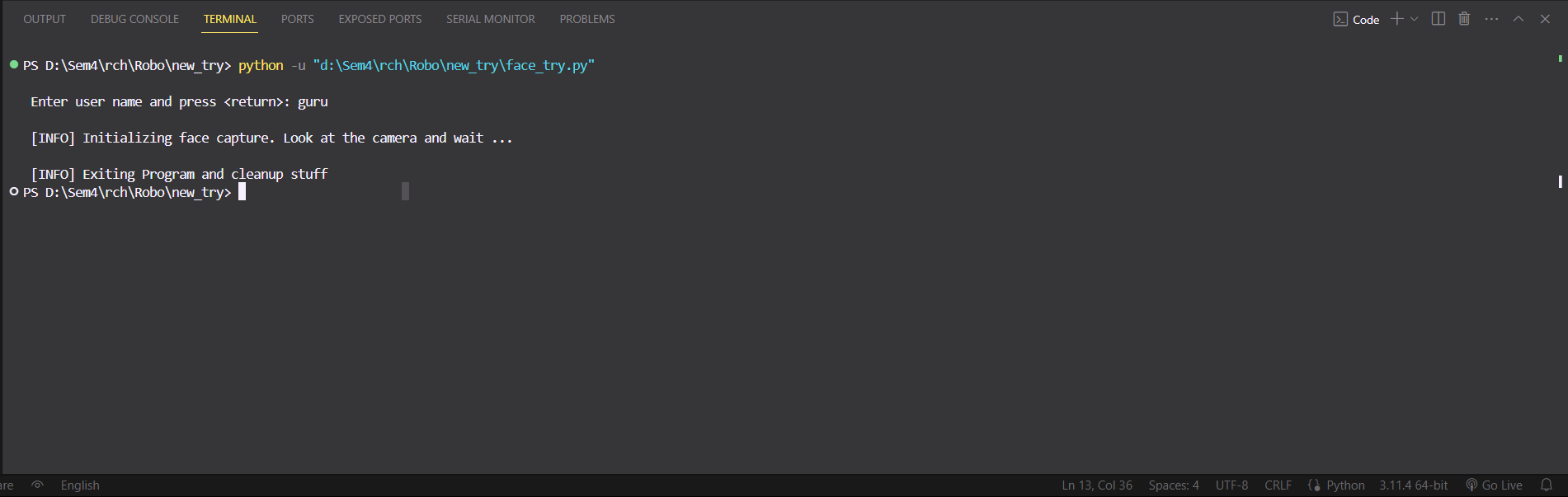
Here I combine the above 3 process in a function and NLP function from shyam and return to frontend

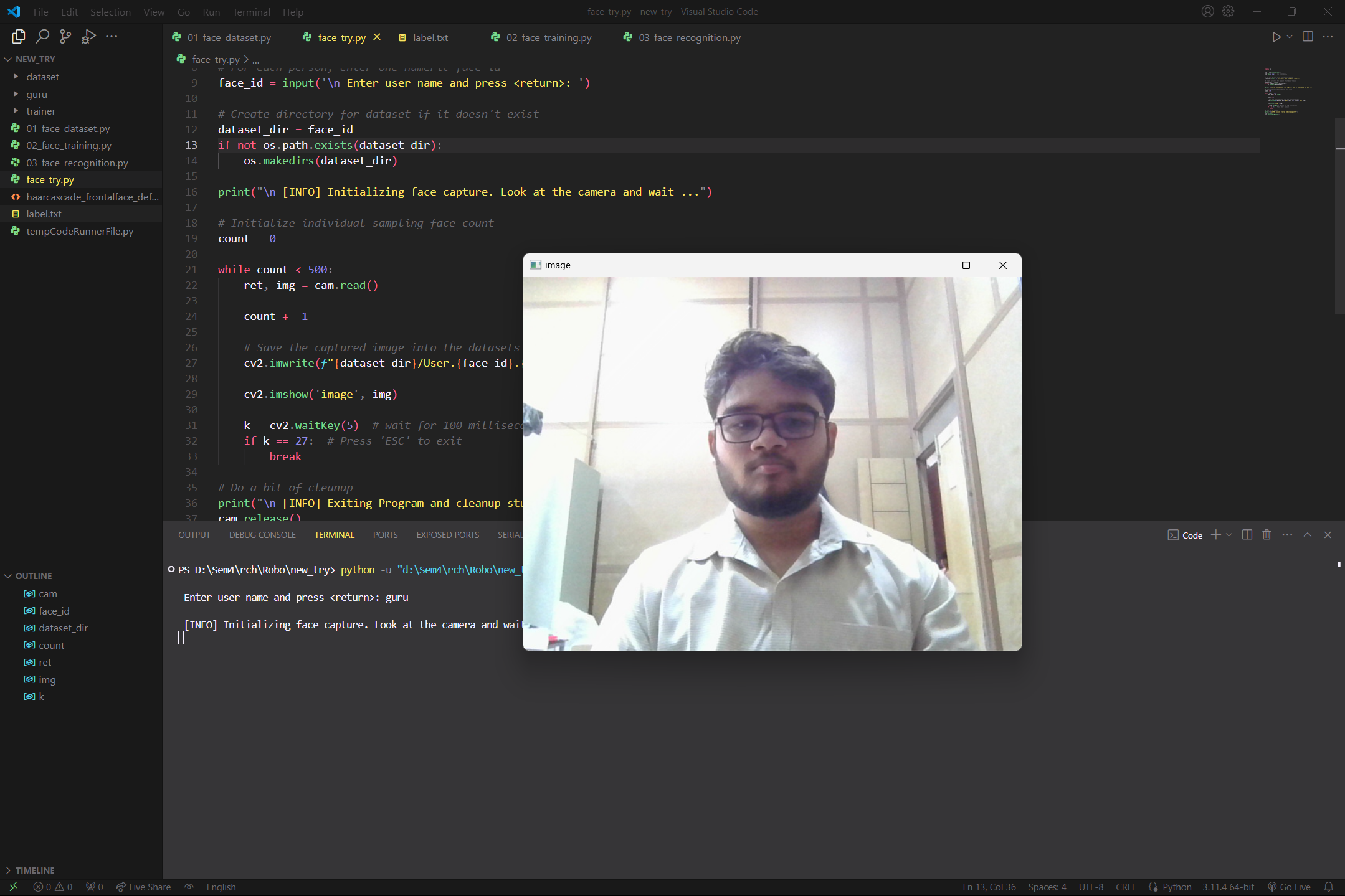
Code

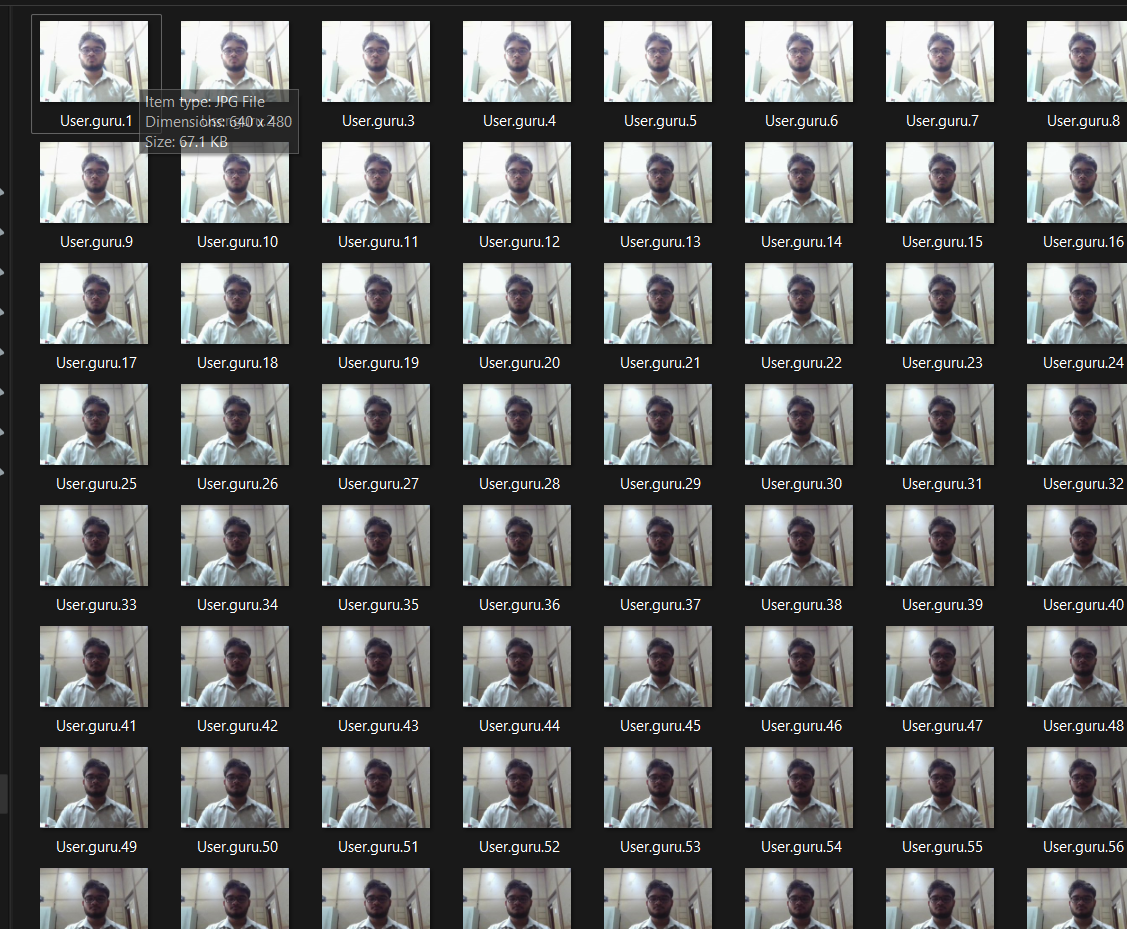
Dataset creation



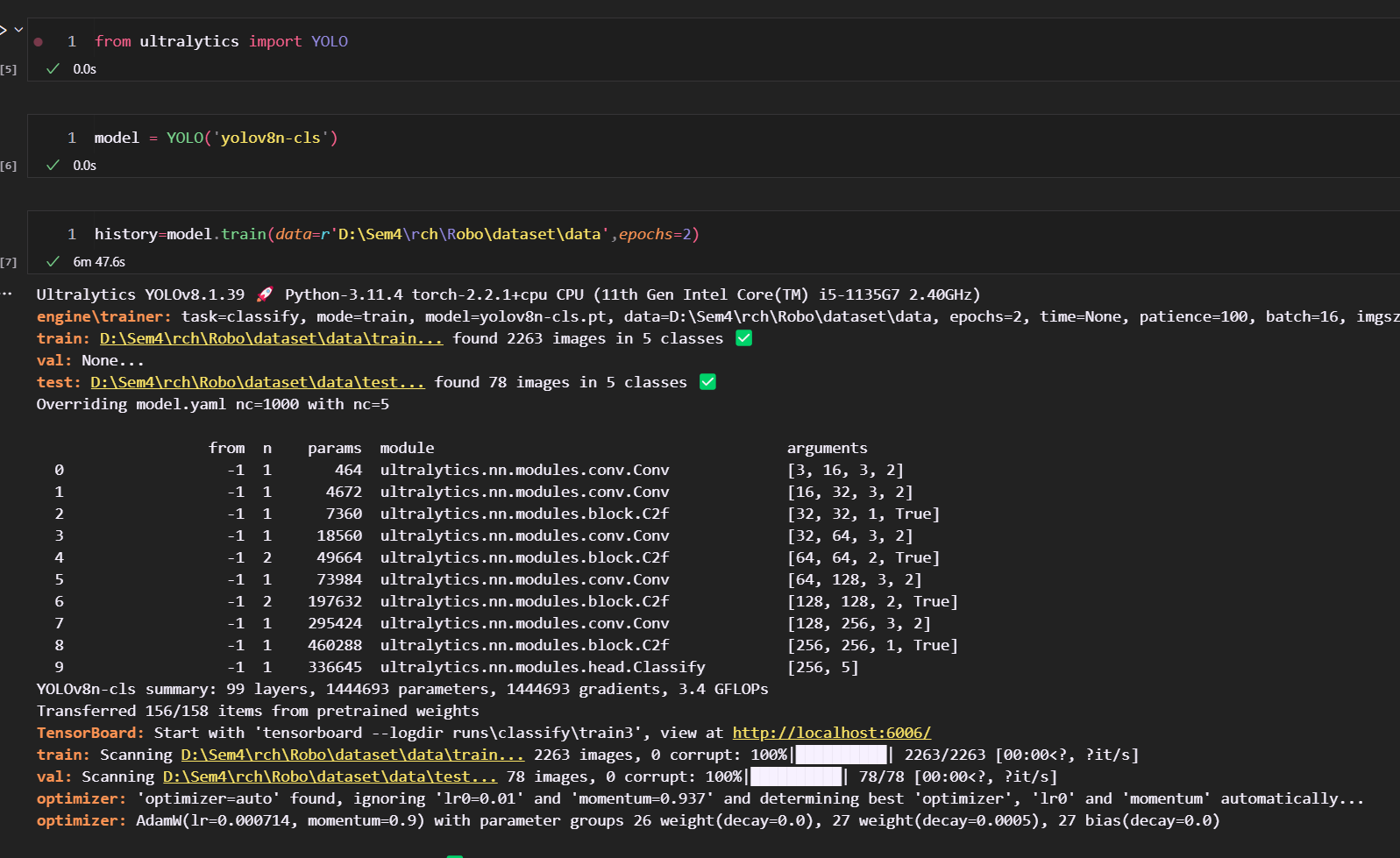
Input and Output

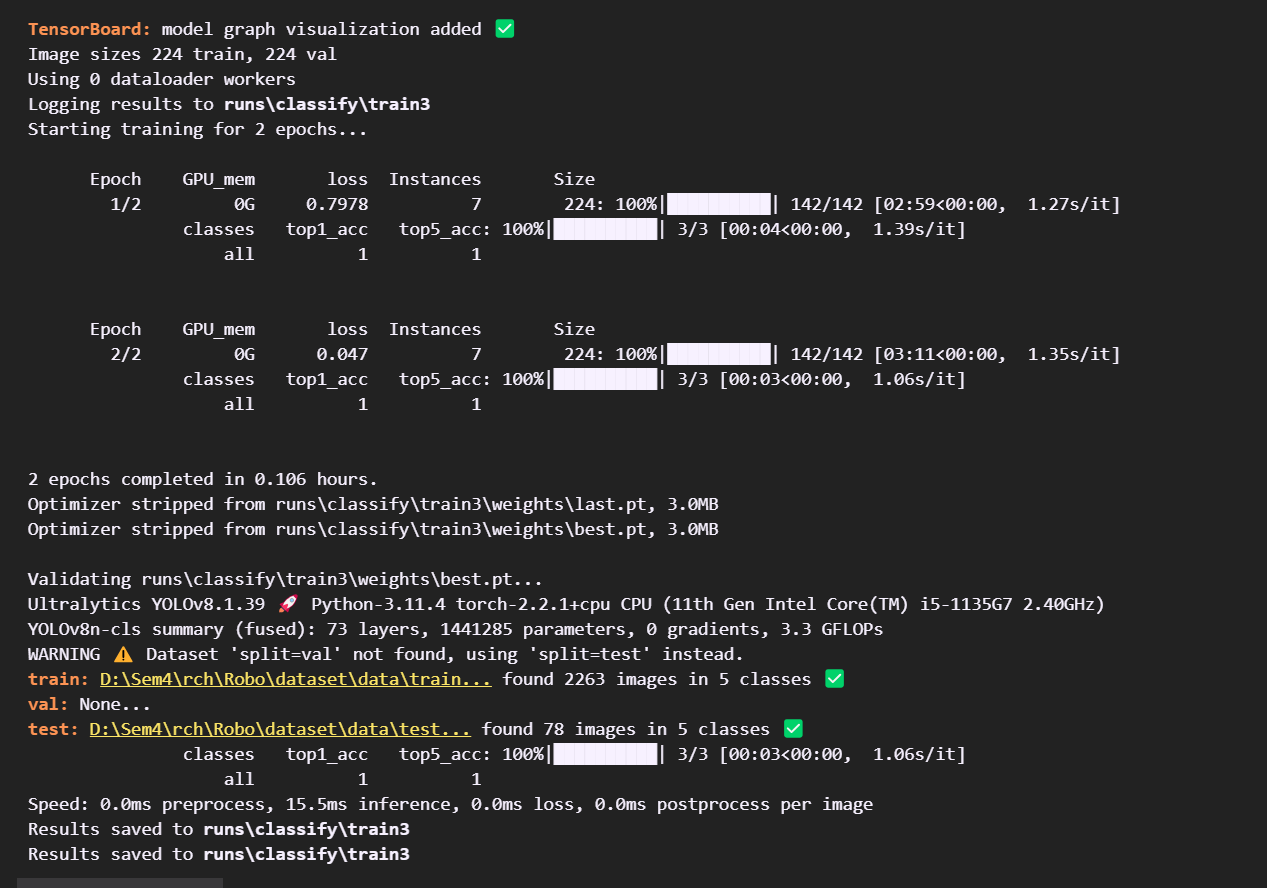


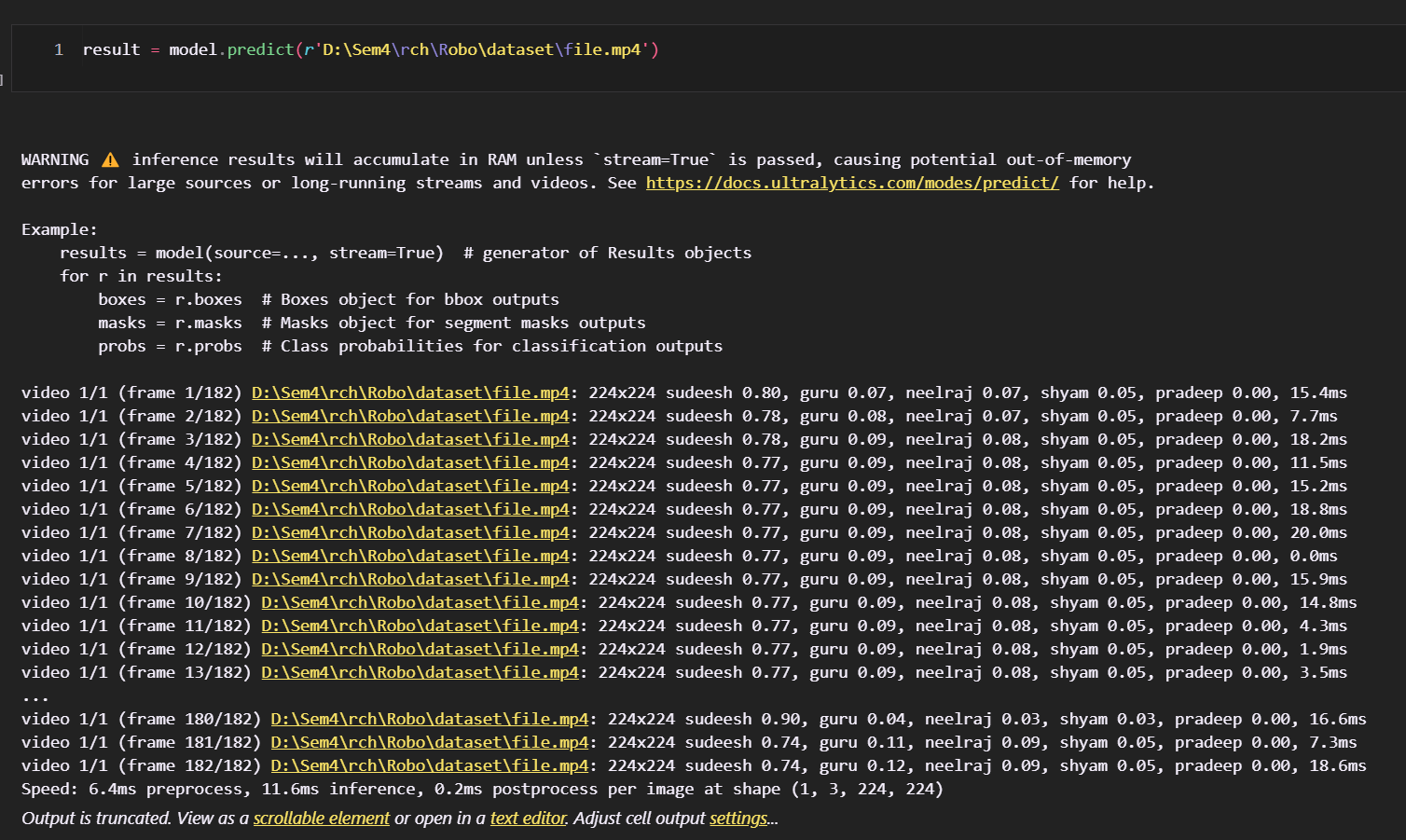


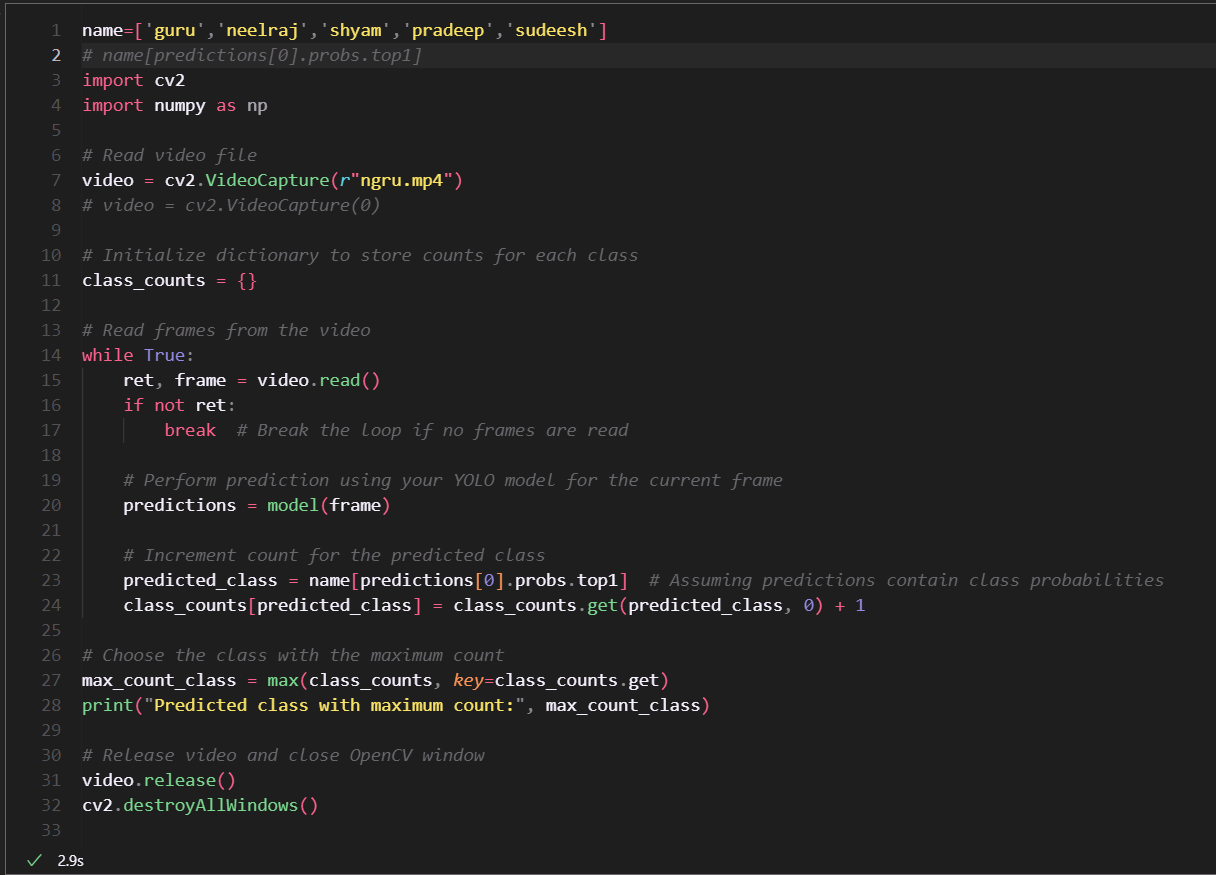


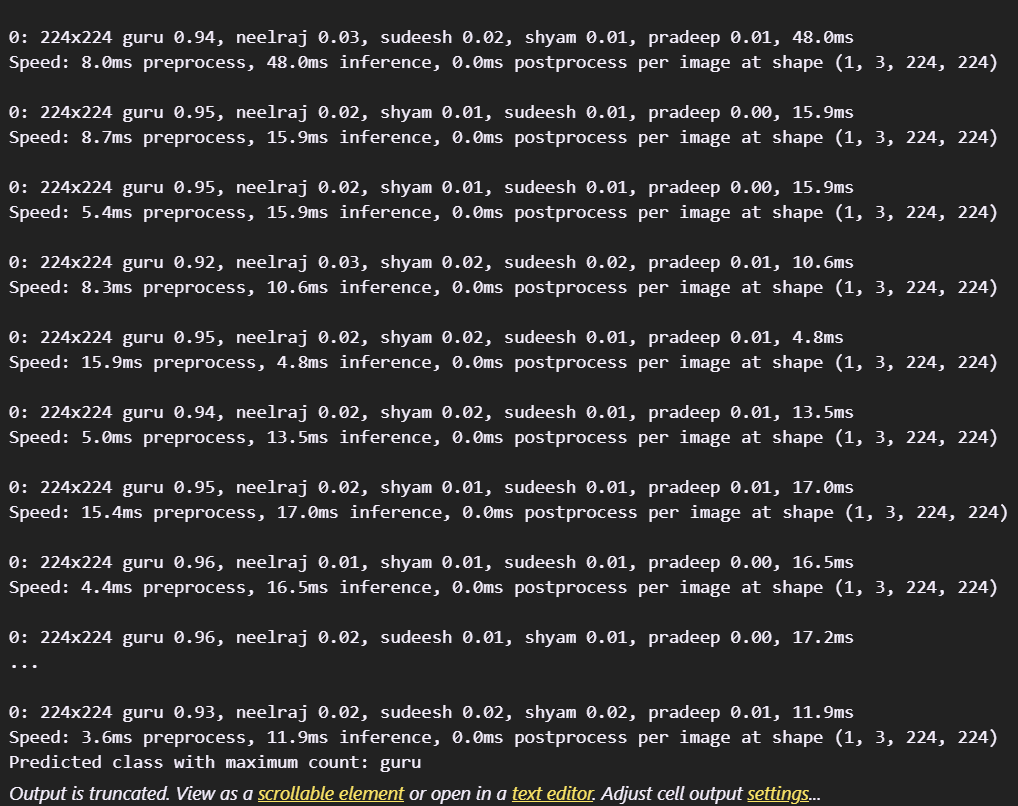
Face Detection











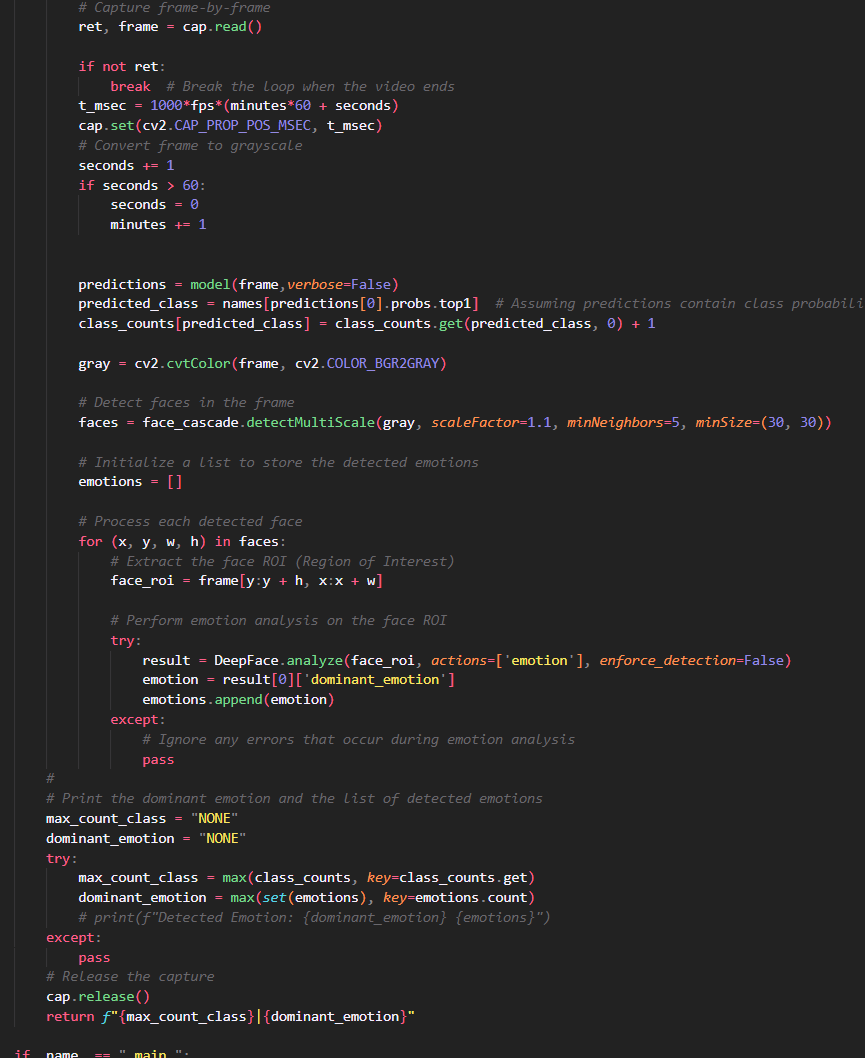
video Used to test



Label :Guru

Emotion Recongintion

Code



Input and Output

