Name: Mai Ngoc Diem My

Course: PB-2048

# **Table of contents:**

Introduction…………………………………………………………………………………1

Details………………………………………………………………...………………1

* Functions………………………………………………………………………………1
* Implementation methods and techniques……………………………………………...5

Conclusion………………………………………………………………………9

# 1 - Introduction

* The objective of this program is to capture an image of an individual’s face, analyze the detected facial features, and compare them against a database of known faces. This process aims to identify whether the captured face matches any of the faces in the database, thereby determining the individual’s identity.
* The program can use a camera or an uploaded photo to identify face traits.

# 2 - Details

**Functions:**

* Identify the features of a person's face in photos or using a webcam, evaluate the results, then cross-reference the results with a database of recognised faces.

# Import photo files and names of people in the known list

known\_images = {

'emma\_watson1.png': "Emma Watson",

'birain1.png': "Bi Rain",

'timothee1.png': "Timotheé Chalamet",

'Asha.jpg': "Asha",

'Anna.jpg': "Anna"

}

**The database of recognised faces.**

**choice = input("Enter '1' to recognize faces from an image or '2' to use the webcam: ")**

**if choice == '1':**

**image\_path = input("Enter the name of the image file to be identified: ")**

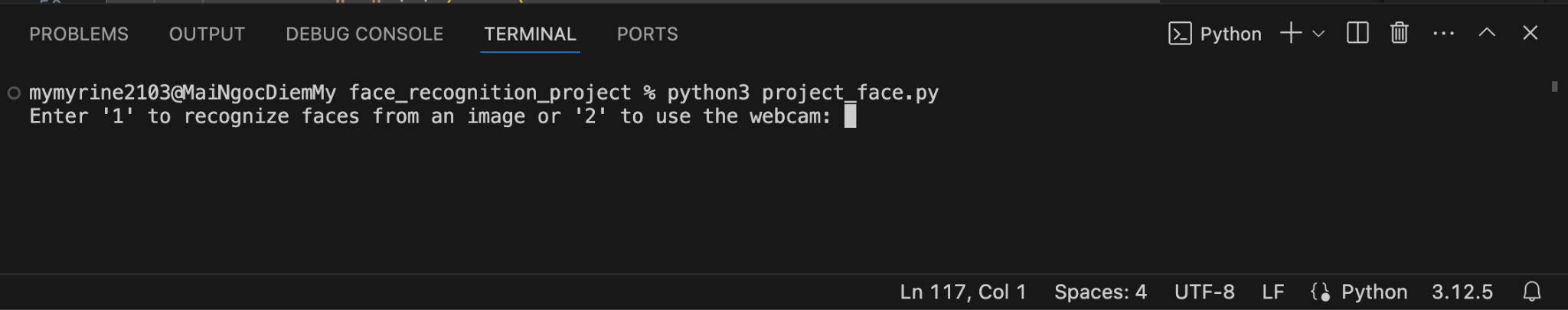
**recognize\_faces\_in\_image(image\_path)**

**elif choice == '2':**

**recognize\_faces\_in\_video()**

**else:**

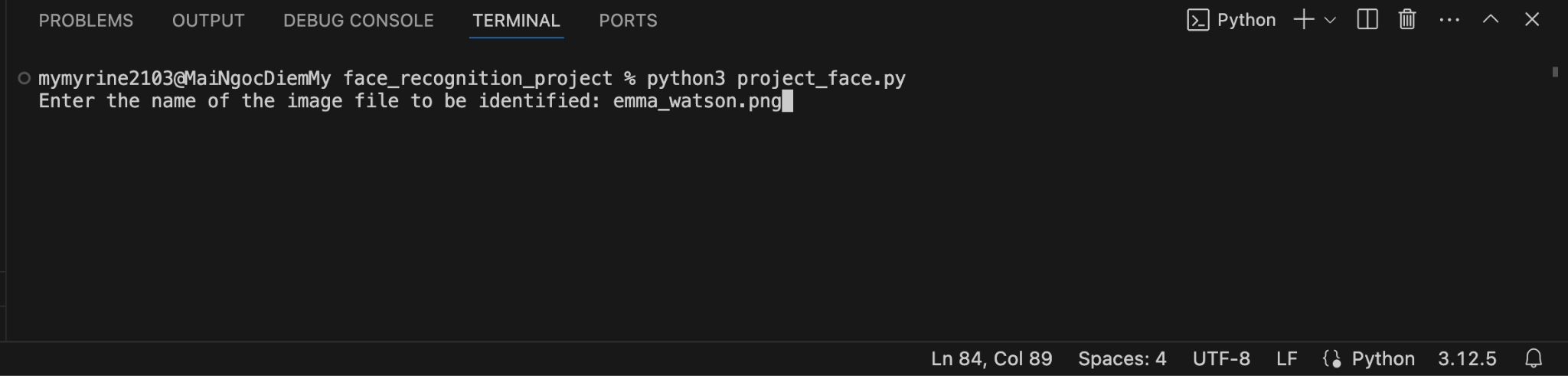
**print("Invalid choice. Please enter '1' or '2'.")**

****

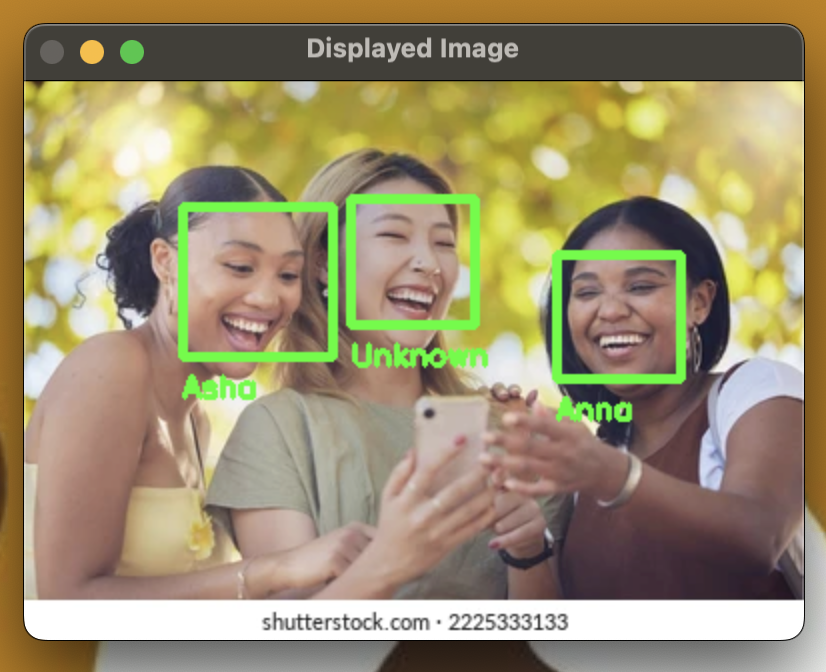
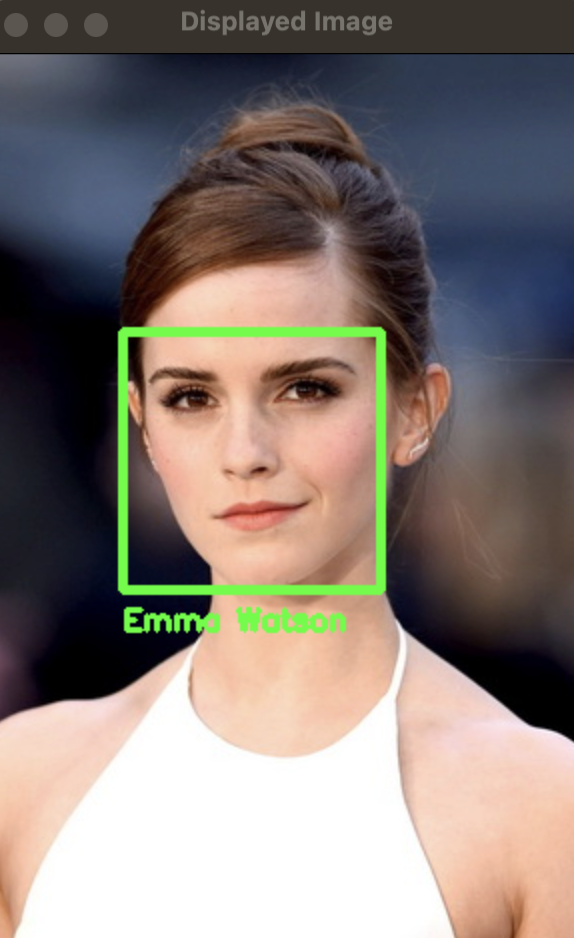
**Users can select recognize faces from an image or use the webcam.**

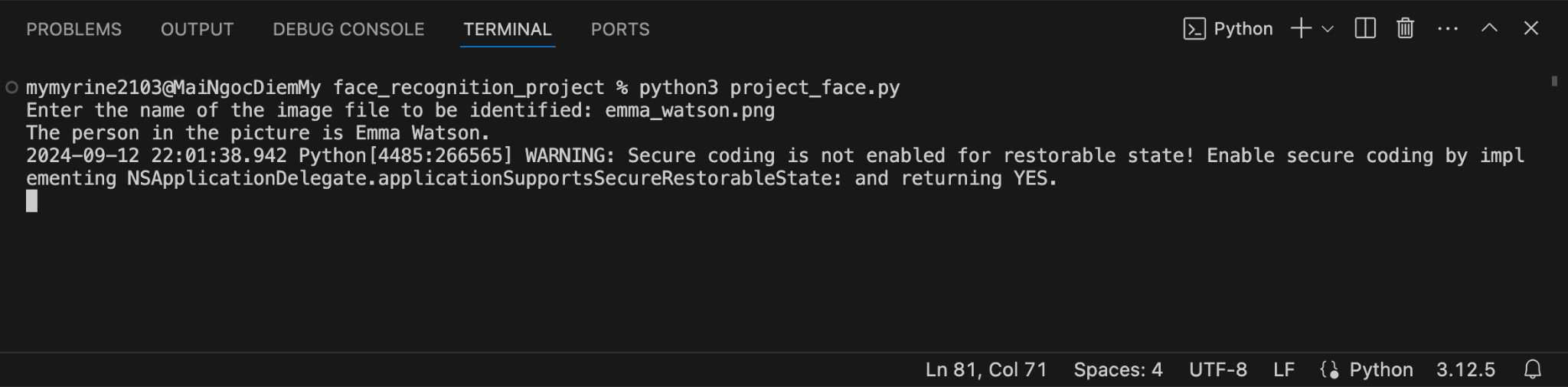
**# Enter the name of the image file to be identified**

**image\_define\_path = input("Enter the name of the image file to be identified: ")**

****

**Get an image of an individual’s face.**

****

****

****

**Identify whether the captured face matches any of the faces in the database**

**Implementation methods and techniques:**

**Load and encode faces of known people:**

*Initialize empty list:*

# Load known faces and their encodings

known\_face\_encodings = []

known\_face\_names = []

* known\_face\_encodings: This list will store the face encodings of known people.
* known\_face\_names: This list will store the names of known people.

*Loop through each item in known\_images:*

for image\_path, name in known\_images.items():

* image\_path: Path to image file.
* name: Name of person in photo.

*Upload photo and encode face:*

image = face\_recognition.load\_image\_file(image\_path)

face\_encoding = face\_recognition.face\_encodings(image)[0]

* image = face\_recognition.load\_image\_file(image\_path): Load image from image\_path.
* face\_encoding = face\_recognition.face\_encodings(image)[0]: Encode the faces in the image. The function face\_recognition.face\_encodings(image) returns a list of face encodings in the image, and [0] takes the encoding of the first face.

*Add face and name encoding to the list:*

known\_face\_encodings.append(face\_encoding)

known\_face\_names.append(name)

* known\_face\_encodings.append(face\_encoding): Adds face encoding to

known\_face\_encodings list.

* known\_face\_names.append(name): Adds name to known\_face\_names list.

**Face recognition function in image:**

*Function definition:*

def recognize\_faces\_in\_image(image\_path):

* This function takes an image\_path parameter, which is the path to the image file to be recognized.

*Upload photo to be identified:*

# Load the image to be recognized

image\_to\_recognize = face\_recognition.load\_image\_file(image\_path)

* Use the face\_recognition.load\_image\_file function to load the image from the image\_path path.

*Convert image to BGR color (OpenCV uses):*

# Convert the image to BGR color (which OpenCV uses)

image\_to\_recognize = cv2.cvtColor(image\_to\_recognize, cv2.COLOR\_RGB2BGR)

*Loop through each face found in the image:*

for (top, right, bottom, left), face\_encoding in zip(face\_locations, face\_encodings):

* zip(face\_locations, face\_encodings): Combines the locations and encodings of each face.
* top, right, bottom, left: Coordinates of the face in the photo.
* face\_encoding: Encoding of the face.

*Compare the found face with known faces:*

matches = face\_recognition.compare\_faces(known\_face\_encodings, face\_encoding)

name = "Unknown"

* matches: A list of boolean values ​​indicating whether the found face matches any of the known faces.
* name = "Unknown": Defaults to “Unknown” if no match is found.

*Check and assign a name to the face if a match is found:*

if True in matches:

matched\_indexes = [i for i, match in enumerate(matches) if match]

names = [known\_face\_names[i] for i in matched\_indexes]

name = ", ".join(names)

print(f"The person in the picture is {name}.")

else:

print("The person in the photo is not on the known list.")

* If there is a match (True in matches), find the indices of the matching faces and get the corresponding names.
* Assign a name to the face and print the result.

*Draw a rectangle around the face and the name below the face:*

# Draw a rectangle around the face

cv2.rectangle(image\_to\_recognize, (left, top), (right, bottom), (0, 255, 0), 3)

# Draw the name below the face

cv2.putText(image\_to\_recognize, name, (left, bottom + 20), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)

*Show image:*

# Display the image

cv2.imshow('Recognized Faces', image\_to\_recognize)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Face recognition function by webcam:**

*Open video stream from webcam:*

video\_capture = cv2.VideoCapture(0)

*While loop:*

while True:

* This loop will continue to run until you press the ‘q’ key to exit.

*Get frames from video:*

ret, frame = video\_capture.read()

* Takes a frame from the video stream.

*Convert frame from BGR to RGB:*

rgb\_frame = frame[:, :, ::-1]

* OpenCV uses BGR color format, while face\_recognition library uses RGB format. This line converts the frame from BGR to RGB.

*Detect and encode faces in the frame:*

face\_locations = face\_recognition.face\_locations(rgb\_frame)

face\_encodings = [face\_recognition.face\_encodings(rgb\_frame, [face\_location])[0] for face\_location in face\_locations]

* Detects the positions of faces in the frame and encodes them.

*Compare detected face with known faces:*

for (top, right, bottom, left), face\_encoding in zip(face\_locations, face\_encodings):

matches = face\_recognition.compare\_faces(known\_face\_encodings, face\_encoding)

name = "Unknown"

if True in matches:

matched\_indexes = [i for i, match in enumerate(matches) if match]

names = [known\_face\_names[i] for i in matched\_indexes]

name = ", ".join(names)

print(f"The person in the picture is {name}.")

else:

print("The person in the photo is not on the known list.")

* This line compares detected faces with known faces and assigns names to them if there is a match.

*Draw a rectangle around the detected faces and display the names:*

cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)

cv2.putText(frame, name, (left, bottom + 20), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (0, 255, 0), 2)

*Show video with detected face:*

cv2.imshow('Video', frame)

*Press ‘q’ to exit:*

if cv2.waitKey(1) & 0xFF == ord('q'):

break

*Release the video stream and close all windows:*

video\_capture.release()

cv2.destroyAllWindows()

# 3 - Conclusion

**Self-assessment of results:**

| **Advantages** | **Disadvantages** |
| --- | --- |
| * The program can detect and recognize individual's face in a photos * The program can connect to the device's webcam for facial recognition. | * The program does not handle a variety of cases and is not convenient for users. * The program has not been processed with a graphical interface. |

**Product development direction:**

1. **Graphics interface design:**

* Develop an intuitive and user-friendly graphical interface for the program.

1. **Database design:**

* Create a robust database to securely store customer information, including pictures, names, birth dates, and other relevant data.

1. **Enhance information Handling:**

* Ensure the program can efficiently manage and process a wide range of customer information.