

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC NGHIỆM/THÍ NGHIỆM
HỌC PHẦN: TRÍ TUỆ NHÂN TẠO – IT6011

Đề tài:

Nhận diện khuôn mặt bằng thuật toán LBPH (Local Binary Patterns Histograms)

Sinh viên thực hiện: Bùi Công Lâm - 2022603110
 Đinh Ngọc Khởi - 2022606346
 Hoàng Quang Kỳ - 2022601978
 Lê Thị Ngọc Lan - 2022602329

Nhóm: 09

Lớp học phần: 20241IT6094003

Khóa: K17

Giảng viên hướng dẫn: Trần Thanh Huân

Hà Nội, tháng 12/2024

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP
HÀ NỘI

CỘNG HÒA XÃ HỘI CHỦ NGHĨA
VIỆT NAM
Độc lập – tự do – hạnh phúc
-----o0o-----

**PHIẾU GIAO ĐỀ TÀI
BÁO CÁO THỰC NGHIỆM/THÍ NGHIỆM**

Nhóm thực hiện: 1) Bùi Công Lâm
2) Đinh Ngọc Khởi
3) Hoàng Quang Kỳ
4) Lê Thị Ngọc Lan

Lớp học phần: 20241IT6094003 Khóa: 17 Khoa Công nghệ thông tin
Ngành học: - Công nghệ thông tin
- Kỹ thuật phần mềm

Tên đề tài: Nhận diện khuôn mặt bằng thuật toán LBPH

Mục đích: Nghiên cứu và ứng dụng thuật toán Local Binary Patterns Histograms (LBPH) trong việc nhận diện khuôn mặt. Đề tài sẽ tập trung vào việc tìm hiểu lý thuyết cơ bản của LBPH, từ đó xây dựng một hệ thống nhận diện khuôn mặt hiệu quả, có thể áp dụng trong các tình huống như xác thực người dùng, bảo mật hệ thống và các ứng dụng an ninh.

Yêu cầu:

- Hoàn thành các nhiệm vụ của đề tài
- Sử dụng kỹ năng làm việc nhóm trong quá trình thực hiện báo cáo
- Trình bày báo cáo đúng yêu cầu, sử dụng kỹ năng “Viết báo cáo” khi thực hiện đề tài
- Sử dụng kỹ năng viết tài liệu kỹ thuật và phi kỹ thuật trong phần mở đầu của báo cáo

Kết quả thu được: Bản báo cáo đề tài (bản cứng và bản mềm); sản phẩm đề tài

Ngày giao đề tài: 20/11/2024

Ngày hoàn thành: 20/12/2024

Giảng viên hướng dẫn: Trần Thanh Huân

Hà nội, Ngày 06 Tháng 11 Năm 2024

GIẢNG VIÊN

Trần Thanh Huân

PHIẾU PHÂN CÔNG NHIỆM VỤ

Nhóm 09, gồm 4 thành viên:

1. Bùi Công Lâm (nhóm trưởng)
2. Đinh Ngọc Khởi
3. Hoàng Quang Kỳ
4. Lê Thị Ngọc Lan

TT	Người thực hiện	Công việc	Kết quả đạt được	Nhận xét của GV
Tuần 11 - Chương 1 : Tổng quan về đề tài				
1	Bùi Công Lâm	1.1. Mục tiêu nghiên cứu	Tìm hiểu và nắm được những mục tiêu cần thiết để hoàn thành nghiên cứu	
2	Hoàng Quang Kỳ	1.2. Phương pháp nghiên cứu	Nghuyên cứu dựa trên các tài liệu thu thập, đánh giá hệ thống qua các trường hợp kiểm thử	
3	Lê Thị Ngọc Lan	1.3. Đối tượng và phạm vi nghiên cứu	Tìm hiểu được một số ứng dụng nhận diện khuôn mặt và quy trình nhận diện khuôn mặt.	
4	Đinh Ngọc Khởi	1.2. Phương pháp nghiên cứu	Tìm hiểu và đưa ra được các phương pháp nghiên cứu về LBHP	
Tuần 12 - Chương 2 : Tìm hiểu chung về phương pháp LBHP				
1	Bùi Công Lâm	2.1. Khái niệm và nguyên lý cơ bản của LBPH	Đã nắm được : Khái Niệm , Nguyên lý cơ bản của thuật toán LBPH	

2	Hoàng Quang Kỳ	2.3. Ưu và nhược điểm của LBPH	Nắm được ưu, nhược điểm của thuật toán LBPH	
3	Lê Thị Ngọc Lan	2.2. Quá trình hoạt động của LBPH	Tìm hiểu và nắm được quá trình hoạt động của LBPH.	
4	Đinh Ngọc Khởi	2.4. Ứng dụng của LBPH	Không hoàn thành	
Tuần 13 - Chương 3 : Ứng dụng LBPH trong nhận diện khuôn mặt				
1	Bùi Công Lâm	3.2 Các công cụ sử dụng 3.3.2 Huấn luyện mô Hình	Nắm được các công cụ sử dụng để áp dụng thuật toán LBPH Áp dụng thuật toán vào thực tiễn để hiểu thêm về mô Hình	
2	Hoàng Quang Kỳ	3.3.1 Phát hiện khuôn mặt và tiền xử lý dữ liệu 3.4 Kết quả	Các bước đầu trong việc xây dựng hệ thống nhận diện khuôn mặt bằng OpenCV Nhận diện được khuôn mặt và biết được nó là của ai	
3	Lê Thị Ngọc Lan	3.1 Nghiên cứu Tổng quan 3.3.2 Huấn luyện mô Hình	Tìm hiểu tổng quan về bài toán nhận diện khuôn mặt, kiến trúc của hệ thống nhận diện và các hạn chế, tồn tại của các phương pháp hiện tại. Áp dụng thuật toán LBPH tạo mô hình nhận diện khuôn mặt.	

4	Đinh Ngọc Khởi	3.3.1 Phát hiện khuôn mặt và tiền xử lý dữ liệu	Nắm được các bước đầu trong việc thực hiện phát hiện khuôn mặt và xử lý tiền dữ liệu	
Tuần 14 - Chương 4 : Kết luận bài học kinh nghiệm				
1	Bùi Công Lâm	4. 1 Kết Luận	Sau khi ứng dụng vào thực tiễn đã rút ra được những thuận lợi ,khó khăn và Tính khả thi của thuật toán LBPH .	
2	Hoàng Quang Kỳ	4.2.1 Kiến Thức	Xây dựng và huấn luyện được mô hình Hiểu hơn về thuật toán LBPH và xử lý hình ảnh với OpenCV	
3	Lê Thị Ngọc Lan	4.2.2 Kỹ năng	Hiểu rõ và trau dồi thêm một số kỹ năng cần thiết như làm việc nhóm, quản lý thời gian, sử dụng một số phần mềm..	
4	Đinh Ngọc Khởi	Kết Luận Tài Liệu Tham Khảo	-Đưa ra kết luận , điểm mạnh và điểm yếu của mô hình nhận diện khuôn mặt LBPH ngoài ra còn những điều kiện thiết để mô hình được phát triển với quy mô lớn hơn -Lấy được tài liệu tham khảo cần thiết cho mô hình	
Tuần 15: Hoàn thành bản mềm				
1	Bùi Công Lâm	Powerpoint	Hoàn thành	
2	Hoàng Quang Kỳ	Powerpoint	Hoàn thành	

3	Lê Thị Ngọc Lan	word	Hoàn thành	
4	Đinh Ngọc Khởi	word	Không hoàn thành	

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ ngày càng phát triển, nhận diện khuôn mặt đang trở thành một trong những ứng dụng thiết yếu của thị giác máy tính, góp phần nâng cao hiệu quả trong nhiều lĩnh vực như an ninh, quản lý, và tự động hóa. Báo cáo này được thực hiện nhằm triển khai một hệ thống nhận diện khuôn mặt cơ bản dựa trên thư viện OpenCV, qua đó tìm hiểu sâu hơn về các thuật toán và phương pháp được sử dụng trong lĩnh vực này.

Trong quá trình thực hiện bài tập lớn này, chúng em xin gửi lời cảm ơn chân thành đến thầy Trần Thanh Huân – giảng viên hướng dẫn, đã nhiệt tình chỉ bảo và hỗ trợ chúng em hoàn thiện đề tài. Chúng em cũng xin chân thành cảm ơn sự đóng góp ý kiến quý báu từ các thầy cô giáo và các bạn sinh viên.

Trong quá trình hoàn thành bài báo cáo nhóm chúng em không thể tránh khỏi những thiếu sót rất mong nhận được sự góp ý từ thầy cô để bài báo cáo được hoàn thiện hơn. Chúng em xin chân thành cảm ơn!

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI.....	11
1.1. Mục tiêu nghiên cứu	11
1.2. Phương pháp nghiên cứu	11
1.2.1. Phương pháp tài liệu	11
1.2.2. Phương pháp quan sát	12
1.2.3. Phương pháp thực nghiệm.....	12
1.3. Đối tượng và phạm vi nghiên cứu	12
1.3.1. Đối tượng.....	12
1.3.2. Phạm vi nghiên cứu	12
CHƯƠNG 2. TÌM HIỂU CHUNG VỀ PHƯƠNG PHÁP LBPH	14
2.1. Khái niệm và nguyên lý cơ bản của LBPH.....	14
2.1.1. Khái niệm về LBPH.....	14
2.1.2. Nguyên lý cơ bản của LBPH.....	14
2.2. Quá trình hoạt động của LBPH	14
2.2.1. Tham số	14
2.2.2. Huấn luyện thuật toán.....	15
2.2.3. Áp dụng thao tác LBP	15
2.2.4. Trích xuất biểu đồ	15
2.2.5. Thực hiện nhận dạng khuôn mặt	15
2.3. Ưu và nhược điểm của LBPH.....	15
2.3.1. Ưu Điểm.....	15
2.3.2. Nhược Điểm.....	16
2.4. Ứng dụng của LBPH	16
CHƯƠNG 3. ỨNG DỤNG LBPH TRONG NHẬN DIỆN KHUÔN MẶT	17
3.1. Nghiên cứu tổng quan.....	17
3.1.1. Bài toán nhận dạng mặt người	17
3.1.2. Tổng quan kiến trúc của một hệ thống nhận diện khuôn mặt.....	17
3.1.3. Hạn chế, tồn tại của các phương pháp	18
3.2. Các công cụ sử dụng	19
3.2.1. Ngôn ngữ và thư viện	19
3.2.2. Môi trường lập trình Visual Studio Code	21
3.3. Các bước thực hiện	22

3.3.1. Phát hiện khuôn mặt và tiền xử lý dữ liệu	22
3.3.2. Huấn luyện mô hình	27
3.4. Kết quả	27
CHƯƠNG 4. KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM.....	28
4.1. Kết luận.....	28
4.1.1. Tính khả thi.....	28
4.1.2. Thuận lợi	28
4.1.3. Khó khăn	28
4.2. Bài học kinh nghiệm	29
4.2.1. Kiến thức	29
4.2.2. Kỹ năng.....	29
KẾT LUẬN	30
TÀI LIỆU THAM KHẢO	31

DANH SÁCH HÌNH ẢNH

Hình 2.1. Hệ thống nhận diện khuôn mặt.....	17
Hình 2.2. Kiến trúc của một hệ thống nhận diện khuôn mặt.....	18
Hình 2.3. OpenCV sử dụng trong nhận diện khuôn mặt	21
Hình 2.4. Môi trường lập trình VS Code	21
Hình 2.5. Xây dựng cơ sở dữ liệu.....	23
Hình 2.6. Các thư viện sử dụng.....	23
Hình 2.7. Kết quả thử nghiệm với webcam	27

PHẦN MỞ ĐẦU

Trong thời đại công nghệ số phát triển vượt bậc, nhận diện khuôn mặt đã trở thành một trong những lĩnh vực quan trọng của thị giác máy tính, được ứng dụng rộng rãi trong các hệ thống an ninh, quản lý và các ứng dụng thông minh. Việc xây dựng hệ thống nhận diện khuôn mặt hiệu quả và chính xác không chỉ giúp cải thiện khả năng tự động hóa mà còn mở ra những cơ hội lớn trong nhiều ngành công nghiệp.

Hiện nay, có nhiều phương pháp đã và đang được sử dụng để giải quyết bài toán nhận diện khuôn mặt. Một số cách tiếp cận phổ biến bao gồm sử dụng mạng nơ-ron tích chập (Convolutional Neural Networks - CNNs), phương pháp dựa trên histogram đặc trưng cục bộ (Local Binary Patterns Histograms - LBPH), và các thuật toán dựa trên học máy. Trong báo cáo này, chúng em sử dụng thư viện OpenCV và phương pháp LBPH để xây dựng một hệ thống nhận diện khuôn mặt đơn giản nhưng hiệu quả.

Báo cáo này nhằm mục tiêu giới thiệu và triển khai một ứng dụng nhận diện khuôn mặt thời gian thực. Đồng thời, nhóm chúng em sẽ xác định và giới hạn phạm vi vấn đề, tập trung vào việc sử dụng LBPH để nhận diện khuôn mặt từ nguồn video đầu vào. Báo cáo cũng sẽ tập trung phân tích các thách thức liên quan đến việc cải thiện độ chính xác và hiệu suất nhận diện trong điều kiện ánh sáng và độ sáng khác nhau.

Nội dung chính của báo cáo bao gồm:

Chương 1: Tìm hiểu chung về phương pháp LBPH.

Chương 2: Ứng dụng LBPH trong nhận diện khuôn mặt.

Chương 3: Kết luận và bài học kinh nghiệm.

Với báo cáo này, nhóm chúng em mong muốn đem đến một góc nhìn toàn diện và các giải pháp khả thi để phát triển hệ thống nhận diện khuôn mặt dựa trên OpenCV, đáp ứng yêu cầu cơ bản về hiệu quả và độ tin cậy.

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

1.1. Mục tiêu nghiên cứu

- **Xây dựng hệ thống nhận diện khuôn mặt hiệu quả:**
 - ✓ Triển khai một hệ thống nhận diện khuôn mặt cơ bản sử dụng thuật toán **LBPH (Local Binary Patterns Histogram)** kết hợp với thư viện Python **OpenCV**.
 - ✓ Tạo ra ứng dụng có khả năng nhận diện khuôn mặt chính xác trong điều kiện thực tế, bao gồm ánh sáng, góc chụp, và biểu cảm khuôn mặt khác nhau.
- **Tối ưu hóa thuật toán LBPH:**
 - ✓ Nghiên cứu và cải tiến thuật toán LBPH để giảm độ phức tạp tính toán, từ đó tăng hiệu suất xử lý.
 - ✓ Đánh giá khả năng sử dụng LBPH trong việc nhận diện ảnh grayscale nhằm cải thiện độ chính xác và tốc độ.
- **Ứng dụng thực tiễn:**
 - ✓ Phát triển ứng dụng nhận diện khuôn mặt với tính năng quản lý dữ liệu, bảo mật.
 - ✓ Hệ thống sẽ có khả năng áp dụng trong các lĩnh vực như kiểm soát ra vào, chấm công tự động, và quản lý nhân sự.
- **Khám phá tiềm năng của trí tuệ nhân tạo trong xử lý hình ảnh:**
 - ✓ Tạo điều kiện để người dùng tiếp cận và hiểu rõ hơn về ứng dụng của trí tuệ nhân tạo trong thị giác máy tính thông qua các công nghệ nhận diện khuôn mặt.

1.2. Phương pháp nghiên cứu

1.2.1. Phương pháp tài liệu

- **Mục tiêu:**
 - ✓ Tìm hiểu các tài liệu học thuật, bài báo khoa học, và hướng dẫn kỹ thuật về thuật toán LBPH và thư viện OpenCV.
- **Hoạt động:**
 - ✓ Phân tích nguyên lý hoạt động của LBPH:
 - Phương pháp chuyển đổi ảnh sang dạng grayscale.
 - Tính toán các mẫu nhị phân cục bộ và xây dựng biểu đồ histogram.
 - ✓ Nghiên cứu ứng dụng thực tế của LBPH trong nhận diện khuôn mặt.

- ✓ Tham khảo các tài liệu so sánh giữa LBPH và các thuật toán khác như EigenFace, FisherFace, và Deep Learning.

1.2.2. Phương pháp quan sát

– Mục tiêu:

- ✓ Đánh giá hiệu suất của hệ thống nhận diện khuôn mặt khi chạy trên các tập dữ liệu và trong điều kiện thực tế.

– Hoạt động:

- ✓ Quan sát hiệu quả của hệ thống trong điều kiện ánh sáng, góc chụp, và chất lượng hình ảnh khác nhau.
- ✓ So sánh khả năng nhận diện với các ứng dụng nhận diện khuôn mặt hiện tại để tìm ra các điểm mạnh và hạn chế.

1.2.3. Phương pháp thực nghiệm

– Mục tiêu:

- ✓ Triển khai, kiểm thử, và cải tiến thuật toán LBPH để tối ưu hóa hiệu suất và độ chính xác.

– Hoạt động:

- ✓ Xây dựng chương trình nhận diện khuôn mặt sử dụng Python và OpenCV.
- ✓ Huấn luyện mô hình với tập dữ liệu có gắn nhãn ID.
- ✓ Chạy thử nghiệm trên dữ liệu thực tế để đánh giá hiệu suất.
- ✓ Điều chỉnh các tham số như kích thước lưới (Grid X, Grid Y), bán kính (Radius), và số điểm lân cận (Neighbors) để tối ưu mô hình.

1.3. Đối tượng và phạm vi nghiên cứu

1.3.1. Đối tượng

- Các ứng dụng nhận diện khuôn mặt, bao gồm:
 - ✓ Hệ thống điểm danh tự động trong trường học, công ty.
 - ✓ Ứng dụng bảo mật như kiểm soát ra vào, mở khóa bằng khuôn mặt.
- Thuật toán LBPH và các phương pháp xử lý hình ảnh.

1.3.2. Phạm vi nghiên cứu

- Quy tắc và quy trình nhận diện khuôn mặt:
 - ✓ Chuyển đổi ảnh màu sang ảnh grayscale.
 - ✓ Phân tích đặc trưng khuôn mặt bằng các mẫu nhị phân cục bộ (LBP).
 - ✓ Biểu diễn khuôn mặt dưới dạng biểu đồ histogram.

- Thuật toán LBPH:
 - ✓ Tối ưu hóa tham số của LBPH để cải thiện độ chính xác và tốc độ nhận diện.
- Xây dựng ứng dụng thực tế:
 - ✓ Triển khai hệ thống trên các thiết bị máy tính, camera.

CHƯƠNG 2. TÌM HIỂU CHUNG VỀ PHƯƠNG PHÁP LBPH

2.1. Khái niệm và nguyên lý cơ bản của LBPH

2.1.1. Khái niệm về LBPH

LBPH là một phương pháp mô tả đặc trưng hình ảnh, tập trung vào các mẫu kết cấu cục bộ. Thuật toán dựa trên việc phát hiện các mẫu nhị phân cục bộ (local binary patterns) trong vùng lân cận của từng điểm ảnh, sau đó tổng hợp các thông tin này dưới dạng histogram để biểu diễn toàn bộ hình ảnh hoặc khu vực quan tâm.

LBPH được ưa chuộng trong nhận diện khuôn mặt bởi tính đơn giản, tốc độ xử lý nhanh, và khả năng hoạt động tốt ngay cả khi điều kiện ánh sáng thay đổi.

2.1.2. Nguyên lý cơ bản của LBPH

LBPH gồm 4 bước chính:

Bước 1: Tính toán các mẫu nhị phân cục bộ (Local Binary Patterns - LBP)

- Mỗi điểm ảnh được so sánh với các điểm ảnh lân cận trong một vùng (thường là 3×3 pixel).
- Một giá trị nhị phân được gán cho từng điểm ảnh lân cận, bằng cách so sánh độ sáng:
 - ✓ Nếu giá trị của điểm ảnh lân cận \geq giá trị điểm trung tâm, gán 1.
 - ✓ Nếu không, gán 0.
- Tập hợp các giá trị nhị phân này tạo thành một số nhị phân (dạng chuỗi), sau đó được chuyển đổi thành số thập phân.

Bước 2: Tính toán histogram

- Sau khi áp dụng LBP cho từng pixel của ảnh, các giá trị LBP thu được được gom thành một histogram.
- Histogram biểu diễn tần suất xuất hiện của từng giá trị LBP, giúp nắm bắt kết cấu cục bộ của hình ảnh.

Bước 3: Chuẩn hóa histogram

- Histogram được chuẩn hóa để đảm bảo tính đồng nhất, giúp cải thiện khả năng phân loại và nhận diện.

2.2. Quá trình hoạt động của LBPH

2.2.1. Tham số

LBPH sử dụng 4 tham số:

- Radius: bán kính được sử dụng để xây dựng mẫu nhị phân cục bộ hình tròn và biểu thị bán kính xung quanh pixel trung tâm. Nó thường được đặt thành 1.

- Neighbors: số lượng điểm mẫu để xây dựng mẫu nhị phân cục bộ hình tròn. Nó thường được đặt thành 8.
- Grid X: số lượng ô theo chiều ngang. Càng nhiều ô, lưới càng mịn thì chiều của vector đặc trưng thu được càng cao. Nó thường được đặt thành 8.
- Grid Y: số lượng ô theo chiều dọc. Càng nhiều ô, lưới càng mịn thì chiều của vector đặc trưng thu được càng cao. Nó thường được đặt thành 8.

2.2.2. Huấn luyện thuật toán

Đầu tiên chúng ta cần huấn luyện thuật toán. Để làm như vậy, chúng ta cần sử dụng tập dữ liệu có hình ảnh khuôn mặt của những người mà chúng ta muốn nhận dạng. Chúng ta cũng cần đặt ID (có thể là số hoặc tên người) cho mỗi hình ảnh, vì vậy thuật toán sẽ sử dụng thông tin này để nhận dạng hình ảnh đầu vào và cung cấp cho bạn đầu ra. Hình ảnh của cùng một người phải có cùng ID. Với tập huấn luyện đã được xây dựng, hãy xem các bước tính toán LBPH.

2.2.3. Áp dụng thao tác LBP

Bước tính toán đầu tiên của LBPH là tạo ra một ảnh trung gian mô tả ảnh gốc một cách tốt hơn bằng cách làm nổi bật các đặc điểm trên khuôn mặt. Để làm như vậy, thuật toán sử dụng khái niệm của số trượt, dựa trên bán kính tham số và các lân cận.

2.2.4. Trích xuất biểu đồ

Bây giờ, bằng cách sử dụng hình ảnh được tạo ra, chúng ta có thể sử dụng tham số Grid X và Grid Y để chia hình ảnh thành nhiều lưới.

2.2.5. Thực hiện nhận dạng khuôn mặt

Ở bước này, thuật toán đã được huấn luyện. Mỗi biểu đồ được tạo sẽ được sử dụng để thể hiện từng hình ảnh từ tập dữ liệu huấn luyện. Vì vậy, với một hình ảnh đầu vào, chúng tôi thực hiện lại các bước cho hình ảnh mới này và tạo biểu đồ đại diện cho hình ảnh.

2.3. Ưu và nhược điểm của LBPH

2.3.1. Ưu Điểm

- **Đơn giản và hiệu quả:**
 - ✓ LBPH là một thuật toán đơn giản, dễ hiểu, và dễ triển khai.
 - ✓ Hiệu quả cao trong nhận dạng khuôn mặt với các tập dữ liệu nhỏ hoặc khi chất lượng hình ảnh không cao.
- **Tốc độ xử lý nhanh:** Do không yêu cầu tính toán phức tạp, LBPH phù hợp với các ứng dụng yêu cầu xử lý thời gian thực, ví dụ: hệ thống giám sát.
- **Khả năng chịu ảnh hưởng của ánh sáng:** LBPH dựa vào các mẫu cục bộ (local patterns), nên ít bị ảnh hưởng bởi sự thay đổi ánh sáng trong hình ảnh.

- **Khả năng hoạt động với các tập dữ liệu nhỏ:** Không cần quá nhiều dữ liệu để đào tạo, thuật toán vẫn có thể hoạt động tốt, đặc biệt trong môi trường kiểm soát.
- **Linh hoạt:** Có thể áp dụng không chỉ cho nhận diện khuôn mặt mà còn cho các bài toán nhận diện hoặc phân loại hình ảnh khác.

2.3.2. Nhược Điểm

- **Nhạy cảm với các biến dạng lớn:** LBPH không hiệu quả khi đối tượng có các biến dạng lớn như xoay, thay đổi góc độ hay cử chỉ khuôn mặt.
- **Phụ thuộc vào thông tin cục bộ:** Thuật toán chỉ dựa trên thông tin cục bộ nên không tận dụng được toàn bộ đặc trưng toàn cục của khuôn mặt.
- **Kém chính xác trên tập dữ liệu lớn và đa dạng:** Với các tập dữ liệu lớn chứa nhiều biến thể về khuôn mặt, độ tuổi, giới tính, và điều kiện môi trường, LBPH không cạnh tranh được với các phương pháp hiện đại như Deep Learning.
- **Không có khả năng tự động học:** LBPH chỉ sử dụng các đặc trưng đã được định nghĩa trước (Local Binary Patterns) mà không thể tự động học các đặc trưng từ dữ liệu như các mô hình học sâu.

2.4. Ứng dụng của LBPH

- **Nhận dạng khuôn mặt trong hệ thống bảo mật:** Dùng trong hệ thống kiểm tra danh tính tại văn phòng, trường học, hoặc nhà riêng.
- **Hệ thống giám sát thời gian thực:** Áp dụng trong camera an ninh để nhận diện khuôn mặt nhanh chóng.
- **Chấm công bằng nhận diện khuôn mặt:** LBPH thường được dùng trong các thiết bị chấm công có quy mô nhỏ.
- **Ứng dụng trong robot và IoT:** Dùng để nhận diện khuôn mặt cho các robot giao tiếp với con người hoặc các hệ thống IoT.
- **Nhận diện khuôn mặt trên thiết bị hạn chế tài nguyên:** Phù hợp với các thiết bị có tài nguyên phần cứng hạn chế như Raspberry Pi, Arduino, hoặc thiết bị nhúng.
- **Phân loại hình ảnh cục bộ:** LBPH không chỉ dùng cho khuôn mặt mà còn có thể ứng dụng trong phân loại hoa, lá, vân tay, hoặc các mẫu vật cục bộ trong y học và nông nghiệp.

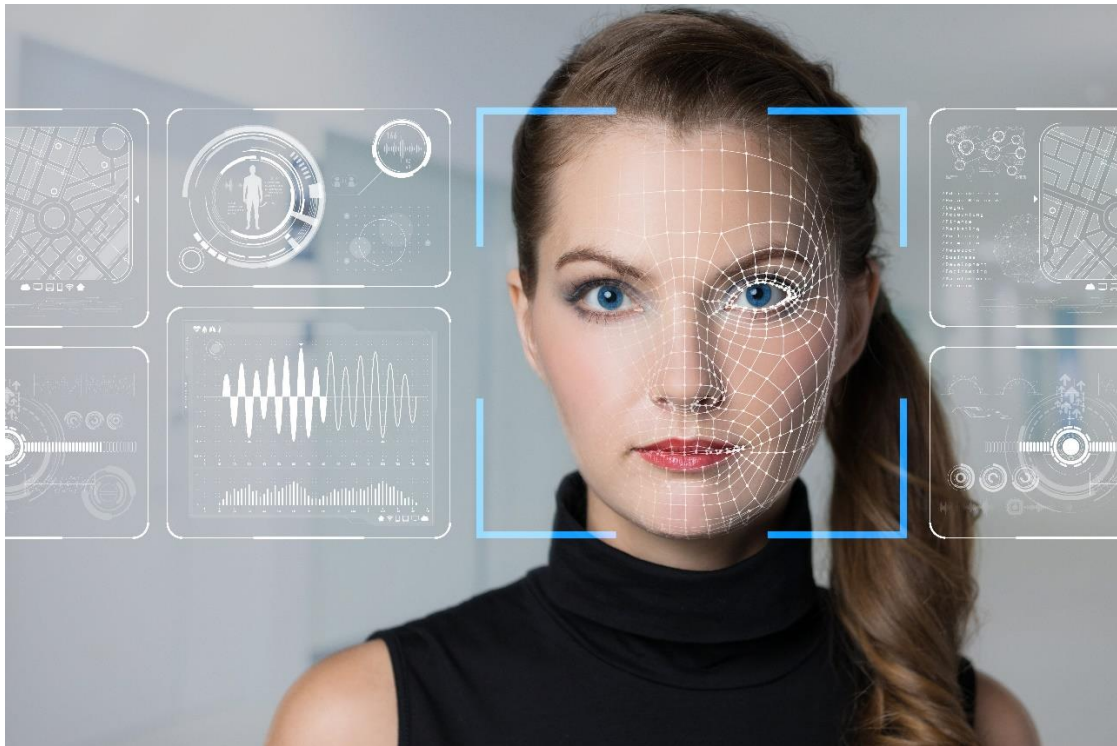
CHƯƠNG 3. ỨNG DỤNG LBHP TRONG NHẬN DIỆN KHUÔN MẶT

3.1. Nghiên cứu tổng quan

3.1.1. Bài toán nhận dạng mặt người

Bài toán nhận diện khuôn mặt là một bài toán phổ biến trong AI, nó có tên tiếng anh thường gọi là Face recognition:

- Input: 1 ảnh chứa mặt người (có thể là một người hoặc nhiều hơn)
- Output: xuất ra tên của người tương ứng có trong k người ở cơ sở dữ liệu.
- Hệ thống nhận dạng mặt người là một hệ thống nhận vào là một ảnh, một đoạn video (một dòng các hình ảnh liên tục) hoặc bằng webcam. Qua xử lý, tính toán hệ thống xác định được vị trí mặt người (nếu có) trong ảnh và xác định là người nào trong số những người mà hệ thống đã được biết (qua quá trình học) hoặc là người lạ.

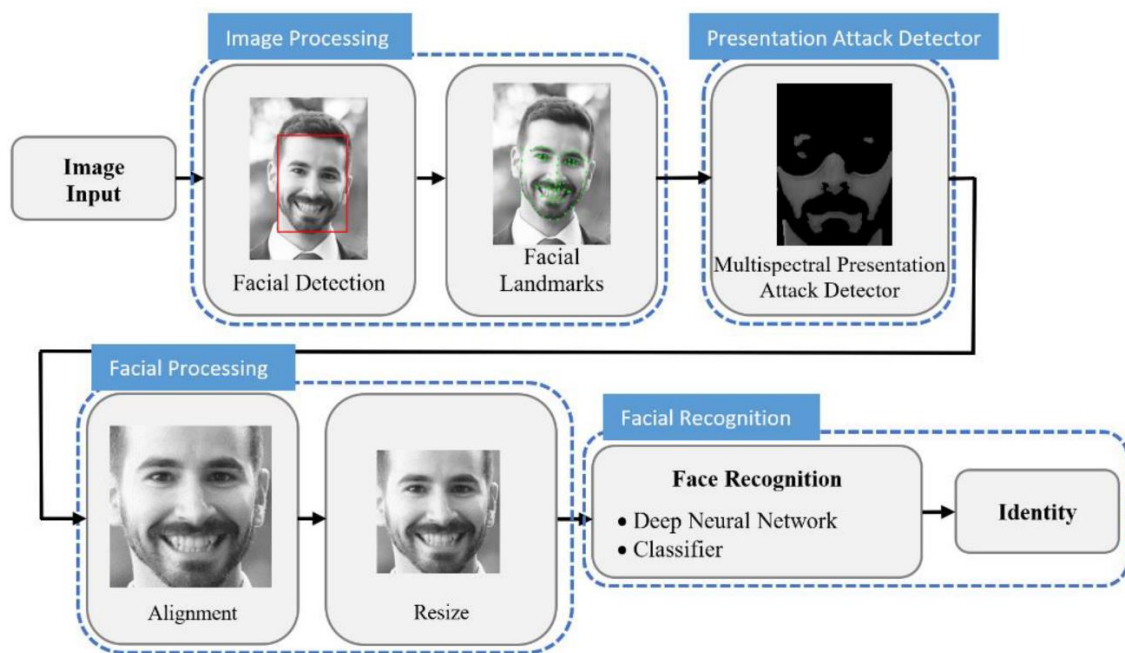


Hình 3.1. Hệ thống nhận diện khuôn mặt

3.1.2. Tổng quan kiến trúc của một hệ thống nhận diện khuôn mặt

Một hệ thống nhận diện khuôn mặt thường bao gồm 4 bước xử lý

- Phát hiện khuôn mặt
- Phân đoạn khuôn mặt
- Trích trong đặc trưng
- Nhận dạng hay phân lớp khuôn mặt



Hình 3.2. Kiến trúc của một hệ thống nhận diện khuôn mặt

3.1.3. Hạn chế, tồn tại của các phương pháp

- Độ chính xác và độ tin cậy bị ảnh hưởng bởi môi trường và yếu tố con người
 - ✓ **Ánh sáng và góc chụp:** Hiệu suất nhận diện giảm trong điều kiện ánh sáng yếu, bóng tối, hoặc khi khuôn mặt không nhìn thẳng vào camera.
 - ✓ **Biểu cảm khuôn mặt:** Những thay đổi như cười, cau mày, hoặc nhắm mắt có thể gây sai lệch.
 - ✓ **Trang phục và phụ kiện:** Mũ, kính mát, khẩu trang, hoặc các yếu tố che khuất khuôn mặt làm giảm độ chính xác.
- Thiếu công bằng
 - ✓ **Chênh lệch độ chính xác:** Các hệ thống thường kém hiệu quả hơn đối với một số nhóm người (dựa trên màu da, giới tính, hoặc độ tuổi), do dữ liệu đào tạo thiếu đa dạng.
 - ✓ **Thiên vị trong thuật toán:** Nếu dữ liệu huấn luyện không cân bằng, hệ thống sẽ thiên vị, dẫn đến sai lệch hoặc phân biệt đối xử.
- Vấn đề bảo mật và quyền riêng tư
 - ✓ **Nguy cơ giả mạo:** Kẻ tấn công có thể sử dụng hình ảnh hoặc video giả mạo (deepfake, ảnh in 2D, hoặc mô hình 3D) để đánh lừa hệ thống.
 - ✓ **Xâm phạm quyền riêng tư:** Việc sử dụng nhận diện khuôn mặt trong giám sát có thể bị lạm dụng, xâm phạm quyền cá nhân.
 - ✓ **Lưu trữ và bảo vệ dữ liệu:** Rủi ro liên quan đến việc lưu trữ dữ liệu khuôn mặt, như bị hack hoặc sử dụng sai mục đích.

- Hiệu suất trên dữ liệu lớn và thực tế
 - ✓ **Hiệu quả trên quy mô lớn:** Khi số lượng người cần nhận diện tăng, thời gian xử lý và khả năng nhận diện chính xác có thể giảm.
 - ✓ **Khả năng chịu lỗi:** Các phương pháp hiện nay không luôn đủ linh hoạt để xử lý dữ liệu không hoàn hảo hoặc bị biến dạng, như ảnh mờ hoặc chất lượng thấp.

3.2. Các công cụ sử dụng

3.2.1. Ngôn ngữ và thư viện

3.2.1.1. Ngôn ngữ

- Python là một ngôn ngữ lập trình bậc cao, mã nguồn mở, đa năng. Các nhà phát triển sử dụng Python vì nó hiệu quả, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Ngôn ngữ này được tạo ra bởi Guido van Rossum vào năm 1991 và được phát triển và duy trì bởi một cộng đồng lớn các lập trình viên trên toàn thế giới.
- Python mang lại những lợi ích gì?
 - ✓ Các nhà phát triển có thể dễ dàng đọc và hiểu một chương trình Python vì ngôn ngữ này có cú pháp cơ bản giống tiếng Anh.
 - ✓ Python giúp cải thiện năng suất làm việc của các nhà phát triển vì so với những ngôn ngữ khác, họ có thể sử dụng ít dòng mã hơn để viết một chương trình Python.
 - ✓ Python có một thư viện tiêu chuẩn lớn, chứa nhiều dòng mã có thể tái sử dụng cho hầu hết mọi tác vụ. Nhờ đó, các nhà phát triển sẽ không cần phải viết mã từ đầu.
 - ✓ Các nhà phát triển có thể dễ dàng sử dụng Python với các ngôn ngữ lập trình phổ biến khác như Java, C và C++.
 - ✓ Cộng đồng Python tích cực hoạt động bao gồm hàng triệu nhà phát triển nhiệt tình hỗ trợ trên toàn thế giới. Nếu gặp phải vấn đề, bạn sẽ có thể nhận được sự hỗ trợ nhanh chóng từ cộng đồng.
 - ✓ Trên Internet có rất nhiều tài nguyên hữu ích nếu bạn muốn học Python. Ví dụ: bạn có thể dễ dàng tìm thấy video, chỉ dẫn, tài liệu và hướng dẫn dành cho nhà phát triển.
 - ✓ Python có thể được sử dụng trên nhiều hệ điều hành máy tính khác nhau, chẳng hạn như Windows, macOS, Linux và Unix.
- Python được sử dụng như thế nào?
 - ✓ Phát triển web phía máy chủ: Python cung cấp nhiều thư viện và khung (framework) như Django và Flask, giúp xây dựng ứng dụng web nhanh chóng, dễ dàng và hỗ trợ kiểm thử tích hợp.

- ✓ Tự động hóa bằng tập lệnh: Python được dùng để tự động hóa các tác vụ lặp đi lặp lại như đổi tên tệp, gửi email, chuyển đổi định dạng tệp, và kiểm tra lỗi trong mã.
- ✓ Khoa học dữ liệu và học máy (ML): Python hỗ trợ làm sạch dữ liệu, trực quan hóa dữ liệu, và xây dựng các mô hình học máy thông qua thư viện mạnh mẽ như Pandas, NumPy, Matplotlib, Scikit-learn, và TensorFlow.
- ✓ Phát triển phần mềm: Python hỗ trợ theo dõi lỗi, tự động hóa xây dựng phần mềm, phát triển nguyên mẫu, và tạo ứng dụng với giao diện đồ họa (GUI). Ngoài ra, Python còn được sử dụng trong phát triển game.
- ✓ Kiểm thử phần mềm: Python có các khung kiểm thử như Unittest và Robot, hỗ trợ kiểm tra tính năng và tự động hóa kiểm thử thông qua các công cụ CI/CD như Jenkins và Travis CI.

3.2.1.2. Thư viện sử dụng OpenCV

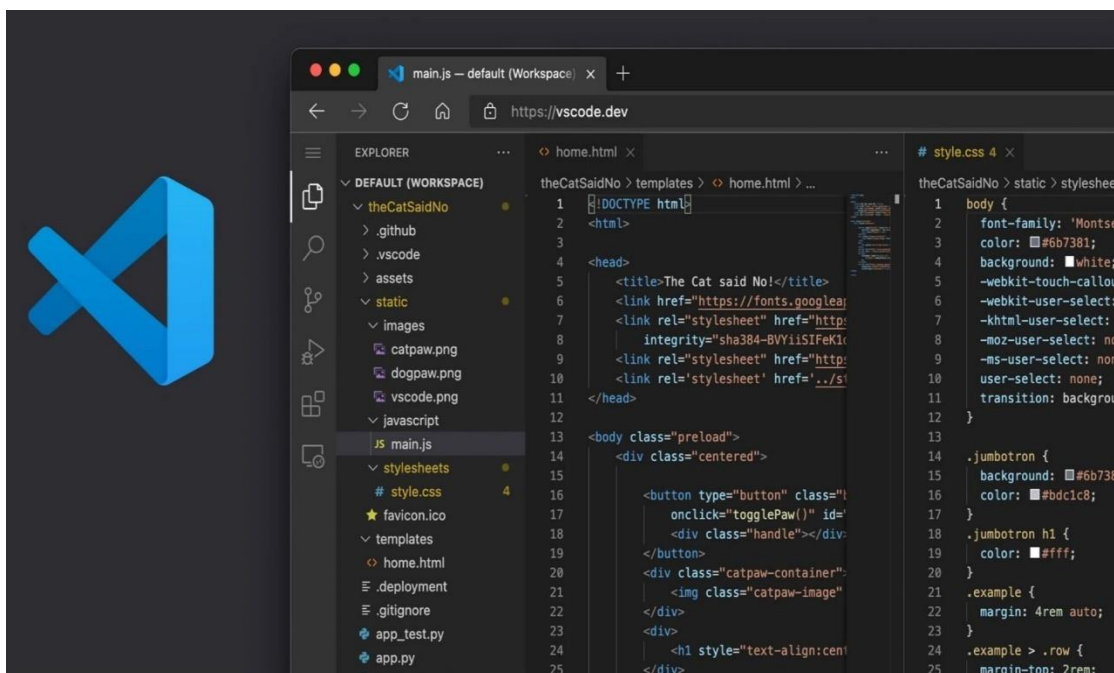
- OpenCV-Python là một thư viện mà các nhà phát triển sử dụng để xử lý hình ảnh cho các ứng dụng thị giác máy tính. Thư viện này cung cấp nhiều hàm cho các tác vụ xử lý hình ảnh như đọc và ghi hình ảnh cùng lúc, xây dựng môi trường 3D từ môi trường 2D cũng như chụp và phân tích hình ảnh từ video.
- OpenCV đang được sử dụng rộng rãi trong các ứng dụng bao gồm:
 Hình ảnh street view Kiểm tra và giám sát tự động Robot và xe hơi tự lái Phân tích hình ảnh y tế Tìm kiếm và phục hồi hình ảnh/video Phim - cấu trúc 3D từ chuyển động Nghệ thuật sắp đặt tương tác.
- Chức năng OpenCV:
 Image/video I/O, xử lý, hiển thị (core, imgproc, highgui) Phát hiện các vật thể (objdetect, features2d, nonfree) Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab) Computational photography (photo, video, superres) Machine learning & clustering (ml, flann) CUDA acceleration (gpu).



Hình 3.3. OpenCV sử dụng trong nhận diện khuôn mặt

3.2.2. Môi trường lập trình Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn miễn phí, đa nền tảng, được phát triển bởi Microsoft. Nó đã nhanh chóng trở thành một trong những trình soạn thảo phổ biến nhất dành cho các nhà phát triển, nhờ giao diện thân thiện, tính năng mạnh mẽ và khả năng mở rộng cao.



Hình 3.4. Môi trường lập trình VS Code

Lý do nên chọn Visual Studio Code (VS Code):

- Miễn phí và mã nguồn mở: Được cấp phép MIT, hoàn toàn miễn phí và cho phép chỉnh sửa, đóng góp mã nguồn.
- Đa nền tảng: Hoạt động trên Windows, macOS và Linux, thuận tiện cho làm việc trên nhiều hệ điều hành.
- Giao diện thân thiện: Đơn giản, trực quan, phù hợp cho cả người mới và lập trình viên chuyên nghiệp.
- Hỗ trợ đa ngôn ngữ lập trình: Hỗ trợ hơn 30 ngôn ngữ như JavaScript, Python, Java, C++, với IntelliSense, gỡ lỗi tích hợp và hỗ trợ TypeScript.
- Hệ thống Extension phong phú: Cho phép tùy chỉnh và thêm tính năng mới, phù hợp mọi nhu cầu lập trình.
- Gỡ lỗi hiệu quả: Hỗ trợ breakpoint, kiểm tra biến, điều khiển luồng mã, với console tích hợp.
- So sánh vượt trội: VS Code hơn Sublime Text ở tiện ích mở rộng, gỡ lỗi và Git; vượt Atom về hiệu suất và ổn định; đa năng hơn IntelliJ IDEA về hỗ trợ nhiều ngôn ngữ.
- Tích hợp dịch vụ phổ biến: Kết nối Git, Azure, GitHub, Docker, giúp quản lý mã và triển khai ứng dụng thuận tiện.

3.3. Các bước thực hiện

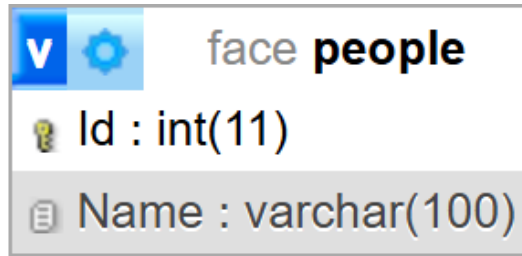
3.3.1. Phát hiện khuôn mặt và tiền xử lý dữ liệu

3.3.1.1. Thư Viện Chương Trình

- Thư viện hệ thống OS
- Thư viện toán học numpy
- Thư viện openCV
- Thư viện PIL (python imaging library)
- Thư viện xử lý dữ liệu QueryDB
- Thư viện kết nối CSDL

3.3.1.2. Xây dựng, kết nối dữ liệu và xử lý dữ liệu trên database

- Xây dựng CSDL



Hình 3.5. Xây dựng cơ sở dữ liệu

- Khai báo thư viện cần thiết

```
import mysql.connector  
import datetime
```

Hình 3.6. Các thư viện sử dụng

- Hàm kết nối dữ liệu

```
def connectdatabase():  
    connection=mysql.connector.connect(  
        host="localhost",  
        user="root",  
        database="face",  
        password=""  
    )  
    return connection
```

- Hàm thêm và update dữ liệu

```
def insertOrUpdate(id, name):  
    con = connectdatabase()  
    query = "select * from people where id =" + str(id)  
    cursor = con.cursor()  
    cursor.execute(query)  
    records = cursor.fetchall()  
    isReccorExit = 0  
    for row in records:  
        isReccorExit = 1  
    if isReccorExit == 0:  
        query = "insert into people (id,name) values (%s,%s)"  
        cursor.execute(query, (id, name))  
    else:  
        query = "update people set name = %s where id = %s"  
        cursor.execute(query, (name, id))  
    con.commit()  
    con.close()  
    cursor.close()
```


- Tìm kiếm theo ID của user

```
def getProfile(id):
    conn = connectdatabase()
    cmd = "select * from people where id = " + str(id)
    cursor = conn.cursor()
    cursor.execute(cmd)
    records = cursor.fetchall()
    profile = None
    for row in records:
        profile = row
    conn.close()
    return profile
```

3.3.1.3. Nhập dữ liệu và ảnh

- Các thư viện cần thiết

```
import cv2
import numpy as np
import os
import queryDB as db
```

- Nhập dữ liệu và chèn lên database

```
# nhập id và name người mình muốn add mặt vào
id = input("Nhập Id :")
name = input("Nhập Name :")
# gọi đến hàm insert vào db để lưu trữ thông tin
db.insertOrUpdate(id, name)
```

- Khởi tạo webcam và lấy ảnh

```
sampleNum = 0
while True:
    ret, img = cam.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = detector.detectMultiScale(gray, 1.3, 5)
    for (x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        if not os.path.exists("dataSet"):
            os.makedirs("dataSet")
        sampleNum = sampleNum + 1
        cv2.imwrite(
            "dataSet/User." + id + "." + str(sampleNum) +
            ".jpg",
            gray[y : y + h, x : x + w],
        )
        cv2.imshow("frame", img)
        # nhấn phím q để kết thúc chương trình
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break
        elif sampleNum > 100:
            break
    cam.release()
    cv2.destroyAllWindows()
```

3.3.1.4. Nhận diện khuôn mặt

- Khai báo thư viện cần thiết

```
import cv2
import numpy as np
from PIL import Image
import queryDB as db
from gtts import gTTS
import os
```

- Khởi tạo webcam

```
# Khởi tạo webcam và đặt độ phân giải của nó thành 900x720
cam = cv2.VideoCapture(0)
cam.set(3, 900)
cam.set(4, 720)
# Khởi tạo một đối tượng CascadeClassifier trong thư viện
OpenCV
# với tệp tin XML chứa thông tin về mô hình Cascade để phát
hiện khuôn mặt trên hình ảnh đầu vào
face_cascade =
cv2.CascadeClassifier("libs/haarcascade_frontalface_default.x
ml")
# Tạo ra một đối tượng nhận dạng khuôn mặt bằng thuật toán
LBPH
recognizer = cv2.face.LBPHFaceRecognizer_create()
# Dùng để đọc dữ liệu được huấn luyện từ tệp
recognizer.read("recognizer/trainingData.yml")
fontface = cv2.FONT_HERSHEY_SIMPLEX
```

- Nhận diện

```
while True:
    ret, frame = cam.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray)

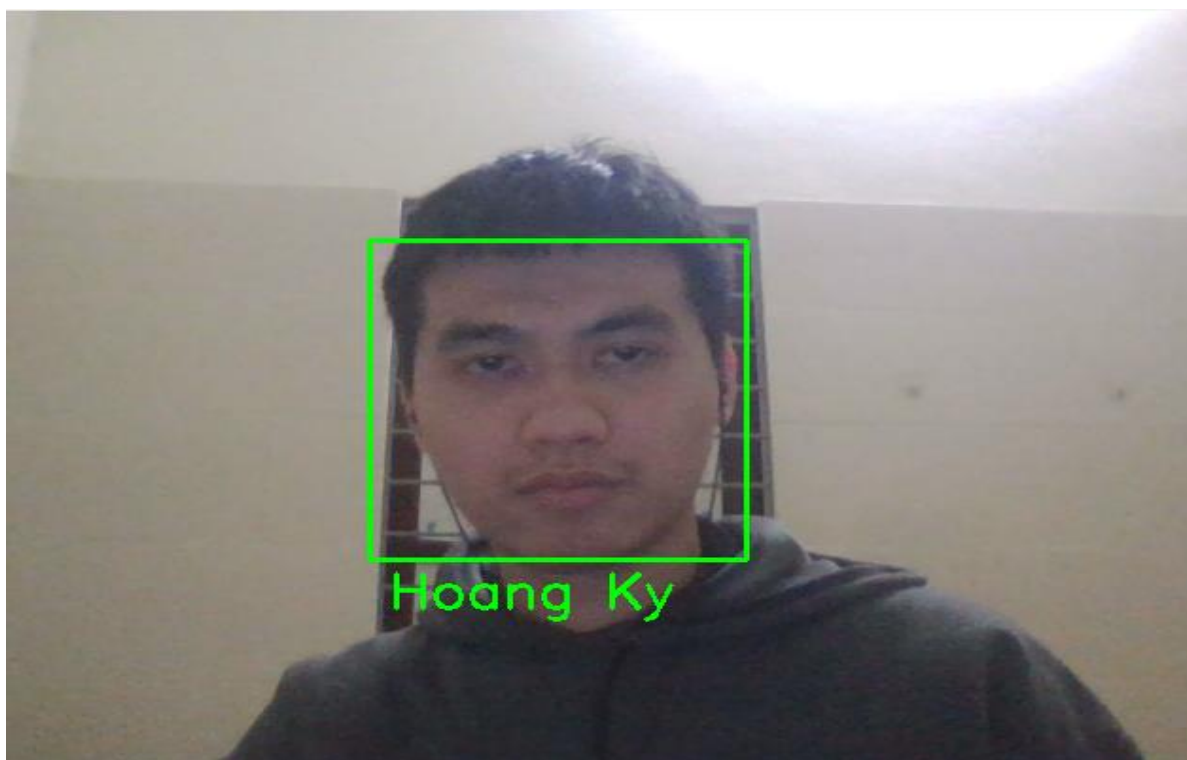
    for (x, y, w, h) in faces:
        # Vẽ một hình chữ nhật xung quanh khuôn mặt được phát hiện
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        roi_gray = gray[y : y + h, x : x + w]
        id, confidence = recognizer.predict(roi_gray)
        if confidence < 40:
            profile = db.getProfile(id)
            # Nếu profile mà không bằng rỗng thì hiển thị tên
            if profile is not None:
                # Hiện thị ra tên trong khung cam
                cv2.putText(
                    frame,
                    "" + str(profile[1]),
                    (x + 10, y + h + 30),
                    fontface,
                    1,
                    (0, 255, 0),
                    2,
                )
            else:
                cv2.putText(
                    frame, "Unknown", (x + 10, y + h + 30), fontface,
                    1, (0, 0, 255), 2
                )
            cv2.imshow("Nhận Diện Khuôn Mặt", frame)
            if cv2.waitKey(1) == ord("q"):
                break

    print("\nThoát...")
    cam.release()
    cv2.destroyAllWindows()
```

3.3.2. Huấn luyện mô hình

3.4. Kết quả

- Test với webcam



Hình 3.7. Kết quả thử nghiệm với webcam

CHƯƠNG 4. KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

4.1. Kết luận

4.1.1. Tính khả thi

Sau thời gian tìm hiểu và nghiên cứu, nhóm chúng em đã xây dựng thành công chương trình “Nhận diện khuôn mặt qua ảnh bằng thuật toán LBPH”. Chương trình đã bước đầu đạt được mục tiêu đề ra, đó là áp dụng thuật toán LBPH vào bài toán nhận diện khuôn mặt. Tuy nhiên, nhóm nhận thấy tính khả thi của chương trình khi áp dụng vào thực tế vẫn còn hạn chế. Một số vấn đề đáng lưu ý bao gồm độ chính xác chưa cao, khả năng nhận diện bị ảnh hưởng bởi điều kiện ánh sáng, góc chụp, và đặc biệt là dữ liệu huấn luyện chưa đủ lớn. Để cải thiện, chương trình cần được tối ưu hóa và bổ sung thêm dữ liệu nhằm tăng độ chính xác và ổn định.

4.1.2. Thuận lợi

Trong quá trình thực hiện, nhóm chúng em đã gặp nhiều thuận lợi đáng kể:

- ✓ **Nguồn tài liệu phong phú:** Có rất nhiều tài liệu chuyên ngành từ giáo trình, bài báo khoa học đến các hướng dẫn trực tuyến, giúp nhóm dễ dàng tham khảo và học tập.
- ✓ **Sự hỗ trợ từ giảng viên:** Giảng viên đã nhiệt tình góp ý, hướng dẫn từng bước, đặc biệt trong việc xử lý các vấn đề kỹ thuật cũng như định hướng nghiên cứu.
- ✓ **Đội ngũ phát triển nhiệt huyết:** Các thành viên trong nhóm luôn cố gắng học hỏi, tìm hiểu, và hỗ trợ lẫn nhau để hoàn thành tốt nhất nhiệm vụ được giao.

4.1.3. Khó khăn

Tuy nhiên, nhóm cũng đối mặt với không ít khó khăn:

- ✓ **Kiến thức chuyên sâu:** Đề tài đòi hỏi sự hiểu biết sâu về Trí tuệ nhân tạo, xử lý hình ảnh, và thuật toán LBPH. Với kinh nghiệm còn hạn chế, nhóm mất nhiều thời gian để nghiên cứu và áp dụng các kiến thức này vào thực tế.
- ✓ **Sự cố kỹ thuật:** Trong quá trình chạy mã, nhóm thường xuyên gặp phải các lỗi liên quan đến đầu vào, đầu ra, hoặc sự không tương thích của các thư viện. Điều này ảnh hưởng đáng kể đến tiến độ thực hiện đề tài.
- ✓ **Khó khăn trong phối hợp nhóm:** Do lịch học và lịch làm việc khác nhau, các thành viên gặp nhiều trở ngại trong việc sắp xếp thời gian để họp nhóm, trao đổi công việc, và cùng xử lý các vấn đề phát sinh.

4.2. Bài học kinh nghiệm

- ✓ Trong quá trình thực hiện đề tài, nhóm chúng em đã học được rất nhiều kiến thức chuyên ngành cũng như kỹ năng mềm.
- ✓ Biết thêm những kỹ thuật trong Trí tuệ nhân tạo.

4.2.1. Kiến thức

- ✓ **Xây dựng và huấn luyện mô hình:** Nhóm đã hiểu rõ hơn về quy trình xây dựng, huấn luyện và đánh giá mô hình học máy. Đây là kiến thức nền tảng quan trọng cho việc phát triển các ứng dụng AI sau này.
- ✓ **Thuật toán LBPH:** Nhóm đã tìm hiểu sâu về cách hoạt động của thuật toán LBPH (Local Binary Pattern Histograms), đặc biệt là khả năng phân loại dữ liệu và ứng dụng trong bài toán nhận diện khuôn mặt. Qua đó, nhóm hiểu thêm về cách tối ưu hóa các tham số trong thuật toán để đạt hiệu suất tốt nhất.
- ✓ **Xử lý hình ảnh với OpenCV:** Nhóm học được cách sử dụng OpenCV để xử lý hình ảnh, từ đó nâng cao khả năng phân tích dữ liệu trực quan và làm sạch dữ liệu đầu vào.

4.2.2. Kỹ năng

- ✓ **Làm việc nhóm:** Kỹ năng phân chia công việc, hợp tác và hỗ trợ lẫn nhau trong nhóm được cải thiện đáng kể.
- ✓ **Giao tiếp:** Việc thuyết trình, trình bày ý tưởng và báo cáo tiến độ giúp nhóm trau dồi khả năng giao tiếp hiệu quả.
- ✓ **Quản lý thời gian:** Dù gặp nhiều khó khăn về lịch trình, nhóm đã biết cách sắp xếp thời gian hợp lý để hoàn thành các mục tiêu đúng tiến độ.
- ✓ **Sử dụng công cụ hỗ trợ:** Việc sử dụng thành thạo các công cụ như Word, PowerPoint, Excel giúp nhóm trình bày nội dung và quản lý dự án một cách chuyên nghiệp.

KẾT LUẬN

Trong bài báo cáo này, chúng em đã đề xuất một mô hình nhận diện khuôn mặt con người dựa trên thuật toán LBPH (Local Binary Patterns Histogram). Mô hình này được thiết kế với độ phức tạp vừa phải, tận dụng khả năng phát hiện và nhận diện khuôn mặt một cách nhanh chóng và chính xác. Đặc điểm nổi bật của mô hình là khả năng hoạt động hiệu quả trên các hệ thống có cấu hình trung bình, giúp giảm thiểu yêu cầu phần cứng mà vẫn đảm bảo độ chính xác ở mức chấp nhận được.

Ngoài ra, mô hình còn được tích hợp khả năng quản lý dữ liệu thông qua cơ sở dữ liệu MySQL, hỗ trợ các tính năng như lưu trữ thông tin người dùng và ghi nhận điểm danh (check-in, check-out). Điều này chứng tỏ tiềm năng ứng dụng cao trong thực tiễn, đặc biệt là trong các lĩnh vực như kiểm soát ra vào, điểm danh tự động, và quản lý nhân sự.

Tuy nhiên, để hoàn thiện hơn, mô hình cần được mở rộng với khả năng xử lý trong điều kiện ánh sáng phức tạp, đa góc nhìn, và quy mô dữ liệu lớn hơn. Chúng em tin rằng mô hình này không chỉ là một giải pháp khả thi trong hiện tại mà còn là nền tảng để phát triển các hệ thống khác tốt hơn.

TÀI LIỆU THAM KHẢO

- [1] Amazon Web Services (AWS), "What is Python?", *Amazon.com*. [Online]. Available: <https://aws.amazon.com/vi/what-is/python/>. [Accessed: Nov. 30, 2024].
- [2] Medium, "Face Recognition: How LBPH works", *Towards Data Science*. [Online]. Available: <https://medium.com/m/global-identity-2?redirectUrl=https%3A%2F%2Ftowardsdatascience.com%2Fface-recognition-how-lbph-works-90ec258c3d6b>. [Accessed: Nov. 30, 2024].
- [3] 200lab.io, "Visual Studio Code là gì? Hướng dẫn sử dụng VSCode cho người mới", *200lab.io*. [Online]. Available: <https://200lab.io/blog/visual-studio-code-la-gi-huong-dan-su-dung-vscode-cho-nguoi-moi/>. [Accessed: Dec. 4, 2024].
- [4] FPT Shop, "OpenCV là gì?", *FPT Shop*. [Online]. Available: <https://fptshop.com.vn/tin-tuc/danh-gia/opencv-la-gi-168801>. [Accessed: Nov. 30, 2024].
- [5] Khoa Điện Tử, Đại học Duy Tân, "Nghiên cứu khoa học: Thuật toán LBPH", *Khoa Điện Tử, Đại học Duy Tân*. [Online]. Available: <https://kdientu.duytan.edu.vn/nghien-cuu-khoa-hoc/thuat-toan-lbph/>. [Accessed: Dec. 4, 2024].
- [6] "Video: kuR7VC9zyZc," *YouTube*. [Online]. Available: <https://youtu.be/kuR7VC9zyZc?si=T1MV-wAYv96l5Ldf>. [Accessed: Dec. 5, 2024].