BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

NIÊN LUẬN CƠ SỞ
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài
MÔ HÌNH NHẬN DIỆN KHUÔN MẶT

Sinh viên: Đinh Thành Đạt
Mã số: DC22V7N552
Khóa: K48

Cần Thơ, 11/202

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
KHOA CÔNG NGHỆ THÔNG TIN

NIÊN LUẬN CƠ SỞ
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài
MÔ HÌNH NHẬN DIỆN KHUÔN MẶT

Người hướng dẫn
PGS, TS. TRẦN CÔNG ÁN

Sinh viên thực hiện
Đinh Thành Đạt
Mã số: DC22V7N552
Khóa: K48

Cần Thơ, 11/2025

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

MỤC LỤC

LỜI CẢM ƠN

Lời đầu tiên, em xin bày tỏ lòng biết ơn sâu sắc đến Giảng viên hướng dẫn – Thầy Trần Công Án đã giảng dạy, cung cấp các kiến thức học tập, đưa ra những đóng góp quan trọng trong quá trình học tập và thực hiện đề tài Niên luận cơ sở Xây dựng mô hình nhận diện khuôn mặt bằng Python. Sự hướng dẫn tận tâm, những góp ý quý báu của Thầy đã giúp em có thêm nhiều kiến thức, kinh nghiệm thực tế và hoàn thành bài báo cáo này.

Do còn hạn chế về kiến thức và kinh nghiệm thực hiện đề tài, bài báo cáo chắc chắn không thể tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp, nhận xét và phê bình từ Thầy để bài báo cáo được hoàn thiện hơn trong tương lai.

Cuối cùng, em xin kính chúc Thầy dồi dào sức khỏe, nhiều niềm vui, thành công trong công tác giảng dạy và nghiên cứu khoa học.

Em xin chân thành cảm ơn!

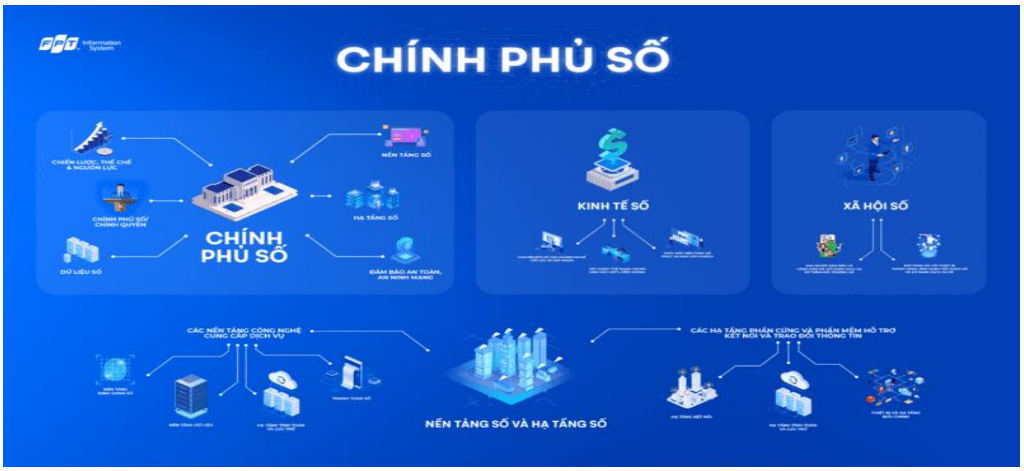
PHẦN GIỚI THIỆU

I. Giới thiệu đề tài

Trong tình hình bối cảnh công nghiệp 4.0 phát triển nhanh chóng và xu hướng toàn cầu hóa, việc ứng dụng trí tuệ nhân tạo, thị giác máy tính và các công nghệ số vào đời sống và sản xuất đang trở thành yêu cầu cấp thiết của cả thế giới nói chung và nước ta nói riêng. Hiện nay, ở nước ta quá trình "chuyển đổi số quốc gia" đã được xác định là một trong ba trụ cột chiến lược phát triển gồm Chính phủ số, kinh tế số và xã hội số. Với định hướng chuyển đổi số quốc gia Chính phủ đã ban hành Chương trình chuyển đổi số quốc gia từ năm 2025 đến năm 2030.

Theo báo cáo, kinh tế số của Việt Nam năm 2024 đạt tỷ trọng khoảng 18,3% trong GDP cả nước, tăng trưởng nhanh so với năm trước. Cũng trong năm này, tỷ lệ hồ sơ dịch vụ công trực tuyến toàn trình của cả nước đạt khoảng 45% vào cuối tháng 11, thể hiện bước tiến rõ rệt trong việc hiện đại hóa hành chính và dịch vụ công.

Ngoài ra, Việt Nam đã vươn lên vị trí thứ 71/193 quốc gia về xếp hạng chính phủ điện tử, tăng 15 bậc so với năm 2022. Xuất phát từ bối cảnh trên, đề tài **"Xây dựng chương trình nhận diện khuôn mặt bằng Python"** được thực hiện với mong muốn kết hợp kiến thức về xử lý ảnh, học máy và công nghệ nhận diện khuôn mặt để tạo nên một ứng dụng có tính thực tiễn và ứng dụng cao trong thời đại số. Hệ thống này hướng tới việc phát hiện và nhận dạng khuôn mặt người dùng thông qua webcam, từ đó mở ra các khả năng ứng dụng như kiểm soát an ninh, điểm danh tự động, hoặc hỗ trợ trong các hệ thống thông minh.



Hình 1. Chuyển đổi số của Chính phủ

Quá trình thực hiện đề tài gồm việc nghiên cứu các thư viện như OpenCV, face_recognition, dlib, os,…; phân tích thuật toán phát hiện và nhận dạng khuôn mặt; thiết kế giao diện và luồng xử lý; lập trình và thử nghiệm thực tế. Trong quá trình đó, sinh viên đã gặp và khắc phục nhiều thách thức về môi trường phát triển, ánh sáng khi quay camera, tốc độ nhận dạng…

II. Mục tiêu đạt được

Xây dựng một chương trình nhận diện khuôn mặt thời gian thực bằng Python, có khả năng phát hiện và nhận dạng khuôn mặt thông qua webcam, đáp ứng yêu cầu về độ chính xác và tốc độ xử lý để có thể ứng dụng trong các hệ thống điểm danh, kiểm soát ra/vào và các ứng dụng an ninh cơ bản.

III. Đối tượng và phạm vi nghiên cứu

1. Đối tượng nghiên cứu

Công nghệ nhận diện khuôn mặt dựa trên một số thuật toán, mô hình máy học (machine learning) và học sâu (deep learning) dùng để phát hiện và nhận diện khuôn mặt. Kết hợp với các thư viện phần mềm trong ngôn ngữ lập trình để hỗ trợ trong quá trình triển khai. Các thư viện phần mềm đều có các chức năng và các thế mạnh riêng ví dụ như trong dự án "Xây dựng mô hình Nhận diện khuôn mặt" thư viện *face_recognition* hỗ trợ cho việc cung cấp một số API đơn giản giúp cho việc phát hiện, nhận diện khuôn mặt, thư viện OpenCV hỗ trợ thị giác máy tính giúp xử lý hình ảnh, video, vẽ đồ họa, …

2. Phạm vi nghiên cứu

Chương trình nhận diện khuôn mặt được áp dụng mới quy mô nhỏ (từ 10 – 30 hình ảnh).

Dữ liệu được tạo trước từ dữ liệu được thu thập từ nhiều nguồn như internet, chụp ảnh,...

Chương trình được chạy trên máy tính cá nhân có Webcam.

Thông tin ghi nhận sự xuất hiện sẽ được thêm vào tệp lưu trữ được tạo ra trong quá trình phát triển.

IV. Phương pháp nghiên cứu

Sử dụng các thư viện có sẵn trong ngôn ngữ lập trình Python hỗ trợ trong việc xử lý hình ảnh, phát hiện khuôn mặt, huấn luyện, phân tích dữ liệu, cung cấp các mảng đa chiều,…, có thể nói đến một số thư viện thường sử dụng như: OpenCV, NumPy, face_recognition, dlib, …

Thu thập dữ liệu: thu thập từ những hình ảnh có sẵn, lưu trữ vào từng tệp, thư mục riêng và gán nhãn cho chúng với mỗi nhãn là khác nhau

Xử lý dữ liệu: sử dụng thư viện OpenCV có sẵn để phát hiện khuôn mặt trong mỗi ảnh, nếu khuôn mặt không có trong ảnh đó sẽ bỏ qua và thực hiện tới ảnh kế tiếp.

Mã hóa đặc trưng khuôn mặt: chuyển đổi ảnh thành Vector đặc trưng thông qua mô hình học máy trong thư viện "face_recognition".

Nhận diện và so sánh: khi hệ thống nhận được khung hình mới, chương trình mã hóa khuôn mặt hiện tại, so sánh với dữ liệu đã lưu và xác định danh tính dựa trên độ tương đồng.

Ghi nhận kết quả: nếu khuôn mặt được nhận dạng thành công, hệ thống lưu tên và thời gian nhận diện vào tệp "thamdu.csv".

V. Nội dung nghiên cứu

Nghiên cứu tập trung vào quá trình xây dựng và phát triển mô hình nhận diện khuôn mặt đơn giản, tự động bằng ngôn ngữ lập trình Python, sử dụng hình ảnh có sẵn, kết hợp với các thư viện phổ biến trong lĩnh vực xử lý ảnh và học máy.

Các nội dung nghiên cứu chính bao gồm:

- Nghiên cứu tổng quan về nhận dạng khuôn mặt
  - Tìm hiểu các phương pháp phát hiện và nhận dạng khuôn mặt
  - Lựa chọn thư viện "face_recognition" dựa trên mô hình học sâu (Deep learning) để xác định độ chính xác và tốc độ xử lý
- Xây dựng mô hình nhận diện khuôn mặt
  - Thu thập, lưu trữ, gán nhãn dữ liệu
  - Mã hóa dữ liệu từng khuôn mặt thành Vector đặc trưng bằng phương pháp face_recognition.
  - Lưu danh sách mã hóa làm cơ sở so sánh khi nhận diện thời gian thực.
- Phát hiện và nhận diện khuôn mặt thời gian thực
  - Sử dụng webcam để ghi nhận hình ảnh đầu vào.
  - Xác định vị trí khuôn mặt bằng hàm "face_locations" của thư viện "face_recognition".
  - So sánh các đặc trưng khuôn mặt hiện tại với dữ liệu đã lưu thông qua hàm "compare_face" và "face_distance" để tìm người trùng khớp.
- Ghi nhận kết quả nhận diện
  - Khi khuôn mặt được nhận dạng, chương trình ghi lại tên người và thời gian xuất hiện vào tệp "thamdu.csv"
  - Nếu khuôn mặt không trùng khớp với bất kỳ dữ liệu nào, hệ thống gán nhãn là "Unknow"**.**
- Kiểm thử và đánh giá kết quả
  - Thực hiện các thử nghiệm nhận diện trong điều kiện ánh sáng, góc nhìn và độ phân giải khác nhau.
  - Đánh giá độ chính xác, tốc độ nhận diện và khả năng ổn định của hệ thống.

VI. Bố cục quyển niên luận

Bố cục của Niên luận cơ sở gồm có ba phần chính là phần giới thiệu, phần nội dung và phần kết luận. Phần giới thiệu sẽ là phần đặt ra vấn đề, mục tiêu nghiên cứu, phạm vi, đối tượng nghiên cứu. Phần nội dung sẽ là mô tả tổng quan của chương

trình, phân tích, mô tả chức năng, cài đặt và thiết kế mô hình. Phần kết luận sẽ là đưa ra kết qua đạt được, hướng phát triển đề tài sau này.

Ngoài ra, bên cạnh 3 phần chính nêu trên thì niên luận còn có phần giới thiệu, mục lục, danh mục bảng, hình ảnh,… và cuối cùng sẽ có phần tài liệu tham khảo.

PHẦN NỘI DUNG
CHƯƠNG 1 ĐẶC TẢ YÊU CẦU

I. Mô tả tổng quan

1. Bối cảnh của chương trình

Trong kỷ nguyên số hóa, công nghệ nhận diện khuôn mặt đang dần trở thành một phần tất yếu trong hoạt động của chính phủ, các doanh nghiệp tư nhân và cả trong đời sống xã hội. Từ việc mở khóa ứng dụng, xác minh danh tính, chấm công cho đến các hệ thống an ninh giám sát phức tạp đã mở ra vô số cơ hội nhưng cũng đặt ra nhiều thách thức mới. Việc ứng dụng nhận dạng khuôn mặt hiện nay đang là tất yếu, hầu hết có ở những nơi có quy mô lớn như sân bay, nhà ga, các khách sạn, công ty, trên những tuyến đường giao thông,…, đến những thiết bị hiện đại trong đời sống hằng ngày như camera, điện thoại, máy tính, máy chấm công,… Nên việc ứng dụng, đẩy mạnh và phát triển nhận diện khuôn mặt trở thành một trong những xu hướng hiện nay. Công nghệ này không những giúp con người có thể nâng cao được mức độ an toàn, tin cậy và tránh rủi ro.

Hình 2. Ứng dụng của nhận diện khuôn mặt trong đời sống

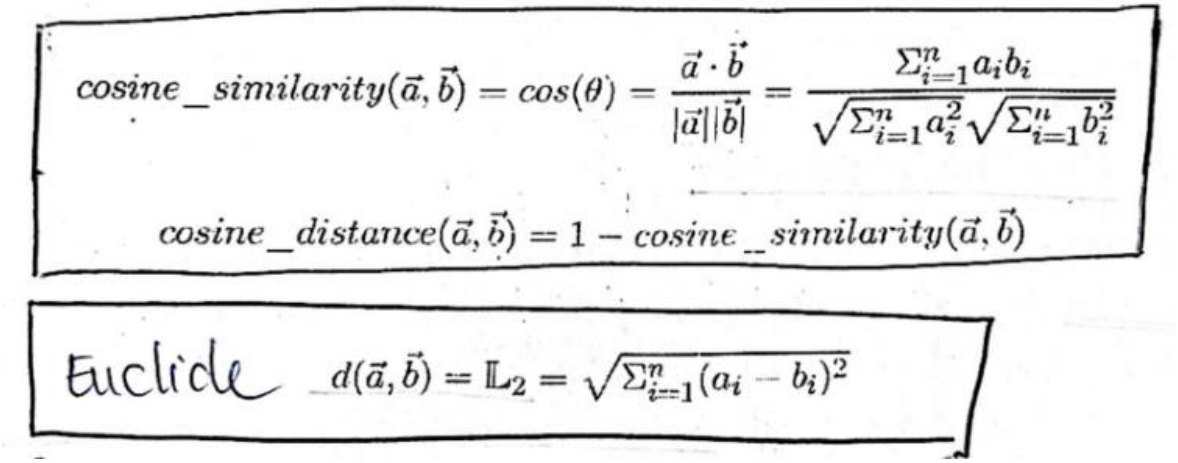
2. Tìm hiểu chung về nhận dạng khuông mặt

Hệ thống nhận dạng khuôn mặt (Face Recognition System) dựa trên học sâu (Deep Learning) hoạt động theo một quy trình nhiều giai đoạn được liên kết và hoạt động liên tục với nhau. Kiến trúc này đảm bảo tính chính xác và mạnh mẽ khi thực hiện xử lý hình ảnh thực tế. Các giai đoạn xử lý được chia làm 3 giai đoạn chính:

- Phát hiện khuôn mặt: mục tiêu của giai đoạn này là xác định và vẽ khung xác định xung quanh các khuôn mặt có trong khung hình.

- Trích xuất đặc trưng: sử dụng thuật toán Neural tích chập (CNN) tự động học một bộ các đặc trưng có tính phân biệt từ dữ liệu hình ảnh thô, thay vì dựa vào việc trích xuất đặc trưng thủ công (như pixel ảnh, hoặc các bộ lọc cạnh truyền thống). Mỗi lớp tích chập sử dụng nhiều bộ lọc. Các bộ lọc này sẽ đi qua ảnh đầu vào để tính toán tích chập, tạo ra các bản đồ đặc trưng. Sau khi quá trình tích chập và gộp hoàn tất, dữ liệu sẽ được chuyển thành các Vector đặc trưng 128 chiều. Với các Vector đặc trưng của những người giống nhau thì vector càng gần nhau và ngược lại.

- Đối chiếu và nhận dạng: với ngưỡng tương đồng giữa Vector đặc trưng đầu vào mới (từ khuôn mặt vừa phát hiện) và tất cả các Vector đặc trưng đã lưu trong cơ sở dữ liệu. Sử dụng khoảng cách Euclide đo độ dài hai vector, giá trị càng nhỏ thì khuôn mặt càng giống nhau. Ngoài ra, còn có thể sử dụng cách đo độ tương đồng Cosine Similarity để đo góc giữa hai vector. Để việc nhận dạng chính xác hơn, cần có một giá trị cố định để làm ngưỡng quyết định. Nếu khoảng cách khuôn mặt nhỏ hơn ngưỡng quyết định thì xác định danh tính và ngược lại.



Hình 3. Khoảng cách cosine_similarity và khoảng cách Euclide

3. Chức năng của chương trình

Chương trình nhận diện khuôn mặt nhằm đáp ứng việc phát hiện và nhận diện khuôn mặt theo dữ liệu đã có sẵn, xác định đúng thông tin, theo dõi vị trí khuôn mặt theo thời gian thực và lưu lại thời gian khuôn mặt xuất hiện cho đến khi kết thúc.

Các chức năng chính của chương trình:

- Đọc dữ liệu khuôn mặt đầu vào.
- Mã hóa dữ liệu khuôn mặt.
- Nhận diện khuôn mặt thời gian thực qua Webcam.
- Ghi nhận thông tin.
- Hiển thị thông tin khuôn mặt.
- Kết thúc chương trình nhận diện.

4. Yêu cầu phi chức năng

- Đạt được yêu cầu về mặt thời gian.
- Phát hiện và nhận diện khuôn mặt trong môi trường thiếu sáng, thiếu thông tin.
- Hiển thị thông tin dễ nhìn.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

I. Thiết kế tổng thể của mô hình

Hệ thống được thiết kế theo kiến trúc xử lý tuần tự theo ba giai đoạn chính:

- Giai đoạn huấn luyện (Data Preparation & Encoding): Đọc dữ liệu khuôn mặt đã biết từ thư mục nguồn và mã hóa chúng thành các vector số học (Face Encodings).

- Giai đoạn xử lý theo thời gian thực (Real-time Processing): Khởi động webcam, đọc từng khung hình (frame), và phát hiện khuôn mặt trên khung hình đó.

- Giai đoạn nhận dạng và ghi nhận (Recognition & Logging): So sánh khuôn mặt được phát hiện với các khuôn mặt đã mã hóa . Nếu khớp, tên sẽ được hiển thị và ghi nhận vào tệp điểm danh.

II. Cơ sở lý thuyết

Mô hình nhận diện khuôn mặt này được xây dựng trên nền tảng thị giác máy tính (Computer Vision), sử dụng các thư viện OpenCV và face_recognition để thực hiện việc phát hiện, nhận dạng khuôn mặt và ghi lại thông tin.

- Phát hiện khuôn mặt (Face Detection): Sử dụng thư viện dlib để xác định vị trí của khuôn mặt trong ảnh, thường thông qua mô hình mạng Neural tích chập (CNN) cho độ chính xác cao hơn.

1. Thuật toán Neural Tích chập (CNN)

Thuật toán mạng Neural Tích chập (CNN: Convolutional Neural Networks): là mô hình học sâu trong lĩnh vực thị giác máy tính nhằm hỗ trợ cho việc xử lý dữ liệu không gian như là hình ảnh, video, …. Có khả năng tự động trích xuất và học các đặt trưng phức tạp của dữ liệu. Mạng Neural tích chập sử dụng phép toán tích chập quét các dữ liệu đầu vào bằng bộ lọc (Kernel) có kích thước nhỏ gồm nhiều Pixel trọng số di chuyển trên toàn bộ ảnh thực hiện phép nhân – cộng tạo ra các bản đồ đặc trưng. Mỗi bộ lọc (Kernel) có một dạng đặc trưng nhất định như cạnh ngang, cạnh dọc, đường cong, hoa văn,… từ đó có khả năng nhận diện các mẫu lặp lại trong ảnh một cách hiệu quả.

Cấu trúc mạng Neural có 3 lớp chính:

- Lớp tích chập: sử dụng các bộ lọc (Kernel) di chuyển trên hình ảnh tìm các đặc trưng cụ thể trong ảnh tạo ra một bộ lọc (Kernel) mới làm nổi bật nơi tìm thấy các mẫu này.

- ReLU: giúp mạng học được mối quan hệ phi tuyến tính giữa các đặc điểm trong hình ảnh, nên làm cho mạng mạnh mẽ hơn để xác định các mẫu khác nhau.

- Lớp gộp: kéo các đặc điểm quan trọng nhất từ ma trận phức tạp, làm giảm chiều của bản đồ đặc trưng (ma trận phức tạp).

2. Thư viện face_recognition

Mô hình nhận diện khuông mặt bằng thư viện Python face_racognition sử dụng kỹ thuật Face Embedding, mỗi khuôn mặt sau khi được xử lý sẽ được biểu diễn dưới

dạng một Vector đặc trưng có kích thước cố định, thường là 128 chiều. Để đo mức độ tương đồng giữa hai khuôn mặt, hệ thống thường sử dụng khoảng cách Euclide (Euclidean Distance). Khoảng cách này được tính giữa hai Vector đặc trưng tương ứng với hai khuôn mặt cần so sánh. Nếu khoảng cách hai vector này nhỏ hơn ngưỡng tương đồng đặt ra thì xem như hai khuôn mặt cùng thuộc về một người.

III. Chức năng của mô hình

Thành phần/ Hàm	Chức năng chi tiết
myList = os.listdir(path)	Tải dữ liệu: Đọc tên tất cả các tệp hình ảnh (dữ liệu huấn luyện) trong thư mục dataSet_1
Mahoa(images)	Mã hóa khuôn mặt: Chuyển đổi hình ảnh từ BGR sang RGB (chuẩn hóa), sau đó sử dụng face_recognition. face_encodings để tạo ra danh sách các vector 128 chiều (encodeListKnow).
thamdu(name)	Ghi nhận điểm danh: Đọc tệp thamdu.csv. Kiểm tra nếu tên (name) chưa có trong danh sách điểm danh, thì ghi thêm tên và thời gian hiện tại (now. strtime("%H:%M:%S")) vào cuối tệp.
Vòng lặp while True	Xử lý thời gian thực: Đọc khung hình từ webcam (cap. read()), giảm kích thước hình ảnh (cv2.resize) để tăng tốc độ xử lý, và chuyển đổi màu sắc (BGR sang RGB).
face_locations và face_encodings	Phát hiện và mã hóa: Tìm vị trí và tạo vector 128D cho mỗi khuôn mặt trong khung hình hiện tại.
compare_faces và face_distance	So sánh: Tính khoảng cách giữa khuôn mặt hiện tại và tất cả khuôn mặt đã biết.
cv2.rectangle và cv2.putText	Hiển thị: Vẽ khung (bounding box) và tên (name) lên khuôn mặt đã nhận dạng trên màn hình hiển thị.

Bảng 1. Chức năng của mô hình

IV. Thiết kế cơ sở dữ liệu

1. Dữ liệu khuôn mặt đã biết

Lưu trữ vật lý: Các tệp ảnh khuôn mặt được đặt trong thư mục *dataSet_1*. Tên tệp được sử dụng làm danh tính (className).

Lưu trữ mã hóa: Vector 128 chiều của các khuôn mặt này được lưu trong biến encodeListKnow (dạng numpy array) trong bộ nhớ, phục vụ cho việc so sánh nhanh chóng.

2. Dữ liệu ghi nhận

Được lưu trong tệp *thamdu.csv,* mỗi dòng trong tệp là một bảng ghi nhận gồm hai trường thông tin tên và thời gian ghi nhận được phân tách bằng dấu phẩy.

CHƯƠNG 3 CÀI ĐẶT GIẢI PHÁP VÀ KẾT QUẢ ĐẠT ĐƯỢC

I. Cài đặt giải pháp

1. Môi trường cài đặt

Mô hình được triển khai hoàn toàn trên nền tảng Python 3.10 trong VScode và yêu cầu phải cài đặt các thư viện cốt lõi như thư viện OpenCV và thư viện dlib cho các tác vụ sinh trắc học. Cấu trúc dự án đơn giản nhưng phản ánh rõ ràng sự tách biệt giữa dữ liệu nguồn và logic xử lý:

- Nhan_Dien_Khuon_Mat.py: Chứa toàn bộ logic xử lý và điều khiển luồng.
- dataSet_1: Thư mục đầu vào chứa ảnh huấn luyện.
- thamdu.csv: thư mục ghi lại thời gian khi phát hiện và nhận diện được khuôn mặt có trong thư mục dữ liệu.

Trước tiên khi bắt tay vào giải quyết mô hình, chúng ta cần thêm các thư viện cần thiết vào trong tệp lập trình như:

- Thư viện cv2: dùng để hỗ trợ xử lý hình ảnh và video: điều khiển Webcam, đọc ghi hình ảnh, chuyển đổi không gian màu của hình ảnh, giảm kích thước ảnh và hiển thị kết quả
- Thư viện os: cho phép duyệt qua các tệp trong trong mục và trích xuất tên tệp chuẩn bị cho quá trình huấn luyện.
- Thư viện face_racognition: hỗ trợ thực hiện các thuận toán nhận dạng khuôn mặt phức tạp như mã hóa 128 chiều, so sánh khuôn mặt, và tính khoảng cách Euclide.
- Thư viện numpy: là thư viện toán học để xử lý Vector: thao tác hiệu quả với các mảng, vector số học đặc biệt với các Vector đặc trưng 128 chiều, tìm kiếm giá trị nhỏ nhất,…
- Thư viện datetime: chung cấp các chức năng quản lý thời gian dùng để ghi nhận thời gian hiện tại đưa vào tệp dữ liệu sau này.

```
1  import cv2
2  import os
3  import face_recognition
4  import numpy as np
5  import datetime
```

Hình 4. Thêm thư viện cần thiết vào lập trình

2. Quá trình triển khai

Tạo kho hình ảnh cần nhận dạng với dữ liệu trong mô hình này được lấy từ ảnh chụp, internet. Các hình ảnh được đưa vào chung một thư mục và đặt tên theo từng ảnh sao cho mỗi ảnh là một tên khác nhau.

Hình 5. Dữ liệu được tạo sẵn trong thư mục

Với mô hình nhận diện khuôn mặt cần chú trọng vào 3 bước chính để đảm bảo mô hình có thể hoạt động chính xác, phát hiện, nhận diện khuôn mặt với dữ liệu sẵn có. Các bước chính gồm:

- Thao tác với kho ảnh mẫu nhận dạng trong đó:

Cần load kho ảnh mẫu nhận dạng. Đầu tiên cần khai báo biết "path" chứa đường dẫn tương đối tới thư mục dữ liệu nguồn, nơi lưu trữ tất cả hình ảnh khuôn mặt đã biết. Khởi tạo lần lượt hai danh sách rỗng "images" và "className", chứa các đối tượng hình ảnh dưới dạng ma trận pixel và chứa danh sách tên của các cá nhân ứng với mỗi hình ảnh. Dùng thư viện "os" để kiểm tra toàn bộ tên hình ảnh có trong thư mục. Tạo vòng lặp duyệt qua từng tên của hình ảnh trong danh sách "myList". Dùng OpenCV để đọc từng đường dẫn của tệp ảnh hiện tại sau đó tệp ảnh vừa được đọc sẽ được thêm vào danh sách "images". Sử dụng "splitext()" để tách phần đuôi mở rộng với phần danh tính của hình ảnh và thêm vào danh sách "className" cho việc hiển thị danh tính khuôn mặt ra màn hình sau này.

```
path = 'dataSet_1'
images = []
className = []
myList = os.listdir(path)
print(myList)
for cl in myList:
    curimg = cv2.imread(f"{path}/{cl}")
    images.append(curimg)
    className.append(os.path.splitext(cl)[0])
```

Hình 6. Khai báo thư mục và đọc tên tệp trong thư mục

- Xác định vị trí khuôn mặt tại một thời điểm bất kỳ: sử dụng lệnh "face_recognition.face_locations" để phát hiện khuôn mặt và sử dụng thư viện dlib tìm tọa độ tất cả khuôn mặt con người trong khung hình, trả về định dạng đầu ra theo thứ tự (trên, phải, dưới, trái). Sau đó tạo ra các Vector đặc trưng cho mỗi khuôn mặt được phát hiện trong khung hình.

```
# Xác định vị trí khuôn mặt
facecurFrame = face_recognition.face_locations(framE)
encodecurFrame = face_recognition.face_encodings(framE)
```

Hình 7. Mã hóa và tạo ra Vector đặc trưng của khuôn mặt

- Mã hóa các điểm trên khuôn mặt: cần khai báo trước một biến "encodeList" chứa các hình ảnh đã được mã hóa. Tạo một vòng lặp duyệt qua từng ma trận hình ảnh. Chuyển đổi màu sắc của hình ảnh từ định dạng BGR (mặc định của thư viện OpenCV) sang RGB. Thực hiên việc mã hóa hình ảnh với lệnh "encode = face_recognition.face_encodings(img)", lệnh này sử dụng mô hình CNNs để trích xuất Vector đặc trưng 128 chiều của khuôn mặt. Thêm Vector đặc trưng 128 chiều vừa tạo ra vào danh sách. Kết thúc trả về danh sách chứa tất cả các Vector đặc trưng đã tạo. Hàm mã hóa thực thi với những Vector đặc trưng đã biết. Khi mã hóa thành công toàn bộ thư viện ảnh thì sẽ in ra màn hình "MÃ HÓA THÀNH CÔNG"

```
#ma hoa
def Mahoa(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)
        encodeList.append(encode[0])
    return encodeList

encodeListKnow = Mahoa(images)
print("MÃ HÓA THÀNH CÔNG")
```

Hình 8. Mã hóa hình ảnh

- Thực hiện khởi động Webcam của mô hình để phát hiện khuôn mặt cho các luồng xử lý tiếp theo.

- Với vòng lặp chính là "while True" thực hiện công việc phát hiện và nhận dạng khuôn mặt theo thời gian thực với nguồn dữ liệu hình ảnh lớn việc xử lý mất nhiều thời gian nên mô hình cần sử dụng "cv2.resize" giảm tỉ lệ khung hình gốc xuống 50% (fx = 0.5, fy = 0.5) để việc tối ưu thời gian xử lý trong thời gian thực và tiếp tục chuyển đổi định dạng màu BGR sang RGB để phù hợp với yêu cầu xử lý của face_recognition. Với việc Webcam đã được khởi động và ghi nhận khuôn mặt theo thời gian thực thư viện "face_recognition" xác định và trích xuất dữ liệu khuôn mặt, tìm ra tọa độ vị trí của tất cả các khuôn mặt có trong khung hình, thực hiện mã hóa khuôn mặt thành các Vector đặc trưng và đây cũng là dữ liệu cho việc so sánh nhận dạng. Hệ thống lặp qua từng khuôn mặt được phát hiện, tính toán khoảng cách khuôn mặt giữa khuôn mặt hiện tại và khuôn mặt đã được mã hóa trước đó. Khoảng cách nhỏ hơn ngưỡng tương đồng thì khuôn mặt được xác định là khớp với cá nhân đã biết. Tên tương ứng sẽ được ghi nhận sự hiện diện và thời gian vào thư mục

"thamdu.csv", nếu khoảng cách lớn hơn ngưỡng tương đồng sẽ được gán nhãn "Unknow". Hiển thị hình chữ nhật bao quanh vị trí khuôn mặt theo thời gian thực và hiển thị tên của khuôn mặt theo dữ liệu đã đưa vào trước đó.

```
# Khởi động Webcam
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    framE = cv2.resize(frame, (0, 0), None, fx=0.5, fy=0.5)  # tỉ lệ hình ảnh hiển thị bằng 50% hình ảnh gốc
    framE = cv2.cvtColor(framE, cv2.COLOR_BGR2RGB)
    # Xác định vị trí khuôn mặt
    facecurFrame = face_recognition.face_locations(framE)  # lay tung vi tri tren khuon mat hien tai
    encodecurFrame = face_recognition.face_encodings(framE)

    for encodeFace, faceLocations in zip(encodecurFrame, facecurFrame):  # lay tung vi tri tren khuon mat hien tai theo cap
        matches = face_recognition.compare_faces(encodeListKnow, encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnow, encodeFace)
        print(faceDis)  # in khoảng cách khuôn mặt hiện tại với khuôn mặt trong ds
        matchIndex = np.argmin(faceDis)
        if matches:
            if faceDis[matchIndex] < 0.50: # ngưỡng khoảng cách khuôn mặt
                name = className[matchIndex].upper()
                thamdu(name)
            else:
                name = 'Unknow_close'
        else:
            name = 'Unknow'

        # vẽ tên ra màn hình
        y1, x2, y2, x1 = faceLocations
        y1, x2, y2, x1 = y1 * 2, x2 * 2, y2 * 2, x1 * 2
        color = (0, 255, 0) if name not in ['Unknow', 'Unknow_close'] else (0, 0, 255)
        cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
        cv2.putText(frame, name, (x2, y2), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 0), 2)
        cv2.putText(frame, f"faceDis: {faceDis[matchIndex]:.2f}", (x1, y1 - 10), cv2.FONT_HERSHEY_COMPLEX, 0.5, (0, 255, 0), 2)
    cv2.imshow("Mô hình Nhận diện khuôn mặt", frame)
    if cv2.waitKey(1) == ord('q'):
        break
```

Hình 9. Quy trình nhận dạng khuôn mặt theo thời gian

- Kết thúc quá trình phát hiện và nhận diện khuôn mặt: sử dụng lệnh cap.release() đảm bảo giải phóng Webcam cho các ứng dụng khác có thể sử dụng camera. Lệnh cv2.destroyAllWindows() giúp đóng tất cả các của số hiển thị của thư viện OpenCV.

- Các khuôn mặt được nhận diện và so sánh với dữ liệu có sẵn, mô hình sẽ ghi nhận lại sự xuất hiện đó và lưu vào tệp "thamdu.csv". Việc ghi nhận sẽ ghi nhận tên và thời gian xuất hiện, hỗ trợ cho các tính năng sau này.

```
from datetime import datetime
with open ("thamdu.csv", 'r+', encoding='utf-8') as f:
        myDatalist = f.readlines()
        print(myDatalist)
        for line in myDatalist:
                entry = line.split(",")
                print(entry)
                print(line)
now = datetime.now()
print(now)
```

Hình 10. Thư mục ghi nhận sự xuất hiện của khuôn mặt

CHƯƠNG 4 ĐÁNH GIÁ KẾT QUẢ KIỂM THỬ

I. Giới thiệu

1. Phạm vi kiểm thử

Kiểm thử mới một số khuôn mặt có sẵn trong dữ liệu, một số khuôn mặt bên ngoài dữ liệu. Ngoài ra còn kiểm thử với các đồ vật không phải là khuôn mặt con người.

2. Mục tiêu kiểm thử

Phát hiện khuôn mặt, xác định vị trí khuôn mặt, nhận diện, so sánh với dữ liệu, hiển thị thông tin khuôn mặt nhận diện được và ghi nhận sự xuất hiện.

II. Đánh giá kết quả kiểm thử

Kết quả kiểm thử đạt yêu cầu với đặc tả yêu cầu đưa ra, tài liệu thiết kế đáp ứng với yêu cầu mong đợi.

Hình 11. Khuôn mặt được nhận diện

thamdu.csv

You, 1 second ago | 1 author (You)

```
1  Name, Time
2  DINH_THANH_DAT,22:03:22
3  DINH_THANH_DAT,22:03:23
4  DINH_THANH_DAT,22:03:23
5  DINH_THANH_DAT,22:03:24
6  DINH_THANH_DAT,22:03:25
7  DINH_THANH_DAT,22:03:25
```

Hình 12. Ghi nhận sự xuất hiện

PHẦN KẾT LUẬN

I. Kết quả đạt được

Sau khi nghiên cứu và tiến hành thực hiện đề tài, em đã hoàn hành mô hình nhận diện khuôn mặt với các yêu cầu sau:

- Phát hiện được khuôn mặt con người
- Xác định được vị trí của khuôn mặt
- Nhận diện được khuôn mặt ở thời gian thực
- So sánh với dữ liệu đã được huấn luyện
- Hiển thị chính xác thông tin
- Lưu trữ được thời gian xuất hiện.

II. Hạn chế

- Mô hình hạn chế nhiều tính năng.
- Khả năng phát hiện và năng nhận diện khuôn mặt trong môi trường thiếu ánh sáng chưa thực hiện được.

III. Hướng phát triển

- Cải thiện độ chính xác cao hơn
- Phát hiện và nhận diện được trong môi trường không hoàn hảo
- Phát triển quy mô, giao diện tốt hơn.
- Hiển thị nhiều thông tin.

TÀI LIỆU THAM KHẢO

[1] C. đ. số, "dx.moj.gov.vn," [Online]. Available: https://dx.moj.gov.vn.

[2] thudna, "Nhìn lại năm 2024: Chuyển đổi số Việt Nam vươn tầm bứt phá," 26 12 2024. [Online]. Available: https://dx.moj.gov.vn/nhin-lai-nam-2024-chuyen-doi-so-viet-nam-vuon-tam-but-pha-828.htm. [Accessed 05 11 2025].

[3] "quantrimang.com," [Online]. Available: quantrimang.com.

[4] 3school, "3school," 06 11 2025. [Online]. Available: http://www.w3schools.com.

[5] https://vdigital.vn/tong-quan-va-dinh-huong-chuyen-doi-so/, "VDigital," 2025. [Online]. Available: https://vdigital.vn/tong-quan-va-dinh-huong-chuyen-doi-so/.

[1] [2]

PHỤ LỤC

I. Hướng dẫn cài đặt môi trường

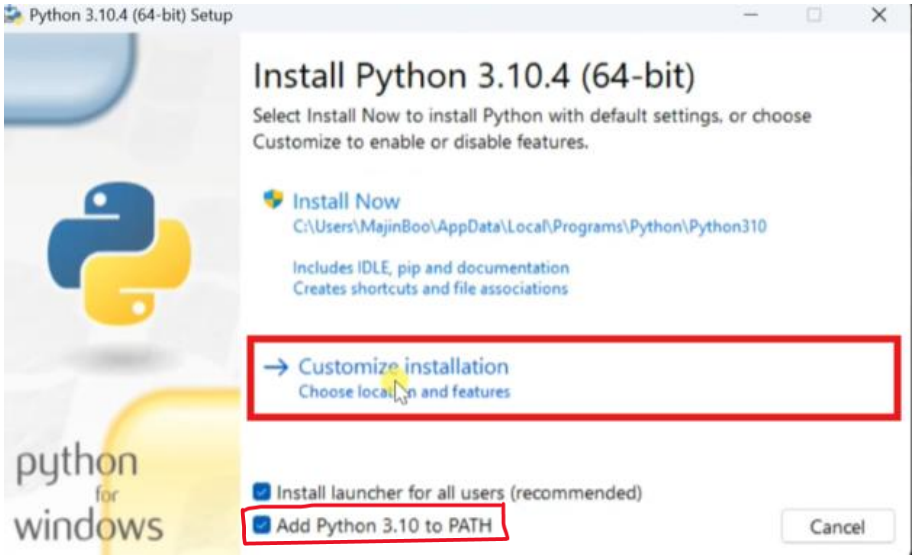
1. Cài đặt ngôn ngữ lập trình

Trong đề tài thực hiện "mô hình nhận diện khuôn mặt" ngôn ngữ chính được sử dụng trong mô hình là ngôn ngữ Python với phiên bản 3.10 *(tùy vào lựa chọn cho phù hợp với máy tính)* Link để tải Python:

https://www.python.org/ftp/python/3.10.6/python-3.10.6-amd64.exe

Các bước thực hiện như sau:

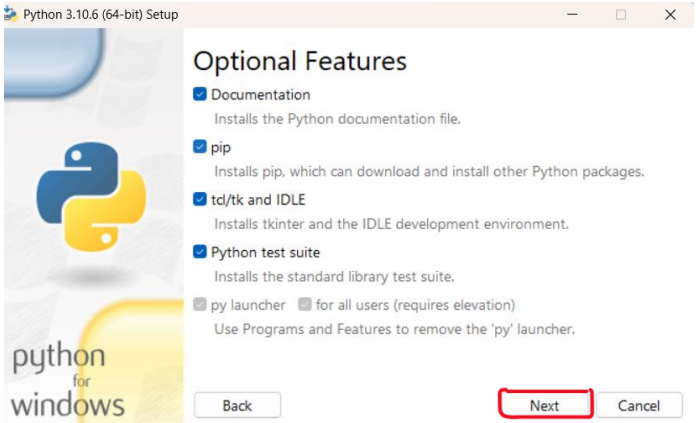
Bước 1: Thực hiện cài đặt ngôn ngữ, Chọn "**Customize installtion**" để tối giản đường dẫn lưu phần mềm.



Hình 13. Cài đặt phần mềm

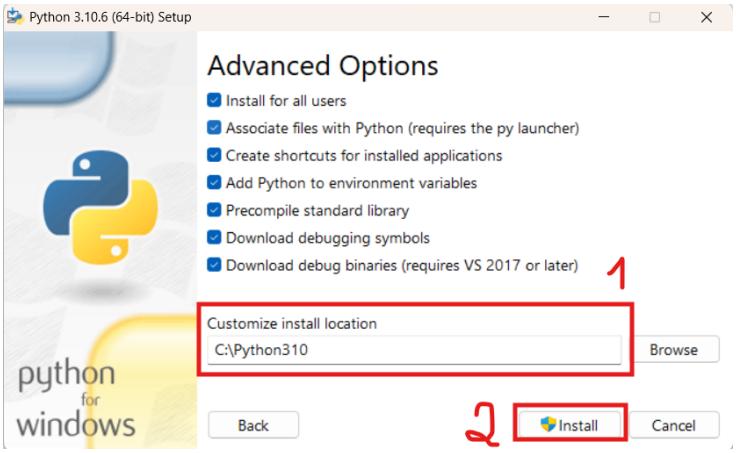
Lưu ý: chọn vào ô "Add python 3.10 to PATH**"*

Bước 2: Chọn next



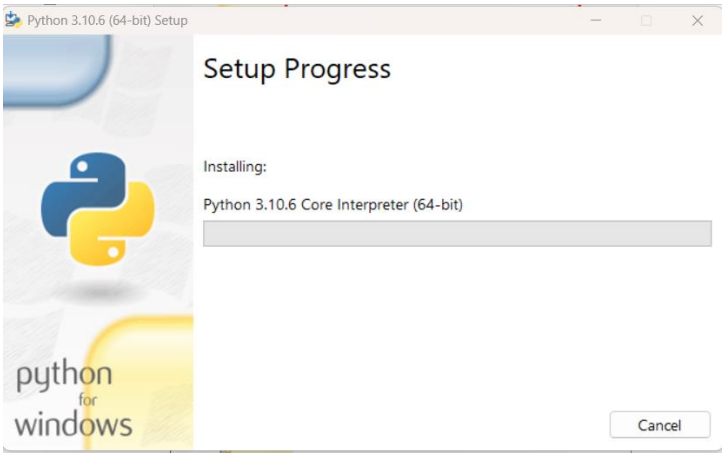
Hình 14. Chọn các cài đặt

Bước 3: Chọn tất cả các mục, sau đó tối giản đường dẫn lưu phần mềm nhất có thể để thuận tiện cho việc lấy dữ liệu sau này.



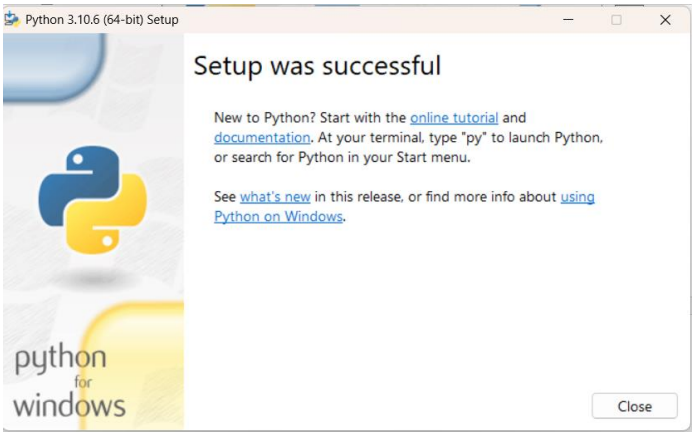
Hình 15. Tối giản đường dẫn lưu phần mềm

Bước 5: Thực hiện cài đặt phần mềm



Hình 16. Phần mềm đang được cài đặt

Phần mềm đã cài đặt thành công. Chọn Close để đóng

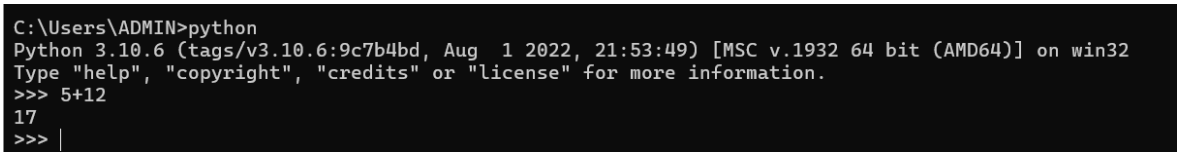


Hình 17. Phần mềm được cài đặt thành công

Kiểm tra xem phần mềm Python đã có trên máy hay chưa, hãy gõ cmd tại thanh tìm kiếm. Thực hiện lệnh python –version, nếu hiện thị như hình … thì đã cài đặt thành công.

Hình 18. Kiểm tra phần mềm đã được cài đặt

Kiểm tra phần mềm có thể chạy trên máy tính được không, ta thực hiện lệnh python sáu đó nhấn Enter, ta giả sử thực hiện 1 phép tính đơn giản để kiểm tra. Nếu kết quả phép tính đúng thì có thể hoạt động bình thường.



Hình 19. Kiểm tra hoạt động của phần mềm

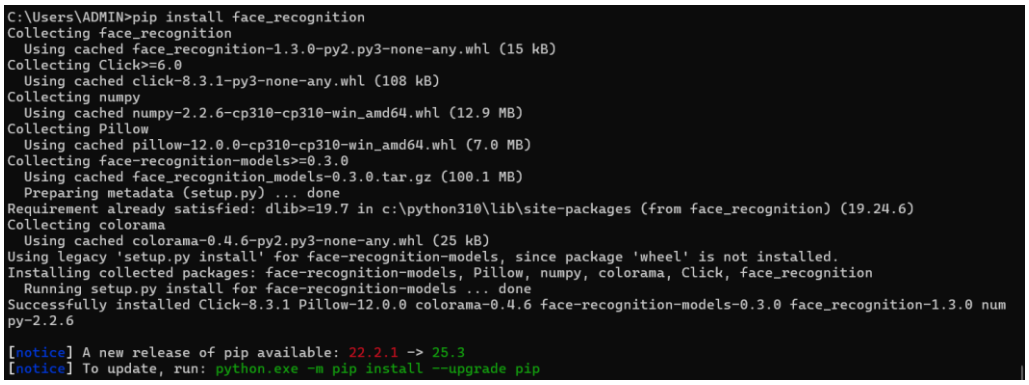
2. Cài đặt thư viện

Để có thể thực hiện "mô hình nhận diện khuôn mặt" cần thêm vào các thư viện cần thiết như: "face_recognition", "OpenCV", "dlib", … Để cài đặt, ta thực hiện một số lệnh sau:

- Tải tệp whl để chuẩn bị cho bước cài đặt thư viện dlib

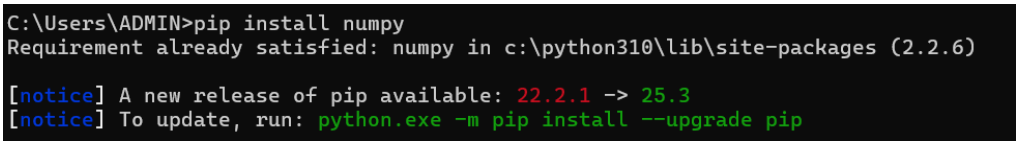
  dlib/dlib-19.24.6-cp310-cp310-win_amd64.whl

- Cài đặt thư viện face_recognition với lệnh "pip install face_recognition"



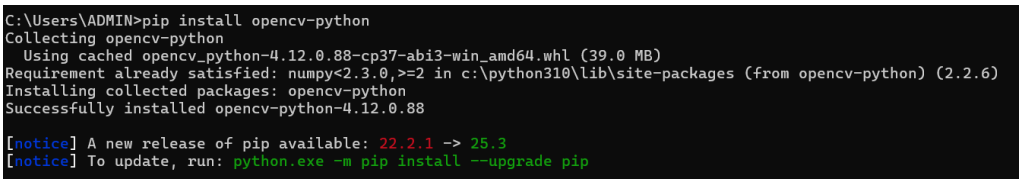
Hình 20. Cài đặt thư viện face_recognition

- Cài đặt thư viện numpy với lệnh "pip install numpy"



Hình 21. Cài đặt thư viện numpy

- Cài đặt thư viện OpenCV với lệnh “pip install opencv-python”



Hình 22. Cài đặt thư viện opencv