**REPORT COMPUTER VISION ASSIGNMENT**

*Môn: CPV301*

*Người thực hiện: Huỳnh Quốc Việt (SE194225)*

## **1. Giới thiệu bài Assignment**

* Kịch bản:
* Một lớp học tại Đại học FPT đang triển khai một hệ thống điểm danh đơn giản.
* Mỗi sinh viên sẽ lần lượt đứng trước camera IP để thực hiện điểm danh.
* Hệ thống yêu cầu ghi lại tên người đang đứng trước camera và thời gian sinh viên vào lớp.
* Mục tiêu của project:
* Xây dựng hệ thống điểm danh bằng nhận diện khuôn mặt realtime với Python và OpenCV
* Output: file csv MSSV\_FullName\_Class + thời gian vào lớp.

## **2. Tổng quan các chức năng chính đã triển khai**

| **Bước** | **Tên chức năng** | **File liên quan** |
| --- | --- | --- |
| 1 | Khởi tạo và lấy frame từ camera | function\_1.py |
| 2 | Kiểm tra format tên thư mục (MSSV\_FullName\_Class) | utils.py |
| 3 | Chụp ảnh sinh viên | function\_2.py |
| 4 | Phát hiện và cắt khuôn mặt | function\_3.py |
| 5 | Tiền xử lý ảnh khuôn mặt (Optional) | preprocessing.py |
| 6 | Huấn luyện model nhận diện khuôn mặt | function\_4.py |
| 7 | Nhận diện khuôn mặt realtime | function\_4.py |
| 8 | Ghi log điểm danh vào file CSV | utils.py |
| 9 | Xây dựng giao diện GUI và Log Output | main.py |

## **3. Mô tả chi tiết từng bước**

#### **Bước 1: Khởi tạo và lấy frame từ camera**

*File: function\_1.py*

* Chức năng:
* Khởi tạo đối tượng camera sử dụng cv2.VideoCapture().
* Liên tục lấy các khung hình (frames) từ camera để hiển thị lên GUI.
* Hỗ trợ giải phóng camera khi đóng chương trình để tránh lỗi chiếm cổng camera.

#### **Bước 2: Kiểm tra format tên thư mục**

*File: utils.py*

* Chức năng:
* Khi sinh viên nhập tên folder lưu ảnh, hệ thống sẽ kiểm tra định dạng.
* Yêu cầu chuẩn format:
* MSSV\_FullName\_Class
* Ví dụ: SE123456\_NguyenVanA\_AI1907
* Nội dung kiểm tra:
* Đúng format MSSV (SE hoặc SS + 2 số xác định K + 4 số).
* Họ tên viết hoa chữ cái đầu.
* Tên lớp đúng định dạng (VD: AI1907, SE2101…)
* Lý do:
* Tránh lỗi dữ liệu, đảm bảo quản lý dễ dàng.

#### **Bước 3: Chụp ảnh sinh viên**

*File: function\_2.py*

* Chức năng:
* Chụp 20 ảnh liên tục từ camera.
* Lưu ảnh vào thư mục theo loại ảnh:
* raw hoặc glasses
* Vị trí lưu:
* data/raw\_images/[image\_type]/[MSSV\_FullName\_Class]/
* Xử lý lỗi:
* Nếu camera chưa được mở hoặc không có frame sẽ báo lỗi.

#### **Bước 4: Phát hiện và cắt khuôn mặt**

*File: function\_3.py*

* Chức năng:
* Dùng Haar Cascade Classifier để detect khuôn mặt trên từng ảnh đã chụp.
* Tham số detect:
  + scaleFactor = 1.05
  + minNeighbors = 6
  + minSize = (20, 20)
* Sau khi detect:
* Cắt vùng chứa khuôn mặt và lưu vào thư mục preprocessed:
  + Nếu Không bật Preprocessing → data/preprocessed\_images/
  + Nếu Có bật Preprocessing → data/preprocessed\_gray\_images/
* Xóa folder raw:
* Sau khi xong sẽ tự động xóa ảnh gốc (raw) để tiết kiệm dung lượng. (Sau khi Preprocessing hoàn thành)

#### **Bước 5: Tiền xử lý ảnh khuôn mặt (Optional)**

*File: preprocessing.py*

* Chức năng:
* Thực hiện các bước nâng cao chất lượng ảnh khuôn mặt:
* Chuyển Grayscale.
* Resize về 150x150 px.
* Histogram Equalization (tăng độ tương phản).
* Gaussian Blur (giảm noise).
* Sharpening (tăng độ nét).
* Mục đích:
* Cải thiện chất lượng dữ liệu đầu vào cho model recognition.

#### **Bước 6: Huấn luyện model nhận diện khuôn mặt**

*File: function\_4.py*

* Chức năng:
* Load toàn bộ ảnh đã preprocessed (cả images/ và images\_with\_glasses/).
* Gán nhãn (label) cho từng người.
* Huấn luyện model LBPHFaceRecognizer với tham số:
  + radius=2
  + neighbors=8
  + grid\_x=6
  + grid\_y=6
* Kiểm tra dữ liệu:
* Nếu dưới 10 ảnh → Cảnh báo log thiếu data để train.
* Output:
* 2 biến quan trọng:
  + recognizer
  + label\_map (Map từ Label ID → Tên người)

#### **Bước 7: Nhận diện khuôn mặt realtime**

*File: function\_4.py*

* Chức năng:
* Dùng camera realtime, lấy từng frame.
* Detect khuôn mặt → Cắt ROI → Preprocess (nếu có).
* Dùng model LBPH đã train để predict label và độ chính xác (confidence).
* Threshold:
  + Điều chỉnh qua GUI slider (default = 50).
* Countdown:
  + Chỉ log điểm danh nếu nhận diện liên tục ≥ 2 giây và mỗi ngày điểm danh 1 lần.
* Hiển thị:
  + Bounding box + Tên + Confidence.

#### **Bước 8: Ghi log điểm danh**

*File: utils.py*

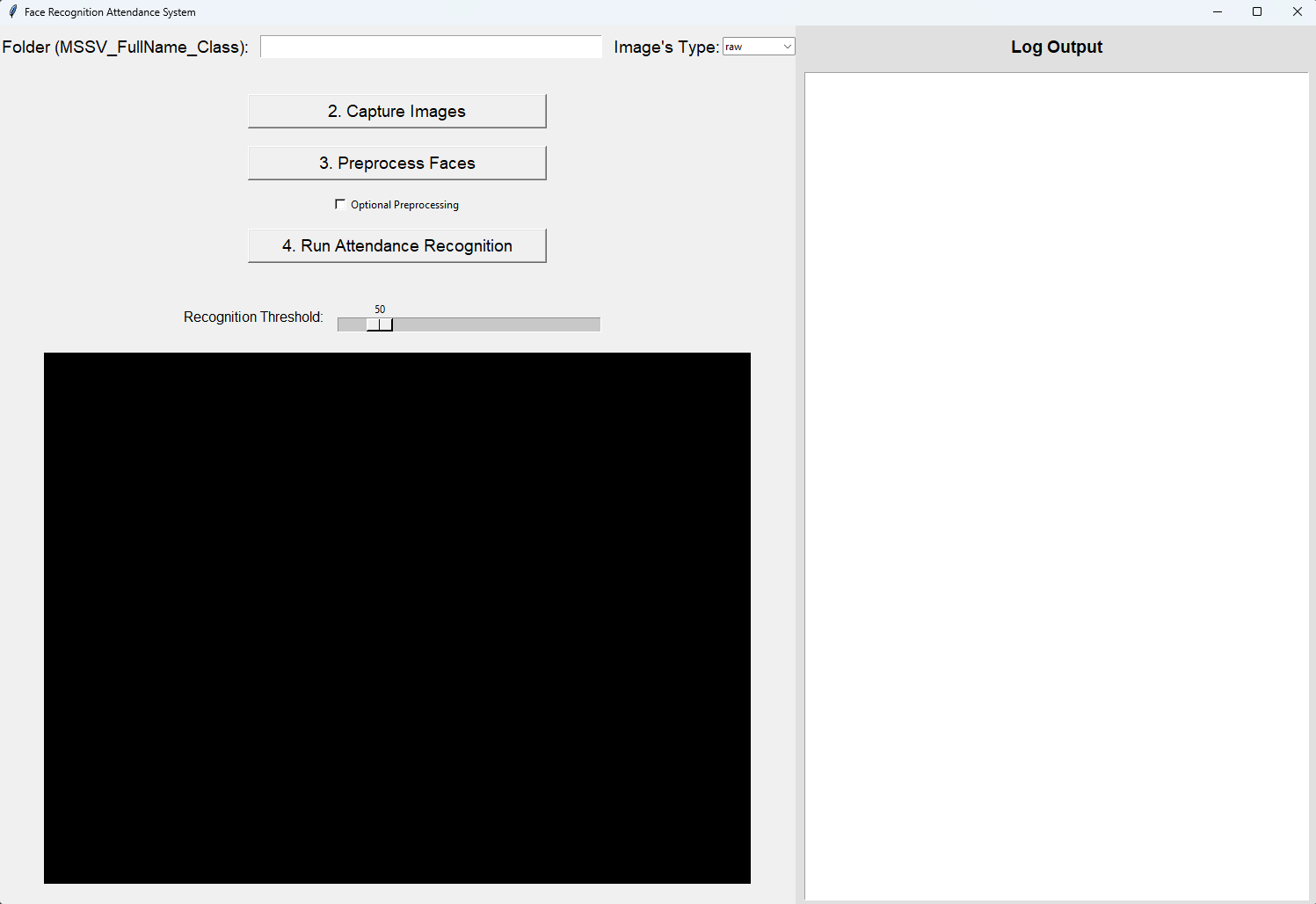
* Chức năng:
* Ghi log vào file CSV theo ngày.
* Ví dụ: logs/attendance\_2025-06-26.csv
* Cơ chế:
* Mỗi sinh viên chỉ được ghi log 1 lần/ngày để tránh trùng lặp.

#### **Bước 9: Xây dựng giao diện GUI và Log Output**

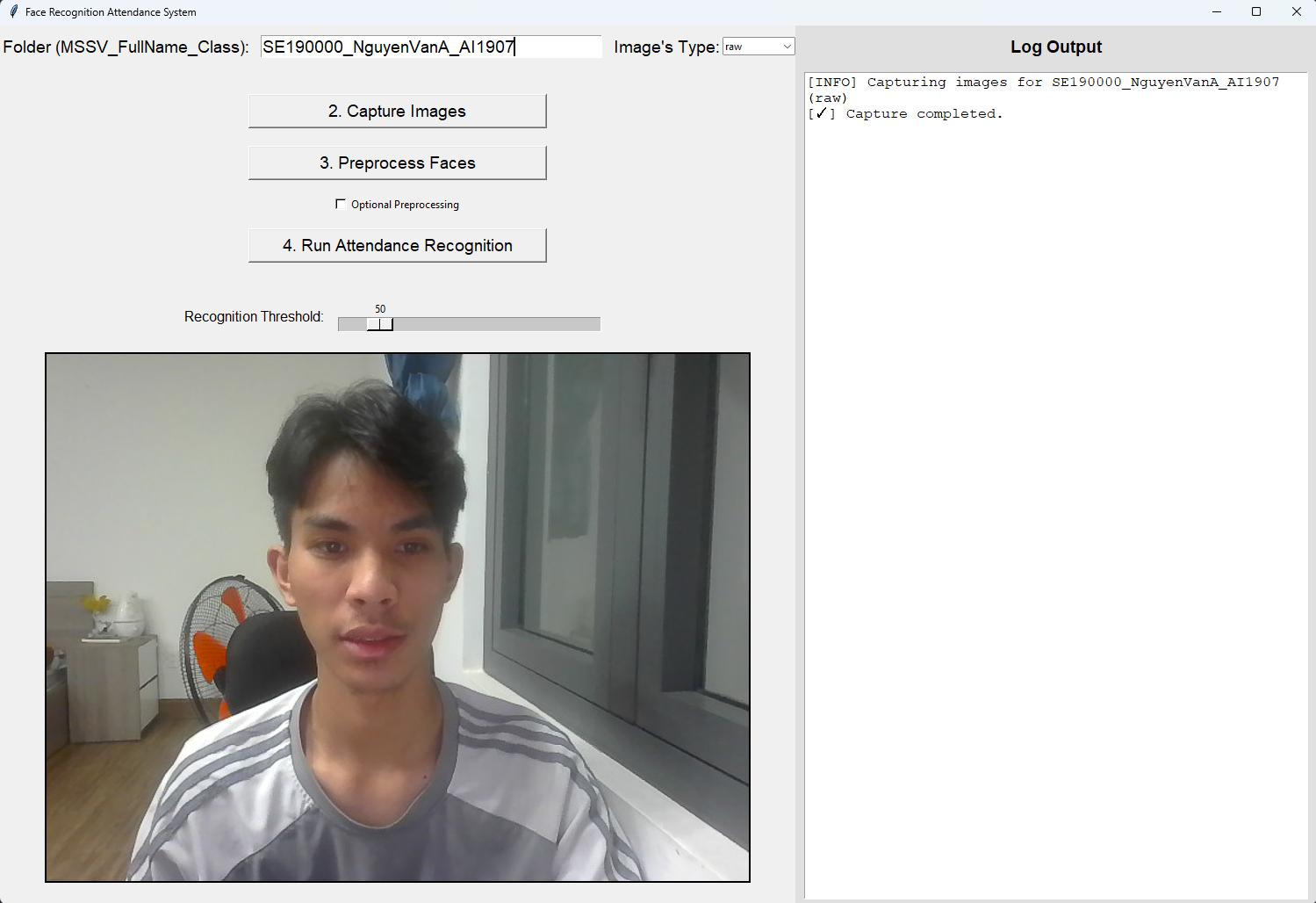
*File: main.py*

* Framework: Tkinter
* Chức năng:
* Hiển thị camera realtime.
* Text box hiển thị log.
* Các button điều khiển từng bước:
  + Capture Images
  + Preprocess Faces
  + Run Training + Attendance Recognition
* Combobox chọn type ảnh (raw, glasses).
* Entry nhập folder name.
* Checkbox bật tắt Preprocessing.
* Slider chỉnh threshold.
* Đa luồng (threading) để không treo GUI khi đang chạy model hoặc chụp ảnh.

## **4. Kết quả thực hiện**



*Giao diện cho Assignment*



*Chụp ảnh người dùng và Log Output sau khi chụp xong*

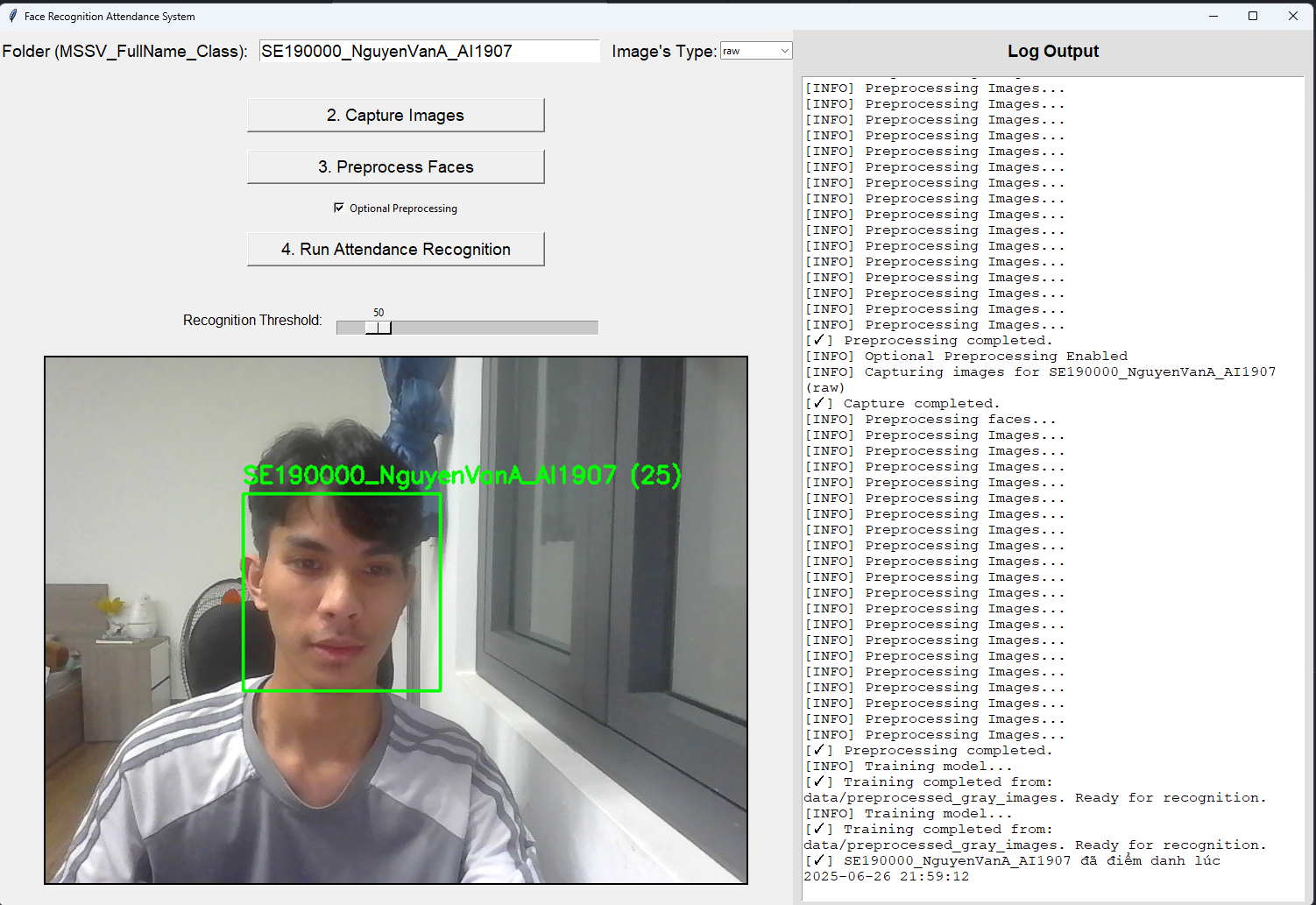


*Ảnh chụp được từ camera*

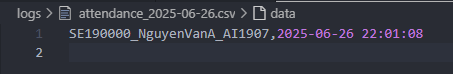


*Không Preprocessing Có Preprocessing*

*Nhận diện mặt và cắt từ ảnh chụp camera*



*Nhận diện khuôn mặt thời gian thực*



*Output là file csv với MSSV\_FullName\_Class + Thời gian điểm danh*