



**PERIYAR  
MANIAMMAI**  
INSTITUTE OF SCIENCE & TECHNOLOGY  
(Deemed to be University)  
Established Under Sec. 3 of UGC Act, 1956 • NAAC Accredited  
think • innovate • transform

**NAME : T.THIBARISHI  
C.KANAGAMALAR**

**REGISTER NUMBER : 123011019032  
123011019015**

**CLASS : B.TECH(CSE) AI&ML**

**SUBJECT CODE : XCSHA1**

**ASSIGNMENT : MINI PROJECT  
FACE RECOGNITION**

## **INTRODUCTION :**

Face recognition is a technology that can identify or verify a person's identity using their facial features. In recent years, this technology has gained widespread popularity across various industries, including security, healthcare, and social media. With Python's rich ecosystem of libraries and frameworks, developing a face recognition system has become relatively straightforward.

A face recognition program in Python typically involves detecting a face in an image or video stream, extracting facial features, and then comparing those features with a database to find a match or verify a person's identity. Python offers powerful tools like OpenCV, dlib, and `face_recognition` libraries, which make it easy to implement face recognition systems.

### **Input:**

A single image of the person you want to recognize (usually a passport-style or clear portrait photo).

A group photo that may contain multiple faces (with the target person's face among them).

### **Face Detection:**

The program detects all faces in the group photo, typically using techniques such as Haar cascades, Dlib, or modern deep learning-based methods (MTCNN, for example).

The goal is to find the coordinates (bounding boxes) for each face in the group photo.

## **PSEUDO CODE :**

```
import face_recognition
```

```
import cv2
```

```
import numpy as np
```

```
group_image_path = '/Users/pathin/Desktop/python/group_image.jpeg'
```

```
group_image = face_recognition.load_image_file(group_image_path)
```

```
group_image_rgb = cv2.cvtColor(group_image, cv2.COLOR_BGR2RGB)
```

```
group_face_encodings = face_recognition.face_encodings(group_image_rgb)
```

```
group_face_locations = face_recognition.face_locations(group_image_rgb)
```

```
if len(group_face_encodings) == 0:
```

```
    print("No faces found in the group image.")
```

```
    exit()
```

```
video_capture = cv2.VideoCapture(0)
```

```
frame_width = 640
```

```
frame_height = 480
```

```
video_capture.set(3, frame_width) # Width
```

```
video_capture.set(4, frame_height) # Height
```

```
frame_skip = 2 # Process every 3rd frame
```

```
frame_count = 0
```

```
print("Starting webcam. Press 'q' to quit.")
```

```
while True:
```

```
    ret, frame = video_capture.read()
```

```
    if not ret:
```

```
        print("Failed to capture frame from webcam. Exiting...")
```

```
        break
```

```
    frame_count += 1
```

```
    if frame_count % frame_skip != 0:
```

```
        continue
```

```
    small_frame = cv2.resize(frame, (0, 0), fx=0.5, fy=0.5)
```

```
    small_frame_rgb = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)
```

```
    frame_face_locations = face_recognition.face_locations(small_frame_rgb)
```

```
    frame_face_encodings = face_recognition.face_encodings(small_frame_rgb,  
    frame_face_locations)
```

```
for frame_face_encoding, frame_face_location in zip(frame_face_encodings,  
frame_face_locations):
```

```
    matches = face_recognition.compare_faces(group_face_encodings,  
frame_face_encoding)
```

```
    match_found = any(matches)
```

```
    top, right, bottom, left = frame_face_location
```

```
        top *= 2
```

```
        right *= 2
```

```
        bottom *= 2
```

```
        left *= 2
```

```
    if match_found:
```

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

```
        cv2.putText(frame, "Match Found", (left, top - 10),  
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)
```

```
    else:
```

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

```
        cv2.putText(frame, "No Match", (left, top - 10), cv2.FONT_HERSHEY_SIMPLEX,  
0.9, (0, 0, 255), 2)
```

```
cv2.imshow('Live Face Recognition', frame)
```

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

```
    break
```

```
video_capture.release()
```

```
cv2.destroyAllWindows()
```

## OUTPUT:

```
pathin — -zsh — 81x24
Last login: Tue Nov 19 17:44:14 on ttys002
(base) pathin@pathins-MacBook-Air ~ % python /Users/pathin/Desktop/python/my_face_recognition_script.py
```

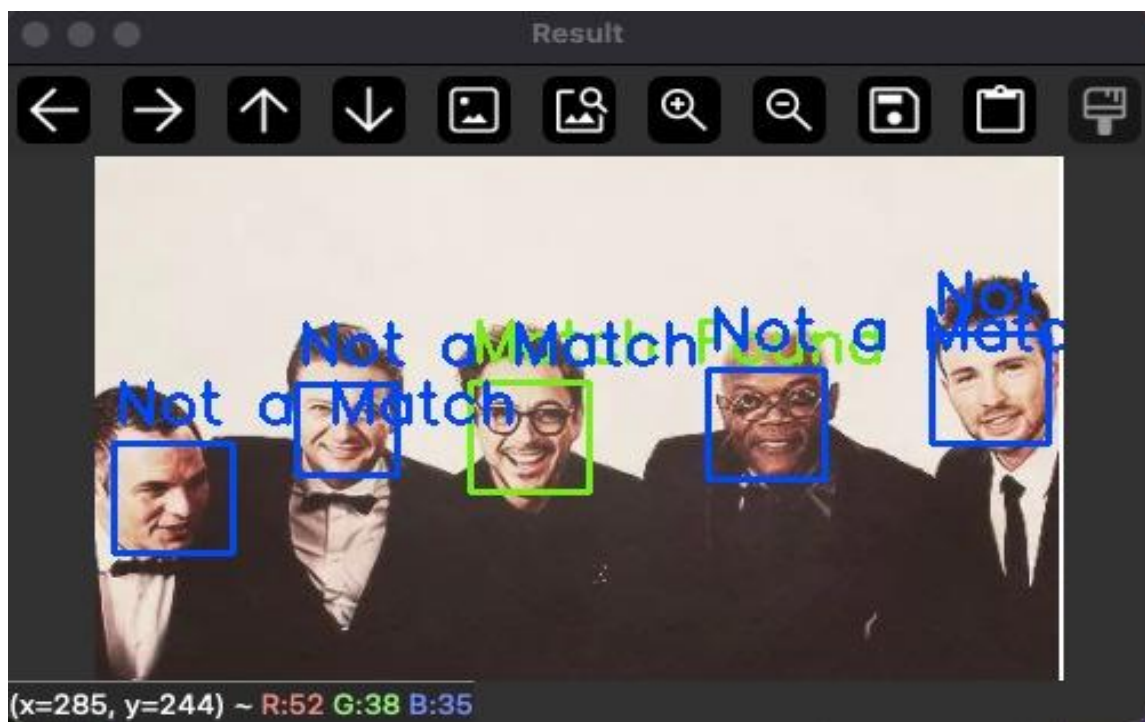
## INPUT GROUP IMAGE:



## IMAGE TO BE SEARCHED(GIVE AS INPUT):



## OUTPUT(AFTER REGONITION):



If the image is present in the group Image then it will be marked in a square box with a message match found if

image is not recognised then the images were marked in an blue rectangular box with a message not a match

### **Conclusion:**

This face recognition program, developed in Python, provides an efficient solution to identifying a specific person from a group photo. By leveraging powerful libraries like `face_recognition` and OpenCV, the program can detect, match, and highlight faces, making it suitable for various real-world applications in security, social media, and beyond.

**THANK**

**YOU.....**