**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----\*\*\*-----**



**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN “Thị giác máy tính”**

***Đề tài:***

***Xây dựng chương trình điểm danh học viên dựa trên nhận diện khuôn mặt***

GVHD: GV. Hồ Thị Hương Thơm

Sinh viên thực hiện: Vũ Trung Kiên – 86313

Phạm Bá Huy – 87726

***Hải Phòng, ngày 23 tháng 11 năm 2023***

**MỤC LỤC**

[**CHƯƠNG 1** 3](#_Toc152156012)

[**GIỚI THIỆU VỀ CHƯƠNG TRÌNH ĐIỂM DANH HỌC VIÊN DỰA TRÊN NHẬN DIỆN KHUÔN MẶT** 3](#_Toc152156013)

[**1.** **Đặt vấn đề** 3](#_Toc152156014)

[**2.** **Mục tiêu và nhiệm vụ** 5](#_Toc152156015)

[**3.** **Cách thức triển khai** 5](#_Toc152156016)

[**4.** **Phương pháp sử dụng, phương tiện và công cụ sử dụng** 6](#_Toc152156017)

[**CHƯƠNG 2** 9](#_Toc152156018)

[**TRÌNH BÀY VÀ PHÂN TÍCH THUẬT TOÁN SỬ DỤNG** 9](#_Toc152156019)

[**1.** **Tổng quan** 9](#_Toc152156020)

[**2.** **Giới thiệu bài toán nhận dạng mặt người** 10](#_Toc152156021)

[**3.** **Cách hoạt động của thuật toán LBPH** 10](#_Toc152156022)

[**4.** **Ưu, nhược điểm của thuật toán LBPH** 14](#_Toc152156023)

[*4.1.* *Một số ưu điểm của thuật toán LBPH* 15](#_Toc152156024)

[*4.2.* *Một số nhược điểm của thuật toán LBPH* 15](#_Toc152156025)

[**5.** **Thuật toán Haar Cascade** 16](#_Toc152156026)

[*5.1.* *Giới thiệu thuật toán Haar Cascade* 16](#_Toc152156027)

[*5.2.* *Phân tích thuâth toán Haar Cascade* 17](#_Toc152156028)

[*5.3.* *Ứng dụng Haar cascade* 19](#_Toc152156029)

[**CHƯƠNG 3** 21](#_Toc152156030)

[**XÂY DỰNG CHƯƠNG TRÌNH** 21](#_Toc152156031)

[**1.** **Công cụ lập trình** 21](#_Toc152156032)

[***2.*** **Các bước thực hiện** 21](#_Toc152156033)

[2.2. *Chức năng lấy ảnh học viên* 22](#_Toc152156034)

[*2.3.* *Chức năng huấn luyện mô hình* 22](#_Toc152156035)

[*2.4.* *Chức năng nhận diện điểm danh học viên* 23](#_Toc152156036)

[**3.** **Kết quả chạy chương trình** 23](#_Toc152156037)

[**KẾT LUẬN** 25](#_Toc152156038)

**MỞ ĐẦU**

Trong thời đại công nghệ 4.0, việc áp dụng hệ thống nhận diện khuôn mặt đã trở thành một giải pháp quan trọng trong nhiều lĩnh vực cuộc sống. Hệ thống này không chỉ giúp chúng ta xác định và phân biệt cá nhân một cách nhanh chóng mà còn mở ra những tiềm năng vô tận trong việc cải thiện đáng kể hiệu suất làm việc cũng như an ninh cho các tổ chức và cộng đồng.

Sự tiến bộ trong lĩnh vực nhận diện khuôn mặt không chỉ hỗ trợ việc quản lý danh tính mà còn tạo ra tiềm năng rộng lớn cho việc tăng cường tính bảo mật và tự động hóa các quy trình hàng ngày.

Việc phát triển của công nghệ và trí tuệ nhân tạo đã mở ra nhiều cơ hội trong việc nâng cao hiệu suất và độ chính xác của các hệ thống nhận diện khuôn mặt, từ đó áp dụng vào các ứng dụng thực tế khác nhau. Một trong những ứng dụng quan trọng là việc sử dụng nhận diện khuôn mặt trong chương trình điểm danh, giúp tự động ghi nhận sự có mặt của các cá nhân một cách chính xác và thuận tiện.

Bằng việc sử dụng công nghệ nhận diện khuôn mặt tiên tiến, chương trình nhận diện khuôn mặt điểm danh học viên không chỉ giúp xác định danh tính một cách chính xác mà còn tạo điều kiện thuận lợi cho việc quản lý lớp học và kiểm soát quá trình học tập. Sự linh hoạt và tính tiện lợi của việc này không chỉ giúp giảng viên tiết kiệm thời gian mà còn tăng cường tính chính xác và đáng tin cậy trong quá trình quản lý học viên.

Với báo cáo này, em hy vọng có thể giới thiệu về một trong những ứng dụng thực tế của trí tuệ nhân tạo và đóng góp vào việc nâng cao nhận thức về tính ứng dụng và tiềm năng của công nghệ trong cuộc sống hàng ngày. Ngoài ra, em cũng mong muốn có thể chia sẻ kiến thức về thuật toán LBPH (Local Binary Pattern Histograms) và sử dụng thư viện OpenCV để xây dựng mô hình nhận diện khuôn mặt điểm danh học viên, giúp mọi người có thể tiếp cận với công nghệ này và sử dụng nó trong các ứng dụng của riêng mình.

# **CHƯƠNG 1**

# **GIỚI THIỆU VỀ CHƯƠNG TRÌNH ĐIỂM DANH HỌC VIÊN DỰA TRÊN NHẬN DIỆN KHUÔN MẶT**

1. **Đặt vấn đề**

Công nghệ điểm danh nhận diện khuôn mặt đang ngày càng trở nên phổ biến và đóng vai trò quan trọng trong việc nâng cao sự thuận tiện và bảo mật trong nhiều lĩnh vực khác nhau và đã được triển khai rộng rãi trong nhiều lĩnh vực khác nhau và mang lại nhiều ứng dụng hữu ích. Dưới đây là một số ứng dụng phổ biến của công nghệ này:

- Hệ thống điểm danh nhận diện khuôn mặt thay thế các phương tiện truyền thống như thẻ chấm công, giúp quản lý nhân sự hiệu quả hơn và giảm thiểu việc giả mạo thông tin.

- Cải thiện an ninh tại cổng vào hoặc trong các khu vực như công ty, trường học, sân bay, và bất kỳ địa điểm nào cần theo dõi người đi lại.

- Thay thế mật khẩu hoặc thẻ từ trong hệ thống kiểm soát truy cập để tăng cường bảo mật và giảm rủi ro mất mát thông tin.

- Sử dụng trong các sự kiện hoặc địa điểm công cộng để quản lý và theo dõi đối tượng tham gia.

- Điểm danh nhận diện khuôn mặt có thể được tích hợp vào hệ thống quản lý đội xe, giúp xác định lái xe và cải thiện an toàn vận tải.

- Trong ngành bán lẻ, các cửa hàng có thể sử dụng để nhận diện khách hàng thân thiện và tối ưu hóa trải nghiệm mua sắm.

- Hỗ trợ quản lý sự kiện bằng cách tự động điểm danh và theo dõi sự tham gia của người tham gia.

- Sử dụng trong hệ thống quản lý học viên để theo dõi sự tham gia, đánh giá hiệu suất, và bảo đảm an toàn trong trường học hoặc trung tâm đào tạo.

- Cải thiện quá trình quản lý bệnh nhân và nhận diện y bác sĩ hoặc nhân viên y tế.

- Tăng cường bảo mật trong các giao dịch tài chính và quản lý thông tin người dùng.

- Giúp ngăn chặn việc sử dụng thông tin giả mạo hoặc thẻ từ của người khác.

- Tích hợp để cải thiện quá trình check-in và check-out, cũng như quản lý dịch vụ cho khách hàng.

Trong lĩnh vực giáo dục và quản lý học viên, việc điểm danh là một hoạt động quan trọng nhằm kiểm soát sự tham gia và thực hiện các khóa học. Tuy nhiên, các phương pháp điểm danh truyền thống thường tốn thời gian và có thể không hiệu quả với các lớp học lớn hoặc khi cần sự tự động hóa.

Trong bối cảnh phát triển mạnh mẽ của công nghệ nhận diện khuôn mặt, việc áp dụng các hệ thống nhận diện này trong việc điểm danh học viên đã mở ra một khả năng mới, tối ưu hóa và nâng cao tính chính xác của quá trình điểm danh. Sự tiện ích và tính tự động của việc nhận diện khuôn mặt đã thu hút sự quan tâm của giới giáo dục và ngành công nghiệp công nghệ thông tin.

Chương trình Điểm Danh Học Viên Dựa Trên Nhận diện Khuôn Mặt nhằm giải quyết các vấn đề liên quan đến phương pháp điểm danh truyền thống. Bằng cách sử dụng công nghệ nhận diện khuôn mặt, chương trình này hứa hẹn mang lại sự chính xác, hiệu quả và tính tự động hóa cao trong việc ghi nhận sự có mặt của học viên trong các buổi học, giúp tối ưu hóa quản lý lớp học và nâng cao trải nghiệm học tập.

Mục tiêu cũng như lý do ra đời của chương trình Điểm Danh Học Viên Dựa Trên Nhận diện Khuôn Mặt, là để đáp ứng nhu cầu cải thiện phương pháp điểm danh truyền thống thông qua việc áp dụng công nghệ nhận diện khuôn mặt. Điều này sẽ giúp tiết kiệm thời gian, tăng tính chính xác và hiệu suất trong việc quản lý học viên, đồng thời cung cấp một giải pháp hiện đại và linh hoạt cho việc điểm danh trong môi trường học tập và giảng dạy.

1. **Mục tiêu và nhiệm vụ**

Mục tiêu của đề tài xây dựng chương trình điểm danh nhận diện khuôn mặt bao gồm:

* Nghiên cứu và phân tích kỹ thuật nhận dạng bằng Haar Cascade và kỹ thuật nhận diện khuôn mặt bằng thuật toán LBPH (Local Binary Pattern Histograms), bao gồm việc hiểu rõ cơ chế hoạt động của thuật toán và các phương pháp tiền xử lý dữ liệu cần thiết để đưa dữ liệu vào chương trình nhận diện khuôn mặt.
* Xây dựng tập dữ liệu nhận diện khuôn mặt đủ lớn, đa dạng và chất lượng cao để huấn luyện và kiểm tra mô hình điểm danh nhận diện khuôn mặt bằng thuật toán LBPH.
* Thiết kế và huấn luyện mô hình điểm danh nhận diện khuôn mặt sử thuật toán LBPH, với mục tiêu đạt được độ chính xác cao nhất có thể.
* Kiểm tra và đánh giá hiệu suất của mô hình điểm danh nhận diện khuôn mặt sử dụng thuật toán LBPH trên tập dữ liệu kiểm tra độc lập và so sánh kết quả với các phương pháp nhận diện khuôn mặt khác.

Các nhiệm vụ cụ thể của đề tài bao gồm:

* + Nghiên cứu, tìm hiểu về các phương pháp nhận diện khuôn mặt.
  + Xây dựng cơ sở dữ liệu cho chương trình nhận diện.
  + Thiết kế và xây dựng mô hình LBPH cho việc nhận diện khuôn mặt.
  + Tiến hành huấn luyện và kiểm thử mô hình trên dữ liệu thực tế.
  + Đánh giá hiệu suất của mô hình nhận diện được xây dựng.
  + Triển khai chương trình nhận diện trên nền tảng ứng dụng.

1. **Cách thức triển khai**

Để triển khai nghiên cứu xây dựng chương trình điểm danh dựa trên nhận diện khuôn mặt, có thể cần các thứ sau:

- Các tập dữ liệu về khuôn mặt: Để huấn luyện và kiểm tra mô hình, cần phải sử dụng các tập dữ liệu chứa ảnh của các khuôn mặt.

- Các công cụ để tiền xử lý dữ liệu ảnh: Các công cụ này được sử dụng để tiền xử lý dữ liệu ảnh trước khi đưa vào huấn luyện mô hình, như tách ảnh ra khỏi các phông nền, thay đổi kích thước ảnh, chuyển đổi định dạng ảnh,...

- Các công cụ để đánh giá mô hình: Để đánh giá độ chính xác của mô hình, có thể sử dụng các công cụ để tính toán độ chính xác của mô hình.

- Phần cứng máy tính có cấu hình tương đối cao: Cần sử dụng máy tính có cấu hình tương đối cao để đảm bảo quá trình huấn luyện và kiểm tra mô hình diễn ra nhanh chóng và hiệu quả.

1. **Phương pháp sử dụng, phương tiện và công cụ sử dụng**

Haar Cascade là một thuật toán học máy phổ biến được sử dụng để phát hiện đối tượng. Thuật toán này ban đầu được sử dụng để xác định các bộ phận cơ thể, sau đó được sử dụng để xác định bất kỳ loại đối tượng nào. Haar Cascade xác định các đối tượng dựa trên các đặc trưng được cung cấp. Haar Cascade là một phần của OpenCV, một thư viện mã nguồn mở phổ biến cho xử lý hình ảnh và học máy. Trong OpenCV, Haar Cascade được sử dụng để phát hiện các đối tượng được chỉ định trong hình ảnh và video. Haar Cascade được lưu dưới dạng các tệp XML, có thể được tải xuống từ kho lưu trữ được đào tạo trước của OpenCV2. Haar Cascade được đề xuất lần đầu tiên vào năm 2001 bởi Paul Viola và Michael Jones trong bài báo của họ, "Rapid Object Detection using a Boosted Cascade of Simple Features". Trong bài báo này, họ đã giới thiệu một phương pháp phát hiện đối tượng hiệu quả dựa trên các đặc trưng Haar. Các đặc trưng Haar được sử dụng để trích xuất các đặc trưng từ hình ảnh, giúp phân biệt giữa các đối tượng và nền. Tóm lại, Haar Cascade là một công cụ mạnh mẽ trong lĩnh vực học máy và xử lý hình ảnh, giúp chúng ta phát hiện và xác định các đối tượng trong hình ảnh và video.

LBPH (Local Binary Pattern Histograms) là một thuật toán thị giác máy tính nổi tiếng và hiệu quả, thường được ứng dụng chủ yếu trong lĩnh vực nhận diện khuôn mặt và đặc trưng hình ảnh. Thiết kế của nó dựa trên việc sử dụng chuỗi nhị phân địa phương (LBP), một kỹ thuật mạnh mẽ cho việc mô tả đặc trưng cục bộ của hình ảnh. Với ưu điểm là hiệu quả tính toán và khả năng đối phó với biến đổi ánh sáng, LBPH thường được áp dụng rộng rãi trong nhận diện khuôn mặt dưới nhiều điều kiện ánh sáng khác nhau. Mặc dù nó có nhược điểm là nhạy cảm với biến đổi hình dạng và hiệu suất kém ở kích thước nhỏ, nhưng với khả năng mô tả tốt đặc trưng cục bộ, LBPH vẫn là một công cụ quan trọng và đáng tin cậy trong lĩnh vực xử lý ảnh và nhận diện đối tượng.

Flask là một framework web được xây dựng trên ngôn ngữ lập trình Python, được thiết kế để xây dựng ứng dụng web nhỏ đến trung bình và các dự án web nhanh chóng. Được tạo ra bởi Armin Ronacher, Flask nổi tiếng với sự đơn giản, linh hoạt và khả năng mở rộng, giúp các nhà phát triển tập trung vào việc xây dựng ứng dụng mà không phải lo lắng về quá trình phức tạp. Một trong những đặc điểm quan trọng của Flask là cấu trúc nhỏ và không giữ chặt vào một số quy tắc nghiêm ngặt, cho phép nhà phát triển lựa chọn các thành phần cần thiết để tích hợp vào ứng dụng của họ. Flask cung cấp một bộ công cụ cơ bản cho việc xử lý định tuyến (routing), kết nối với cơ sở dữ liệu, và quản lý phiên làm việc, đồng thời hỗ trợ mở rộng thông qua việc tích hợp các extension hoặc middleware.

Pandas là một thư viện mã nguồn mở quan trọng trong ngôn ngữ lập trình Python, được sử dụng rộng rãi trong lĩnh vực phân tích dữ liệu và xử lý dữ liệu có cấu trúc. Được phát triển bởi Wes McKinney, Pandas cung cấp các cấu trúc dữ liệu mạnh mẽ và công cụ xử lý dữ liệu linh hoạt, giúp người phân tích dữ liệu và nhà khoa học dữ liệu thực hiện các thao tác phức tạp một cách dễ dàng. Một trong những cấu trúc dữ liệu quan trọng của Pandas là DataFrame, một bảng dữ liệu hai chiều với các hàng và cột có thể được gán nhãn. DataFrame cho phép người dùng thực hiện các phép toán mạnh mẽ như lọc dữ liệu, thống kê, và kết hợp dữ liệu từ nhiều nguồn khác nhau. Pandas cũng hỗ trợ Series, một cấu trúc dữ liệu một chiều tương tự như mảng một chiều, giúp thực hiện các phép toán trên dữ liệu 1D một cách hiệu quả. Thư viện này không chỉ giúp xử lý dữ liệu có cấu trúc mà còn hỗ trợ đọc và ghi dữ liệu từ và đến nhiều định dạng khác nhau như CSV, Excel, SQL, và nhiều định dạng dữ liệu khác.

OpenCV (Open Source Computer Vision) là một thư viện mã nguồn mở được sử dụng rộng rãi trong lĩnh vực thị giác máy tính và xử lý ảnh. OpenCV cung cấp các công cụ và thuật toán để xử lý và phân tích ảnh, bao gồm các chức năng như lọc ảnh, trích xuất đặc trưng, phát hiện đối tượng, theo dõi vật thể và nhận diện khuôn mặt. Nó được viết bằng C ++ và có thể được sử dụng trên nhiều hệ điều hành, bao gồm Windows, Linux và MacOS. OpenCV được viết bằng C ++, nhưng cũng có thể được sử dụng trong các ngôn ngữ khác như Python, Java và MATLAB. Nó cung cấp các hàm và công cụ để thực hiện các tác vụ như xử lý ảnh, phân tích ảnh, và nhận diện đối tượng. Với OpenCV, người dùng có thể xử lý ảnh từ các nguồn khác nhau như camera, video, tệp ảnh và ảnh từ mạng. OpenCV cũng có thể được sử dụng để phát hiện các đối tượng trong video và ảnh, như khuôn mặt, mắt, xe, người, động vật, và nhiều hơn nữa. Ngoài ra, OpenCV còn có thể được sử dụng để xử lý ảnh y tế, phát hiện giao thông, xử lý hình ảnh trong robot và nhiều ứng dụng khác. OpenCV được cập nhật liên tục và được sử dụng rộng rãi trong các lĩnh vực như nhận diện khuôn mặt, xử lý hình ảnh trong ô tô tự lái, và xử lý ảnh trong robot học.

Môi trường phát triển tích hợp (IDE) Visual Studio Code cùng với các tiện ích mở rộng khác nhau. Visual Studio Code được tích hợp với nhiều tính năng hữu ích, bao gồm cú pháp kiểm tra, đánh dấu lỗi, đánh giá mã và gợi ý hoàn thành, giúp cho việc viết mã trở nên nhanh chóng và dễ dàng hơn. Nó cũng có tính năng gỡ lỗi tích hợp và tích hợp Git để quản lý phiên bản mã nguồn.

Một trong những ưu điểm của Visual Studio Code là có khả năng mở rộng thông qua các tiện ích mở rộng và chủ đề. Các tiện ích mở rộng cung cấp các tính năng bổ sung như lập trình cho các ngôn ngữ khác nhau, quản lý dự án và triển khai, trong khi các chủ đề cung cấp các cài đặt giao diện người dùng tùy chỉnh để làm việc với Visual Studio Code trở nên dễ dàng hơn. Visual Studio Code hỗ trợ nhiều ngôn ngữ lập trình như C++, C#, Java, Python, JavaScript, HTML, CSS, v.v. Nó có sẵn trên nhiều nền tảng, bao gồm Windows, macOS và Linux. Với các tính năng đa nền tảng và đa ngôn ngữ này, Visual Studio Code đã trở thành một trong những trình biên tập mã phổ biến nhất trong cộng đồng lập trình.

# **CHƯƠNG 2**

# **TRÌNH BÀY VÀ PHÂN TÍCH THUẬT TOÁN SỬ DỤNG**

1. **Tổng quan**

LBPH (Local Binary Pattern Histograms) là một trong những thuật toán quan trọng trong lĩnh vực thị giác máy tính và xử lý ảnh. Được thiết kế để nhận diện đặc trưng hình ảnh, LBPH được sử dụng rộng rãi trong nhiều ứng dụng khác nhau. Mục tiêu chính của nó là đơn giản hóa quá trình nhận diện khuôn mặt và đối tượng trong ảnh, giúp giảm bớt độ phức tạp tính toán.

LBPH nổi bật với sự linh hoạt và tính đơn giản trong thiết kế, cho phép nhà phát triển tập trung vào việc triển khai ứng dụng mà không cần lo lắng về chi tiết kỹ thuật phức tạp. Điều này làm cho LBPH trở thành một công cụ hữu ích đối với các dự án có phạm vi từ nhỏ đến trung bình.

LBPH không chỉ đơn giản là một công cụ nhận diện đặc trưng mà còn là một phương tiện mạnh mẽ trong xây dựng các hệ thống nhận diện khuôn mặt và đối tượng. Với khả năng tích hợp linh hoạt và hiệu suất đáng kể, thuật toán này thường được sử dụng trong nhiều ứng dụng, bao gồm an ninh, giám sát, và thậm chí trong các ứng dụng di động.

Sự hiểu biết sâu sắc về cách LBPH xử lý ảnh dưới nhiều điều kiện khác nhau, từ ánh sáng đến biến đổi hình dạng, làm cho nó trở thành một công cụ đáng tin cậy trong việc đảm bảo độ chính xác và ổn định của các hệ thống nhận diện. Đặc biệt, khả năng hoạt động tốt trong các tình huống ánh sáng biến đổi giúp LBPH giải quyết một số thách thức phổ biến khi làm việc với dữ liệu hình ảnh thực tế.

LBPH không chỉ là một công cụ nhận diện hình ảnh mà còn là một giải pháp linh hoạt và mạnh mẽ cho nhiều ứng dụng trong lĩnh vực xử lý ảnh và thị giác máy tính. Sự đơn giản và khả năng tích hợp của nó làm cho LBPH trở thành một lựa chọn hấp dẫn cho các dự án có nhu cầu nhận diện đặc trưng một cách hiệu quả và ổn định.

1. **Giới thiệu bài toán nhận dạng mặt người**

Có ba bước đơn giản để máy tính mã hóa nhận dạng khuôn mặt, tương tự như các bước mà bộ não của chúng ta sử dụng để nhận dạng khuôn mặt. Các bước như sau:

* Thu thập dữ liệu: Thu thập dữ liệu khuôn mặt (trong trường hợp này là ảnh khuôn mặt) của những người muốn nhận dạng.
* Đào tạo bộ nhận dạng: Cung cấp dữ liệu khuôn mặt đó và tên tương ứng của từng khuôn mặt cho bộ nhận dạng để nó có thể học.
* Nhận dạng: Cung cấp các khuôn mặt mới của những người đó và xem liệu trình nhận dạng khuôn mặt vừa đào tạo có nhận ra họ hay không.

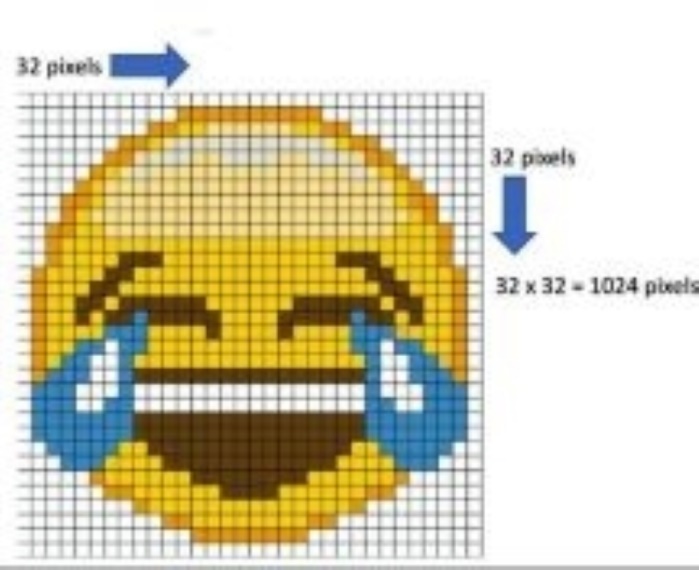
OpenCV có ba trình nhận dạng khuôn mặt tích hợp sẵn. Dưới đây là tên của các trình nhận dạng khuôn mặt đó và các lệnh gọi OpenCV của chúng:

* EigenFaces - cv2.face.createEigenFaceRecognizer ()
* FisherFaces - cv2.face.createFisherFaceRecognizer ()
* Local Binary Patterns Histograms (LBPH) - cv2.face.createLBPHFaceRecognizer ()

Trong báo cáo này chúng ta sẽ sử dụng phương pháp LBPH

1. **Cách hoạt động của thuật toán LBPH**

Thuật toán LBPH (Local Binary Pattern Histograms) là một thuật toán nhận dạng khuôn mặt dựa trên toán tử nhị phân cục bộ, được thiết kế để nhận dạng cả mặt trước và mặt bên của con người. Dưới đây là cách hoạt động của thuật toán LBPH:

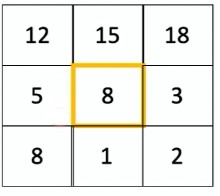
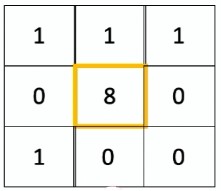
* *Hiểu về hình ảnh và pixel*: Tất cả các hình ảnh đều được biểu diễn dưới dạng ma trận, bao gồm các hàng và cột. Thành phần cơ bản của một hình ảnh là pixel. Một hình ảnh được tạo thành từ một tập hợp các pixel. Mỗi pixel nhỏ này khi được đặt cạnh nhau sẽ tạo thành hình ảnh hoàn chỉnh. Một pixel đơn lẻ được coi là thông tin nhỏ nhất có thể có trong một hình ảnh. Đối với mỗi hình ảnh, giá trị của các pixel nằm trong khoảng từ 0 đến 255.

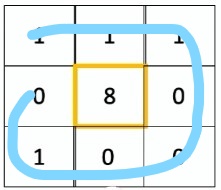
Hình : Ảnh ví dụ về pixel

* *LBPH (Local Binary Patterns Histograms)*: Bắt đầu bằng việc phân tích một ma trận đại diện cho một phần của hình ảnh. Trong ví dụ này, chúng ta có ba hàng và ba cột và tổng số pixel là chín. Chọn pixel trung tâm ở đây, giá trị tám, và áp dụng một điều kiện. Nếu giá trị lớn hơn hoặc bằng 8, kết quả là ‘1’, ngược lại, nếu giá trị nhỏ hơn tám, kết quả là không.

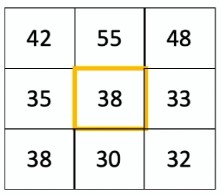
Hình 3: ma trận sau khi tính toán

Hình 2: ma trận biểu thị mức sáng của ảnh



* *Tạo giá trị nhị phân*: Giá trị nhị phân = 11100010. Thuật toán sẽ bắt đầu áp dụng điều kiện từ phần tử góc trên cùng bên trái và đi theo hình tròn như vậy. Sau khi chuyển đổi giá trị nhị phân thành giá trị thập phân, chúng ta nhận được giá trị thập phân = 226. Điều này cho thấy tất cả các pixel xung quanh giá trị trung tâm bằng 226.

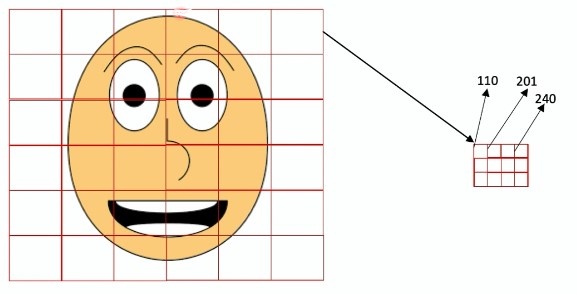
Hình : Cách tiính giá trị nhị phân của ảnh

* *Dưới điều kiện sáng khác nhau:* Nếu bạn đặt đèn pin lên hình ảnh, giá trị của các pixel sẽ tăng lên.

Hình : Ma trận dưới cường độ sáng cao hơn

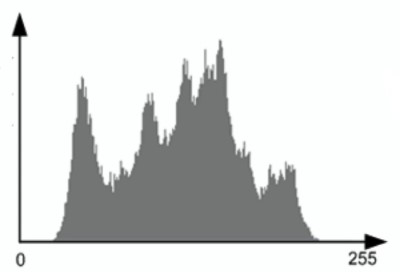
Giá trị càng cao thì hình ảnh càng sáng và khi giá trị thấp hơn thì hình ảnh sẽ tối hơn. Vì lý do này, thuật toán này có kết quả tốt với hình ảnh sáng và tối vì khi hình ảnh trở nên sáng hơn hoặc tối hơn, tất cả các pixel trong vùng lân cận ở đây sẽ thay đổi. Sau khi đặt ánh sáng lên hình ảnh, ma trận sẽ trông như thế này. Sau khi áp dụng điều kiện trên, chúng ta sẽ nhận được giá trị nhị phân giống như trên, tức là 11100010.

* *Áp dụng với mặt người:* Hãy xem xét hình ảnh khác ở đây để hiểu rõ hơn về cách thuật toán sẽ nhận dạng khuôn mặt của một người.



Hình : Ma trận với mặt người

Chúng ta có hình ảnh của một khuôn mặt ở đây, và những gì thuật toán sẽ làm là tạo ra nhiều hình vuông, như bạn có thể thấy ở đây. Và trong mỗi hình vuông này, chúng ta có biểu diễn của ánh sáng trước đó. Ví dụ, hình vuông này ở đây không chỉ đại diện cho một pixel mà là tập hợp với nhiều pixel là ba hàng và bốn cột. Ba nhân bốn bằng mười hai pixel tổng cộng trong những hình vuông này ở mỗi hình vuông này có mười hai pixel. Và sau đó, chúng ta áp dụng điều kiện đó cho mỗi một. Xem xét pixel trung tâm.

* *Tạo ra một biểu đồ histogram*: đó là một khái niệm của thống kê sẽ đếm xem mỗi màu xuất hiện bao nhiêu lần trong mỗi hình vuông. Đây là biểu diễn của biểu đồ histogram.

Hình : Biểu đồ histogram

Dựa trên việc so sánh các biểu đồ histogram, thuật toán sẽ có thể xác định các cạnh và cũng các góc của hình ảnh. Ví dụ, trong hình vuông đầu tiên này, chúng ta không có thông tin về khuôn mặt của người đó. Vì vậy, biểu đồ histogram sẽ khác biệt so với hình vuông khác có biên của khuôn mặt. Nói ngắn gọn, thuật toán biết biểu đồ histogram nào đại diện cho biên và biểu đồ histogram nào đại diện cho các đặc điểm chính của người đó, chẳng hạn như màu mắt, hình dạng của miệng, v.v.

* *Ứng dụng của LBPH*: Thuật toán LBPH được sử dụng rộng rãi trong nhận dạng khuôn mặt. Nó có thể biểu diễn các đặc trưng cục bộ trong các hình ảnh. Nó có khả năng đạt được kết quả tốt (chủ yếu trong môi trường kiểm soát). Nó có khả năng chống lại các biến đổi mức xám đơn điệu.

Tuy nhiên, tốc độ nhận dạng của thuật toán LBPH có hạn, nếu các điều kiện như sự đa dạng biểu cảm, sự mất hướng và thay đổi hiệu suất chiếu sáng xuất hiện.

1. **Ưu, nhược điểm của thuật toán LBPH**
   1. *Một số ưu điểm của thuật toán LBPH*

* *Hiệu quả với kết cấu đơn giản*: LBPH sử dụng toán tử Local Binary Pattern (LBP) để gắn nhãn các pixel của hình ảnh, làm nổi bật các đặc điểm trên khuôn mặt.
* *Biểu diễn dữ liệu đơn giản*: Sử dụng LBP kết hợp với biểu đồ, LBPH có thể biểu diễn hình ảnh khuôn mặt bằng một vectơ dữ liệu đơn giản, giúp việc nhận dạng trở nên dễ dàng hơn.
* *Tùy chỉnh linh hoạt*: LBPH cho phép tùy chỉnh các tham số như bán kính, số lượng điểm mẫu, và số ô lưới, giúp tối ưu hóa thuật toán cho các ứng dụng cụ thể.
* *Phù hợp với các hệ thống nhận dạng khuôn mặt*: LBPH thường được sử dụng trong các hệ thống nhận dạng khuôn mặt do khả năng trích xuất và so sánh các đặc điểm khuôn mặt một cách hiệu quả.
* *Khả năng* *thích nghi tốt*: LBPH có khả năng thích nghi tốt với sự thay đổi về góc nhìn và biểu cảm khuôn mặt, giúp nó hoạt động ổn định trong các điều kiện thực tế.
* *Tính toán nhanh*: LBPH có thể thực hiện tính toán nhanh chóng, nhờ vào việc sử dụng các mẫu nhị phân đơn giản và hiệu quả.
* *Dễ dàng mở rộng*: LBPH có thể dễ dàng mở rộng để xử lý các bộ dữ liệu lớn và đa dạng, làm cho nó trở thành một công cụ linh hoạt trong việc nhận dạng khuôn mặt.
* *Phù hợp với các ứng dụng thực tế*: LBPH đã được chứng minh là phù hợp với nhiều ứng dụng thực tế, từ hệ thống an ninh cho đến các ứng dụng di động và trò chơi điện tử.

Những ưu điểm này giúp LBPH trở thành một lựa chọn phổ biến trong các ứng dụng nhận dạng khuôn mặt, đặc biệt là khi cần một giải pháp có khả năng tùy chỉnh cao và hiệu quả với các đặc điểm kết cấu đơn giản. Tuy nhiên, cần lưu ý rằng LBPH có thể không phù hợp với mọi tình huống và cần được kết hợp với các kỹ thuật khác để đạt được kết quả tốt nhất.

* 1. *Một số nhược điểm của thuật toán LBPH*

Thuật toán LBPH (Local Binary Patterns Histograms) có một số nhược điểm sau:

* *Độ chính xác*: LBPH có thể không hoạt động tốt với các hình ảnh có độ phân giải thấp hoặc khi có sự thay đổi lớn về độ chiếu sáng.
* *Chi phí tính toán*: Nếu sử dụng nhiều điểm mẫu hoặc ô lưới, chi phí tính toán sẽ tăng lên, làm giảm hiệu quả của thuật toán.
* *Biến đổi khuôn mặt*: LBPH có thể không nhận dạng chính xác khi khuôn mặt có sự biến đổi do biểu cảm hoặc góc nhìn.
* *Tắc nghẽn*: Các vật thể che khuất khuôn mặt có thể làm giảm độ chính xác của thuật toán.
* *Khả năng thích ứng*: LBPH có thể không thích ứng tốt với các biến đổi về kích thước hoặc xoay của khuôn mặt.
* *Phụ thuộc vào dữ liệu*: Hiệu suất của LBPH phụ thuộc nhiều vào chất lượng và đa dạng của dữ liệu huấn luyệ.
* *Giới hạn trong phân loại*: LBPH không phải là lựa chọn tốt nhất cho việc phân loại khuôn mặt với số lượng lớn các lớp hoặc danh mục.
* *Cần cân nhắc khi sử dụng trong môi trường đa văn hóa*: Do sự khác biệt về đặc điểm khuôn mặt giữa các dân tộc, LBPH có thể cần được điều chỉnh để tăng cường khả năng nhận dạng trên đa dạng dân tộc.

Những hạn chế này cần được xem xét kỹ lưỡng khi triển khai LBPH trong các ứng dụng thực tế, đặc biệt là trong các hệ thống nhận dạng khuôn mặt quy mô lớn hoặc đa dạng văn hóa. Các nghiên cứu và phát triển tiếp theo có thể tập trung vào việc cải thiện các nhược điểm này để tăng cường hiệu suất và độ chính xác của thuật toán.

1. **Thuật toán Haar Cascade**
   1. *Giới thiệu thuật toán Haar Cascade*

Haar Cascade là một phương pháp mạnh mẽ trong lĩnh vực nhận diện đối tượng và khuôn mặt, nổi tiếng với khả năng nhanh chóng và hiệu quả trong việc xác định vị trí của các đối tượng trong ảnh và video. Được phát triển bởi nhóm nghiên cứu của Viola và Jones, Haar Cascade là một kỹ thuật được ứng dụng rộng rãi trong nhiều ứng dụng công nghệ, đặc biệt là trong lĩnh vực nhận diện khuôn mặt.

Nguyên tắc cơ bản của Haar Cascade là sử dụng các bộ lọc Haar để phát hiện các đặc trưng quan trọng của đối tượng cần nhận diện. Bằng cách xây dựng một loạt các bộ phân lớp được tổ chức thành một cấp độ Cascade, thuật toán này có khả năng nhanh chóng loại bỏ các vùng không quan trọng trong ảnh và tập trung vào các vùng có khả năng chứa đối tượng.

Haar Cascade đã đạt được thành công lớn trong các ứng dụng như nhận diện khuôn mặt trong ảnh và video, giúp tạo nên các hệ thống an ninh, ứng dụng camera giám sát, và nhiều ứng dụng khác. Sự kết hợp giữa tốc độ cao, hiệu quả, và khả năng đào tạo linh hoạt đã làm cho Haar Cascade trở thành một công cụ quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo.

* 1. *Phân tích thuật toán Haar Cascade*

Thuật toán Haar Cascade là một phương pháp trong lĩnh vực thị giác máy tính được sử dụng để nhận diện đối tượng trong ảnh hoặc video. Phương pháp này được giới thiệu bởi Paul Viola và Michael Jones trong bài báo năm 2001 với tên đầy đủ là "Rapid Object Detection using a Boosted Cascade of Simple Features." Haar Cascade thường được sử dụng nhiều trong việc nhận diện khuôn mặt, nhưng nó cũng có thể được áp dụng để nhận diện các đối tượng khác.

Dưới đây là các bước chính của thuật toán Haar Cascade:

* Tạo Đặc Trưng Haar:

- Các đặc trưng Haar là những khu vực hình chữ nhật với giá trị pixel khác nhau.

 - Mỗi đặc trưng này là sự khác biệt giữa tổng giá trị pixel trong các khu vực chữ nhật khác nhau.

Hình : Đặc trưng kiểu haar

* Bộ Phân Loại AdaBoost:

Hình : Những đặc trưng mạnh

- Một bộ phân loại AdaBoost được sử dụng để chọn ra các đặc trưng quan trọng và tạo ra các bộ phân loại "yếu" từ chúng.

- AdaBoost gán trọng số cao cho những đặc trưng quan trọng và sử dụng chúng để xây dựng các bộ phân loại yếu.

* Xây Dựng Cascade:

- Các bộ phân loại yếu được kết hợp thành một bộ phân loại mạnh theo dạng cascade.

- Cascade là một loạt các bộ phân loại, với mỗi bộ được đánh giá theo dạng tuần tự từ dễ đến khó.

- Các cửa sổ không chứa đối tượng có thể được loại bỏ sớm trong quá trình này.

* Kiểm Tra Cửa Sổ Trượt:

- Cửa sổ trượt được áp dụng lên ảnh để kiểm tra xem có chứa đối tượng hay không.

- Mỗi cửa sổ được đưa qua tất cả các bộ phân loại trong cascade. Nếu cửa sổ không chứa đối tượng, quá trình dừng lại, và nếu chứa, tiếp tục với bộ phân loại tiếp t

heo.

* Tối Ưu Hóa:

- Nhiều kỹ thuật tối ưu hóa được sử dụng để làm cho việc nhận diện nhanh chóng và chính xác.

- Sử dụng integral images để tính toán đặc trưng Haar nhanh chóng và giảm thiểu số lượng tính toán cần thiết.

Thuật toán Haar Cascade thích hợp cho ứng dụng thời gian thực và có thể được triển khai trên các thiết bị có tài nguyên hạn chế. Tuy nhiên, độ chính xác của nó có thể bị giảm khi đối mặt với biến động lớn trong hình dạng và ánh sáng của đối tượng.

* 1. *Ứng dụng Haar cascade*

Thuật toán Haar Cascade được sử dụng chủ yếu để nhận diện đối tượng trong ảnh hoặc video. Công dụng chính của Haar Cascade là trong lĩnh vực thị giác máy tính và xử lý ảnh. Dưới đây là một số ứng dụng phổ biến của Haar Cascade:

* Nhận diện Khuôn Mặt:

- Một trong những ứng dụng phổ biến nhất của Haar Cascade là nhận diện khuôn mặt trong ảnh hoặc video.

- Các mô hình Haar Cascade cho nhận diện khuôn mặt được sử dụng rộng rãi trong các ứng dụng như máy ảnh an ninh, nhận dạng khuôn mặt trực tuyến, và trong các ứng dụng máy ảnh trên điện thoại di động.

* Nhận diện Đối Tượng Khác:

- Ngoài khuôn mặt, Haar Cascade cũng có thể được huấn luyện để nhận diện các đối tượng khác như xe ô tô, máy bay, và các đối tượng khác dựa trên đặc trưng của chúng.

* Tracking Đối Tượng:

- Haar Cascade có thể được sử dụng để theo dõi đối tượng trong video. Một khi đối tượng đã được nhận diện, cascade có thể được áp dụng liên tục để giữ theo dõi vị trí của đối tượng trong các frame tiếp theo.

* Nhận diện Gương Mặt trong Ứng Dụng Mạng Xã Hội:

- Nhiều ứng dụng mạng xã hội sử dụng Haar Cascade để nhận diện và nhãn dán khuôn mặt trong ảnh, đặc biệt là khi người dùng muốn tag (gắn nhãn) bạn bè của họ trong các bức ảnh.

* Ứng Dụng trong Robot và Trí Tuệ Nhân Tạo:

- Haar Cascade có thể được tích hợp vào robot và hệ thống trí tuệ nhân tạo để giúp robot nhận diện và tương tác với môi trường xung quanh.

* Kiểm Tra An Ninh và Quản lý Đám Đông:

- Các hệ thống an ninh sử dụng Haar Cascade để theo dõi và nhận diện các sự kiện an ninh, chẳng hạn như theo dõi đám đông hoặc phát hiện hành vi đáng ngờ trong các khu vực công cộng.

Tuy Haar Cascade có nhược điểm là không đủ linh hoạt trong việc nhận diện các đối tượng có biến động lớn về hình dạng và ánh sáng, nhưng nó vẫn được sử dụng rộng rãi do khả năng thực hiện nhanh chóng và có thể triển khai trên nhiều nền tảng.

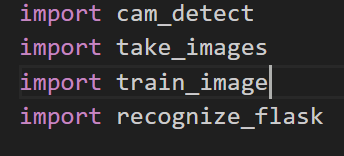
# **CHƯƠNG 3**

# **XÂY DỰNG CHƯƠNG TRÌNH**

1. **Công cụ lập trình**

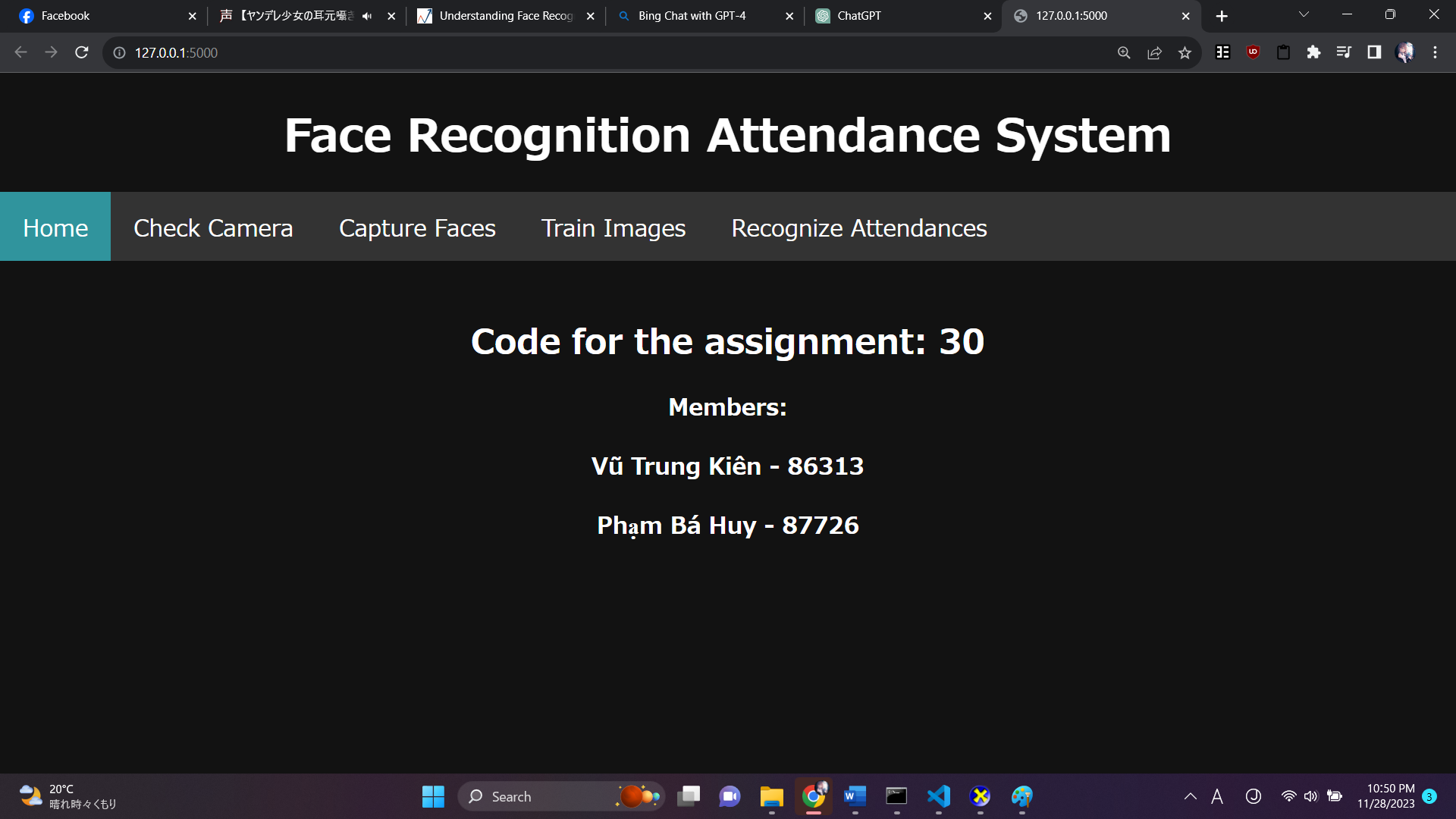
Trong đề tài này, em sử dụng ngôn ngữ lập trình lập trình Python 3.11.4 (thư viện OpenCV 4.8.1, thư viện Flask 3.0.0, thư viện pandas 2.1.1. Bộ công cụ viết code là Visual Studio Code. Các thư viện được em build lại trên hệ thống phần cứng thử nghiệm là 1 laptop sử dụng chip Core i5-11400H, 16GB RAM, có trang bị 1 card GPU Geforce GTX 3050.

1. **Các bước thực hiện**

Đầu tiên em sẽ code ra 4 chức năng chính của chương trình lần lượt là các file cam\_detect.py, take\_images.py, train\_image.py, recognize\_flask.py

Ảnh 1 import chức năng

Sau đó em import tất cả vào app.py, được em xây dựng dựa trên web framework Flask để làm giao diện cho chương trình



Ảnh 2 Giao diện chương trình

Tiếp theo em xin giải thích từng file một hoạt động như nào và chức năng của nó là gì.

* 1. *Chức năng check camera*

Đây là hoạt động của file cam\_detect.py

Ở đây em chỉ đơn giản là dùng haarcascade\_frontalface\_default.xml, import nó vào và dùng Haar Cascade để kiểm tra khuôn mặt thử qua camera

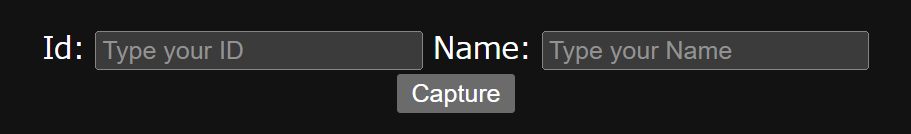
Sau đó với mỗi khuôn mặt phát hiện được em vẽ hình vuông bao quanh khuôn mặt để đánh dấu. Sau đó em dùng imshow của opencv để show kết quả ra.

Bước này chỉ đơn giản là bước kiểm tra hoạt động ổn định của camera và haarcascade để sẵn sàng cho bước tiếp theo

* 1. *Chức năng lấy ảnh học viên*

Đây là hoạt động của file take\_images.py

Đầu tiên cần điền tên và id của học viên cần lấy ảnh, sau đó nhấn vào nút capture



Ảnh 3 Nhập thông tin học viên

Sau đó đầu tiên em sẽ kiểm tra xem id có phải là số và tên có phải là chữ không. Sau đó em sẽ mở camera lên, dùng Haar Cascade để chỉ lấy mỗi khuôn mặt, chuyển ảnh khuôn mặt đấy thành ảnh xám.

Sau đó dùng cv2.imwrite để ghi lại ảnh khuôn mặt vào trong folder TrainingImage với định dạng: name + "." + id + '.' + str(sampleNum) + ".jpg"

Trong đó sampleNum là số thứ tự của ảnh chụp được. Em chỉ lấy đúng 100 ảnh mỗi học viên sau đó dừng chương trình. Sau đó em ghi lại thông tin học viên vừa rồi vào file StudentDetails nằm trong folder StudentDetails

Dưới đây là tập training data mà em đã thu thập được, gồm ảnh của 10 học viên khác nhau:



Ảnh 4 Tập training

* 1. *Chức năng huấn luyện mô hình*

Đây là hoạt động của file train\_image.py.

Đầu tiên em đọc folder TrainingImage nơi lưu tất cả các ảnh của học viên, em lấy từng ảnh một và id của nó dựa vào tên file từ định dạng name + "." + id + '.' + str(sampleNum) + ".jpg" khi em lấy ảnh.

Sau đó em dùng recognizer.train(faces, np.array(id)) để train mô hình nhận diện rồi lưu lại dưới tên model.yml

Nhóm em có 2 bạn nên thời gian train tầm khoảng 10 giây.

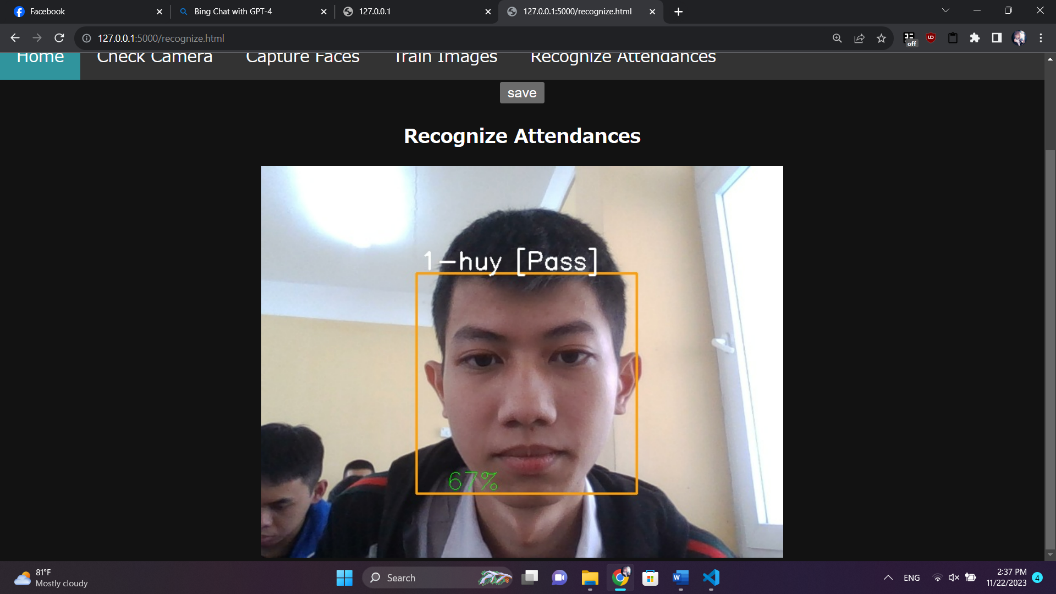
* 1. *Chức năng nhận diện điểm danh học viên*

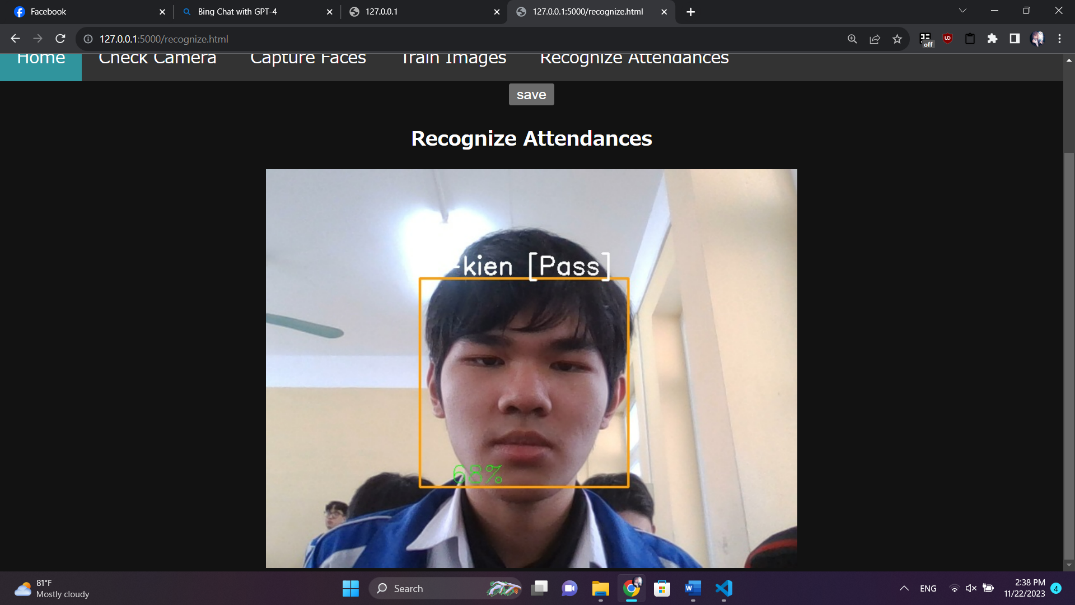
Đây là chức năng của file recognize\_flask.py

Ở đây em sử dụng model mình vừa tạo là model.yml để dự đoán khuôn mặt trên camera. Nếu độ chính xác trên 67% thì em sẽ dùng dataframe của thư viên pandas với các cột id, name, date, time để lưu lại id, tên, ngày, thời gian điểm danh của học viên.

Sau đó em lưu lại kết quả của điểm danh vào folder Attendance

## **Kết quả chạy chương trình**

Để đánh giá kết quả của mô hình em đã xây dựng, em đã thu thập dữ liệu là các ảnh của thành viên nhóm em. Các dữ liệu thu thập đều có chất lượng tốt và có khả năng đánh giá độ chính xác của hệ thống.



Ảnh 5 Kết quả nhận diện

Dựa trên kết quả thử nghiệm, có thể kết luận rằng việc sử dụng mô hình nhận diện khuôn mặt đã đạt được độ chính xác tương đối tốt. Tuy nhiên, vẫn còn một số vấn đề cần được giải quyết để cải thiện hiệu quả của mô hình:

**Cải thiện độ chính xác:** Mặc dù độ chính xác của mô hình đã tốt, nhưng việc tiếp tục tăng độ chính xác là một thách thức lớn. Để giải quyết vấn đề này, có thể sử dụng các kỹ thuật như thêm dữ liệu mới, tăng độ sâu của mô hình hoặc tinh chỉnh siêu tham số của mô hình.

**Xử lý ảnh nhiễu và nhỏ:** Mô hình vẫn gặp khó khăn trong việc nhận diện các khuôn mặt nhỏ hoặc bị nhiễu. Để giải quyết vấn đề này, có thể sử dụng các kỹ thuật xử lý ảnh, như lọc ảnh hoặc tăng độ phân giải của ảnh đầu vào.

**Tốc độ xử lý:** Trong một số trường hợp, mô hình có thể gặp khó khăn trong việc xử lý ảnh nhanh chóng, đặc biệt là khi sử dụng trên các thiết bị có tài nguyên hạn chế. Để giải quyết vấn đề này, có thể sử dụng các kỹ thuật tối ưu hóa hoặc triển khai mô hình trên các nền tảng phần cứng đặc biệt để tăng tốc độ xử lý.

# **KẾT LUẬN**

Trong đề tài này, em đã xây dựng một chương trình nhận diện khuôn mặt bằng cách sử dụng thuật toán LBPH. Chương trình được huấn luyện trên tập dữ liệu lấy từ ảnh học viên để thực hiện nhận diện và điểm danh.

Kết quả thử nghiệm cho thấy, chương trình nhận diện đã đạt được độ chính xác khá trong việc nhận diện các khuôn mặt phổ biến. Tuy nhiên, vẫn còn tồn tại một số hạn chế như khả năng nhận diện còn kém khi nhận diện các khuôn mặt có độ phân giải thấp hoặc bị mờ.

Cụ thể về phần điểm danh, các học viên được điểm danh khi độ nhận diện chính xác trên 60%. Phần lớn các thử nghiệm đều đạt kết quả tốt. Bên cạnh đó, còn tồn đọng một số nhận diện bị sai lệch do nhiều yếu tố như ánh sáng, màu sắc của tóc với nền tường, độ xa gần khi nhận dạng cũng ảnh hưởng không nhỏ. Kế đến là trong quá trình nhập dữ liệu, do ảnh được chụp liên tục, không tránh khỏi những lúc người được chụp di chuyển mặt quá nhanh, hay có tay che miệng, mặt….

Do đây là một trong những đề tài không hề nhỏ trong lĩnh vực. Đồng thời, thời gian còn hạn hẹp nên nhóm chúng em vẫn còn những thiếu sót cần khác phục để có thể hoàn thiện. Chúng em cảm ơn những đóng góp ý kiến của cô cùng các bạn đã hỗ trợ để đạt kết quả như hiện tại và phát triển hơn nữa trong lĩnh vực này.

Tuy nhiên, đây là một bước quan trọng trong việc ứng dụng công nghệ vào giải quyết vấn đề điểm danh trong các trường học và công ty. Trong tương lai, các em hy vọng có thể cải thiện chương trình bằng cách bổ sung thêm dữ liệu và nâng cao khả năng nhận diện của mô hình LBPH.