PRE-PROJECT REPORT ON

FACEREC

Department of Computer Science

(MCA 4th Sem)

SPU Mandi

**SUBMITTED BY:**  **SUBMITTED TO:**

Name : Aditya Rana Mrs. Ruchi Thakur

Class : MCA 4th

Roll No : MCA-23-01

Uni Roll No : 202301504127001

Date :



**CONTENT**

* Place of working
* About the project
  + Introduction to the project
* Objective
* Hardware and software tools used
* Language used
* Output
* Remaining work



During a comprehensive six-month internship program undertaken at Excellence Technology in Mohali, my primary focus was dedicated to a significant project centered on face recognition technology. Working under the expert guidance and mentorship of **Mr. Naresh Kumar**, I was afforded the invaluable opportunity to gain hands-on experience. This involved engaging directly with various aspects of the face recognition project, allowing me to apply theoretical knowledge in a practical setting and develop key skills in this specialized domain.

**Corporate Office**

C-133, First Floor, Industrial Area, Phase-8, Sector-72, Mohali. PH.: 0172-4698755, 93177-88822, 93562-55522 E-MAIL: excellencehead@gmail.com URL: www.excellencetechnology.in

**INTRODUCTION TO PROJECT**

This project developed a basic system to identify specific celebrities from a live video feed using a webcam. It involves two main steps: teaching the computer what different celebrity faces look like (training) and then using that knowledge to spot and name them in real-time video (recognition).

**OBJECTIVES OF PROPOSED SYSTEM**

The main goals of this project were:

* To find human faces in a video stream.
* To match the detected face to a known celebrity from a list.
* To show the celebrity's name and how sure the system is about the match in real-time.
* To demonstrate simple computer vision techniques.

**METHODOLOGY**

The project uses standard computer vision methods and a specific face recognition technique.

How it Works:

1. Getting Face Examples (Data):

The system needs pictures of the celebrities it should recognize. These pictures are organized in folders, one for each celebrity. Having many pictures of each person from different angles and in different lights helps the system learn better.

1. Finding Faces:

First, the system looks for faces in the video using a tool called a Haar Cascade classifier. This tool is good at quickly finding face-like shapes. It draws a box around where it thinks a face is. This isolates the face area for the next step.

1. Teaching the System (Training):
   1. A script (createmodel.py) is used for this.
   2. It takes the celebrity pictures, finds the faces, and turns them grayscale (black and white).
   3. The face pictures are all resized to the same small size (100x100 pixels).
   4. These processed face pictures and the celebrity names are used to train a face recognition method called LBPH (Local Binary Patterns Histograms). LBPH looks at the texture patterns on the face.
   5. After learning, the trained model is saved in a file (face\_recogonizer.yml) so it can be used later.
2. Recognizing Faces (Real-time):
   * Another script (read\_face.py) does the real-time recognition.
   * It loads the saved model.
   * It uses your computer's webcam to get live video frames.
   * In each frame, it finds faces using the same face finder.
   * It gives the detected face to the trained model.
   * The model predicts which celebrity it is and gives a "confidence" score. A lower score means a better match.
   * The system then shows the celebrity's name and the confidence score next to the face in the video window. This continues until you press the 'e' key to stop.

**HARDWARE AND SOFTWARE TOOLS USED**

**Software:**

* Application Name – Visual Studio Code
* Operating System – Windows 11

**Hardware:**

* Processor – intel i5-12500H
* Disk – 512 GB SSD
* Memory – 16 GB RAM

**LANGUAGE USED**

The code for this project is written entirely in **Python**.

* **Python:** The programming language used.
* **OpenCV** **(cv2):** A library for handling images and video, finding faces, and doing face recognition.
* **NumPy (numpy):** A library for working with numbers, used for handling image data.
* **OS Module (os):** A standard Python tool to work with files and folders.
* **Haar Cascade File** (haarcascade\_frontalface\_default.xml): A file that helps find faces.
* **Trained Model File** (face\_recogonizer.yml): The file created after training, containing what the system learned.

**SCREENSHOTS**

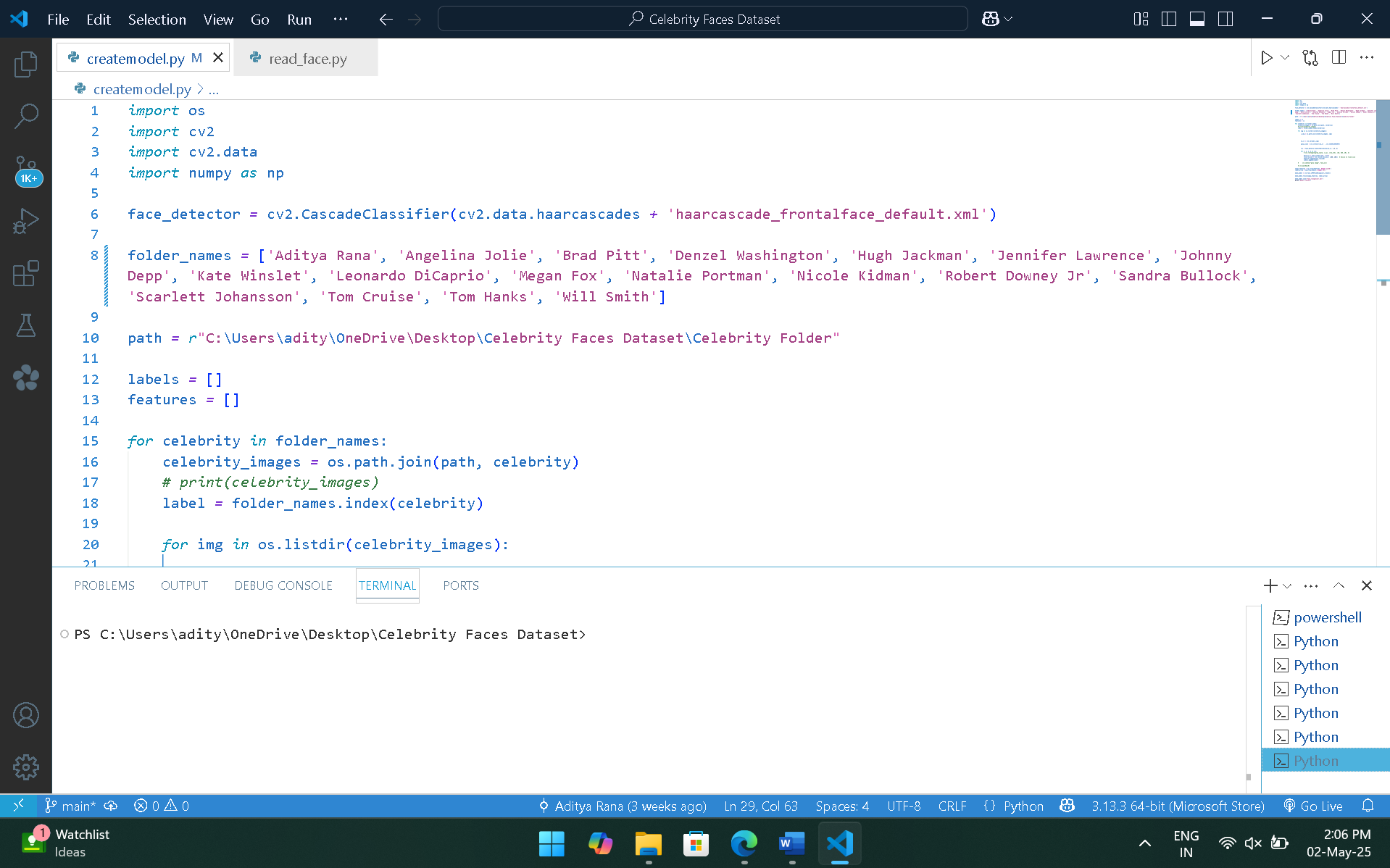


Figure : Model Training

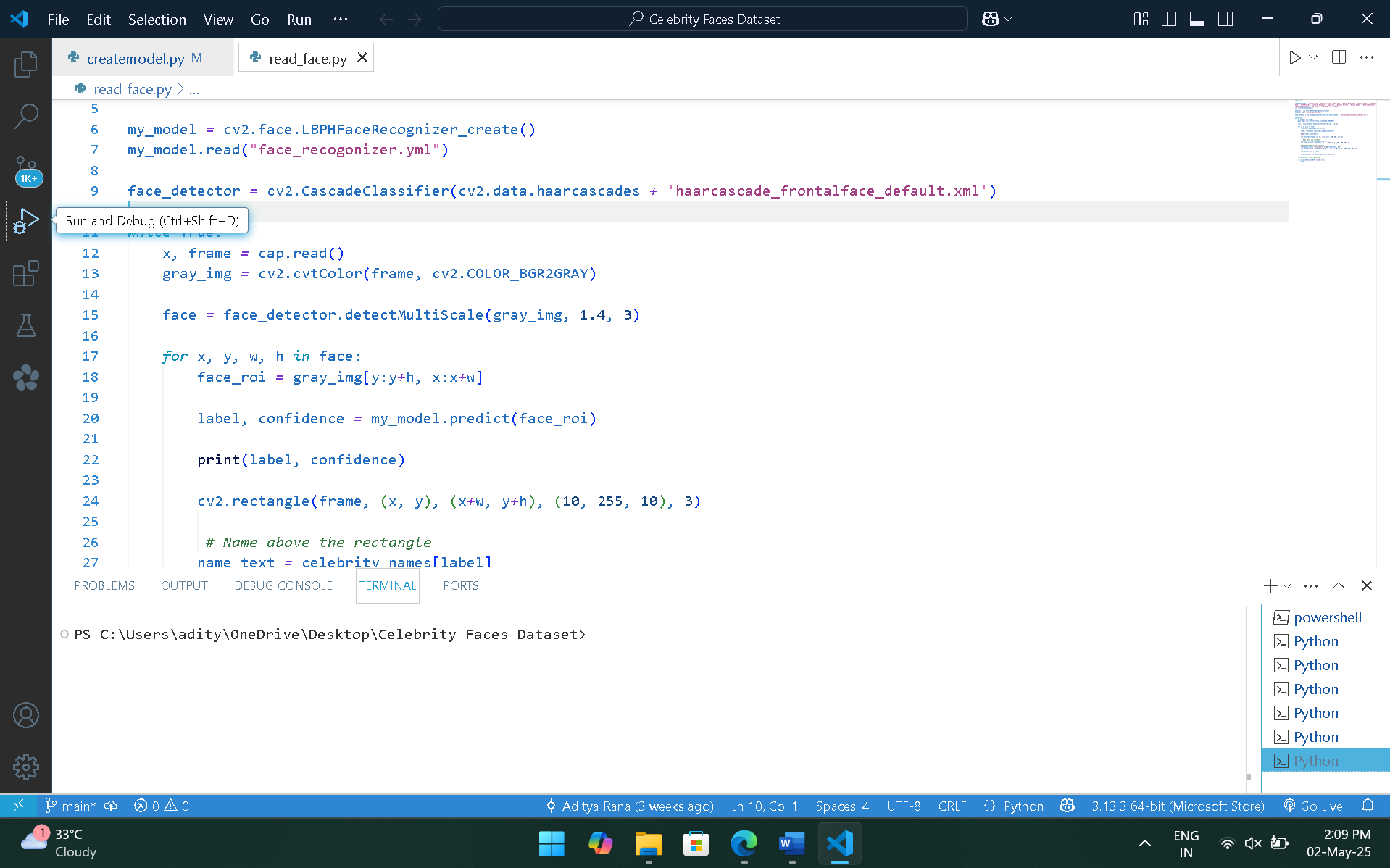


Figure : Output Screen Code

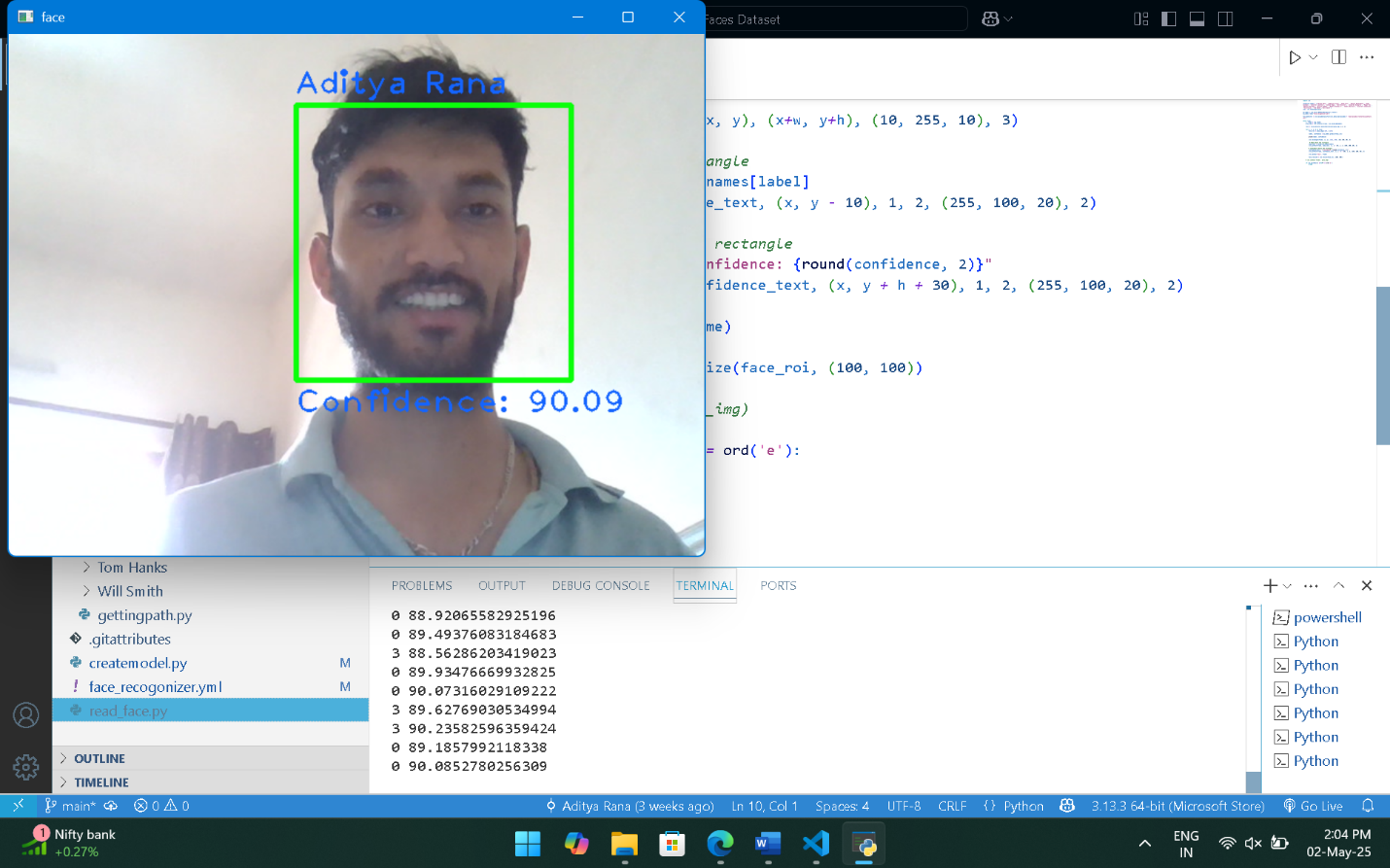


Figure : Output

**REMAINING WORK:**

Adding Gender and Age Prediction: A significant next step is to expand the system's capabilities beyond just identification. This could involve integrating additional models or techniques to estimate the gender and approximate age of the detected face. This would provide more descriptive information about the person being observed, adding another layer of analysis to the system's output.