

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG



ĐỒ ÁN MÔN HỌC 2

Face detection and identity with machine learning

NGÀNH KỸ THUẬT MÁY TÍNH

Sinh viên: **NGUYỄN HIỀN**
MSSV: 20161190
PHẠM TRẦN NHẬT TÂN
MSSV: 20119277

Hướng dẫn: TS. **HUỲNH THẾ THIỆN**

TP. HỒ CHÍ MINH, 08/2023

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG



ĐỒ ÁN MÔN HỌC 2

Face detection and identity with machine learning

NGÀNH KỸ THUẬT MÁY TÍNH

Sinh viên: **NGUYỄN HIỀN**
MSSV: 20161190
PHẠM TRẦN NHẬT TÂN
MSSV: 20119277

Hướng dẫn: TS. **HUỲNH THẾ THIỆN**

TP. HỒ CHÍ MINH, 08/2023

LỜI CẢM ƠN

Để hoàn thành đề tài đồ án môn học này , lời đầu tiên cho phép nhóm được gửi lời cảm ơn chân thành đến toàn thể quý thầy cô Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM nói chung và các thầy cô trong Khoa Điện - Điện tử nói riêng, những người đã tận tình dạy dỗ, trang bị cho nhóm những kiến thức nền tảng và kiến thức chuyên ngành quan trọng, giúp nhóm có được cơ sở lý thuyết vững vàng và đã luôn tạo điều kiện giúp đỡ tốt nhất cho nhóm trong quá trình học tập và nghiên cứu.

Đặc biệt, nhóm xin chân thành cảm ơn thầy Huỳnh Thế Thiện đã tận tình giúp đỡ, đưa ra những định hướng nghiên cứu cũng như hướng giải quyết một số vấn đề để nhóm có thể thực hiện tốt đề tài. Trong thời gian làm việc với thầy, nhóm đã không ngừng tiếp thu thêm nhiều kiến thức được chỉ dạy từ thầy, luôn thể hiện một thái độ nghiên cứu nghiêm túc, hiệu quả và đây cũng là điều rất cần thiết trong quá trình học tập và làm việc sau này đối với nhóm.

Mặc dù đã cố gắng hết sức, song do điều kiện thời gian và kinh nghiệm của nhóm còn ít, cho nên để tài không thể tránh khỏi thiếu sót. Vì vậy, nhóm rất mong nhận được sự đóng góp ý kiến của quý thầy, cô giáo.

Nhóm xin chân thành cảm ơn

Nhóm thực hiện đồ án môn học 2
(Ký và ghi rõ họ tên)

Nguyễn Hiền Phạm Trần Nhật Tân

LỜI CAM ĐOAN

Nhóm xin cam đoan thực hiện cam đoan rằng bài báo cáo này là kết quả của công trình nghiên cứu và biên soạn chân thực của chúng nhóm. Nhóm cam đoan rằng mọi thông tin, dữ liệu, số liệu và tài liệu tham khảo được trích dẫn trong báo cáo này là hoàn toàn chính xác và được sử dụng một cách đúng mực.

Nhóm cam đoan rằng nhóm đã thực hiện tất cả các công đoạn nghiên cứu và viết báo cáo này một cách trung thực và không vi phạm bất kỳ quy tắc đạo đức hoặc chuẩn mực nào liên quan đến việc tham khảo và sử dụng thông tin từ nguồn khác. Chúng tôi không sao chép, biên tập lại hoặc sử dụng thông tin từ nguồn khác mà không ghi rõ nguồn gốc. Các nội dung tham khảo đã được trích dẫn đầy đủ.

Nhóm thực hiện đồ án môn học 2
(Ký và ghi rõ họ tên)

Nguyễn Hiền Phạm Trần Nhật Tân

TÓM TẮT NỘI DUNG

Với sự phát triển không ngừng của khoa học và công nghệ, đặc biệt là các thiết bị được hỗ trợ công nghệ xử lý ảnh ngày càng hiện đại và được sử dụng phổ biến trong đời sống con người đã làm cho lượng thông tin thu được bằng hình ảnh ngày càng tăng và phổ biến. Theo đó, lĩnh vực xử lý ảnh cũng được chú trọng phát triển, ứng dụng rộng rãi trong đời sống xã hội hiện đại. Không chỉ dừng lại ở việc chỉnh sửa, tăng chất lượng hình ảnh mà với công nghệ xử lý ảnh hiện nay chúng ta có thể giải quyết các bài toán nhận dạng chữ viết, nhận dạng dấu vân tay, đặc biệt là nhận dạng khuôn mặt...

Công nghệ nhận diện khuôn mặt (Facial Recognition Technology) hiện là một công nghệ đang được sử dụng khá phổ biến tại các quốc gia phát triển. Công nghệ này có khả năng xác định hoặc nhận một người từ hình ảnh kỹ thuật số được lấy mẫu trước đó hoặc từ một khung hình trong một nguồn video khác. Đây là một phương pháp xác minh độc đáo khi thiết bị sẽ dựa vào những điểm khác nhau tiêu biểu nhất trên khuôn mặt của một người để tiến hành phân biệt giữa người này với người khác. Do vậy đối với các trường hợp như song sinh thì người dùng có thể yên tâm rằng máy vẫn sẽ phát hiện ra. Chính vì đặc điểm này thì ngoài được ứng dụng trong việc quản lý nhân sự ra thì nó còn là sự lựa chọn của rất nhiều đơn vị hoạt động trong lĩnh vực an ninh, bảo mật, gia dịch.

Xây dựng một ứng dụng nhận dạng khuôn mặt nhằm giúp việc quản lý, điểm danh, giao dịch hay thống kê theo từng yêu cầu phục vụ cho các mục đích khác nhau là cấp thiết. Ngoài ra, giúp sinh viên tìm hiểu về thư viện OpenCV, tìm hiểu các phương pháp xác định khuôn mặt (Face Detection).

Mục lục

1 TỔNG QUAN	1
1.1 Giới thiệu	1
1.2 Mục tiêu	2
1.3 Tình hình nghiên cứu	2
1.4 Phương pháp nghiên cứu	2
1.5 Bố cục nội dung	3
2 CƠ SỞ LÝ THUYẾT	4
2.1 Bài toán nhận diện khuôn mặt	4
2.1.1 Nhận diện khuôn mặt là gì?	4
2.1.2 Các bước xây dựng thuật toán nhận diện khuôn mặt	5
2.2 Machine Learning (ML)	6
2.2.1 Machine Learning là gì ?	6
2.2.2 Machine Learning Workflow	7
2.2.3 Phân loại Machine learning	7
2.3 K-nearest neighbor (KNN)	8
2.4 Thuật toán HOG (Histogram of oriented gradient)	10
2.4.1 Các bước tạo ra vector HOG	10
2.5 Haar Cascade Classifier	13
2.5.1 Haar Cascade là gì?	13
2.5.2 Cách hoạt động của Haar Cascade	13
3 THIẾT KẾ HỆ THỐNG	17

3.1	Giới thiệu	17
3.2	Sơ đồ khái niệm và chi tiết từng khái	17
3.2.1	Sơ đồ khái	17
3.2.2	Chi tiết từng khái	18
3.3	Giao diện hệ thống	22
3.3.1	Flask python là gì?	22
3.3.2	Các tính năng cơ bản của Flask	23
3.3.3	Phương thức Flask Python hoạt động trong hệ thống điểm danh bằng nhận diện khuôn mặt.	23
4	KẾT QUẢ	28
4.1	Kết quả nghiên cứu	28
4.2	Hệ thống điểm danh sinh viên	28
4.2.1	Giao diện người dùng	29
4.2.2	Hoạt động của hệ thống	30
4.3	Video demo hệ thống	32
5	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	33
5.1	Kết luận	33
5.2	Hướng phát triển	34

Danh sách bảng

Danh sách hình vẽ

2.1	Ví dụ hệ thống nhận diện khuôn mặt	4
2.2	Sơ đồ Machine Learning	6
2.3	KNN tính khoảng cách k điểm gần nhất (k=4)	8
2.4	Liệt kê và chọn nhãn xuất hiện nhiều nhất	9
2.5	Tạo 2 ma trận gradient [1]	11
2.6	Tìm Histogram bins cho mỗi cell	11
2.7	Biểu đồ Histogram of Gradient	12
2.8	Biểu diễn phân phối HOG trên ảnh	13
2.9	Đặc trưng Haar-like	14
2.10	Các vùng đặc trưng trên khuôn mặt	14
2.11	Sơ đồ bộ phân loại cascade gồm các bộ phân loại đơn	16
3.1	Sơ đồ khối hệ thống nhận diện khuôn mặt.	17
3.2	Phát hiện khuôn mặt bằng Haar Cascade.	18
3.3	Lưu đồ thuật toán huấn luyện.	20
3.4	Lưu đồ thuật toán nhận dạng.	21
3.5	Flask python.	22
4.1	Kết quả hiển thị trên terminal.	29
4.2	Giao diện hệ thống điểm danh.	29
4.3	Cửa sổ phát hiện và chụp ảnh khuôn mặt làm ảnh huấn luyện.	30
4.4	Thư mục data\faces chứa ảnh đã chụp và huấn luyện.	31
4.5	Cửa sổ phát hiện khuôn mặt và điểm danh.	31
4.6	Thông tin người dùng điểm danh.	32

Danh sách các hàm

3.1	Hàm sử dụng thuật toán HOG để trích xuất đặc trưng	19
3.2	Hàm khởi tạo Flask	23
3.3	Routes và Templates	24
3.4	Hàm khởi động ứng dụng Flask	24
3.5	Hàm home()	25
3.6	Hàm start()	25
3.7	Hàm add()	26

Danh mục các từ viết tắt

Dưới đây là danh mục các từ viết tắt được sử dụng trong luận văn.

Các từ viết tắt	Định nghĩa
AI	Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
KNN	K-nearest neighbor
CNN	Convolutional Neural Network
HOG	Histogram of oriented gradients
AdaBoost	Adaptive Boosting

Chương 1

TỔNG QUAN

1.1 Giới thiệu

Trong thời đại của sự phát triển vượt bậc của khoa học và công nghệ, hay ngành trí tuệ nhân tạo (AI). Trong đó học sâu (DL) và máy học (ML) đã nhanh chóng trở thành một công cụ mạnh mẽ trong việc giải quyết các vấn đề phức tạp trong nhiều lĩnh vực. Trong lĩnh vực xử lý ảnh và thị giác máy tính, học sâu đã tạo ra những bước tiến đáng kể trong việc nhận diện đối tượng và khuôn mặt. Sự phát triển nhanh chóng của học sâu đã mở ra những cơ hội mới để nghiên cứu và ứng dụng trong việc phát hiện khuôn mặt và nhận diện danh tính.

Nhận diện khuôn mặt và xác định danh tính từ ảnh hoặc video đã trở thành một trong những lĩnh vực quan trọng trong thị giác máy tính. Các ứng dụng như hệ thống an ninh, quản lý thời gian, giao thông, trải nghiệm người dùng cá nhân và nhiều lĩnh vực khác đều đòi hỏi sự hiện đại và chính xác trong việc nhận dạng con người thông qua khuôn mặt. Với sự phát triển của các mô hình học sâu và học máy ví dụ như KNN hay CNN..., việc phát triển các hệ thống phát hiện khuôn mặt và nhận diện danh tính đã trở nên tiềm năng hơn bao giờ hết.

Năm bắt được sự phát triển và với mong muốn tìm hiểu trao đổi thêm kiến thức quan trọng về ngành trí tuệ nhân tạo (AI) cụ thể là về máy học (ML) nhóm quyết định chọn “Face detection and identity with Machine learning” để làm đề tài ở môn Đồ Án 2 lần này.

1.2 Mục tiêu

Vận dụng kiến thức đã học từ các môn học như xử lý ảnh, cơ sở ứng dụng AI, đồ án 1 để nghiên cứu và thiết kế. Đề tài “Face detection and identity with machine learning” gồm có các mục tiêu chính như sau:

- Tìm hiểu tổng quan về Machine learning (ML), các thuật toán nhận dạng và phân loại, ứng dụng thực tế vào mô hình của hệ thống.
- Nhận dạng được khuôn mặt trong các frame được nhận từ webcam.
- Trích xuất đặc trưng và phân loại từng khuôn mặt. Sau đó gán nhãn thành các nhóm tương ứng với từng cá nhân khác nhau.
- Sử dụng dữ liệu đã được gán nhãn và đặc trưng, tiến hành huấn luyện mô hình học máy để nhận diện khuôn mặt sao cho nó có thể phân loại và nhận diện khuôn mặt một cách chính xác.
- Xây dựng giao diện hệ thống điểm danh sinh viên để ứng dụng đề tài vào thực tế.

1.3 Tình hình nghiên cứu

Hiện nay, phát hiện và nhận diện khuôn mặt là một chủ đề nghiên cứu quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Nhiều nghiên cứu đã được thực hiện để tăng cường hiệu suất và độ chính xác của các hệ thống phát hiện và nhận diện khuôn mặt. Đây là một số công trình nghiên cứu liên quan: Face Detection and Recognition Using Face Mesh and Deep Neural Network [2], Tiềm năng từ công nghệ nhận diện khuôn mặt Make in Vietnam đạt chuẩn Mỹ [3],..

1.4 Phương pháp nghiên cứu

Để đạt được mục tiêu của đề tài, nhóm sử dụng các phương pháp thu thập số liệu, thực nghiệm và phân tích tổng kết kinh nghiệm.

- Phương pháp thu thập số liệu: thu thập các số liệu về khuôn mặt từ nhiều nguồn khác nhau. Dữ liệu bao gồm ảnh khuôn mặt của nhiều người, có chứa nhiều biến thể như

góc chụp, ánh sáng, môi trường, biểu cảm khuôn mặt, v.v. Điều này giúp đảm bảo tính đa dạng và đại diện cho dữ liệu.

- Phương pháp thực nghiệm: Nhóm tiến hành thiết kế và triển khai hệ thống "Điểm danh sinh viên" từ các thuật toán phát hiện và nhận diện khuôn mặt.
- Phương pháp phân tích tổng kết kinh nghiệm:
 - Đánh giá hiệu quả: Nhóm tiến hành đánh giá hiệu quả của hệ thống "Điểm danh sinh viên" từ các kết quả về độ chính xác thu được.
 - Tổng kết kinh nghiệm: Dựa trên kết quả phân tích, nhóm tổng kết kinh nghiệm thu được sau khi thực hiện đề tài

1.5 Bố cục nội dung

Báo cáo đồ án môn học 2 của nhóm sẽ bao gồm 5 chương:

- Chương 1: Tổng quan
- Chương 2: Cơ sở lý thuyết
- Chương 3: Thiết kế hệ thống
- Chương 4: Kết quả
- Chương 5: Kết luận và hướng phát triển.

Chương 2

CƠ SỞ LÝ THUYẾT

2.1 Bài toán nhận diện khuôn mặt

2.1.1 Nhận diện khuôn mặt là gì?

Nhận diện khuôn mặt là thuật toán xác định danh tính của một cá nhân bằng đặc trưng khuôn mặt của họ. Hệ thống nhận diện khuôn mặt thường được sử dụng để xác định một người thông qua hình ảnh hoặc video trong thời gian thực.



Hình 2.1: Ví dụ hệ thống nhận diện khuôn mặt

Công nghệ nhận diện khuôn mặt mang lại nhiều ứng dụng hữu ích và thiết thực trong cuộc sống hàng ngày như phần mềm mở khóa điện thoại, hệ thống chấm công không chạm, hay các hệ thống camera giám sát an ninh, an toàn trong các trung tâm thương mại, cửa hàng, trường học, bệnh viện...

2.1.2 Các bước xây dựng thuật toán nhận diện khuôn mặt

Nhiều người đã quen thuộc với công nghệ nhận diện khuôn mặt thông qua các phần mềm được sử dụng để mở khóa điện thoại thông minh (tuy nhiên, đây chỉ là một ứng dụng nhỏ của công nghệ này). Thông thường, tính năng được sử dụng để mở khóa điện thoại không dựa vào cơ sở dữ liệu ảnh khổng lồ nhằm xác định danh tính của một cá nhân – nó chỉ xác định và công nhận một người là chủ sở hữu duy nhất của thiết bị, đồng thời hạn chế quyền truy cập của những người khác.

Ngoài việc mở khóa điện thoại, tính năng này cũng hoạt động nhằm mục đích khớp khuôn mặt của những người đi ngang qua máy ảnh với hình ảnh của những người trong danh sách đang theo dõi. Danh sách theo dõi là một tập dữ liệu mở được đến từ các nguồn được cung cấp công khai. Các hệ thống công nghệ trên khuôn mặt có thể khác nhau, nhưng nhìn chung, chúng có xu hướng hoạt động theo một số bước như sau:

- **Bước 1: Nhận diện khuôn mặt**

Máy ảnh phát hiện và định vị hình ảnh của một khuôn mặt khi đối tượng đang ở một mình hoặc trong một đám đông. Từ đó ghi nhận hình ảnh của đối tượng.

- **Bước 2: Phân tích khuôn mặt**

Hình ảnh khuôn mặt đối tượng được chụp lại và tiến hành phân tích. Hầu hết công nghệ nhận diện khuôn mặt dựa trên hình ảnh 2D thay vì 3D vì nó có thể khớp hình ảnh 2D với ảnh công khai trong cơ sở dữ liệu một cách thuận tiện và chính xác hơn. Sau đó đưa vào phần mềm để tiến hành đọc hình dạng khuôn mặt của đối tượng. Các yếu tố chính cần xử lý bao gồm khoảng cách giữa hai mắt, độ sâu của hốc mắt, khoảng cách từ trán đến cằm, hình dạng của gò má và đường viền của môi, tai và cằm. Mục đích là để xác định các điểm mốc chính trên khuôn mặt nhằm phân biệt nhiều người khác nhau

- **Bước 3: Chuyển đổi hình ảnh thành dữ liệu**

Chuyển đổi ảnh chụp khuôn mặt thành một tập hợp thông tin kỹ thuật số (dữ liệu) dựa trên các đặc điểm khuôn mặt của người đó. Những thuộc tính khuôn mặt sẽ được xử lý dưới dạng các thuật toán và công thức, và mỗi người sẽ sở hữu một dữ liệu khuôn mặt riêng biệt.

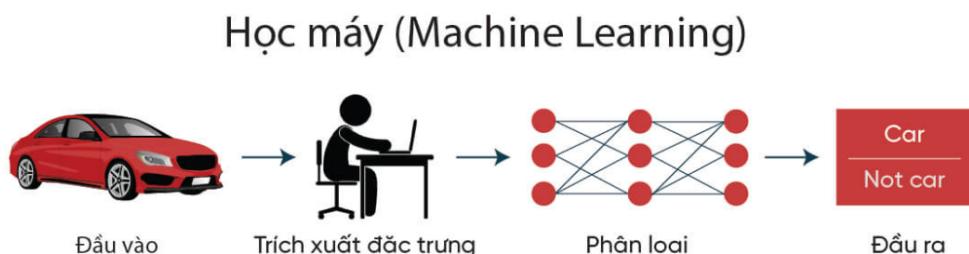
- Bước 4: Tìm kết quả nhận diện khuôn mặt phù hợp

Bước tiếp theo tiến hành so sánh với cơ sở dữ liệu khuôn mặt sẵn có. Nếu khuôn mặt được ghi lại khớp với một hình ảnh trong cơ sở dữ liệu, thì việc xác định sẽ hoàn tất.

2.2 Machine Learning (ML)

2.2.1 Machine Learning là gì ?

Machine learning (ML) hay máy học là một nhánh của trí tuệ nhân tạo (AI), nó là một lĩnh vực nghiên cứu cho phép máy tính có khả năng cải thiện chính bản thân chúng dựa trên dữ liệu mẫu (training data) hoặc dựa vào kinh nghiệm (những gì đã được học). Machine learning có thể tự dự đoán hoặc đưa ra quyết định mà không cần được lập trình cụ thể.



Hình 2.2: Sơ đồ Machine Learning

Hình 2.2 thể hiện sơ đồ xử lý của Machine Learning với ví dụ đầu vào là một vật (chiếc xe) đầu ra là kết quả dự đoán hay phân loại là xe hay không phải là xe. [4]

Bài toán machine learning thường được chia làm hai loại là dự đoán (prediction) và phân loại (classification). Các bài toán dự đoán như dự đoán giá nhà, giá xe... Các bài toán phân loại như nhận diện chữ viết tay, nhận diện đồ vật...

Các ứng dụng của ML đã quá quen thuộc với con người: xe tự hành của Google và Tesla, hệ thống tự gắn thẻ khuôn mặt trên Facebook, hệ thống gợi ý sản phẩm của

Amazon, hệ thống gợi ý phim của Netflix..., chỉ là một vài trong vô vàn những ứng dụng của trí tuệ nhân tạo và cụ thể là Machine learning.

2.2.2 Machine Learning Workflow

Cụ thể từng bước trong machine learning workflow như sau như sau:

1. Data collection – thu thập dữ liệu: để máy tính có thể học được cần có một bộ dữ liệu (dataset), có thể tự thu thập chúng hoặc lấy các bộ dữ liệu đã được công bố trước đó. Lưu ý là phải thu thập từ nguồn chính thống, có như vậy dữ liệu mới chính xác và máy có thể học một cách đúng đắn và đạt hiệu quả cao hơn.

2. Preprocessing – tiền xử lý: bước này dùng để chuẩn hóa dữ liệu, loại bỏ các thuộc tính không cần thiết, gán nhãn dữ liệu, mã hóa một số đặc trưng, trích xuất đặc trưng, rút gọn dữ liệu nhưng vẫn đảm bảo kết quả... Bước này tốn thời gian nhất tỉ lệ thuận với số lượng dữ liệu ta có. Bước 1 và 2 thường chiếm hơn 70% tổng thời gian thực hiện.

3. Training model – huấn luyện mô hình: bước này là bước huấn luyện cho mô hình hay chính là cho nó học trên dữ liệu chúng ta đã thu thập và xử lý ở hai bước đầu.

4. Evaluating model – đánh giá mô hình: sau khi đã huấn luyện mô hình xong, chúng ta cần dùng các độ đo để đánh giá mô hình, tùy vào từng độ đo khác nhau mà mô hình cũng được đánh giá tốt hay không khác nhau. Độ chính xác của mô hình đạt trên 80% được cho là tốt.

5. Improve – cải thiện: sau khi đã đánh giá mô hình, các mô hình đạt độ chính xác không tốt thì cần được train lại, chúng ta sẽ lặp lại từ bước 3, cho đến khi đạt độ chính xác như kỳ vọng. Tổng thời gian của 3 bước cuối rơi vào khoảng 30% tổng thời gian thực hiện.

2.2.3 Phân loại Machine learning

Có rất nhiều cách phân loại machine learning, thông thường thì machine learning sẽ được phân làm hai loại chính sau:

- Học có giám sát: Thuật toán dự đoán đầu ra của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data,

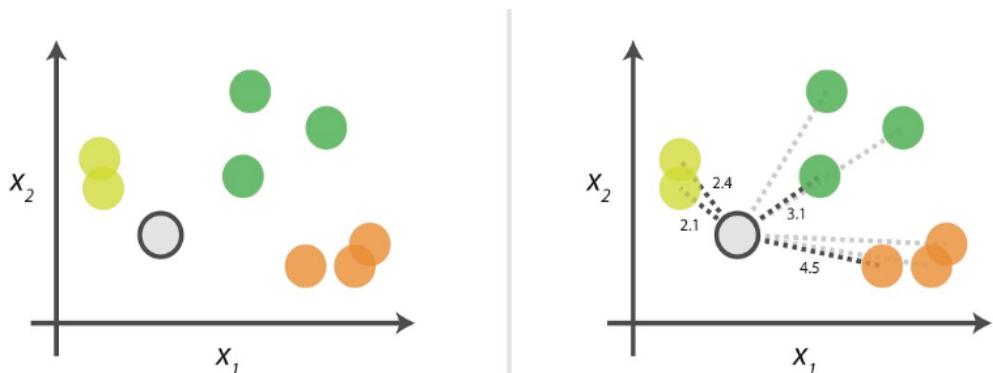
label), tức (dữ liệu, nhãn). Học có giám sát là nhóm phổ biến nhất trong các thuật toán Machine learning.

- Học không giám sát: Là một kỹ thuật của máy học nhằm tìm ra một mô hình hay cấu trúc bị ẩn bởi tập dữ liệu không được gắn nhãn trước. Khác với học có giám sát là không thể xác định trước output từ tập dữ liệu huấn luyện được. Tập dữ liệu huấn luyện của cách học này không do con người gán nhãn, máy tính sẽ phải tự học hoàn toàn. Ứng dụng phổ biến của học không giám sát là bài toán phân cụm

2.3 K-nearest neighbor (KNN)

K-nearest neighbor là một trong những thuật toán "học có giám sát" trong Machine Learning. KNN là một kỹ thuật phi tham số lưu trữ tất cả các trường hợp có sẵn và phân loại các trường hợp mới dựa trên phép đo (hàm khoảng cách). Do đó, khi phân loại một tập dữ liệu mới bằng mô hình KNN đã được huấn luyện, thuật toán này sẽ so sánh dữ liệu huấn luyện và tìm k mẫu huấn luyện gần nhất với mẫu mới. Sau đó, nó gán nhãn cho mẫu mới dựa trên số lượng nhãn nhiều nhất giữa k mẫu huấn luyện đó. Điều này có nghĩa là nếu k bằng 1, nhãn sẽ được chỉ định dựa trên điểm dữ liệu gần nhất. Tuy nhiên, nếu K bằng 4, thuật toán sẽ chọn bốn điểm dữ liệu gần nhất cho từng trường hợp và phân loại nó dựa trên số nhãn xuất hiện nhiều nhất trong bốn điểm dữ liệu đó. [5]

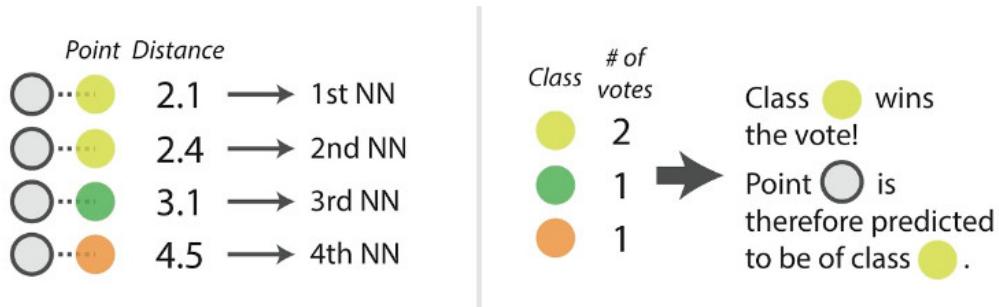
KNN có thể áp dụng được vào cả hai loại bài toán của học có giám sát là Classification (Phân loại) và Regression (hồi quy). KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.



Hình 2.3: KNN tính khoảng cách k điểm gần nhất ($k=4$)

Điểm xám biểu đồ trên chính là dữ liệu mới cần được phân loại, xung quanh là các điểm dữ liệu đã được huấn luyện và gán nhãn (xanh, vàng, cam). K ở đây được ví dụ là 4, thuật toán sẽ tính toán khoảng cách của 4 điểm dữ liệu gần nhất dựa trên công thức Eucli Distance:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.1)$$



Hình 2.4: Liệt kê và chọn nhãn xuất hiện nhiều nhất

Từ các giá trị khoảng cách vừa tính được, KNN sẽ liệt kê 4 dữ liệu gần nhất sau đó chọn ra dữ liệu có số nhãn xuất hiện nhiều nhất và dự đoán dữ liệu mới sẽ thuộc lớp nhãn ấy. Cụ thể ở trường hợp trên là nhãn vàng, vậy điểm xám sẽ thuộc lớp vàng.

Ví dụ về các khái niệm tương ứng giữa người và máy trong mô hình K-nearest neighbor (KNN):

Ngôn ngữ người	Ngôn ngữ Máy Học	in Machine Learning
Câu hỏi	Điểm dữ liệu	Data point
Đáp án	Đầu ra, nhãn	Output, Label
Ôn	Huấn luyện	Training
Tập tài liệu mang vào phòng thi	Tập dữ liệu tập huấn	Training set
Đề thi	Tập dữ liệu kiểm thử	Test set
Câu hỏi trong đề thi	Dữ liệu kiểm thử	Test data point
Câu hỏi có đáp án sai	Nhiều	Noise, Outlier
Câu hỏi gần giống	Điểm dữ liệu gần nhất	Nearest Neighbor

Bảng 2.1: Bảng ví dụ các khái niệm tương ứng giữa người và máy.

Trường hợp trên là khi một thí sinh đang tham gia một cuộc thi. Các câu hỏi và đáp án trong quá trình ôn thi chính là dữ liệu được huấn luyện. Tập tài liệu mang vào phòng thi chính là mô hình lưu trữ các câu hỏi với các đáp án tương ứng. Việc kiểm tra dữ liệu mới tương trưng cho việc cuộc thi bắt đầu, các câu hỏi trong đề thi chính là dữ liệu cần kiểm

thử, đáp án chính là nhãn của dữ liệu. Thí sinh làm bài thi sẽ so sánh câu hỏi trong đề bài và câu hỏi trong tài liệu để tìm ra đáp án đúng nhất cho đề bài. Cách làm này cũng giống như cách hoạt động của mô hình học máy KNN. Các câu hỏi có nội dung gần giống nhất với tài liệu chính là các điểm dữ liệu có khoảng cách gần nhất với dữ liệu mới (đề bài).

2.4 Thuật toán HOG (Histogram of oriented gradient)

Phương pháp mô tả đặc trưng (Histogram of Oriented Gradient – HOG) là một trong những phương pháp phát hiện đối tượng lâu đời nhất. Nó được giới thiệu lần đầu tiên vào năm 1986. HOG được sử dụng như 1 thuật toán trích chọn đặc trưng của đối tượng trong ảnh.

Mục đích của HOG là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh. HOG dựa trên việc chia ảnh đầu vào thành các ảnh con, tính toán histogram của ảnh để tổng hợp và trích rút ra các vector gọi là vector đặc trưng ứng với từng đối tượng.

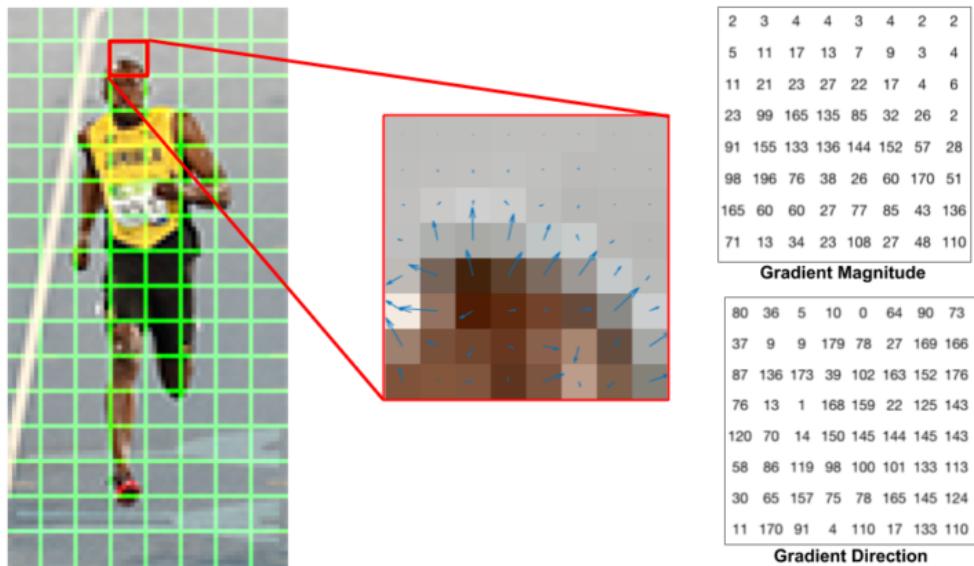
2.4.1 Các bước tạo ra vector HOG

Để tạo được một vector đặc trưng của một ảnh, thuật toán HOG thực hiện lần lượt các bước sau:

- Bước 1: Đặc trưng của mỗi bức ảnh được biểu diễn thông qua 2 thông số đó là mức độ thay đổi cường độ màu sắc (ma trận gradient magnitude) và hướng thay đổi cường độ màu sắc (ma trận gradient direction). Do đó cần tạo ra được một bộ mô tả (feature descriptor) sao cho biến đổi bức ảnh thành một véc tơ mà thể hiện được cả 2 thông tin này.

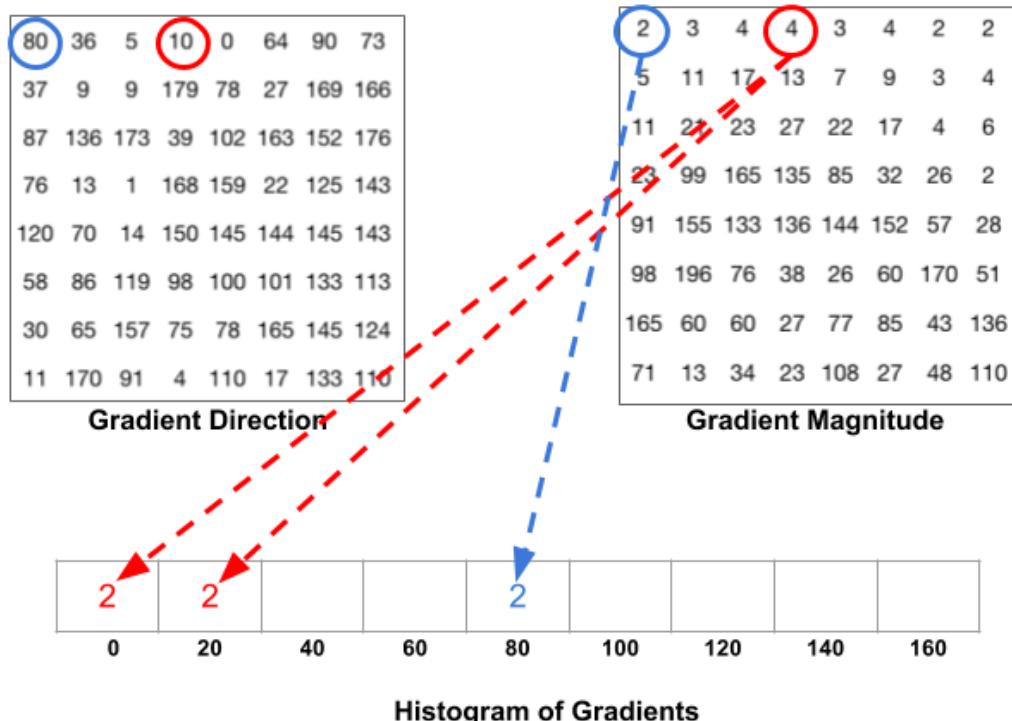
Hình 2.5 là hình ảnh vận động viên được chia thành các lưới ô vuông, mỗi ô vuông có kích thước 8x8 pixels. Trên mỗi ô thực hiện tính đạo hàm thông qua bộ lọc Sobel để thu được 2 ma trận bên phải là gradient magnitude và gradient direction.

- Bước 2: Mapping độ lớn gradient vào các bins tương ứng của phương gradient.



Hình 2.5: Tạo 2 ma trận gradient [1]

Mỗi một phương gradient sẽ ghép cặp với một độ lớn gradient ở cùng vị trí tọa độ. Khi biết được phương gradient thuộc bins nào trong véc tơ bins, ta sẽ điền vào giá trị giá trị của độ lớn gradient vào chính bin đó.



Hình 2.6: Tìm Histogram bins cho mỗi cell

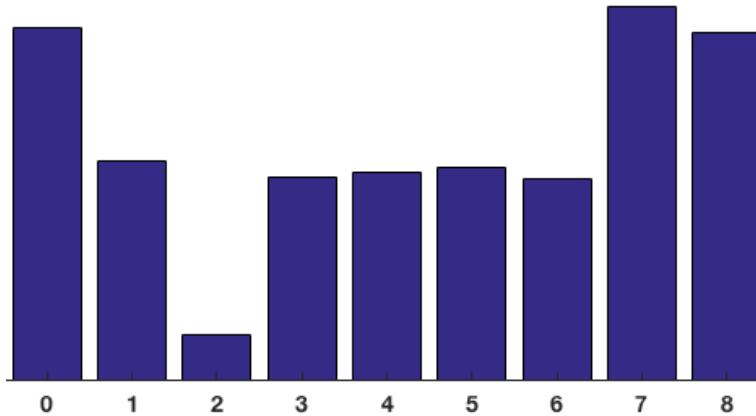
Trong trường hợp độ lớn phương gradients không rơi vào các đầu mút, ta sẽ sử dụng

linear interpolation để phân chia độ lớn gradient về 2 bins liền kề mà giá trị phương gradient rơi vào. Công thức linear interpolation như sau:

$$x_{l-1} = \frac{(x_1 - x)}{x_1 - x_0} * y \quad (2.2)$$

$$x_l = \frac{(x - x_0)}{x_1 - x_0} * y \quad (2.3)$$

Tính tổng tất cả các độ lớn gradient thuộc cùng 1 bins của véc tơ bins ta thu được biểu đồ Histogram of Gradients như bên dưới gồm các bins tương ứng với một ô vuông trong lưới ô vuông:



Hình 2.7: Biểu đồ Histogram of Gradient

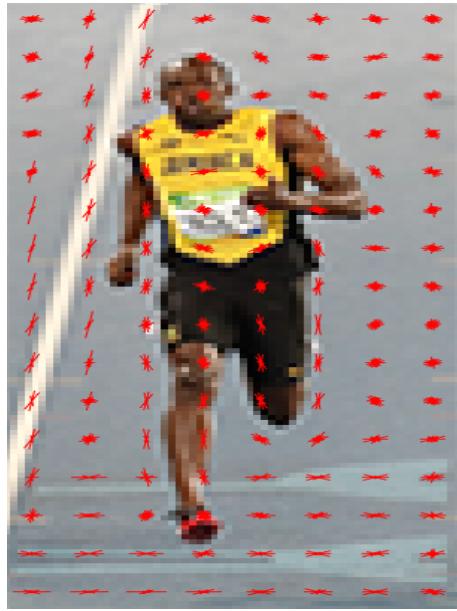
- Bước 3: Chuẩn hóa và tính toán HOG feature vector.

Với 2 bức ảnh có cùng nội dung nhưng bức ảnh biến thể tối hơn được tạo thành từ ma trận ảnh gốc nhân $1/2$. Khi đó giá trị véc tơ histogram của ảnh gốc cũng sẽ gấp đôi vector histogram của ảnh biến thể. Chính vì thế cần chuẩn hóa vector histogram để cả 2 bức ảnh có cùng một véc tơ biểu diễn.

Công thức chuẩn hóa bậc 2:

$$\text{normalize}(\mathbf{h}) = \frac{\mathbf{h}}{\|\mathbf{h}\|_2} \quad (2.4)$$

Sau khi chuẩn hóa các vector histogram, chúng ta nối liền các vector này thành một vector lớn. Đây chính là vecrt HOG đại diện cho toàn bộ hình ảnh.



Hình 2.8: Biểu diễn phân phối HOG trên ảnh

Hình 2.7 biểu diễn nhóm vector histogram trên các lưới ô vuông của hình ảnh gốc. Các phương vector phổ biến là chiều dọc trùng với chiều bức ảnh.

Tương tự như vậy, hình ảnh khuôn mặt sau khi được phát hiện trên khung hình sẽ được thuật toán HOG trích xuất ra những đặc trưng trước khi đưa vào mô hình KNN để training và phân loại.

2.5 Haar Cascade Classifier

2.5.1 Haar Cascade là gì?

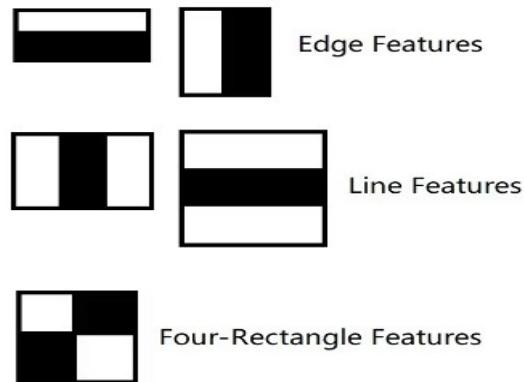
Haar Cascade là thuật toán phát hiện đối tượng được giới thiệu bởi Paul Viola và Michael Jones nhằm phát hiện khuôn mặt trong ảnh hoặc video. Một chức năng phân tầng được đào tạo bằng cách sử dụng nhiều bức ảnh âm và dương, sau này có thể được sử dụng để xác định bất kỳ đối tượng hoặc khuôn mặt nào.

2.5.2 Cách hoạt động của Haar Cascade

Thuật toán này sử dụng phương thức tiếp cận cửa sổ trượt, một cửa sổ có kích thước cố định sẽ lặp lại hình ảnh từ trái sang phải, từ trên xuống dưới. Ở mỗi giai đoạn, cửa sổ

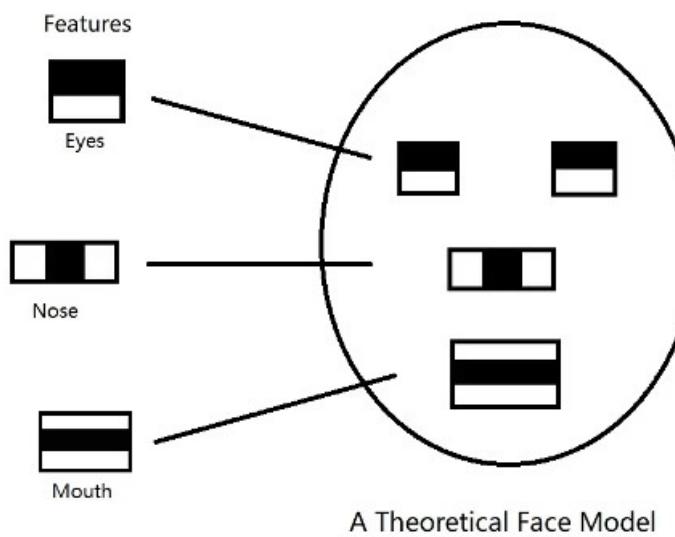
dùng lại và phân loại xem khu vực có chứa khuôn mặt hay không.

Đặc trưng Haar-like của thuật toán được tính bằng tổng cường độ pixel vùng trắng trừ cho tổng cường độ pixel vùng đen. Các hình chữ nhật ở hình 2.8 được trượt ở các vị trí khác nhau với các kích thước khác nhau, thay đổi trên ba kênh màu ở vùng cần lấy đặc trưng và ta sẽ thu được một vector đặc trưng của vùng.



Hình 2.9: Đặc trưng Haar-like

Khi quan sát, vùng mắt có vẻ sẫm màu hơn vùng má trong khi vùng mũi sáng hơn vùng mắt. Có thể hình dung các đối tượng trong hình bên dưới.



Hình 2.10: Các vùng đặc trưng trên khuôn mặt

Bộ phân loại Cascade tăng hiệu suất của việc nhận dạng đối tượng bằng cách kết hợp

nhiều bộ phân loại đơn giản sử dụng thuật toán AdaBoost để giảm số lượng vùng không cần thiết cần kiểm tra trong quá trình phát hiện đối tượng. [6]

AdaBoost (Adaptive Boosting) là một thuật toán học máy dùng để kết hợp nhiều bộ phân loại yếu để tạo thành một bộ phân loại mạnh. Ý tưởng chính của AdaBoost là gán trọng số khác nhau cho các mẫu đào tạo và tạo ra nhiều bộ phân loại yếu. Các bộ phân loại này sau đó được kết hợp lại để tạo ra bộ phân loại mạnh. Thuật toán AdaBoost có thể được mô tả chi tiết thành các bước sau:

- Cho một tập các mẫu $(x_1, y_1), \dots, (x_n, y_n)$; trong đó y_i là nhãn của mẫu x_i . Trường hợp này, x_i là các ảnh và $y_i \in \{0, 1\}$ tương ứng cho ảnh x_i có chứa đối tượng người hay không.
- Khởi tạo các giá trị trọng số để cho $w_{l,i} = \frac{1}{2m}, \frac{1}{2l}$ tương ứng với $y_i = 0, 1$ trong đó m là số ảnh không chứa người hoặc trường hợp âm (negative) và l là số trường hợp dương (positive) hay số ảnh chứa người.
- Lặp lại với $t = 1, \dots, T$:
 - Chuẩn hóa các trọng số để cho w_t là một phân phối xác suất

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=l}^n w_j} \quad (2.5)$$

- Với mỗi đặc trưng j , huấn luyện bộ phân loại h_j chỉ được hạn chế để sử dụng cho một đặc trưng đơn. Sai số được đánh giá cho w_t

$$\varepsilon_t = \sum_i w_i |h_j(x_i) - y_i| \quad (2.6)$$

- Chọn bộ phân loại h_t với các sai số ε_t nhỏ nhất
- Cập nhật lại các trọng số

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\varepsilon_t} \quad (2.7)$$

Trong đó $e_i = 0$ nếu x_i được phân loại chính xác, ngược lại bằng 1 và

$$\beta_t = \frac{\varepsilon_t}{1 - \varepsilon_t} \quad (2.8)$$

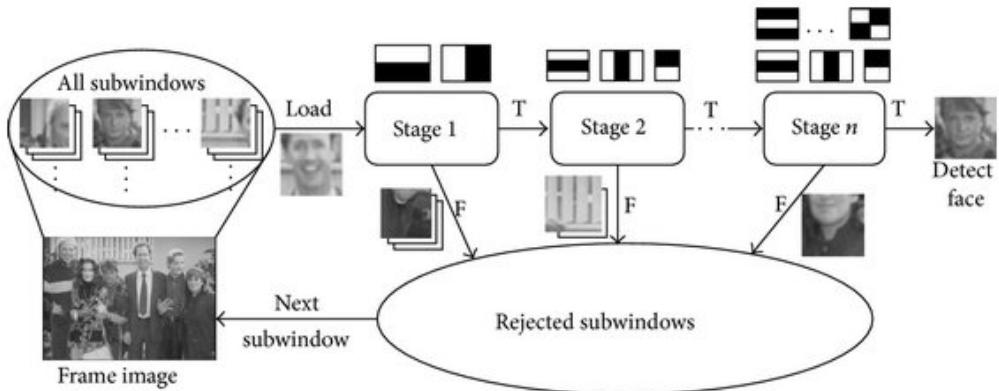
- Bộ phân loại mạnh cuối cùng là

$$h_j(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{else} \end{cases} \quad (2.9)$$

với

$$\alpha_t = \log \frac{1}{\beta_t} \quad (2.10)$$

Bộ phân loại cascade có cấu trúc tương tự như một cây quyết định và mỗi node là một bộ phân loại đơn. Bằng cách sử dụng bộ phân loại có cấu trúc như hình dưới, quá trình phát hiện khuôn mặt sẽ được cải thiện về thời gian chạy do các ảnh chắc chắn không phải khuôn mặt sẽ có các đặc trưng rất khác biệt và có thể được phát hiện sớm từ các bộ phân loại đơn đầu tiên.



Hình 2.11: Sơ đồ bộ phân loại cascade gồm các bộ phân loại đơn

Hình 2.10 thể hiện một cấu trúc của bộ phân loại cascade gồm nhiều tầng. Mỗi tầng sẽ có một số bộ phân loại yếu. Các tầng đầu tiên sẽ kiểm tra các vùng ảnh với ít đặc trưng hơn để nhanh chóng loại bỏ các vùng không phải đối tượng. Các tầng sau sẽ kiểm tra các vùng ảnh còn lại để tăng độ chính xác.

Chương 3

THIẾT KẾ HỆ THỐNG

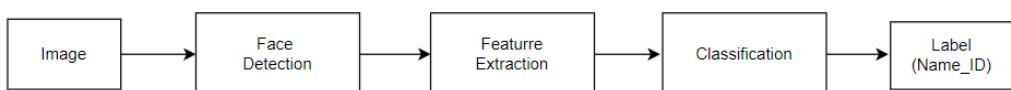
3.1 Giới thiệu

Theo như mục tiêu đã đề ra, nhóm sẽ cố gắng hoàn thiện đề tài “Face detection and identity with machine learning” với chức năng nhận dạng khuôn mặt trong các frame được nhận từ webcam từ đó trích xuất đặc trưng và phân loại từng khuôn mặt. Huấn luyện mô hình học máy để nhận diện và phân loại khuôn mặt với độ chính xác cao. Cuối cùng là xây dựng giao diện hệ thống điểm danh ứng dụng mô hình nhận diện khuôn mặt ấy vào thực tế.

3.2 Sơ đồ khái niệm hệ thống và chi tiết từng khôi

3.2.1 Sơ đồ khái niệm

Hệ thống điểm danh sinh viên gồm có các khôi liên kết với nhau như sau:



Hình 3.1: Sơ đồ khái niệm hệ thống nhận diện khuôn mặt.

Hình 3.1 thể hiện sơ đồ khái niệm hệ thống điểm danh sinh viên với hai quá trình chính là quá trình huấn luyện và nhận dạng.

3.2.2 Chi tiết từng khối

1. Images (Dữ liệu ảnh):

Hệ thống thu thập hình ảnh dữ liệu khuôn mặt bằng cách sử dụng chính webcam của máy tính, hoặc có thể là hình ảnh từ nhiều nguồn khác. Các ảnh được thu thập cần đảm bảo các yếu tố như điều kiện ánh sáng, các góc độ khác nhau của khuôn mặt, tuổi tác,... Và khuôn mặt không nên có các vật cản như kính.

Ngoài ra, để đảm bảo độ chính xác cho hệ thống, đối với mỗi người dùng cần thu thập một số lượng ảnh nhất định không quá ít, và mỗi bức ảnh chỉ chứa duy nhất một khuôn mặt.

2. Face Detection (Phát hiện khuôn mặt):

Sau khi nhận được các hình ảnh có người bên trong. Hệ thống tìm ra vị trí khuôn mặt trong bức hình bằng cách sử dụng bộ phân loại Haar Cascade. Để sử dụng được module này, nhóm tải tệp "haarcascade_frontalface_default.xml", là một tệp chuẩn được cung cấp bởi OpenCV để phát hiện khuôn mặt. Sau đó tạo hàm phát hiện khuôn mặt để sử dụng trong cả hai trường hợp huấn luyện và nhận dạng.



Hình 3.2: Phát hiện khuôn mặt bằng Haar Cascade.

Như hình 3.2 thể hiện. Khuôn mặt của một người được phát hiện ra trong video dù ở trong điều kiện ngược sáng. Vậy nên đề tài này nhóm sử dụng Haar Cascade để trích xuất khuôn mặt.

3. Feature Extraction (Trích chọn đặc trưng):

Trước khi đưa vào hàm trích chọn đặc trưng của khuôn mặt. Hệ thống cần chuyển đổi ảnh màu sang ảnh xám để tăng hiệu suất và giảm độ phức tạp trong việc xử lý. Sau đó cài đặt các thông số như tỷ lệ thu nhỏ và ngưỡng phát hiện được thiết lập là 1.3 và 5 tương ứng. Các thông số này quyết định mức độ nhạy cảm và độ chính xác của việc phát hiện khuôn mặt.

Sau khi ảnh đầu vào được xử lý, hệ thống cần tạo ra những điểm đặc trưng của từng khuôn mặt để huấn luyện và nhận dạng khuôn mặt ấy. Ở đây nhóm đã sử dụng thuật toán HOG (Histogram of Oriented Gradients) để tạo ra các vector đặc trưng cho mỗi khuôn mặt.

```
1 def extract_hog_features(img):
2     winSize = (64, 128)
3     blockSize = (16, 16)
4     blockStride = (8, 8)
5     cellSize = (8, 8)
6     nbins = 9
7     hog = cv2.HOGDescriptor(winSize, blockSize, blockStride,
8                             cellSize, nbins)
9     features = hog.compute(img)
10    return features.flatten()
```

Hàm 3.1: Hàm sử dụng thuật toán HOG để trích xuất đặc trưng

Thuật toán sẽ tạo ra các cửa sổ trượt khác nhau và cho trượt trên hình ảnh đầu vào để tạo ra các vector đặc trưng dựa trên hướng và cường độ của các pixel có trên ảnh. Kết quả là một mảng chứa các giá trị đặc trưng HOG đã được làm phẳng thông qua hàm "feature.flatten(). Mảng này sẽ được sử dụng để huấn luyện mô hình nhận dạng khuôn mặt và thực hiện việc nhận dạng trong quá trình điểm danh.

4. Classification (Phân loại)

Nhóm sử dụng thuật toán học máy KNN (K-Nearest Neighbors) để phân loại và lưu trữ các điểm dữ liệu với các nhãn tương ứng của chúng. Mô hình huấn luyện sẽ ở dạng file có tên là "face_recognition_model.pkl" để sử dụng cho việc nhận dạng.

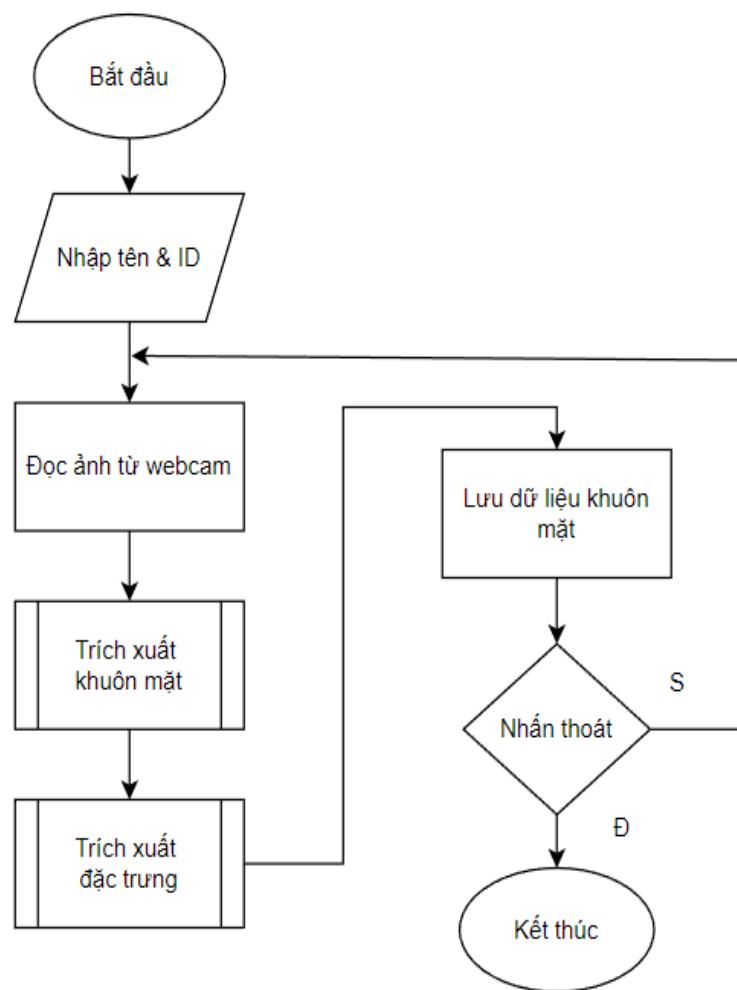
Việc nhận dạng sẽ hoạt động theo nguyên lý của thuật toán KNN. KNN hoạt động

dựa trên nguyên tắc rằng các điểm dữ liệu có tính chất tương tự thường nằm gần nhau trong không gian đặc trưng. KNN sẽ tìm ra K điểm dữ liệu gần nhất với điểm dự đoán.

Sau đó dự đoán nhãn cho điểm dự đoán dựa trên các nhãn của K điểm láng giềng gần nhất và chọn nhãn xuất hiện nhiều nhất trong K điểm đó. Như vậy, các điểm đặc trưng của khuôn mặt khi điểm danh sẽ được so sánh và phân loại theo nhãn sau khi đưa qua mô hình đã được huấn luyện.

Lưu đồ thuật toán

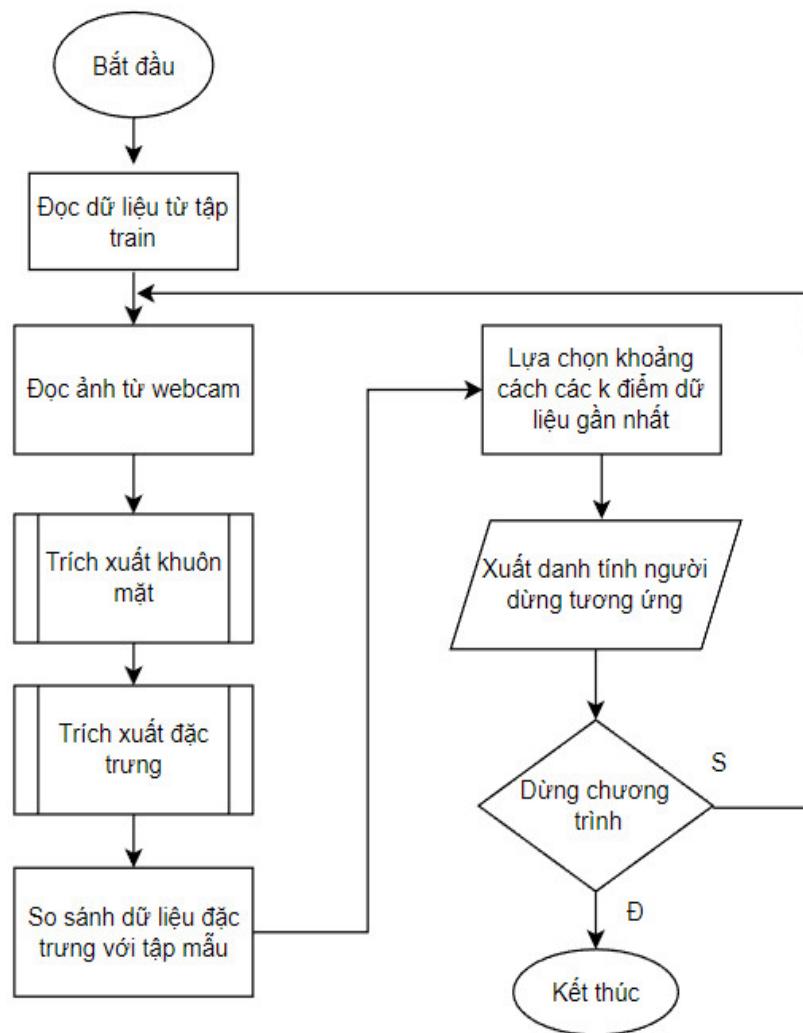
Lưu đồ quá trình huấn luyện dữ liệu



Hình 3.3: Lưu đồ thuật toán huấn luyện.

Hệ thống sẽ bắt đầu bằng việc nhập tên và id định danh cho hình ảnh chứa khuôn mặt cần thêm vào mô hình huấn luyện. Tiếp theo hình ảnh sẽ được đọc liên tục và đưa vào hàm xử lý phát hiện khuôn mặt. Ở đây, hàm sử dụng thuật toán Haar cascade để tìm ra

khuôn mặt trong các ảnh. Ảnh sẽ được tiền xử lý trước khi sử dụng thuật toán HOG để tạo ra các vector đặc trưng cho mỗi khuôn mặt. Dữ liệu được tạo ra sẽ được gán nhãn theo tên và ID khi nhập. Nếu nhấn "ESC" để thoát, chương trình sẽ dừng lại và ngược lại chương trình sẽ tiếp tục đọc ảnh từ camera của máy tính. Quy trình huấn luyện này xảy ra khi người dùng cần "Thêm sinh viên" vào lớp học.



Hình 3.4: Lưu đồ thuật toán nhận dạng.

Khác với quá trình huấn luyện, hệ thống sau khi trích xuất đặc trưng từ những khuôn mặt được lấy từ webcam trong quá trình điểm danh sẽ so sánh các dữ liệu này với tập mẫu đã được huấn luyện trước đó. Ở đây, mô hình học máy KNN sẽ dùng tính chất lấy các điểm dữ liệu có tính chất gần giống với dữ liệu đầu vào nhất để phân loại vào nhãn tương ứng. Sau đó sẽ hiển thị tên và ID của người đang thực hiện điểm danh lên cửa sổ webcam. Chương trình sẽ dừng lại nếu nhấn nút "ESC" trên bàn phím, ngược lại hệ

thống sẽ tiếp tục đọc ảnh từ camera.

3.3 Giao diện hệ thống

Để người dùng có thể tương tác với hệ thống, nhóm đã xây dựng giao diện điểm danh gồm có các chức năng điểm danh và thêm sinh viên. Danh sách sinh viên bao gồm tên, ID, ngày điểm danh cũng sẽ được hiển thị lên giao diện để người dùng có thể nắm bắt được. Để làm được điều này, nhóm đã lựa chọn sử dụng Flask để thiết kế web cho hệ thống với file html.

3.3.1 Flask python là gì?

Flask Python là một framework ứng dụng cho website được tạo ra từ ngôn ngữ lập trình web Python. Công cụ này có dung lượng khá nhẹ nhưng lại rất linh hoạt trong công dụng. Nó được ứng dụng trong thiết kế website đơn giản và tạo lập các ứng dụng cho những trang web lớn và phức tạp.



Hình 3.5: Flask python.

Nền tảng chính của Flask Python là Werkzeug WSGI và Jinja2. Hiện framework này đã trở thành công cụ chính trong phát triển ứng dụng web của Python

3.3.2 Các tính năng cơ bản của Flask

Flask được sử dụng rộng rãi với các tính năng như sau:

- Xử lý Yêu cầu HTTP: Flask cho phép người dùng định nghĩa các tuyến (routes) để xử lý các yêu cầu HTTP như GET, POST, PUT, DELETE. Có thể quy định hành động cụ thể khi người dùng gửi yêu cầu tới các địa chỉ URL cụ thể.
- Quản lý Sesi và Cookie: Flask hỗ trợ quản lý phiên (session) của người dùng và tạo cookie để theo dõi thông tin đăng nhập và trạng thái của người dùng.
- Kết nối Cơ sở dữ liệu: Flask cho phép kết nối với các cơ sở dữ liệu phổ biến như SQLite, MySQL, PostgreSQL. Bạn có thể thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên dữ liệu.
- Giao diện người dùng và Mẫu (Templates): Flask hỗ trợ việc tạo các giao diện người dùng động bằng cách sử dụng các mẫu Jinja2. Điều này giúp tạo ra các trang web với dữ liệu động một cách dễ dàng.
- Triển khai ứng dụng trực tuyến: Flask có thể được triển khai trên các môi trường máy chủ thực sự để ứng dụng của bạn có thể truy cập trực tiếp qua mạng Internet.
- RESTful API: Có thể sử dụng Flask để xây dựng các API RESTful để cung cấp dữ liệu cho các ứng dụng khác, thường là ứng dụng di động hoặc web khác.

3.3.3 Phương thức Flask Python hoạt động trong hệ thống điểm danh bằng nhận diện khuôn mặt.

- Khởi tạo ứng dụng Flask: Đây là bước đầu tiên trong việc tạo một ứng dụng Flask. Đối tượng app sẽ đại diện cho ứng dụng web. [7]

Cụ thể:

```
1 app = Flask(__name__)
```

Hàm 3.2: Hàm khởi tạo Flask

- Routes và Templates: Flask sử dụng cơ chế routes để liên kết các URL với các hàm xử lý tương ứng. Các templates (giao diện người dùng) được sử dụng để hiển thị dữ

liệu trong giao diện web. Ở đây, các hàm 'render_template()' được sử dụng để kết hợp dữ liệu từ Python vào các template HTML để hiển thị trên trình duyệt web.

Cụ thể:

```
1 @app.route('/')
2 def home():
3     names,rolls,times,l = extract_attendance()
4     return render_template('giaodien.html', names=names,rolls=
5         rolls,times=times,l=l,totalreg=totalreg(),datetoday2=
6         datetoday2)
```

Hàm 3.3: Routes và Templates

- Truyền các biến dữ liệu từ máy chủ Python vào trong template HTML để có thể sử dụng trong giao diện người dùng. Ở đây các biến dữ liệu là: names, rolls, times và l: lần lượt là tên, số thứ tự, thời gian và số lượng điểm danh.
- 'totalreg()' là một hàm được gọi để lấy tổng số người dùng đã đăng ký và truyền giá trị này vào biến totalreg.
- 'datetoday2' chứa ngày hiện tại trong định dạng "dd-mm-yyyy" (ngày-tháng-năm).

Sau khi các giá trị được truyền vào, hàm render_template() sẽ tạo ra nội dung HTML dựa trên template "giaodien.html" và thay thế các biến trong template bằng các giá trị thích hợp. Kết quả là một trang web được hiển thị với dữ liệu được lấy từ máy chủ Flask và được trình bày theo cấu trúc và giao diện đã được định nghĩa trong file templates.

- Hàm đảm nhiệm khởi động ứng dụng Flask và chạy trên máy chủ cục bộ với tùy chọn 'debug=True':

Cụ thể:

```
1 if __name__ == '__main__':
2     app.run(debug=True)
```

Hàm 3.4: Hàm khởi động ứng dụng Flask

- Tạo các hàm quản lý giao diện người dùng: Các hàm home(), start(), add() được sử dụng làm các chức năng của các tuyến đường trong ứng dụng Flask.

- Hàm home() cụ thể:

```

1 @app.route('/')
2 def home():
3     names,rolls,times,l = extract_attendance()
4     return render_template('giaodien.html', names=names,rolls
=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=
datetoday2)

```

Hàm 3.5: Hàm home()

Đây là trang chủ của ứng dụng. Khi truy cập trang chủ, hàm home() sẽ được thực thi. Nó gọi hàm extract_attendance() để lấy thông tin điểm danh và render template giaodien.html với dữ liệu điểm danh (tên, mssv, thời gian điểm danh, tổng sinh viên) để hiển thị trên giao diện web.

- Hàm start() cụ thể:

```

1 @app.route('/start',methods=['GET'])
2 def start():
3     if 'face_recognition_model.pkl' not in os.listdir('data'):
4         :
5             return render_template('giaodien.html',totalreg=
totalreg(),datetoday2=datetoday2, mess='There is no
trained model in the data folder. Please add a new face to
continue.')
6
7     ret = True
8     cap = cv2.VideoCapture(0)
9     while ret:
10         ret,frame = cap.read()
11         if len(extract_faces(frame))>0:
12             (x,y,w,h) = extract_faces(frame)[0]
13             cv2.rectangle(frame,(x, y), (x+w, y+h), (255, 0,
20), 2)
14             face = cv2.resize(frame[y:y + h, x:x + w], (64,
128))
15             #face = cv2.resize(frame[y:y+h,x:x+w], (50, 50))
16             features = extract_hog_features(face)

```

```

16         identified_person = identify_face(features.
17
18         reshape(1, -1))[0]
19
20         #identified_person = identify_face(face.reshape
21         (1,-1))[0]
22
23         add_attendance(identified_person)
24
25         cv2.putText(frame,f'{identified_person}',(30,30),
26         cv2.FONT_HERSHEY_SIMPLEX,1,(255, 0, 20),2,cv2.LINE_AA)
27
28         cv2.imshow('Attendance',frame)
29
30         if cv2.waitKey(1)==27:
31
32             break
33
34         cap.release()
35
36         cv2.destroyAllWindows()
37
38         names,rolls,times,l = extract_attendance()
39
40         return render_template('giaodien.html',names=names,rolls=
41         rolls,times=times,l=l,totalreg=totalreg(),datetoday2=
42         datetoday2)

```

Hàm 3.6: Hàm start()

Khi truy cập vào "/start", hàm start() sẽ được thực thi. Trong hàm này, hệ thống khởi tạo webcam (cv2.VideoCapture(0)) và bắt đầu vòng lặp vô hạn để chụp hình từ webcam. Trong mỗi vòng lặp, hình ảnh được chụp, khuôn mặt được nhận diện và xác định bằng cách gọi các hàm liên quan. Hình ảnh được hiển thị trên giao diện và tên và mã số sinh viên của người được nhận diện cũng được hiển thị. Vòng lặp này tiếp tục cho đến khi người dùng bấm phím ESC để thoát.

- Hàm add cụ thể:

```

1 @app.route('/add',methods=['GET','POST'])
2 def add():
3     newusername = request.form['newusername']
4     newuserid = request.form['newuserid']
5     userimagefolder = 'data/faces/' + newusername + '_' + str(
6         newuserid)
7
8     if not os.path.isdir(userimagefolder):
9         os.makedirs(userimagefolder)
10    i,j = 0,0
11    cap = cv2.VideoCapture(0)

```

```

10     while 1:
11         _,frame = cap.read()
12         faces = extract_faces(frame)
13         for (x,y,w,h) in faces:
14             cv2.rectangle(frame,(x, y), (x+w, y+h), (255, 0,
20), 2)
15             cv2.putText(frame,f'Images Captured: {i
} / 50', (30,30), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 20), 2,
cv2.LINE_AA)
16             if j%10==0:
17                 name = newusername+'_'+str(i)+'.jpg'
18                 cv2.imwrite(userimagefolder+'/'+name, frame[y:y+h,x:x+w])
19                 i+=1
20             j+=1
21             if j==500:
22                 break
23             cv2.imshow('Add new user',frame)
24             if cv2.waitKey(1)==27: #esc to exit
25                 break
26             cap.release()
27             cv2.destroyAllWindows()
28             print('Training Model')
29             train_model()
30             names,rolls,times,l = extract_attendance()
31             return render_template('giaodien.html',names=names,rolls=
rolls,times=times,l=l,totalreg=totalreg(),datetoday2=
datetoday2)

```

Hàm 3.7: Hàm add()

Khi truy cập vào "/add", hàm add() được thực thi. Hàm này giúp thêm một sinh viên mới vào danh sách điểm danh. Thông tin sinh viên mới (tên và mssv) được yêu cầu lúc nhập và sau đó hình ảnh khuôn mặt của họ được chụp và lưu vào thư mục tương ứng. Sau đó, mô hình nhận diện khuôn mặt được huấn luyện lại với dữ liệu mới.

Chương 4

KẾT QUẢ

4.1 Kết quả nghiên cứu

Trong quá trình nghiên cứu và thực hiện đồ án môn học, nhóm đã đạt được những kết quả như sau:

- Tìm hiểu tổng quan về Machine learning (ML), các thuật toán nhận dạng và phân loại, ứng dụng thực tế vào mô hình của hệ thống.
- Nhận dạng được khuôn mặt trong các frame được nhận từ webcam.
- Trích xuất đặc trưng và phân loại từng khuôn mặt. Sau đó gán nhãn thành các nhóm tương ứng với từng cá nhân khác nhau.
- Sử dụng dữ liệu đã được gán nhãn và đặc trưng, tiến hành huấn luyện mô hình học máy để nhận diện khuôn mặt sao cho nó có thể phân loại và nhận diện khuôn mặt một cách chính xác.
- Xây dựng giao diện hệ thống điểm danh sinh viên để ứng dụng để tài vào thực tế.

4.2 Hệ thống điểm danh sinh viên

Khi khởi chạy hệ thống, màn hình terminal sẽ hiển thị trạng thái ứng dụng Flask bắt đầu hoạt động. Một địa chỉ IP để truy cập trang web hiển thị giao diện người dùng. Khi nhấn vào đường dẫn này người dùng có thể bắt đầu tương tác với hệ thống điểm danh sinh viên, các tương tác cũng sẽ được thông báo trên màn hình terminal.

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\4rdUniver\face-recognition-based-attendance-system-master> python -u "d:\4rdUniver\face-recognition-based-attendance-system-master\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 129-464-984

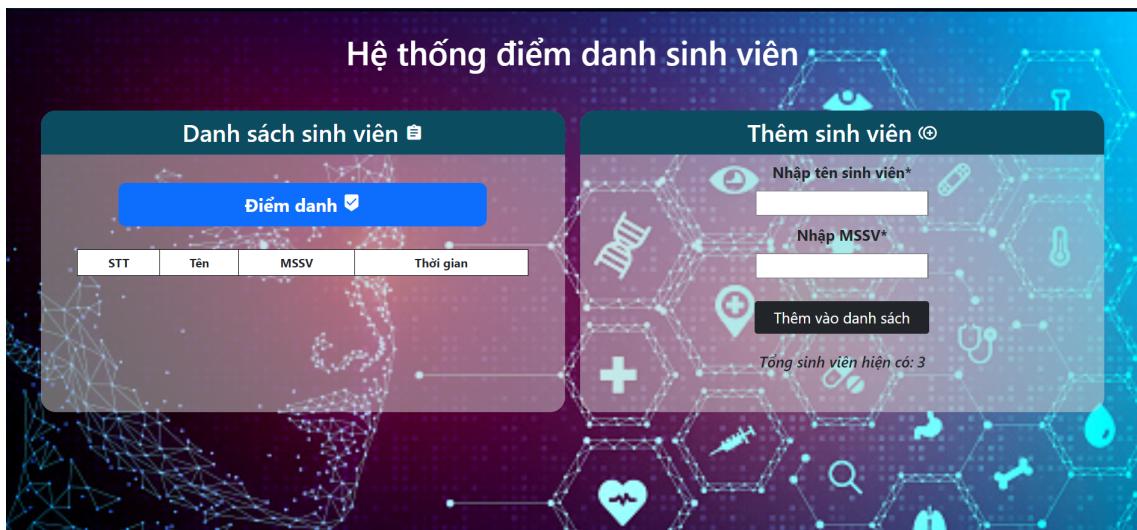
```

Hình 4.1: Kết quả hiển thị trên terminal.

Nếu nhấn tổ hợp phím Ctrl+C, ứng dụng Flask sẽ ngừng lại và hệ thống không thể hoạt động được nữa.

4.2.1 Giao diện người dùng

Hệ thống điểm danh sinh viên có giao diện gồm 2 phần. Khung bên trái chức năng chính là điểm danh sinh viên và hiển thị thông tin sinh viên đã điểm danh. Khung bên phải với chức năng là thêm sinh viên mới vào danh sách lớp học, thông tin sinh viên được thêm bao gồm tên, mã số sinh viên và tổng số sinh viên hiện có trong lớp.



Hình 4.2: Giao diện hệ thống điểm danh.

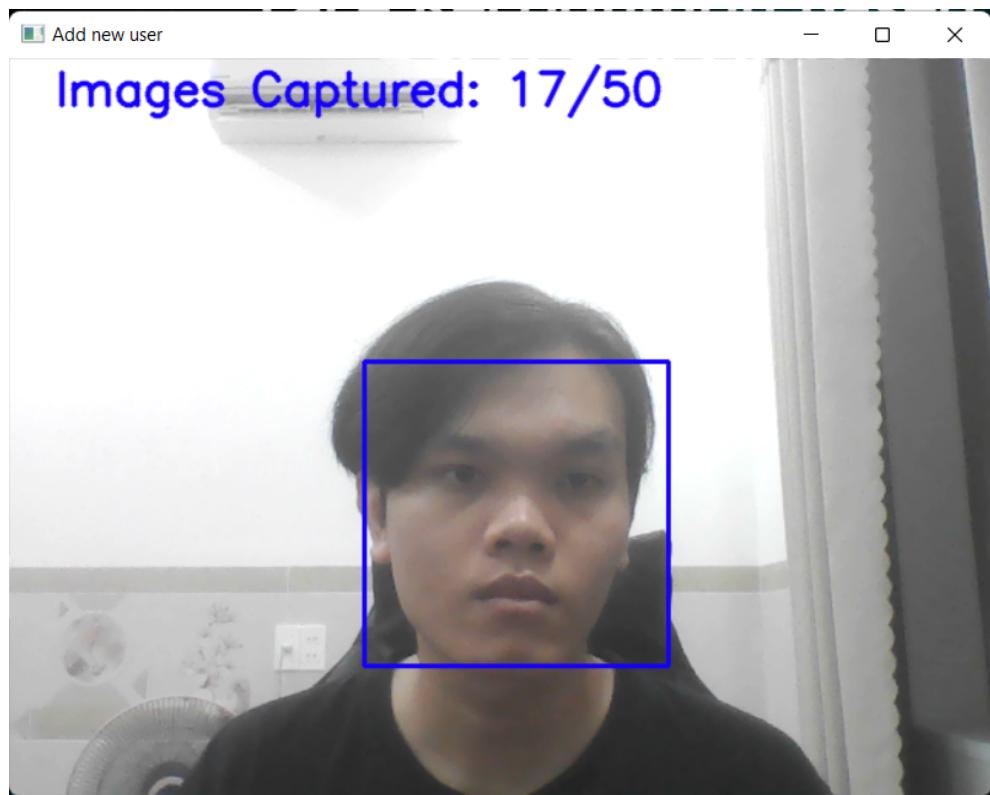
Ngoài ra, nút nhấn "Điểm danh" khi nhấn sẽ mở webcam của máy tính để điểm danh sinh viên có trong danh sách lớp học, và nút nhấn "Thêm vào danh sách" cũng sẽ mở webcam để lấy dữ liệu ảnh khuôn mặt mới để huấn luyện.

4.2.2 Hoạt động của hệ thống

Hệ thống điểm danh sinh viên hoạt động gồm 2 chức năng chính là điểm danh và thêm sinh viên mới:

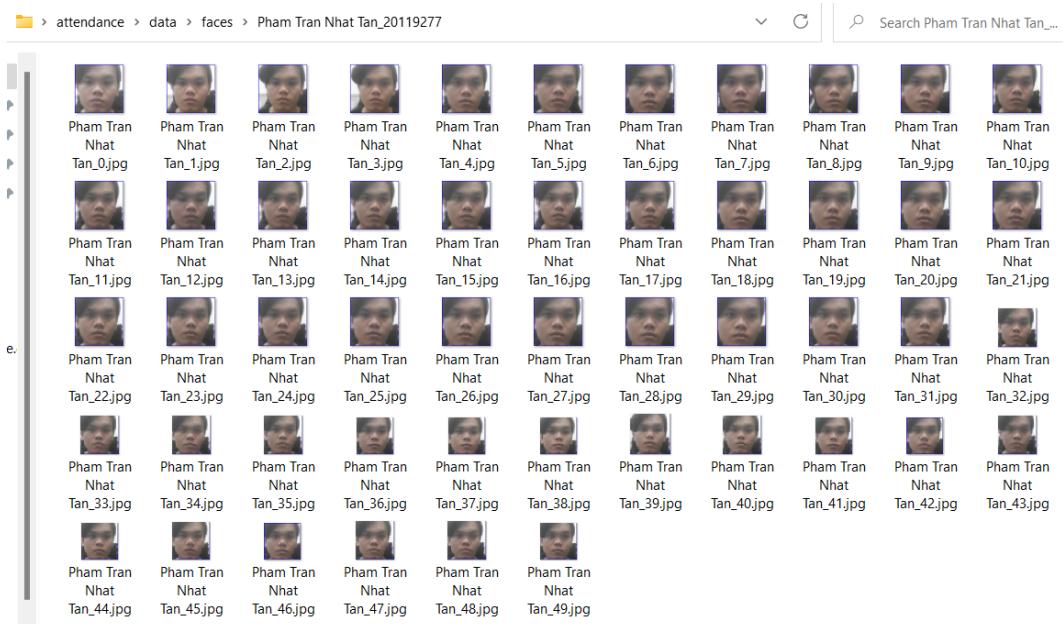
1. Chức năng thêm sinh viên mới:

Để thêm sinh viên mới vào danh sách. Người dùng cần nhập tên sinh viên và nhập mã số sinh viên, sau đó nhấp vào ô "Thêm vào danh sách". Một cửa sổ sẽ hiện ra đồng thời webcam của máy sẽ bật lên và hàm phát hiện khuôn mặt sẽ hoạt động. Hệ thống sẽ lưu liên tục 50 tấm ảnh mà phát hiện được khuôn mặt để làm ảnh huấn luyện. Ảnh huấn luyện sẽ được trích xuất đặc trưng và dán nhãn, sau đó lưu vào thư mục data\faces. Như vậy, thông tin sinh viên mới đã có trong cơ sở dữ liệu hệ thống và có thể tiến hành điểm danh bất cứ lúc nào.



Hình 4.3: Cửa sổ phát hiện và chụp ảnh khuôn mặt làm ảnh huấn luyện.

Tập dữ liệu ảnh khuôn mặt đã chụp và huấn luyện được gắn nhãn của hệ thống:



Hình 4.4: Thư mục data\faces chứa ảnh đã chụp và huấn luyện.

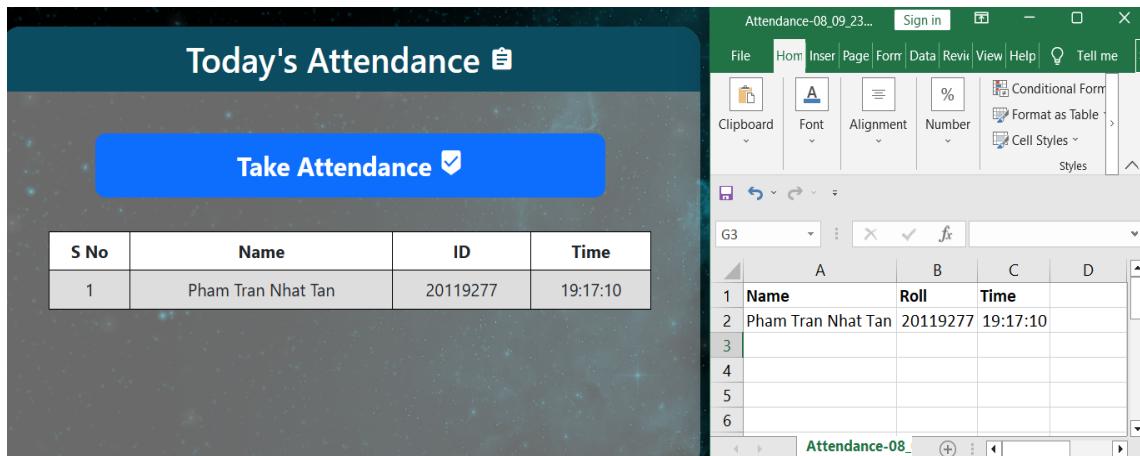
2. Chức năng điểm danh:

Những sinh viên đã có dữ liệu trong cơ sở dữ liệu thì đã có thể tiến hành điểm danh. Để tiến hành điểm danh chọn vào ô "Điểm danh" màu xanh ở khung bên trái của giao diện web. Khi ấy một cửa sổ sẽ hiện ra đồng thời webcam của máy sẽ bật lên và phát hiện khuôn mặt, tên và mã số sinh viên cũng sẽ được hiển thị ở bên trên khung hình.



Hình 4.5: Cửa sổ phát hiện khuôn mặt và điểm danh.

Sau khi nhấn "ESC" để thoát, thông tin của sinh viên bao gồm tên, mssv và thời gian điểm danh sẽ xuất hiện trên khung điểm danh này. Ngoài ra hệ thống còn lưu những thông tin ấy vào file excel theo ngày để người dùng dễ dàng tổng hợp danh sách sinh viên của ngày điểm danh ấy.



Hình 4.6: Thông tin người dùng điểm danh.

4.3 Video demo hệ thống

Để hiểu rõ hơn về cách sử dụng và hoạt động của hệ thống điểm danh sinh viên do nhóm xây dựng, dưới đây là đường dẫn video demo hệ thống do nhóm trình bày.
https://drive.google.com/drive/folders/10Ef1oP0NrLvViPHs6W3n9IMnJlvprBzb?usp=drive_link

Chương 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Trong bối cảnh sự phát triển nhanh chóng của công nghệ thông tin và trí tuệ nhân tạo, đề tài "Face detection and identity with machine learning" đã mang lại những tiềm năng và ứng dụng vô cùng quan trọng trong nhiều lĩnh vực thực tế. Qua quá trình nghiên cứu và thực hiện dự án, chúng ta đã nhận thấy tầm quan trọng và khả năng biến đổi mà công nghệ nhận diện khuôn mặt mang lại.

Hệ thống nhận diện khuôn mặt điểm danh đã chứng minh sự hiệu quả của việc sử dụng các thuật toán học máy để phát hiện và nhận diện khuôn mặt trong thời gian thực. Không chỉ đơn thuần giúp xác định danh tính của cá nhân một cách nhanh chóng và chính xác, mà nó còn mở ra nhiều cơ hội mở rộng ứng dụng đáng kể.

Hệ thống này có thể được ứng dụng trong việc quản lý nhân sự, tiết kiệm thời gian và tối ưu hóa quá trình điểm danh. Ngoài ra, nó còn có thể được sử dụng để cải thiện trải nghiệm của khách hàng thông qua các ứng dụng tương tác cá nhân hóa. Đồng thời, việc mở rộng để nhận biết cảm xúc, tình cảm và thái độ của khuôn mặt mở ra cánh cửa cho việc nghiên cứu về tâm lý con người và ứng dụng trong lĩnh vực chăm sóc sức khỏe tinh thần.

Tuy nhiên, cùng với những lợi ích vượt trội, cần phải chú ý đến các vấn đề liên quan đến bảo mật dữ liệu và quyền riêng tư của người sử dụng. Việc xử lý dữ liệu nhận diện khuôn mặt đòi hỏi sự cân nhắc kỹ lưỡng để đảm bảo tính bảo mật và tuân thủ các quy định về quyền riêng tư.

Tóm lại, đề tài đồ án môn học 2 "Face detection and identity with machine learning" đã thể hiện rõ sự khả thi và tiềm năng to lớn của việc ứng dụng công nghệ nhận diện khuôn mặt trong nhiều lĩnh vực thực tế. Với sự phát triển liên tục của trí tuệ nhân tạo, chúng ta có thể kỳ vọng rằng các ứng dụng của công nghệ này sẽ tiếp tục mở ra những cơ hội mới và đóng góp tích cực cho xã hội và nền kinh tế

5.2 Hướng phát triển

Với mong muốn mở rộng đề tài để hệ thống "Face detection and identity with machine learning" có thể ứng dụng trong thực tế tốt hơn thì nhóm đưa ra thêm một số ý tưởng để phát triển thêm hệ thống như:

- Tích hợp thêm khả năng chống giả mạo việc điểm danh bằng việc đưa hình ảnh.
- Phân loại tình cảm và trạng thái tâm trạng: Điều này có thể hữu ích trong các ứng dụng như phân tích phản hồi của người dùng đối với sản phẩm hoặc dịch vụ, hỗ trợ tâm lý và sức khỏe tinh thần.
- Theo dõi sự tương tác xã hội: Sử dụng công nghệ nhận diện khuôn mặt để theo dõi tương tác xã hội trong thời gian thực. Điều này có thể được ứng dụng để nghiên cứu về hành vi xã hội, phân tích hiệu suất trình diễn trong các cuộc họp hoặc sự kiện, hoặc cải thiện trải nghiệm khách hàng.
- Dịch vụ tương tác với khách hàng: Xây dựng các chatbot hoặc trợ lý ảo có khả năng tương tác bằng cách nhận diện khuôn mặt và dựa trên đó để tạo trải nghiệm cá nhân hóa hơn cho khách hàng trong các lĩnh vực như bán lẻ hoặc dịch vụ khách hàng.
- Ứng dụng trong xe tự hành: Sử dụng công nghệ nhận diện khuôn mặt để phát hiện và theo dõi tình trạng của người lái, như sự tập trung hoặc mệt mỏi, để đảm bảo an toàn trong xe tự hành.
- Phát hiện dấu hiệu mệt mỏi và căng thẳng: Phát triển mô hình học máy để phát hiện

dấu hiệu của mệt mỏi và căng thẳng dựa trên khuôn mặt, từ đó đưa ra gợi ý và khuyến nghị để cải thiện sức khỏe và tình trạng tâm lý của người dùng.

Tài liệu tham khảo

- [1] S. Mallick. Histogram of oriented gradients explained using opencv. Accessed: 2023-07-30. [Online]. Available: <https://learnopencv.com/histogram-of-oriented-gradients/>
- [2] N. N. Shivalila Hangaragi, Tripty Singh, “Face detection and recognition using face mesh and deep neural network,” in *Procedia Computer Science*, 2023, pp. 741–749.
- [3] L. Mỹ. Tiềm năng từ công nghệ nhận diện khuôn mặt make in vietnam đạt chuẩn mỹ. Accessed: 2023-07-27. [Online]. Available: <https://vietnamnet.vn/tiem-nang-tu-cong-nghe-nhan-dien-khuon-mat-make-in-vietnam-dat-chuan-my-i399041.html>
- [4] Tanca. Machine learning là gì? tìm hiểu về cách hoạt động và ứng dụng. Accessed: 2023-07-25. [Online]. Available: <https://tanca.io/blog/machine-learning-la-gi-tim-hieu-ve-cach-hoat-dong-va-ung-dung>
- [5] S. Kellerhals. Intro to machine learning in r (k nearest neighbours algorithm). Accessed: 2023-07-30. [Online]. Available: <https://ourcodingclub.github.io/tutorials/machine-learning/>
- [6] N. T. Thanh. Phát hiện khuôn mặt với thuật toán adaboost. Accessed: 2023-07-29. [Online]. Available: https://www.academia.edu/14013728/Bao_cao_dò_É_an_I
- [7] Envatotuts. Creating a web app from scratch using python flask and mysql. Accessed: 2023-07-27. [Online]. Available: <https://code.tutsplus.com/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972t>