

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP

ỨNG DỤNG KHÓA CỬA THÔNG MINH NHẬN DẠNG KHUÔN MẶT NGƯỜI

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **ĐOÀN HOÀNG PHÚC**

MSSV : 14119042

Sinh viên: **PHẠM THANH TRƯỜNG**

MSSV : 14119059

TP. HỒ CHÍ MINH – 06/2019

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN ĐIỆN TỬ

BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN TỐT NGHIỆP

**ỨNG DỤNG KHÓA CỬA THÔNG MINH
NHẬN DẠNG KHUÔN MẶT NGƯỜI**

NGÀNH CÔNG NGHỆ KỸ THUẬT MÁY TÍNH

Sinh viên: **ĐOÀN HOÀNG PHÚC**

MSSV : 14119042

Sinh viên: **PHẠM THANH TRƯỜNG**

MSSV : 14119059

Hướng dẫn: **ThS. LÊ MINH**

TP. HỒ CHÍ MINH – 06/2019

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

KHOA ĐIỆN ĐIỆN TỬ

BẢN NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(Dành cho giảng viên hướng dẫn)

Đề tài: ƯNG DỤNG KHOA CẢM, TÍNH MINH NHÂN DẠNG
LAUÔNG MẮT NGƯỜI VÀ CẢNH BÁO NGƯỜI LẠ QUA EMAIL
Sinh viên: + Đoàn Hoàng Thu MSSV: 141190
+ Phạm Thanh Tường MSSV: 1414059
Hướng dẫn: TRẦN LÊ MINH

Nhận xét bao gồm các nội dung sau đây:

1. Tính hợp lý trong cách đặt vấn đề và giải quyết vấn đề; ý nghĩa khoa học và thực tiễn:

Đặt vấn đề rõ ràng, mục tiêu cụ thể; đề tài có tính mới, cấp thiết; đề tài có khả năng ứng dụng, tính sáng tạo.

2. Phương pháp thực hiện/ phân tích/ thiết kế:

Phương pháp hợp lý và tin cậy dựa trên cơ sở lý thuyết; có phân tích và đánh giá phù hợp; có tính mới và tính sáng tạo.

3. Kết quả thực hiện/ phân tích và đánh giá kết quả/ kiểm định thiết kế:

Phù hợp với mục tiêu đề tài: phân tích và đánh giá/ kiểm thử thiết kế hợp lý; có tính sáng tạo/ kiểm định chặt chẽ và đảm bảo độ tin cậy.

4. Kết luận và đề xuất:

Kết luận phù hợp với cách đặt vấn đề, đề xuất mang tính cải tiến và thực tiễn; kết luận có đóng góp mới mẻ, đề xuất sáng tạo và thuyết phục.

5. Hình thức trình bày và bố cục báo cáo:

Văn phong nhất quán, bố cục hợp lý, cấu trúc rõ ràng, dùng định dạng mẫu; có tính hấp dẫn, thể hiện năng lực tốt, văn bản trau chuốt.

6. Kỹ năng chuyên nghiệp và tính sáng tạo:

Thể hiện các kỹ năng giao tiếp, kỹ năng làm việc nhóm, và các kỹ năng chuyên nghiệp khác trong việc thực hiện đề tài.

7. Tài liệu trích dẫn

Tính trung thực trong việc trích dẫn tài liệu tham khảo; tính phù hợp của các tài liệu trích dẫn; trích dẫn theo đúng chỉ dẫn APA.

8. Đánh giá về sự trùng lặp của đề tài

Cần khẳng định đề tài có trùng lặp hay không? Nếu có, đề nghị ghi rõ mức độ, tên đề tài, nơi công bố, năm công bố của đề tài đã công bố.

9. Những nhược điểm và thiếu sót, những điểm cần được bổ sung và chỉnh sửa*

10. Nhận xét tinh thần, thái độ học tập, nghiên cứu của sinh viên

Đề nghị của giảng viên hướng dẫn

Ghi rõ: "Báo cáo đạt/ không đạt yêu cầu của một khóa luận tốt nghiệp kỹ sư, và được phép/ không được phép bảo vệ khóa luận tốt nghiệp"

Được phép bảo vệ

Tp. HCM, ngày 6 tháng 6 năm 2019

Người nhận xét
(Ký và ghi rõ họ tên)

Trần Lê Minh

LỜI CẢM ƠN

Trong quá trình thực hiện đồ án, nhóm chúng tôi đã nhận được sự giúp đỡ của rất nhiều người.

Trước tiên, nhóm xin gửi lời cảm ơn chân thành tới thầy giáo **Ths. Lê Minh** đã hướng dẫn, chỉ bảo tận tình giúp đỡ trong suốt quá trình nghiên cứu và xây dựng đồ án.

Đồng thời, nhóm xin gửi lời cảm ơn tới thầy cô trong Khoa Điện – Điện tử đã tận tình giảng dạy, truyền đạt cho nhóm những kiến thức quý báu trong suốt những năm học qua, giúp nhóm có thêm nhiều kiến thức và kỹ năng để vận dụng vào thực hiện đồ án tốt nghiệp.

Nhóm cũng gửi lời cảm ơn đến sự giúp đỡ của các bạn bè trong và ngoài lớp đã tận tình giúp đỡ trong thời gian thực hiện luận văn.

TÓM TẮT

Những năm trở lại đây, nhận dạng khuôn mặt nói riêng và các vấn đề về nhận dạng dựa trên trắc sinh học của con người ngày càng được chú trọng và phát triển một cách mạnh mẽ, ứng dụng hứa hẹn đạt được nhiều thành công trong lĩnh vực xử lý ảnh.

Hệ thống nhận dạng trắc sinh học được ứng dụng nhiều trong các lĩnh vực như hệ thống an ninh và bảo mật, các dịch vụ thanh toán trực tuyến.... Trong đề tài này, lĩnh vực nhận dạng khuôn mặt người sẽ được áp dụng. Với mục đích nhận dạng khuôn mặt người từ bức ảnh được chụp đưa vào, xác định vị trí khuôn mặt, so sánh với các khuôn mặt mẫu trong CSDL với thuật toán Local Binary Pattern. Sử dụng ngôn ngữ lập trình Python, thư viện tạo app giao diện người dùng Tkinter, thư viện Opencv. Sau đó xây dựng nên mô hình căn nhà để mở khóa cửa tự động sử dụng nhận dạng khuôn mặt người.

MỤC LỤC

DANH MỤC HÌNH	I
CÁC TỪ VIẾT TẮT.....	II
CHƯƠNG 1 GIỚI THIỆU.....	1
1.1 GIỚI THIỆU.....	1
1.2 MỤC TIÊU ĐỀ TÀI.....	2
1.3 GIỚI HẠN ĐỀ TÀI.....	2
1.4 PHƯƠNG PHÁP NGHIÊN CỨU	3
1.5 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	3
1.6 BỐ CỤC QUYỀN BÁO CÁO	5
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	6
2.1 TÌNH HÌNH NGHIÊN CỨU.....	6
2.2 CÁC HƯỚNG TIẾP CẬN ĐỂ XÁC ĐỊNH MẶT NGƯỜI.....	8
2.3 PHƯƠNG PHÁP PHÁT HIỆN KHUÔN MẶT VỚI HAAR-LIKE	
ADABOOST	9
2.3.1 ĐẶC TRƯNG HAAR-LIKE	9
2.3.2 ADABOOST	13
2.3.2 ÁP DỤNG ADABOOST CHO BÀI TOÁN PHÁT HIỆN	
KHUÔN MẶT.....	13
2.3.4 SƠ ĐỒ KHỐI QUÁ TRÌNH TÌM KIẾM KHUÔN MẶT	
NGƯỜI.....	17

2.4 THUẬT TOÁN TRÍCH RÚT ĐẶC TRƯNG LBP (LOCAL BINARY PATTERN).....	18
2.4.1 LÝ THUYẾT VỀ LBP.....	18
2.4.2 THUẬT TOÁN LBP:.....	19
2.4.3 VI DỤ VỀ TRÍCH RÚT ĐẶC TRƯNG LBP TRÊN ẢNH SỐ .	21
2.4.4 LBPH – MỞ RỘNG TỪ THUẬT TOÁN LBP.	23
2.5 NHẬN DẠNG KHUÔN MẶT.....	24
CHƯƠNG 3 THIẾT KẾ HỆ THỐNG.....	26
3.1 MÔ HÌNH HỆ THỐNG	26
3.2 THIẾT KẾ PHẦN CỨNG.....	27
3.2.1 CHỨC NĂNG CỦA PHẦN CỨNG.....	27
3.2.2 SƠ ĐỒ KHỐI PHẦN CỨNG	28
3.2.3 LỰA CHỌN PHẦN CỨNG	32
3.2.4 KẾT NỐI PHẦN CỨNG	32
3.3 THIẾT KẾ PHẦN MỀM.....	34
3.3.1 CHỨC NĂNG HOẠT ĐỘNG CỦA PHẦN MỀM	34
3.3.2 QUY TRÌNH XỬ LÝ CỦA HỆ THỐNG.....	34
3.3.3 LƯU ĐỒ HOẠT ĐỘNG.....	34
CHƯƠNG 4 KẾT QUẢ	41
4.1 MÔ HÌNH CỦA HỆ THỐNG.....	41
4.2 HOẠT ĐỘNG CỦA HỆ THỐNG.....	43
4.3 ĐÁNH GIÁ HOẠT ĐỘNG.....	44
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	45
5.1 KẾT LUẬN.....	45

5.2	HƯỚNG PHÁT TRIỂN.....	46
	TÀI LIỆU THAM KHẢO.....	47

DANH MỤC HÌNH

Hình 2.1: Nguồn gốc đặc tính sinh trắc của con người.....	6
Hình 2.2: Đặc trưng Haar-like cơ bản	9
Hình 2.3: Đặc trưng Haar-like mở rộng	10
Hình 2.4: Công thức Integral Image của ảnh.....	12
Hình 2.5: Công thức tính mức xám của vùng D.....	13
Hình 2.6 : Cascade của các bộ phân lớp mạnh. Nơi một vùng ứng viên có thể bị loại tại bất kì giai đoạn nào	15
Hình 2.7: Quá trình hình thành strong classifiers	16
Hình 2.8: Sơ đồ khối quá trình tìm kiếm khuôn mặt người.....	17
Hình 2.9: Cách tính LBP và độ tương phản cục bộ C.	19
Hình 2.10: Minh họa toán tử LBP mở rộng	20
Hình 2.11: Các biến thể của LBP: LBP đồng dạng.	21
Hình 2.12 : Ví dụ về quá trình tính toán đặc trưng LBP	22
Hình 2.13 : Công thức tính độ tương tự giữa hai histograms	24
Hình 2.14: Mô tả các đặc trưng cục bộ của một khuôn mặt với LBPH	25
Hình 3.1: Sơ đồ mô hình hệ thống.....	26
Hình 3.2: Sơ đồ khối phần cứng của hệ thống.....	27
Hình 3.3: Raspberr Pi Camera Module V2.....	28
Hình 3.4: Màn hình cảm ứng 7inch	30
Hình 3.5: Raspberry Pi 3 Model B	30
Hình 3.6: Servo MG996R.....	32
Hình 3.7: Module relay 1 kênh.	32
Hình 3.8: Cách kết nối phần cứng.	33
Hình 3.9: Quy trình của hệ thống	34
Hình 3.10: Lưu đồ nhận diện khuôn mặt.....	35
Hình 3.11: Lưu đồ huấn luyện dữ liệu gốc.....	36
Hình 3.12: Lưu đồ nhận diện khuôn mặt.....	37

Hình 3.13: Lưu đồ giải thuật khối xử lý	38
Hình 4.1: Mô hình của hệ thống	41
Hình 4.2: Màn hình hiển thị.....	42
Hình 4.3: App hiển thị điều khiển	43

CÁC TỪ VIẾT TẮT

LPB	Local Binary Patterns
LBPH	Local Binary Patterns Histogram
CSDL	Cơ sở dữ liệu
AI	Artificial Intelligence
SF	Split faces
FE	Featured Excerpt

Chương 1

GIỚI THIỆU

1.1 GIỚI THIỆU

Nhận dạng mặt người (Face recognition) là một lĩnh vực nghiên cứu của ngành Computer Vision, và cũng được xem là một lĩnh vực nghiên cứu của ngành Biometrics (tương tự như nhận dạng vân tay – Fingerprint recognition, hay nhận dạng mống mắt – Iris recognition). Xét về nguyên tắc chung, nhận dạng mặt có sự tương đồng rất lớn với nhận dạng vân tay và nhận dạng mống mắt, tuy nhiên sự khác biệt nằm ở bước trích chọn đặc trưng (feature extraction) của mỗi lĩnh vực.

Trong khi nhận dạng vân tay và mống mắt đã đạt tới độ chín, tức là có thể áp dụng trên thực tế một cách rộng rãi thì nhận dạng mặt người vẫn còn nhiều thách thức và vẫn là một lĩnh vực nghiên cứu thú vị với nhiều người. So với nhận dạng vân tay và mống mắt, nhận dạng mặt có nguồn dữ liệu phong phú hơn (bạn có thể nhìn thấy mặt người ở bất cứ tấm ảnh, video clip nào liên quan tới con người trên mạng) và ít đòi hỏi sự tương tác có kiểm soát hơn (để thực hiện nhận dạng vân tay hay mống mắt, dữ liệu input lấy từ con người đòi hỏi có sự hợp tác trong môi trường có kiểm soát).

Hiện nay các phương pháp nhận dạng mặt được chia thành nhiều hướng theo các tiêu chí khác nhau: nhận dạng với dữ liệu đầu vào là ảnh tĩnh 2D (still image based FR) là phổ biến nhất, 3D FR (vì việc bố trí nhiều camera 2D sẽ cho dữ liệu 3D và đem lại kết quả tốt hơn, đáng tin cậy hơn), cũng có thể chia thành 2 hướng là: làm với dữ liệu ảnh và làm với dữ liệu video.

Trên thực tế người ta hay chia các phương pháp nhận dạng mặt ra làm 3 loại: phương pháp tiếp cận toàn cục (global, như Eigenfaces-PCA, Fisherfaces-LDA), phương pháp tiếp cận dựa trên các đặc điểm cục bộ (local feature based, như LBP, Gabor wavelets) và phương pháp lai (hybrid, là sự kết hợp của hai phương pháp toàn cục và local feature). Phương pháp dựa trên các đặc điểm cục bộ đã được chứng minh là ưu việt hơn khi làm việc trong các điều kiện không có kiểm

soát và có thể nói rằng lịch sử phát triển của nhận dạng mặt là sự phát triển của các phương pháp trích chọn đặc trưng (feature extraction methods) được sử dụng trong các hệ thống dựa trên các cơ sở tính năng.

Nó có thể giải quyết các bài toán như giám sát tự động phục vụ trong cơ quan, ngân hàng, kho bạc, ... hoặc trong việc giám sát giao thông tự động phục vụ tại bãi đỗ xe, trạm thu phí tự động hoặc việc phát hiện và nhận dạng mặt người phục vụ trong công tác quân sự, an ninh v.v...

Nhận dạng khuôn mặt không phải là bài toán mới nhưng nó vẫn là một thách thức lớn vì một bài toán nhận dạng mặt người chứa nhiều các bài toán khác như: phát hiện mặt người, đánh dấu, rút trích đặc trưng, gán nhãn, phân lớp. Ngoài ra, ảnh khuôn mặt trong thực tế chứa đựng nhiều vấn đề như: độ sáng, độ nhòe độ mờ, độ nhiễu, độ phân giải, góc ảnh, ...

Luận văn chọn đặc trưng Local Binary Pattern (LBP) là một đặc trưng rất phổ biến trong các bài toán liên quan đến ảnh khuôn mặt nói chung và nhận dạng khuôn mặt nói riêng. tác giả chọn đề tài nghiên cứu: “Ứng dụng khóa cửa thông minh nhận dạng khuôn mặt người”.

1.2 MỤC TIÊU ĐỀ TÀI

Đề tài: “Ứng dụng khóa cửa thông minh nhận dạng khuôn mặt người” được thực hiện với mục tiêu ban đầu gồm:

- Mở cửa bằng cách nhận diện khuôn mặt người có chứa trong CSDL.
- Có thể cập nhật thêm người nhận dạng vào CSDL.
- Điều khiển các chức năng trên màn hình cảm ứng điện dung.

1.3 GIỚI HẠN ĐỀ TÀI

Đề tài: “Ứng dụng khóa cửa thông minh nhận dạng khuôn mặt người” được thực hiện với mục tiêu ban đầu gồm:

- Mở cửa bằng cách nhận diện khuôn mặt người có chứa trong CSDL.
- Có thể cập nhật thêm người nhận dạng vào CSDL.
- Điều khiển các chức năng trên màn hình cảm ứng điện dung.

Do tính chất của các đặc trưng và thuật toán nhận diện khuôn mặt quá nhiều và khó nên đề tài này chỉ tập trung vào phần ứng dụng phát triển thực tế.

Một số khó khăn của bài toán nhận diện khuôn mặt: Góc chụp, các thành phần xuất hiện trên khuôn mặt, độ sáng của ảnh, ... vì thế trong luận văn này nhóm đưa ra các ràng buộc và giả định sau để làm giảm độ phức tạp của bài toán nhận diện khuôn mặt:

- Khuôn mặt nhận diện cần nhìn thẳng vào camera hoặc góc nghiêng không quá rộng (nhỏ hơn 20°).
- Với camera có độ phân giải kém cùng với ánh sáng yếu sẽ làm cho việc nhận dạng khuôn mặt không thực hiện ổn định.
- Khoảng cách người nhận diện đến camera trong khoảng 30cm đến 60cm.

1.4 PHƯƠNG PHÁP NGHIÊN CỨU

- Tìm hiểu phương pháp phát hiện khuôn mặt Thuật Toán Viola-Jones và phương pháp rút trích đặc trưng khuôn mặt Local Binary Pattern Histograms (LBPH).
- Tìm hiểu các lý thuyết có liên quan như ngôn ngữ Python, Raspberry Pi, thư viện tạo app Tkinter và thư viện OpenCV.
- Phương pháp thực nghiệm: xây dựng chương trình nhận diện khuôn mặt trên máy tính nhúng Raspberry Pi để có thể điều khiển mở cửa thông minh tự động và cảnh báo với những khuôn mặt lạ không nhận dạng được. Hiện thị những khuôn mặt vừa được nhận dạng lên màn hình.

1.5 ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

Đối tượng nghiên cứu:

- Các phương pháp, giải thuật phát hiện và nhận dạng khuôn mặt người trên ảnh.
- Máy tính nhúng Raspberry Pi, camera pi, màn hình cảm ứng điện dung 7inch, relay điều khiển bật tắt đèn và đèn dùng để cung cấp ánh sáng cần thiết khi nhận dạng.
- Ngôn ngữ lập trình Python, thư viện Tkinter, thư viện OpenCV.

Phạm vi nghiên cứu:

- Để xử lý và nhận diện khuôn mặt thì cần phải thỏa các điều kiện sau đây :
 - Ánh sáng đều, không có chiều sáng, không có ánh sáng mạnh hoặc trong môi trường thiếu ánh sáng.
 - Góc ảnh: Trục diện với camera hoặc nghiêng không quá 1 góc 20 độ.
 - Khuôn mặt không bị quá tối hoặc che khuất.
 - Ảnh có chất lượng tốt.
 - Khoảng cách người nhận diện đến camera trong khoảng 30cm – 60cm.

1.6 BỐ CỤC QUYỀN BẢO CÁO

Nội dung chính của đề tài được trình bày với năm chương như sau:

Chương 1: Giới thiệu

Trong chương này sẽ giới thiệu sơ lược về đề tài trong thực tế hiện nay, quá trình nghiên cứu đề tài, lý do và mục tiêu chọn đề tài, đối tượng và phạm vi nghiên cứu của đề tài, phương pháp nghiên cứu và giới hạn của đề tài.

Chương 2: Cơ sở lý thuyết

Bao gồm các lý thuyết về các thuật toán nhận dạng khuôn mặt người và xử lý hình ảnh.

Chương 3: Thiết kế hệ thống

Trong chương này mục đích là thiết kế phần cứng và xây dựng phần mềm cho hệ thống với những yêu cầu đặt ra:

- Thiết kế phần cứng: Từ sơ đồ khối tổng quát và sơ đồ khối chi tiết để tiến hành lựa chọn các linh kiện cho các khối.
- Thiết kế phần mềm : Xây dựng chương trình nhận diện khuôn mặt và xuất tín hiệu điều khiển trên Raspberry Pi sử dụng ngôn ngữ lập trình Python, thư viện Tkinter và thư viện OpenCV.

Chương 4: Kết quả

Trình bày về kết quả thi công phần cứng, phần mềm và đánh giá ưu nhược điểm của hệ thống.

Chương 5: Kết luận và hướng phát triển

Đưa ra các kết luận về những vấn đề mà trong quá trình nghiên cứu đã đạt được.

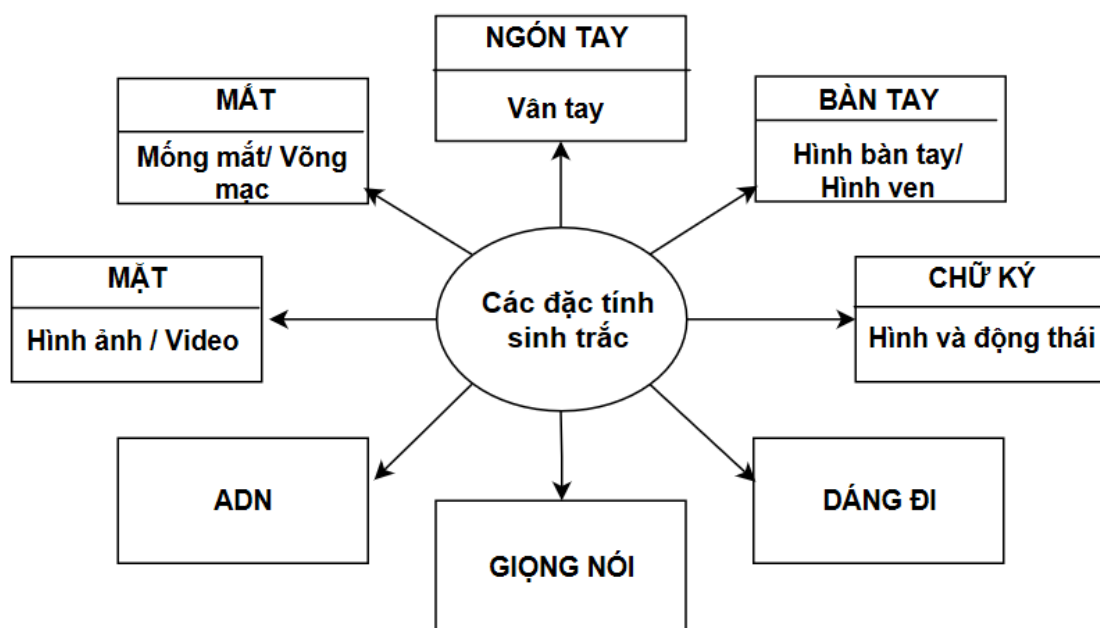
Chương 2

CƠ SỞ LÝ THUYẾT

2.1 Khái quát về nhận dạng mặt người

2.1.1 Phân tích các đặc tính trắc sinh học của người

Đặc tính sinh trắc của người là đặc tính đo được các nét hành vi riêng của con người. Nhận diện được thực hiện trong quá trình kiểm tra tính đồng nhất của đặc tính sinh trắc giữa người cần kiểm tra với đặc tính tương tự của người đã được lưu sẵn trong CSDL. Có thể nhận diện người dựa trên các đặc điểm sinh trắc tĩnh trên người như khuôn mặt, mắt, vân tay, bàn tay, gen... hay các đặc trưng hành vi như dáng đi, chữ viết, giọng nói...



Hình 2.1: Nguồn gốc đặc tính sinh trắc của con người

Dữ liệu sinh trắc lý tưởng của người cần phải có những đặc tính cơ bản như :

- *Tính tổng quát* : Mỗi người đều được thể hiện bởi đặc tính này.
- *Tính duy nhất* : Không thể tồn tại 2 người có đặc tính giống nhau.
- *Tính thường xuyên* : Là sự độc lập của đặc tính đối với thời gian.

- *Tính thu thập được* : Là đặc tính được thu thập một cách tương đối đơn giản và nhanh chóng từ mỗi cá nhân và có thể được chi tiết hóa.

2.1.2 Hệ thống nhận dạng tổng quát

Nhiệm vụ của hệ thống nhận dạng mặt cười là xử lý tự động thông tin từ các ảnh để tìm ra độ tương tự giữa các khuôn mặt và đưa ra quyết định về tính đồng nhất giữa chúng. Cấu trúc của hệ thống nhận dạng mặt người được thể hiện qua các bước sau :

- Tiền xử lý (Pre-Processing): Chức năng này để chuẩn hóa ảnh cần tìm giúp cho việc tìm kiếm được hiệu quả hơn. Các công việc trong bước tiền xử lý có thể là : Chuẩn hóa kích cỡ giữa ảnh trong CSDL và ảnh cần tìm, hiệu chỉnh độ sáng tối của ảnh, lọc nhiễu, chuẩn hóa về vị trí, tư thế ảnh mặt.
- Tách khuôn mặt (SF): Chức năng này làm nhiệm vụ xác định vị trí, kích cỡ của một hoặc nhiều khuôn mặt trên ảnh chụp từ đó tách ra phần mặt. Phần ảnh mặt được tách ra thường nhỏ hơn nhiều so với ảnh chụp ban đầu, nó sẽ là các khuôn mặt cần tìm và chức năng trích chọn đặc trưng sẽ sử dụng các ảnh được tách ra này.
- Trích chọn đặc trưng(FE): Tìm ra các đặc trưng chính của ảnh mặt, các đặc trưng này dựa trên các điểm ảnh được phân nhỏ 3x3 pixel , đối với mỗi pixel trong ảnh đều có tỉ lệ xám sẽ được sử dụng để đối sánh sự giống nhau giữa ảnh mặt cần tìm với ảnh mặt trong CSDL.
- Đối sánh (Comparison): Thực hiện việc so sánh và tìm ra khoảng cách ngắn nhất giữa các biểu đồ có trong ảnh đầu vào với ảnh trong tập huấn luyện.

2.2 Các hướng tiếp cận để xác định mặt người

Dựa vào tính chất của các phương pháp xác định mặt người trên ảnh, các phương pháp này được chia thành bốn loại chính, tương ứng với bốn hướng tiếp cận khác nhau [9]:

- Hướng tiếp cận dựa trên tri thức: mã hoá hiểu biết của con người về các loại khuôn mặt và tạo ra các tập luật để xác định khuôn mặt.
- Hướng tiếp cận dựa trên những đặc trưng không thay đổi: mục tiêu các thuật toán tìm ra các đặt trưng mô tả cấu trúc khuôn mặt người (các đặc trưng không thay đổi với tư thế, vị trí đặt thiết bị thu hình, độ sáng tối thay đổi...).
- Hướng tiếp cận dựa trên so khớp mẫu: Dùng các mẫu khuôn mặt và đặc trưng chuẩn (các mẫu được lưu trong cơ sở dữ liệu) để so sánh các khuôn mặt hay các đặc trưng của khuôn mặt.
- Hướng tiếp cận dựa trên diện mạo: Trái ngược với hướng tiếp cận dựa trên khuôn mẫu, các mô hình (hay các mẫu) sẽ được học từ một tập ảnh huấn luyện mà thể hiện tính chất tiêu biểu của sự xuất hiện của mặt người trong ảnh. Sau đó hệ thống (mô hình) sẽ xác định mặt người. Phương pháp này còn được biết đến với tên gọi tiếp cận theo các phương pháp học máy.

Ngoài ra cũng có rất nhiều nghiên cứu mà phương pháp xác định mặt người không chỉ dựa vào một hướng mà có liên quan đến nhiều hướng.

2.3 Phương pháp phát hiện khuôn mặt với Haar-like – Adaboost

2.3.1 Đặc trưng Haar-like

Phát hiện mặt người là bài toán cơ bản được xây dựng từ nhiều năm nay, có nhiều phương pháp được đưa ra như sử dụng template matching, neuron network ... Cho tới nay bài toán này hầu như được giải quyết dựa trên phương pháp sử dụng các đặc trưng haar like. Phương pháp này được cho là đơn giản và kết quả phát hiện là tương đối cao, lên tới trên 98%, Các hãng sản xuất máy ảnh như Canon, Samsung ... cũng đã tích hợp nó vào trong các sản phẩm của

mình. Trong bài này tôi giới thiệu sơ qua về phương pháp này và chương trình để phát hiện mặt người.

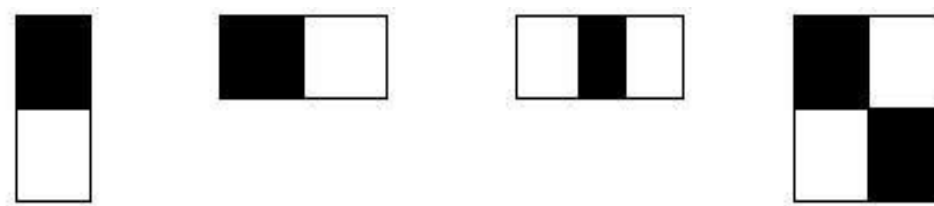
Trên ảnh, vùng khuôn mặt là tập hợp các điểm ảnh có những mối quan hệ khác biệt so với các vùng ảnh khác, những mối quan hệ này tạo lên các đặc trưng riêng của khuôn mặt. Tất cả khuôn mặt người đều có chung những đặc điểm sau khi đã chuyển qua ảnh xám, ví dụ như:

- Vùng hai mắt sẽ tối hơn vùng má và vùng trán, nghĩa là mức xám của vùng này cao hơn vượt trội so với hai vùng còn lại.
- Vùng giữa sống mũi cũng tối hơn vùng hai bên mũi.

Còn rất nhiều những đặc điểm khác của khuôn mặt và các đặc trưng Haar like dựa vào các đặc điểm này để nhận dạng.

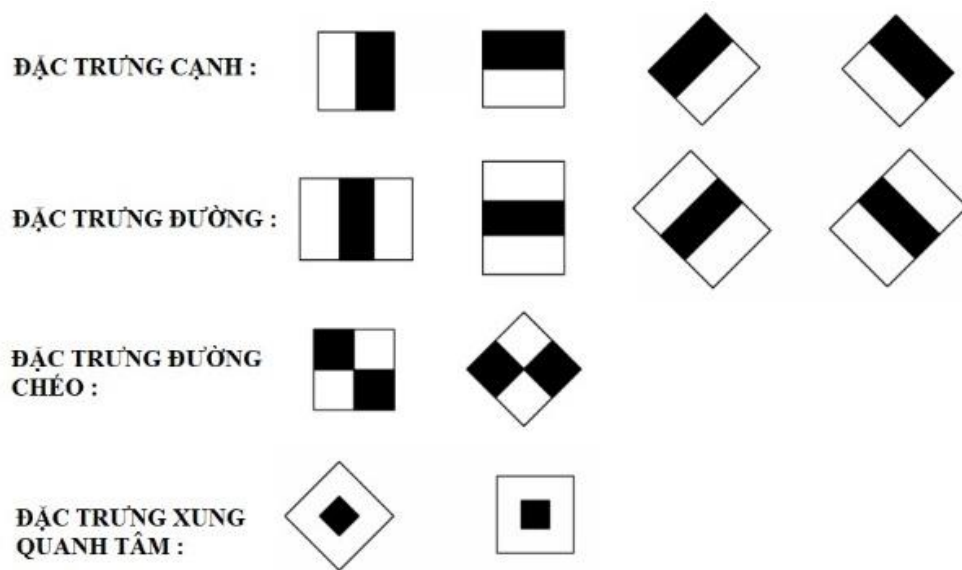
Đầu tiên, ta phải huấn luyện dựa vào các đặc trưng haar like sau đó dựa vào tập huấn luyện này để phát hiện khuôn mặt. Để làm như được vậy, ảnh đầu vào trước hết phải được xám hóa, sau đó cho một hoặc nhiều hình chữ nhật chứa đặc trưng haar-like chạy khắp bức ảnh, những chỗ khác biệt sẽ được lưu trữ lại. Tập hợp hàng nghìn điểm khác biệt của hàng trăm khuôn mặt khác nhau sẽ cho ta một tập dữ liệu dùng cho việc phát hiện mặt người

Các đặc trưng Haar-like do Viola và Jones công bố [5][6][7]. Chúng gồm 4 đặc trưng cơ bản để xác định khuôn mặt người. Mỗi đặc trưng Haar-like là sự kết hợp của hai hay ba hình chữ nhật "trắng" hay "đen" như trong hình sau:



Hình 2.2: Đặc trưng Haar-like cơ bản

Để sử dụng các đặc trưng này vào việc xác định khuôn mặt người, 4 đặc trưng Haar-like cơ bản được mở rộng ra, và được biểu diễn trong Hình 2.2:



Hình 2.3: Đặc trưng Haar-like mở rộng

Dùng các đặc trưng trên, ta có thể tính được giá trị của đặc trưng Haar-like là sự chênh lệch giữa tổng của các pixel của các vùng đen và các vùng trắng như trong công thức sau:

$$f(x) = \text{TổngVùngĐen(pixel)} - \text{TổngVùngTrắng(pixel)} \quad (1)$$

Sử dụng giá trị này, so sánh với các giá trị của các giá trị pixel thô, các đặc trưng Haar-like có thể tăng/giảm sự thay đổi in-class/out-of-class (bên trong hay bên ngoài lớp khuôn mặt người), do đó sẽ làm cho bộ phân loại dễ hơn.

Như vậy để phát hiện khuôn mặt, hệ thống sẽ cho một cửa sổ con(sub-window) có kích thước cố định quét lên toàn bộ ảnh đầu vào. Như vậy sẽ có rất nhiều ảnh con ứng với từng cửa sổ con, các đặc trưng Haar-like sẽ được đặt lên các cửa sổ con này để từ đó tính ra giá trị của đặc trưng. Sau đó các giá trị này được bộ phân loại xác nhận xem khung hình đó có phải khuôn mặt hay không.

Ứng với mỗi một đặc trưng , một bộ phân lớp yếu (weak classifier) $h_k(x)$ sẽ được định nghĩa như sau :

$$h_k = 1 \text{ nếu } p_k f_k(x) < p_k \theta_k \text{ ngược lại thì } h_k = 0 \quad (2)$$

Trong đó :

- x : cửa sổ con cần xét
- θ_k : ngưỡng
- f_k : giá trị của đặc trưng Haar-like
- p_k : hệ số quyết định chiều của phương trình

Ta hiểu công thức trên đơn giản như sau : Khi giá trị của đặc trưng Haar-like $k:f_k$ tại cửa sổ con x vượt qua một ngưỡng thì bộ phân lớp $h_k(x)$ sẽ kết luận cửa sổ con x là khuôn mặt ($h_k(x) = 1$), còn f_k không vượt qua ngưỡng đó thì không là khuôn mặt.

Ngưỡng là giá trị đã rút ra sau quá trình huấn luyện bộ phân lớp.

Ta thấy đặc trưng Haar-like k có kích thước và vị trí đặt ở trong cửa sổ con x trên hình sẽ có giá trị f_k rất lớn (vì mức xám ở vùng mắt lớn hơn nhiều so với vùng má). Giá trị f_k này lớn hơn nhiều với hầu hết các giá trị f_k khác khi ta di chuyển cửa sổ con x sang các vị trí khác (quét trên toàn ảnh đầu vào), và với ngưỡng phù hợp, kết quả bộ phân lớp $h_k(x)$ sẽ cho cửa sổ con ở vị trí như trên là khuôn mặt còn ở vị trí khác không.

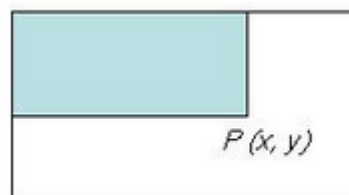
Sẽ có một vài vị trí khác trên ảnh đầu vào vô tình cho ra f_k vượt ngưỡng và bộ phân lớp yếu $h_k(x)$ kết luận đây là khuôn mặt, nhưng ta không chỉ dùng một đặc trưng haar-like mà dùng rất nhiều đặc trưng ở những vị trí và kích thước khác nhau trong cửa sổ con làm giảm đi sự trùng hợp ngẫu nhiên.

Việc xác định một cửa sổ con có phải là mặt hay không, phải sử dụng rất nhiều đặc trưng Haar-like. Ứng với mỗi kích thước, kiểu đặc trưng và vị trí đặt trong cửa sổ con cho ta một đặc trưng ứng với một bộ phân lớp yếu x . Vì vậy tập hợp đầy đủ số lượng các đặc trưng trong một cửa sổ con là rất lớn. Theo như sự tính toán của hai tác giả, với một cửa sổ con ở kích thước cơ bản 24x24 pixel số lượng đầy đủ các đặc trưng Haar-like là hơn 160000.

Có hai vấn đề đặt ra ở đây:

- Thứ nhất là giá trị của mỗi một đặc trưng Haar-like được tính bằng tổng giá trị các điểm ảnh vùng đen trừ đi tổng các điểm ảnh vùng trắng, như vậy là với số lượng lớn các đặc trưng sẽ tạo ra một khối lượng tính toán khổng lồ. Điều này là không phù hợp để đáp ứng thời gian thực vì thời gian xử lý rất lâu.
- Thứ hai là trong số hàng trăm ngàn đặc trưng đó thì không phải đặc trưng nào cũng có ích cho công việc phân loại. Nếu chúng ta không tìm ra cách loại bỏ những đặc trưng không có ích để tập trung vào những đặc trưng có hiệu quả cao thì cũng sẽ mất thời gian xử lý một cách vô ích.

Để tính các giá trị của đặc trưng Haar-like, ta phải tính tổng của các vùng pixel trên ảnh. Nhưng để tính toán các giá trị của các đặc trưng Haar-like cho tất cả các vị trí trên ảnh đòi hỏi chi phí tính toán khá lớn, không đáp ứng được cho các ứng dụng đòi hỏi tính run-time. Do đó Viola và Jones đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích của ảnh cần tính các đặc trưng Haar-like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó. Bắt đầu từ vị trí trên, bên trái đến vị trí dưới, phải của ảnh, việc tính toán này đơn thuần chỉ dựa trên phép cộng số nguyên đơn giản, do đó tốc độ thực hiện rất nhanh.



$$P(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (3)$$

Hình 2.4: Công thức Integral Image của ảnh

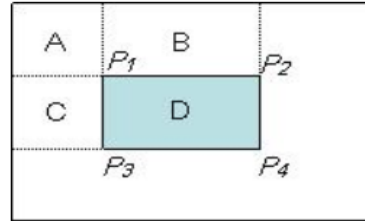
Sau khi đã tính được Integral Image, việc tính tổng các giá trị mức xám của một vùng bất kỳ nào đó trên ảnh thực hiện rất đơn giản theo cách sau:

Giả sử ta cần tính tổng các giá trị mức xám của vùng D như trong hình 4, ta có thể tính như sau:

$$D = P_4 - P_2 - P_3 + P_1 \quad (4)$$

Với $A + B + C + D$ chính là giá trị tại điểm P_4 trên Integral Image, tương tự như vậy $A+B$ là giá trị tại điểm P_2 , $A+C$ là giá trị tại điểm P_3 , và A là giá trị tại điểm P_1 . Vậy ta có thể viết lại biểu thức tính D ở trên như sau:

$$D = \underbrace{(x_4, y_4)}_{A+B+C+D} - \underbrace{(x_2, y_2)}_{A+B} - \underbrace{(x_3, y_3)}_{A+C} + \underbrace{(x_1, y_1)}_A \quad (5)$$



Hình 2.5: Công thức tính mức xám của vùng D

Tiếp theo, để chọn các đặc trưng Haar-like dùng cho việc thiết lập ngưỡng, Viola và Jones sử dụng một phương pháp máy học được gọi là AdaBoost. AdaBoost sẽ kết hợp các bộ phân loại yếu để tạo thành một bộ phân loại mạnh. Với bộ phân loại yếu chỉ cho ra câu trả lời chính xác chỉ hơn viện đoán một cách ngẫu nhiên một chút, còn bộ phân loại mạnh có thể đưa ra câu trả lời chính xác trên 60%.

2.3.2 Tổng quan về AdaBoost

AdaBoost là một thuật toán boosting dùng để xây dựng bộ phân lớp (classifier). Ý tưởng cơ bản của Adaboost cũng hoạt động trên nguyên tắc kết hợp các bộ phân loại yếu để xây dựng bộ phân lớp mạnh[8].

2.3.3 Áp dụng Adaboost cho bài toán phát hiện khuôn mặt

Để huấn luyện bộ phát hiện Adaboost, một tập các ảnh có khuôn mặt và không có khuôn mặt sẽ được sử dụng.

Để phát hiện khuôn mặt, cách tiếp cận dựa trên AdaBoost gồm hai giai đoạn chính.

Trong giai đoạn đầu, các bộ phân lớp mạnh được xây dựng từ các bộ phân lớp yếu.

Trong giai đoạn hai, các bộ phân lớp mạnh sẽ được kết hợp tuần tự để tạo thành cascade of boosted classifier (cascade tree)

Cách xây dựng các bộ phân lớp mạnh :

- Đầu vào : Dữ liệu huấn luyện X , $X = \{x_1, \dots, x_n\}$. Với $x_k = \{x_1^k, x_2^k, \dots, x_m^k\}$ thuộc tập X là 1 vector đặc trưng. y_k thuộc tập $Y = \{-1, 1\}$ là một nhãn tương ứng (1 tương ứng với khuôn mặt, -1 tương ứng với không phải khuôn mặt). T các phân lớp yếu $h_j: X \rightarrow \{-1, +1\}$. Chúng tôi sử dụng thuật toán AdaBoost để xây dựng một bộ phân lớp mạnh.
- Đầu ra: Bộ phân lớp mạnh :

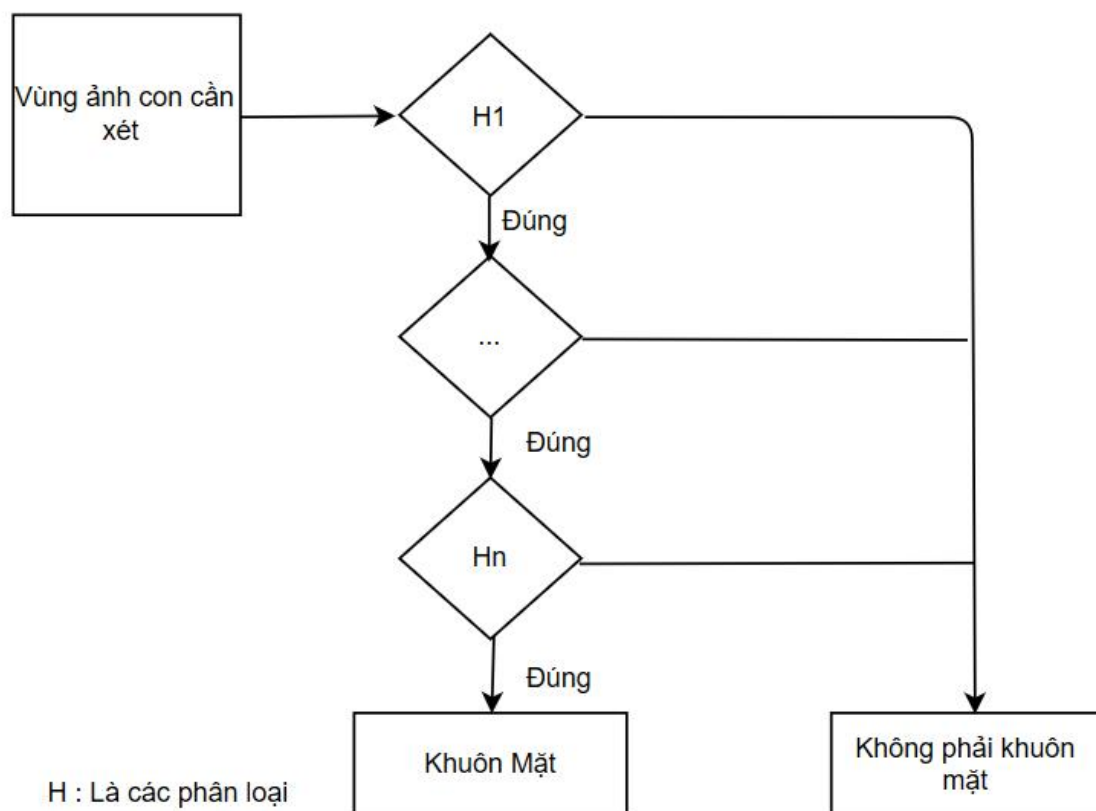
$$H(x) = \text{sign}(\sum_{j=1}^T a_j h_j(x)) \quad (6)$$

- Các bộ phân lớp yếu $h_j(x)$ là các đặc trưng Haar – like đơn giản. Một tập rất lớn các bộ phân lớp yếu được sử dụng để phân lớp một cửa sổ ảnh là khuôn mặt hoặc không phải khuôn mặt. Viola và Jones đề xuất 4 mẫu đặc trưng vô hướng cơ bản cho phát hiện khuôn mặt. Sau đó Rainer Lienhart đã mở rộng thành 14 mẫu đặc trưng [3]. Trong [1], Viola và Jones đã đề xuất một sự biểu diễn ảnh gọi là “hàm ảnh”, đề xuất này cho phép các đặc trưng Haar – like được tính toán cực kì nhanh. Tuy nhiên, để đạt được kết quả tốt, một hệ thống dựa trên AdaBoost cần một số lượng khổng lồ các đặc trưng. Ví dụ, trong một cửa sổ quét kích thước 20x20, có khoảng 10.000 các đặc trưng cho các hình dáng, kích thước và vị trí khác nhau. Điều này làm giảm đáng kể tốc độ của bộ phát hiện nếu chúng ta sử dụng nhiều bộ phân lớp mạnh (strong classifier) trong cascade tree. Hơn nữa, sự chính xác của bộ phân lớp cuối cùng phụ thuộc vào các bộ phân lớp yếu (các

đặc trưng Haar – like). Vì vậy nếu khả năng phân tách (khuôn mặt/ không khuôn mặt) các mẫu background khó của các đặc trưng Haar – like không tốt thì kết quả phát hiện thường có nhiều phát hiện sai.

Xây dựng cascade của các bộ phân lớp mạnh :

- Hầu hết các bộ phân lớp tốt cần nhiều thời gian để có các kết quả phân lớp bởi vì chúng cần xem xét đến một số lượng lớn các đặc trưng của mẫu. Cấu trúc cascade của các bộ phân lớp mạnh đã được đề xuất để giảm thời gian thực thi và giảm tỉ lệ phát hiện sai.

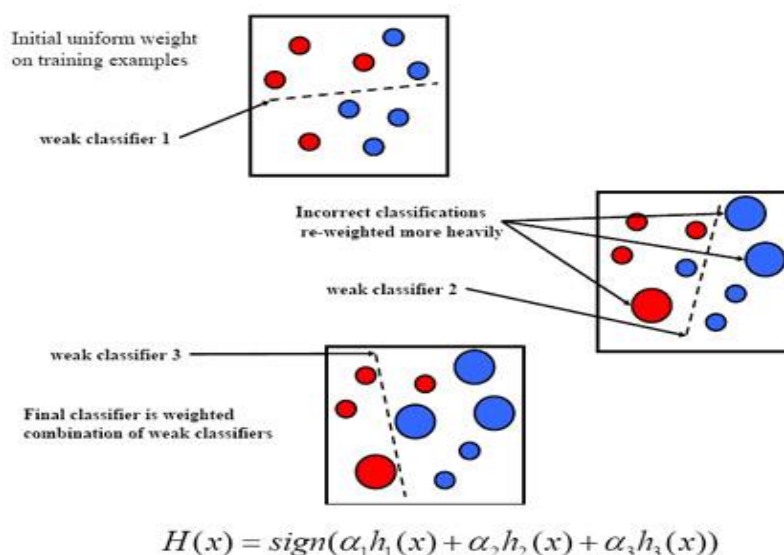


Hình 2.6 : Cascade của các bộ phân lớp mạnh. Nơi một vùng ứng viên có thể bị loại tại bất kì giai đoạn nào

- Cascade tree có một số thuận lợi. Mỗi giai đoạn là một bộ phân lớp mạnh. Trong suốt quá trình phát hiện khuôn mặt. Mỗi cửa sổ được phân tích tuần tự bởi mỗi bộ phân lớp giai đoạn. Một cửa sổ có thể bị loại (kết luận không có khuôn mặt) hoặc đi qua bộ phân lớp giai

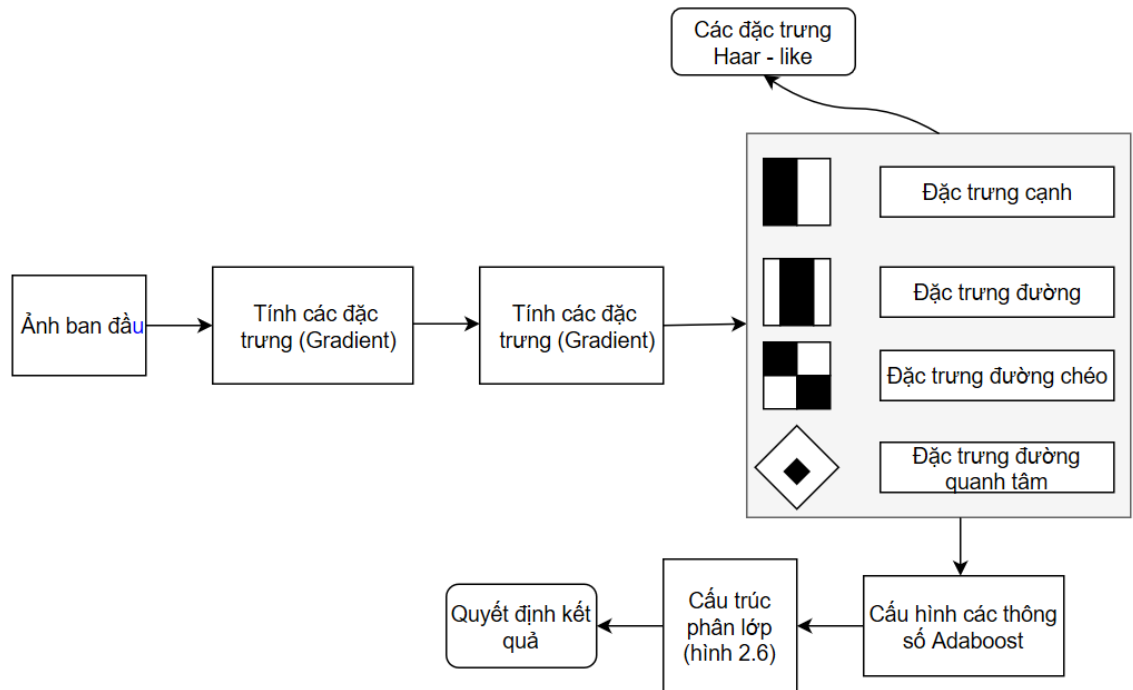
đoạn này. Trong suốt quá trình huấn luyện, mỗi bộ phân lớp được huấn luyện bởi các mẫu bị phân lớp sai ở bộ phân lớp trước. Nghĩa là chúng sẽ tập trung để học các mẫu background khó. Vì vậy sự kết hợp của các bộ phân lớp trong cascade sẽ giảm tỉ lệ phát hiện sai. Hơn nữa cấu trúc này, các bộ phân lớp dễ dàng nhận ra các background dễ và loại bỏ chúng ở các bộ phân lớp đầu. Vì vậy cấu trúc cascade giải quyết được bài toán thời gian thực thi. Một cách tổng quát, cấu trúc cascade cải tiến đáng kể thời gian thực thi và các kết quả phát hiện phụ thuộc hoàn toàn vào các bộ phân lớp yếu hay các đặc trưng Haar – like.

AdaBoost sẽ kết hợp các bộ phân loại yếu thành bộ phân loại mạnh như sau:



Hình 2.7: Quá trình hình thành strong classifiers (Tập phân lớp mạnh)

2.3.4 Sơ đồ khối quá trình tìm kiếm khuôn mặt người



Hình 2.8: Sơ đồ khối quá trình tìm kiếm khuôn mặt người

Như trong hình 2.7, từ ảnh gốc ban đầu, ta sẽ được tính Integral Image, là mảng 2 chiều với phần tử (x, y) sẽ được tính bằng tổng của các phần tử (x', y') với $x' < x$ và $y' < y$, mục đích là để tính nhanh tổng của các giá trị mức xám của một vùng hình chữ nhật bất kỳ trên ảnh gốc. Các vùng ảnh con này sẽ được đưa qua các hàm Haar cơ bản để ước lượng đặc trưng, kết quả ước lượng sẽ được đưa qua bộ điều chỉnh AdaBoost để loại bỏ nhanh các đặc trưng không có khả năng là đặc trưng của khuôn mặt người. Chỉ có một tập nhỏ các đặc trưng mà bộ điều chỉnh AdaBoost cho là có khả năng là đặc trưng của khuôn mặt người mới được chuyển sang cho bộ quyết định kết quả (là tập các bộ phân loại yếu có cấu trúc như trong Hình 2.5). Bộ quyết định sẽ tổng hợp kết quả là khuôn mặt người nếu kết quả của các bộ phân loại yếu trả về là khuôn mặt người.

Mỗi bộ phân loại yếu sẽ quyết định kết quả cho một đặc trưng Haar-like, được xác định ngưỡng đủ nhỏ sao cho có thể vượt được tất cả các bộ dữ liệu mẫu trong tập dữ liệu huấn luyện (số lượng ảnh khuôn mặt trong tập huấn luyện có thể rất lớn). Trong quá trình xác định khuôn mặt người, mỗi vùng ảnh con sẽ

được kiểm tra với các đặc trưng trong chuỗi các đặc trưng Haar-like, nếu có một đặc trưng Haar-like nào cho ra kết quả là khuôn mặt người thì các đặc trưng khác không cần xét nữa. Thứ tự xét các đặc trưng trong chuỗi các đặc trưng Haar-like sẽ được dựa vào trọng số (weight) của đặc trưng đó do AdaBoost quyết định dựa vào số lần và thứ tự xuất hiện của các đặc trưng Haar-like.

2.4 THUẬT TOÁN TRÍCH RÚT ĐẶC TRƯNG LBP (LOCAL BINARY PATTERN)

2.4.1 Lý thuyết về LBP

LBP là viết tắt của Local Binary Pattern hay là mẫu nhị phân cục bộ được Ojala trình bày vào năm 1996 như là một cách đo độ tương phản cục bộ của ảnh [9]. Là một toán tử kernel 3×3 , nó tổng quát hóa cấu trúc không gian cục bộ của một ảnh. Ojala và các đồng nghiệp đã giới thiệu phương pháp LBP và chỉ ra khả năng phân tách cao của chúng cho sự phân lớp vân. Tại một vị trí pixel (x_c, y_c) và 8 pixel lân cận của nó. Như vậy mỗi pixel sẽ được biểu diễn bởi một chuỗi nhị phân, giá trị thập phân của chuỗi nhị phân này chính là giá trị của pixel trung tâm trong sự biểu diễn bởi toán tử LBP.

Kể từ khi được đưa ra, theo định nghĩa là bất biến với những thay đổi đơn điệu trong ảnh đen trắng. Để cải tiến phương pháp, bổ sung thêm phương pháp tương phản trực giao địa phương. Hình dưới minh họa cách tính độ tương phản trực giao (C) là ký hiệu cấp độ xám trung bình của các điểm ảnh lớn hơn hoặc bằng ngưỡng với các điểm ảnh thấp hơn ngưỡng. Phân phối hai chiều của mã LBP và độ tương phản cục bộ được lấy làm đặc trưng gọi là LBP/C.

Vi dụ	Lấy ngưỡng	Trọng số																											
<table border="1"> <tr><td>9</td><td>7</td><td>1</td></tr> <tr><td>5</td><td>8</td><td>9</td></tr> <tr><td>9</td><td>6</td><td>3</td></tr> </table>	9	7	1	5	8	9	9	6	3	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td></td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> </table>	1	0	0	0		1	1	0	0	<table border="1"> <tr><td>1</td><td>2</td><td>4</td></tr> <tr><td>128</td><td></td><td>8</td></tr> <tr><td>64</td><td>32</td><td>16</td></tr> </table>	1	2	4	128		8	64	32	16
9	7	1																											
5	8	9																											
9	6	3																											
1	0	0																											
0		1																											
1	0	0																											
1	2	4																											
128		8																											
64	32	16																											

$$\text{Pattern} = 10010010$$

$$\text{LBP} = 1 + 8 + 64 = 73$$

$$C = (9 + 9 + 9) / 3 - (7 + 1 + 3 + 6 + 5) / 5 = 4.3$$

Hình 2.9: Cách tính LBP và độ tương phản cục bộ C.

2.4.2 Thuật toán LBP:

Thông tin LBP của pixel tại trung tâm của mỗi khối ảnh sẽ được tính dựa trên thông tin của các pixel lân cận. Có thể tóm tắt các bước tiến hành như sau:

- Bước 1: Xác định bán kính làm việc.
- Bước 2: Tính giá trị LBP cho pixel ở trung tâm (x_c, y_c) khối ảnh dựa trên thông tin của các pixel lân cận :

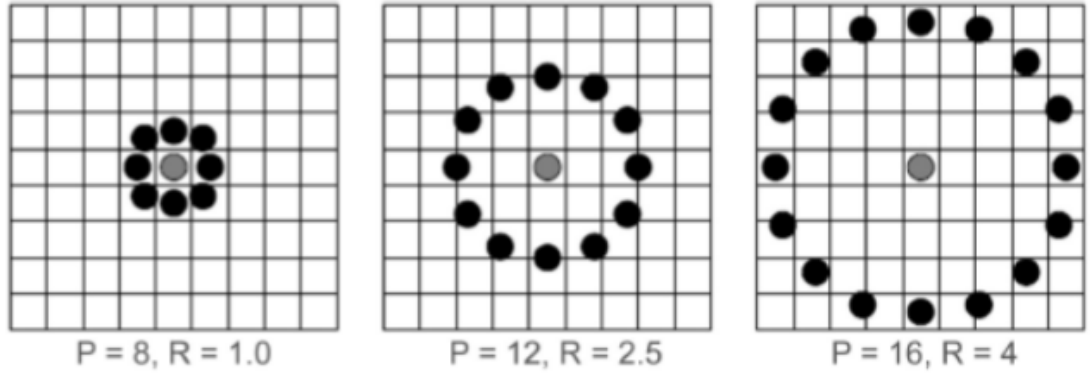
Trong đó, (g_c) là giá trị độ xám của pixel trung tâm, (g_n) là giá trị độ xám của pixel thứ n trong lân cận 8 của pixel trung tâm và (s) là hàm nhị phân được xác định như sau:

$$s(x) = 1 \text{ nếu giá trị } x \geq 0 \quad (7)$$

$$\text{Ngược lại } s(x) = 0.$$

Bằng cách định nghĩa này, toán tử LBP bất biến đối với bất kỳ sự biến đổi độ sáng đều và bảo toàn trật tự mật độ các pixel trong một lân cận cục bộ.

Sau đó, Ojala và đồng nghiệp của ông [10] đã mở rộng toán tử LBP đến 1 lân cận tròn với các bán kính khác nhau. $\text{LBP}_{P,R}$ kí hiệu sự xem xét đến P pixels lân cận trên một vòng tròn có bán kính R.



Hình 2.10: Minh họa toán tử LBP mở rộng với các giá trị P và R khác nhau.

Nếu tọa độ của pixel tâm là (x_c, y_c) thì tọa độ của P pixel lân cận trên đường tròn tâm (x_c, y_c) bán kính R (tính theo đường tròn lượng giác) là:

$$x_p = x_c + R \cos(2\pi p/P)$$

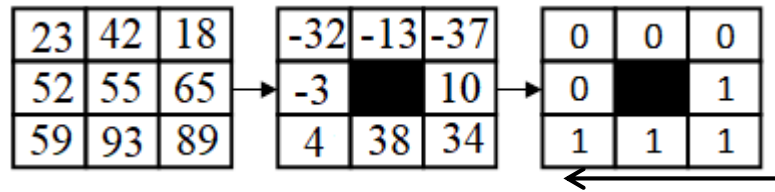
$$y_p = y_c + R \sin(2\pi p/P), p = \{0, 1, \dots, P-1\} \quad (8)$$

Để xác định giá trị của các pixel lân cận không nằm trong tâm của một pixel, chúng tôi sử dụng phép nội suy song tuyến tính.

Trong [4] các tác giả đã kết luận rằng : hầu hết các thông tin vân (texture) trong ảnh được chứa trong một tập con của các mẫu LBP. Các mẫu này được gọi là các mẫu đồng nhất (uniform patterns). Các mẫu đồng nhất chứa tối đa hai sự chuyển đổi bit từ 0 đến 1 hoặc từ 1 đến 0. 11111111, 00000110 hoặc 10000111 là ví dụ cho các mẫu đồng nhất. Toán tử $LBP_{P,R}$ đồng nhất được kí hiệu là $LBP_{P,R}^{U2}$. Với chuỗi LBP có chiều dài P thì số mẫu có tối đa hai sự chuyển đổi (Mẫu đồng nhất là) $P(P-1)$. Có hai mẫu không có sự chuyển đổi nào là mẫu toàn 0 hoặc 1. Việc sử dụng mẫu LBP đồng nhất có hai lợi điểm quan trọng. Thứ nhất là tiết kiệm bộ nhớ, vì trường hợp LBP tổng quát chúng ta có 2^P mẫu có thể, nhưng nếu chỉ xét mẫu đồng nhất thì chúng ta có tối đa là $P(P+1) + 2$ nếu chúng ta sử dụng toán tử $LBP_{P,R}^{U2}$ chỉ phát hiện những mẫu vân cục bộ quan trọng như các điểm, các điểm cuối đường thẳng, biên cạnh và các góc.

Bởi vì khả năng phân tách và chi phí tính toán thấp, LBP được sử dụng rất nhiều trong nhận dạng mẫu, LBP được áp dụng cho việc phát hiện khuôn mặt [1][2], nhận dạng khuôn mặt [2][10].

Ví dụ 1 :



$$LBP = 1 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 + 0 \cdot 2^5 + 0 \cdot 2^6 + 0 \cdot 2^7 = 15$$

Hình 2.11: Các biến thể của LBP: LBP đồng dạng.

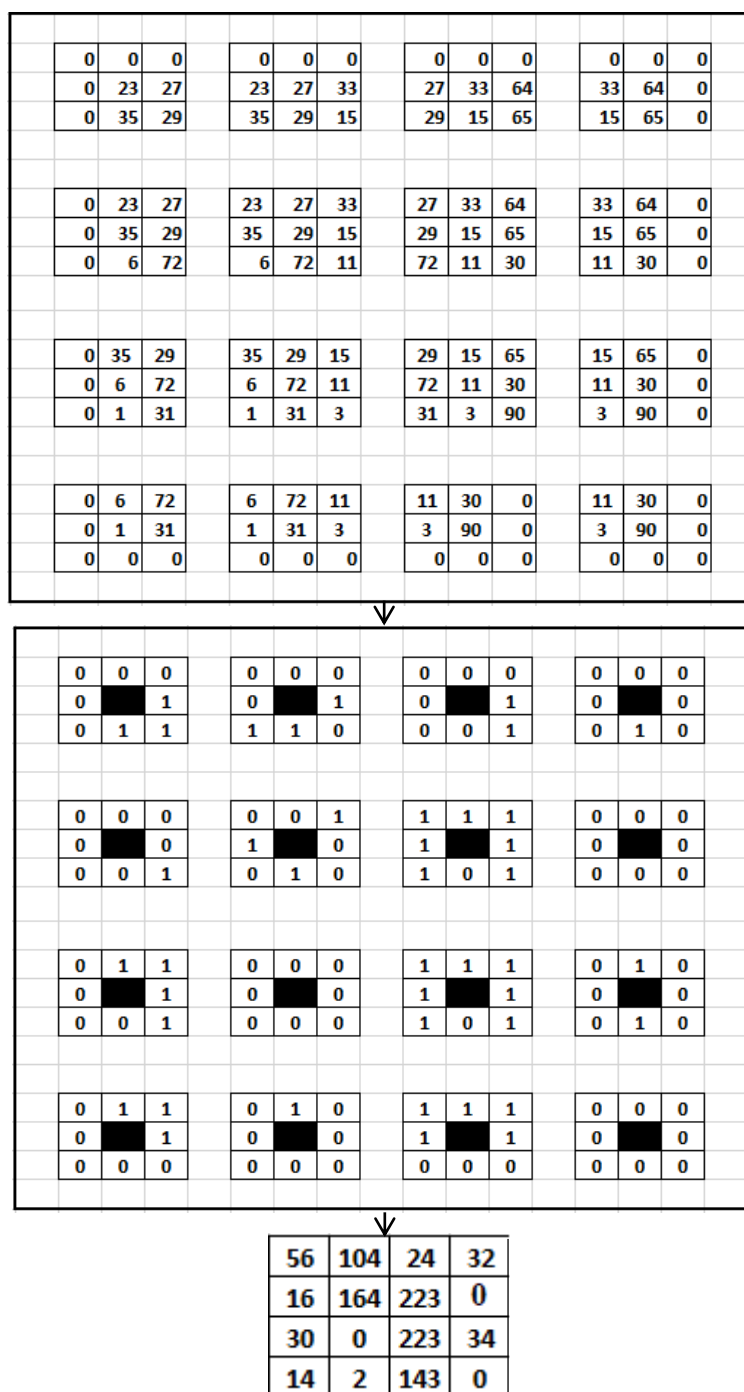
Một mẫu nhị phân được gọi là đồng dạng khi xét chuỗi bit xoay vòng thì có nhiều nhất là 2 lần thay đổi (transitions) từ giá trị bit 0 sang 1 hoặc từ giá trị bit 1 sang 0. Ví dụ: 00000000 có 0 transitions, 01110000 có 2 transitions, 11001111 có 2 transitions nên đây là uniform LBP. 11001001 có 4 transitions, 01010011 có 5 transitions nên không phải là uniform LBP.

2.4.3 Ví dụ về trích rút đặc trưng LBP trên ảnh số :

Ví dụ với một ảnh có kích thước 4x4 :

23	27	33	64
35	29	15	65
6	72	11	30
1	31	3	90

Chúng ta sẽ tính được giá trị đặc trưng LBP như sau:



Hình 2.12 : Ví dụ về quá trình tính toán đặc trưng LBP

2.4.4 LBPH – Mở rộng từ thuật toán LBP:

LBPH là một thuật toán mở rộng từ LBP và chúng ta sẽ định hình thuật toán theo từng bước [4] như sau:

- Thông số: LBPH gồm 4 thông số:
 - Bán kính: được sử dụng để xây dựng mô hình nhị phân cục bộ tròn và biểu thị bán kính xung quanh pixel trung tâm. Nó thường được đặt thành 1.
 - Neighbors: là số lượng điểm mẫu chọn xung quanh để xây dựng mô hình nhị phân cục bộ tròn, thường được đặt là 8.
 - Trục X: số lượng ô theo hướng ngang. Càng nhiều ô, lưới càng mịn, thì kết quả của vector đặc trưng càng cao. Nó thường được đặt thành 8.
 - Trục Y: số lượng ô theo hướng dọc. Càng nhiều ô, lưới càng mịn, thì kết quả của vector đặc trưng càng cao. Nó thường được đặt thành 8.
- Thuật toán training dữ liệu :
 - Sử dụng tập dữ liệu với hình ảnh khuôn mặt của những người mà ta cần đưa vào huấn luyện và nhận dạng. Với mỗi bộ dữ liệu hình ảnh ta cũng cần đặt ID (có thể là số hoặc tên của người tương ứng với mỗi bộ ảnh) cho mỗi hình ảnh, vì vậy thuật toán sẽ sử dụng thông tin này để nhận ra hình ảnh đầu vào và cung cấp cho bạn đầu ra. Hình ảnh của cùng một người phải có cùng ID.
- Áp dụng các thao tác từ thuật toán LBP:
 - Bước tính toán đầu tiên của LBPH là tạo ra một hình ảnh trung gian mô tả hình ảnh gốc theo cách tốt hơn, bằng cách làm nổi bật các đặc điểm khuôn mặt. Để làm như vậy, thuật toán sử dụng khái niệm của sổ trượt, dựa trên bán kính tham số và Neighbors.
- Trích xuất histograms:
 - Bằng cách sử dụng hình ảnh được tạo ở bước cuối cùng, chúng ta có thể sử dụng các tham số Grid X và Grid Y để chia hình ảnh thành nhiều lưới. Khi chúng ta có một hình ảnh ở thang độ xám (grayscale), mỗi biểu đồ (từ mỗi trục) sẽ chỉ chứa 256 vị trí (0 ~ 255) đại diện cho

sự xuất hiện của từng cường độ pixel. Sau đó, nối từng biểu đồ để tạo ra biểu đồ mới và lớn hơn. Biểu đồ cuối cùng đại diện cho các đặc điểm hình ảnh của hình ảnh gốc.

- Nhận dạng khuôn mặt:
 - Thuật toán đã được huấn luyện. Mỗi histograms (biểu đồ) tạo ra được sử dụng để đại diện cho mỗi hình ảnh từ tập dữ liệu đã được huấn luyện. Vì vậy, với một hình ảnh đầu vào, chúng ta thực hiện lại các bước cho hình ảnh mới này và tạo ra một biểu đồ đại diện cho hình ảnh.
 - Vì vậy, để tìm hình ảnh phù hợp với hình ảnh đầu vào, chúng ta chỉ cần so sánh hai biểu đồ và trả lại hình ảnh với biểu đồ gần nhất.
 - Chúng ta có thể sử dụng các cách tiếp cận khác nhau để so sánh biểu đồ (tính khoảng cách giữa hai biểu đồ).
 - Thuật toán đầu ra là ID từ hình ảnh với biểu đồ gần nhất. Thuật toán cũng sẽ trả về khoảng cách được tính toán, có thể được sử dụng như một phép đo độ tin cậy. Lưu ý: đừng nhầm lẫn về độ tin cậy, vì độ tin cậy thấp hơn sẽ tốt hơn bởi vì điều đó có nghĩa là khoảng cách giữa hai biểu đồ gần hơn.
 - Sau cùng, chúng ta có thể sử dụng ngưỡng và độ tin cậy để tự động ước tính nếu thuật toán đã nhận dạng chính xác hình ảnh.

2.5 NHẬN DẠNG KHUÔN MẶT

Nguyên tắc chung của các phương pháp nhận dạng khuôn mặt hiện nay là thực hiện trích chọn đặc trưng của ảnh cần nhận dạng và đối chiếu với đặc trưng của tập ảnh mẫu đã huấn luyện. Sau đó dựa trên giá trị ngưỡng cho phép về sự khác nhau, quyết định sẽ được đưa ra. Trong nghiên cứu này, tác giả xem xét phương pháp tiếp cận dựa trên các đặc điểm cục bộ LBP.

Thuật toán LBPH (Local Binary Pattern Histogram) [4] không coi ảnh như một vector không gian nhiều chiều, mà thay vào đó mô tả các đặc trưng cục bộ của một đối tượng. Thuật toán này được mở rộng ra từ thuật toán LBP. Ý tưởng cơ bản LBP là tính giá trị cấu trúc cục bộ của một hình ảnh bằng cách so sánh từng pixel với các lân cận của nó, xem điểm ảnh trung tâm như một ngưỡng. Theo định nghĩa này, toán tử LBP bất biến đối với bất kỳ sự biến đổi độ sáng đều và bảo toàn trật tự mật độ các pixel trong một lân cận cục bộ. Sau khi biểu diễn gương mặt thu được thành biểu đồ LBP, cần sử dụng thuật toán so sánh 2 biểu đồ

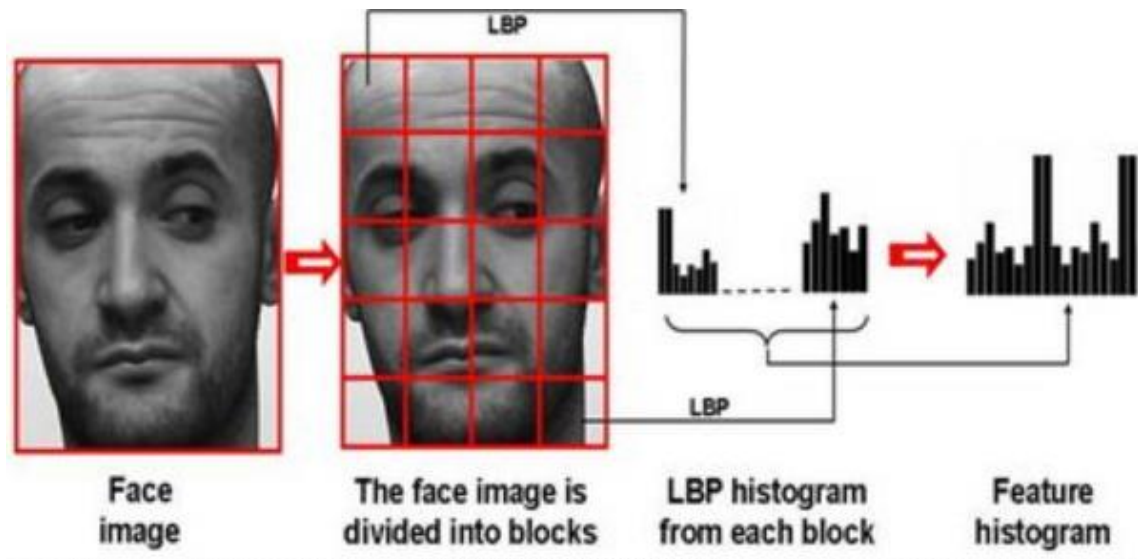
để tìm ra khoảng cách ngắn nhất giữa biểu đồ từ hình ảnh đầu vào với các biểu đồ có được từ tập huấn luyện.

Đại lượng d biểu diễn độ tương tự giữa hai histograms (biểu đồ), (hist1 và hist2) có thể được tính dựa trên bốn phương pháp so sánh khác nhau như : Correlation, Chi-Square, Intersection, Bhattacharyya.

$$D = \sqrt{\sum_{i=1}^n (\text{hist1}_i - \text{hist2}_i)^2} \quad (9)$$

Hình 2.13 : Công thức tính độ tương tự giữa hai histograms.

Một số nghiên cứu cho thấy phương pháp Correlation và Intersection phù hợp với tập dữ liệu lớn, trong khi đó với Chi-Square và Bhattacharyya tỏ ra hiệu quả hơn với tập dữ liệu trung bình và nhỏ.



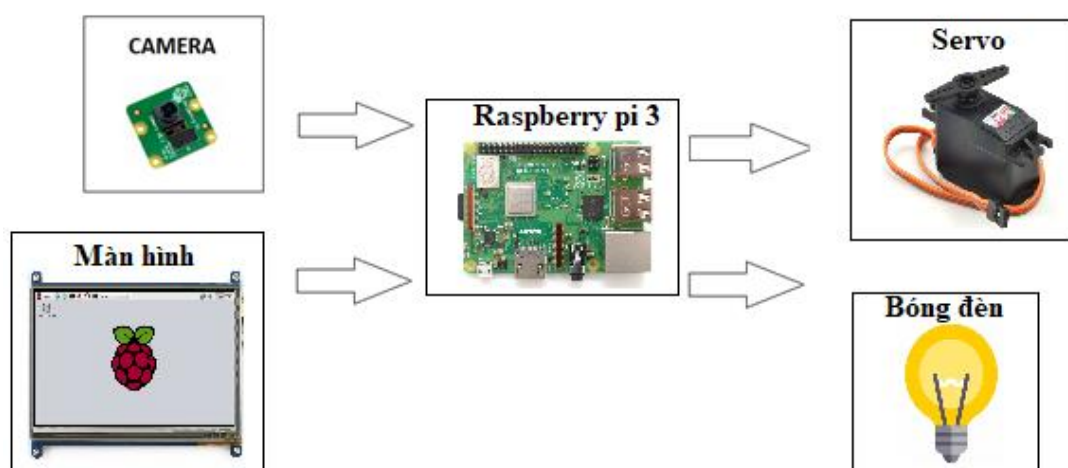
Hình 2.14: Mô tả các đặc trưng cục bộ của một khuôn mặt với LBPH

Chương 3

THIẾT KẾ HỆ THỐNG

3.1 MÔ HÌNH HỆ THỐNG

Với mục tiêu tạo ra một hệ thống hoạt động ổn định về phần cứng và tương tác dễ dàng với người dùng ở phần mềm, nên trước khi đưa ra những giải pháp cụ thể, tác giả đã xây dựng khái quát mô hình hoạt động chung của hệ thống như hình sau:



Hình 3.1: Sơ đồ mô hình hệ thống.

Theo như mô hình đã đề ra, hệ thống sẽ có năm khối chính:

- CAMERA pi có nhiệm vụ dựa trên video livestream lấy hình ảnh nhận dạng được và đưa vào vi xử lý.
- MÀN HÌNH có chức năng lựa chọn các chức năng cho việc xử lý trên vi xử lý.
- BỘ XỬ LÝ xử lý các lệnh đưa vào từ người dùng trên màn hình cảm ứng và xử lý liên tục dữ liệu hình ảnh, nhận diện hình ảnh chụp được so sánh với tập dữ liệu training và đưa ra lệnh điều khiển cho servo và bóng đèn.
- SERVO điều khiển cửa đóng mở cho căn nhà.
- BÓNG ĐÈN để tăng độ sáng khi nhận dạng và thêm dữ liệu hình ảnh.

3.2 THIẾT KẾ PHẦN CỨNG

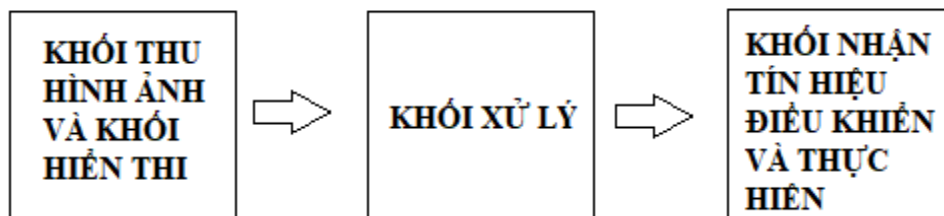
3.2.1 Chức năng của phần cứng

Phần cứng bao gồm một máy tính chạy hệ điều hành Raspbian (ở đây, tác giả sử dụng máy tính nhúng Raspberry Pi 3 Model B), camera pi v2, servo và bóng đèn.

Phần cứng đảm nhận chức năng điều khiển với các chức năng được chọn, và xử lý các dữ liệu hình ảnh nhận vào so sánh với tập dữ liệu hiện tại.

3.2.2 Sơ đồ khối phần cứng

Sơ đồ khối của hệ thống được mô tả như sau:



Hình 3.2: Sơ đồ khối phần cứng của hệ thống

Chức năng các khối:

- Khối thu hình ảnh: Lấy hình ảnh trực tiếp từ môi trường gửi về khối xử lý.
- Khối hiển thị : Lựa chọn các tính năng để gửi về cho khối xử lý.
- Khối xử lý: nhận dữ liệu ảnh từ khối thu hình ảnh, xử lý nhận diện khuôn mặt. Sau khi nhận diện xong sẽ xuất tín hiệu điều khiển đến khối động cơ điều khiển.
- Khối nhận dữ liệu điều khiển: nhận tín hiệu từ Khối xử lý, điều khiển bật tắt thiết bị và mở khóa cửa.

3.2.3 Lựa chọn phần cứng

Để đáp ứng được mô hình so với thiết kế phần cứng như đã đưa ra phía trên, tác giả đã lựa chọn các phần cứng như sau :

3.2.3.1 Camera

- Hiện nay có rất nhiều loại Camera có thể được sử dụng trong việc lấy hình ảnh và livestream như Camera IP, Camera IP Wifi,... tuy nhiên, kích thước của chúng khá lớn, cũng như giá thành cao. Vì vậy, nghiên cứu sử dụng Raspberry Pi Camera Module V1, có giá vừa phải, kích thước nhỏ, rất phù hợp để xây dựng mô hình, và đồng thời giá thành lại rất phù hợp để làm nghiên cứu.



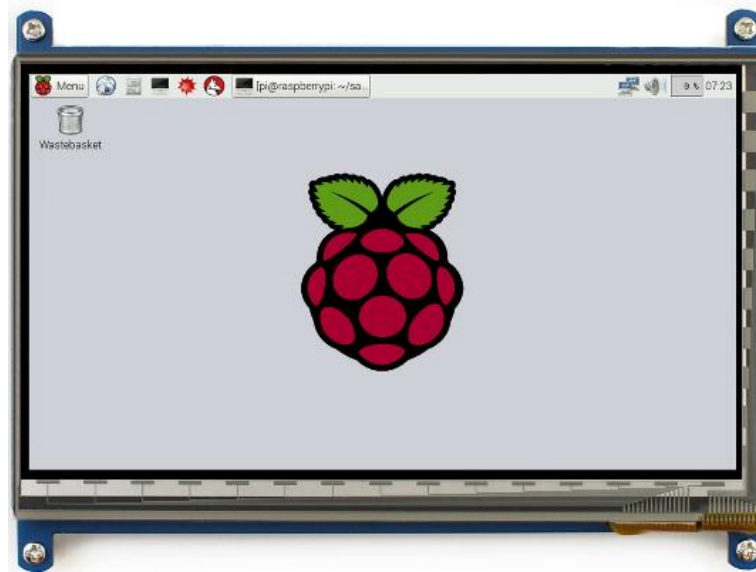
Hình 3.3: *Raspberr Pi Camera Module V1*

- Raspberry Pi Camera Module V1 là loại Module chất lượng cao 5 megapixel. Sử dụng cảm biến hình ảnh OV5647 tùy chỉnh được thiết kế thêm vào bảng cho Raspberry Pi, có một ống kính tập trung cố định. Nó có khả năng chụp ảnh tĩnh 2592x1944 pixel và cũng hỗ trợ video 1080p30, 720p60 và VGA 640x480p60. Nó gắn với Raspberry Pi bằng một trong những lỗ nhỏ trên bề mặt bảng và sử dụng giao diện CSI dành riêng, được thiết kế đặc biệt để kết nối với máy ảnh.

- Tính năng, đặc điểm :

- + Máy ảnh 5 megapixel có khả năng chụp ảnh 2592 x 1944 pixel.
- + Chụp video với độ phân giải 1080p30, 720p60 và VGA 640x480p60.
- + Tất cả phần mềm được hỗ trợ trong phiên bản mới nhất của Hệ điều hành Raspbian.

3.2.3.2 Khối hiển thị



Hình 3.4: Màn hình cảm ứng 7inch.

- Màn hình 7inch HDMI LCD cảm ứng điện dung Waveshare còn có chức năng cảm ứng điện dung và cấp nguồn qua cổng Micro USB, ta có thể xây dựng vô số các ứng dụng điều khiển và hiển thị qua màn hình cảm ứng với các Mini PC như Raspberry Pi, Orange Pi, ... Với giá thành vừa phải cho việc thiết kế hệ thống đề tài.

- Thông số kỹ thuật :

- Model : LCD 7 inch HDMI LCD cảm ứng điện dung Waveshare.
- Điện áp sử dụng : 5VDC từ cổng micro USB.
- Ngõ vào : HDMI.
- Hỗ trợ cảm ứng điện dung và cấp nguồn 5VDC qua cổng Micro USB.
- Độ phân giải 800x480 Pixels.
- Hỗ trợ sử dụng trên Raspberry Pi, Banana Pro / Banana Pi tương thích với các hệ điều hành Ubuntu, Raspbian images.

3.2.3.3 Khối xử lí

- Với đề tài nghiên cứu trên thì có rất nhiều thiết bị có thể được sử dụng làm khối xử lý như : Laptop, Raspberry Pi, FPGA.... Tuy nhiên, nghiên cứu sử dụng đề tài tác giả đã chọn ra kit Raspberry Pi làm khối điều khiển với kích thước hợp lý, giá thành vừa phải làm nhiệm vụ xử lý ảnh nhận được từ Raspberry Pi Camera và màn hình điều khiển để điều khiển thiết bị điện và nhận diện khuôn mặt mở cửa thông minh.



Hình 3.5: *Raspberry Pi 3 Model B*

- Raspberry Pi là một trong những kit nhúng được ứng dụng nhiều nhất như một máy tính mini với hệ điều hành riêng hỗ trợ tối ưu cho việc lập trình điều khiển ngõ ra theo ý muốn của người lập trình. Với lượng thư viện hỗ trợ lớn và tối ưu Raspberry Pi rất thích hợp cho người sử dụng muốn phát triển nhúng có kết hợp đồ họa, âm thanh...

- Raspberry Pi 3 Model B là thế hệ thứ 3 của gia đình Raspberry Pi, nó ra đời vào tháng 2 năm 2016. Thông tin cấu hình Raspberry Pi 3:

- Broadcom BCM2837 chipset running at 1.2 GHz.
- 64-bit quad-core ARM Cortex-A53.
- 802.11 b/g/n Wireless LAN.
- Bluetooth 4.1 (Classic & Low Energy).
- Dual core Videocore IV® Multimedia co-processor.
- 1 GB LPDDR2 memory.

- Supports all the latest ARM GNU/Linux distributions and Windows 10 IoT.
- MicroUSB connector for 2.5 A power supply.
- 1 x 10/100 Ethernet port.
- 1 x HDMI video/audio connector.
- 1 x RCA video/audio connector.
- 4 x USB 2.0 ports.
- 40 GPIO pins.
- Chip antenna.
- DSI display connector.
- MicroSD card slot.
- Dimensions: 85 x 56 x 17 mm.

3.2.3.4 Khối xử lý tín hiệu điều khiển

Servo:

- Để điều khiển đóng mở cửa tự động tác giả sử dụng động cơ RC Servo MG996, được sử dụng nhiều trong các thiết kế robot hoặc các việc sử dụng cần lực kéo mạnh. Với tầm giá rẻ hoạt động tốt.



Hình 3.6: Servo MG996R

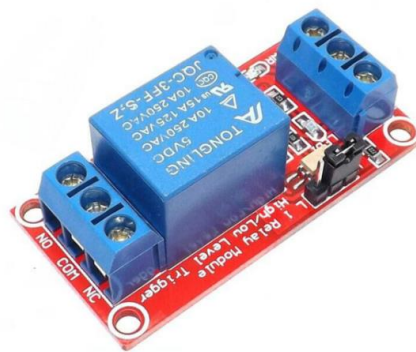
- Thông số kỹ thuật :

- Chủng loại : Analog RC Servo.
- Kích thước: 40mm x 20mm x 43mm.
- Trọng lượng: 55g.
- Điện áp hoạt động : 4.8 – 6.6 VDC.

- Lực kéo :
 - 3.5 Kg-cm với 4.5V – 1.5A.
 - 5.5 Kg-cm với 6.5V – 1.5A.
- Tốc độ quay :
 - 0.17sec / 60 degrees (4.8v)
 - 0.13sec / 60 degrees (6v)

Relay:

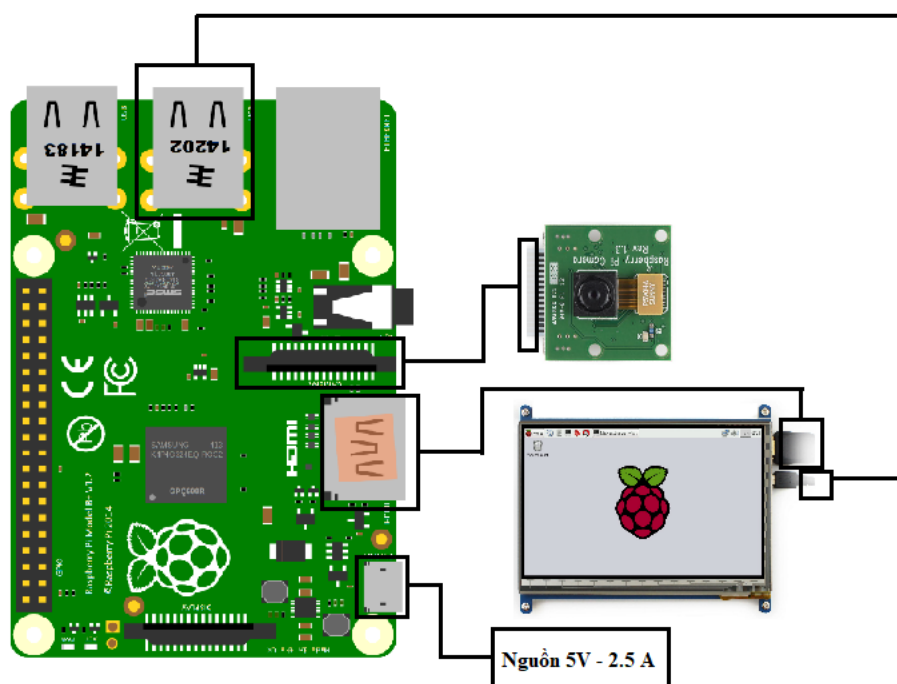
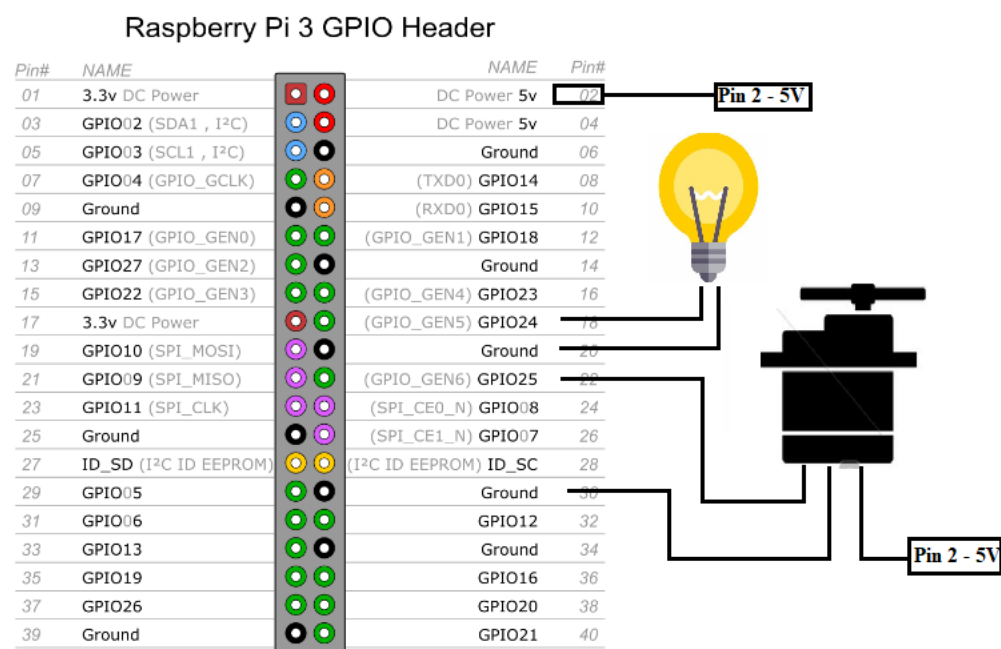
- Để bật tắt đèn cung cấp ánh sáng cho quá trình thu thập ảnh nhận dạng hình ảnh trong môi trường thiếu ánh sáng. Nhóm sử dụng module Relay 1 kênh để điều khiển đèn.



Hình 3.7: Module relay 1 kênh

3.2.4 Kết nối phần cứng:

Phần cứng được thiết kế gồm một Raspberry Pi 3, một camera Raspberry Camera V1 dùng để lấy hình ảnh và nhận dạng trực tiếp từ môi trường, một màn hình cảm ứng điện dung 7inch dùng làm màn điều khiển chọn các chế độ điều khiển, 1 servo dùng để điều khiển mở khóa cửa, 1 bóng đèn để cung cấp ánh sáng khi nhận dạng khuôn mặt trong điều kiện thiếu ánh sáng. Sau đây là hình ảnh sau khi kết nối phần cứng:



Hình 3.8: Cách kết nối phần cứng.

3.3 THIẾT KẾ PHẦN MỀM

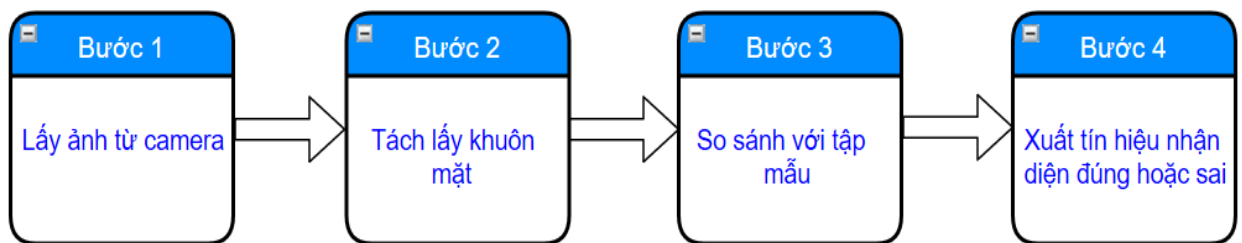
3.3.1 Chức năng hoạt động của phần mềm

Phần mềm được thiết kế gồm hai phần chính:

- Nhận diện khuôn mặt từ camera và hiển thị app điều khiển trên màn hình : chức năng chính là phát hiện khuôn mặt trong ảnh thu từ camera, nhận diện khuôn mặt được phát hiện sau đó đưa ra tín hiệu điều khiển để điều khiển servo hoạt động đóng mở cửa. Sử dụng mô hình mã nhị phân cục bộ (LBPH) để huấn luyện cơ sở dữ liệu.
- Chương trình xử lý trên Raspberry Pi: chức năng chính là nhận hình ảnh từ camera, sau đó tiến hành xử lý nhận dạng khuôn mặt.

3.3.2 Quy trình xử lý của hệ thống

Hệ thống được xử lý theo quy trình sau:



Hình 3.9 : Quy trình của hệ thống

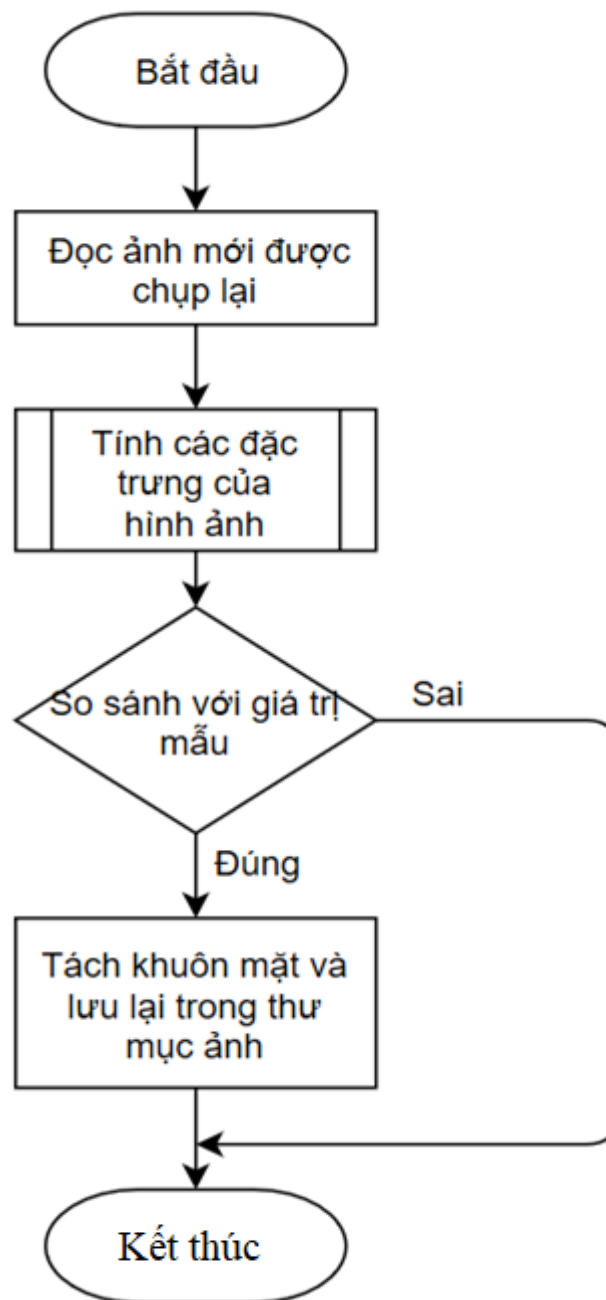
Xử lý nhận diện khuôn mặt: đầu tiên ảnh người nhận diện được thu vào từ camera, tiếp theo chương trình sẽ tìm khuôn mặt có trong ảnh theo phương pháp Viola-Jones. Khuôn mặt sau đó được căn chỉnh và đưa chương trình nhận diện theo phương pháp LBPH.

Sau khi nhận diện xong, nếu kết quả nhận diện đúng thì Raspberry Pi sẽ xuất tín hiệu điều khiển xuống các thiết bị, và ngược lại sẽ báo hiệu sai.

3.3.3 Lưu đồ hoạt động

3.3.3.1 Lưu đồ phát hiện khuôn mặt.

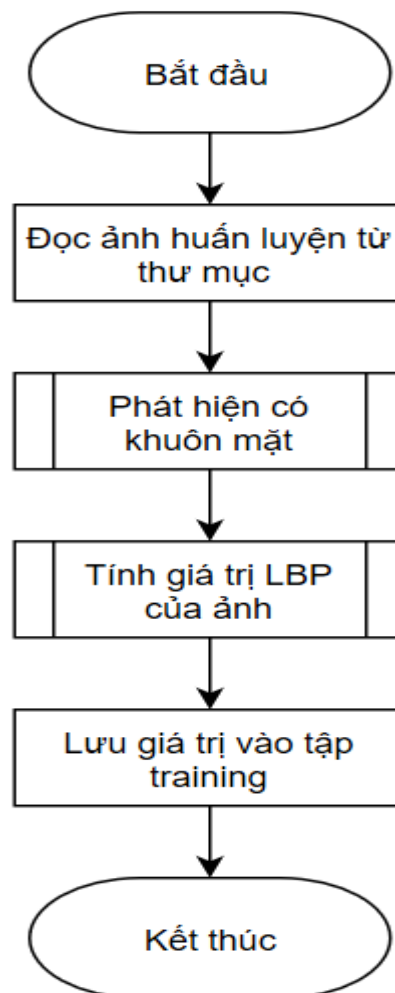
Phát hiện khuôn mặt là bước đầu tiên trong quá trình nhận diện khuôn mặt. Bản chất của phát hiện khuôn mặt là tìm kiếm và định vị trí khuôn mặt trên khung ảnh. Phương pháp phát hiện khuôn mặt được sử dụng là phương pháp sử dụng các đặc trưng Haar-like.



Hình 3.10: Lưu đồ phát hiện khuôn mặt

Ảnh đọc được trực tiếp từ camera sẽ được hóa xám từ ảnh màu RGB, sau đó được tiến hành cân bằng histograms trên ảnh xám để cân bằng độ nhạy cảm với mức sáng. Sau khi phát hiện được khuôn mặt, ta sẽ xác định vị trí khuôn mặt trên ảnh bằng cách vẽ xung quanh khuôn mặt bằng 1 ô vuông (tiến hành tách chọn khuôn mặt có kích thước lớn nhất nếu trong ảnh có nhiều hơn một khuôn mặt). Sau cùng ta sẽ lưu ảnh khuôn mặt tìm được để sử dụng cho việc training dữ liệu.

3.3.3.2 Lưu đồ training dữ liệu.

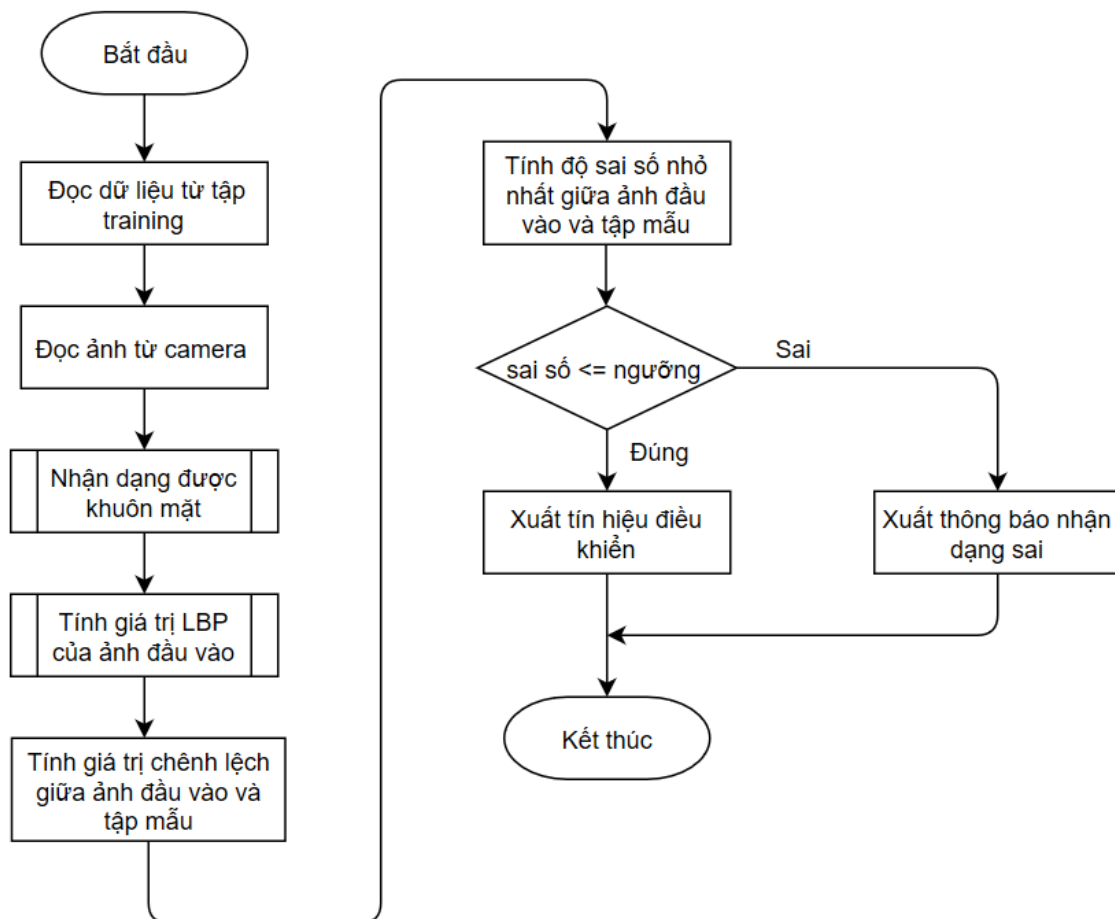


Hình 3.11: Lưu đồ training dữ liệu gốc.

Ảnh huấn luyện được đọc lần lượt từng ảnh. Với mỗi ảnh sẽ được thực hiện các bước sau:

- Phát hiện khuôn mặt: chương trình sẽ dò tìm các khuôn mặt trên ảnh, sau đó đánh dấu khuôn mặt lớn nhất trong ảnh (ảnh có nhiều khuôn mặt) sau đó tách khuôn mặt vừa tìm được.
- Ảnh có chứa khuôn mặt sẽ được cho tính giá trị LBP.
- Sau khi tính giá trị LBP của từng ảnh ta sẽ lưu tất cả các giá trị đó vào một tập tin dữ liệu làm tập cơ sở dữ liệu với đuôi tập là .yaml.

3.3.3.3 Lưu đồ nhận diện khuôn mặt.



Hình 3.12: Lưu đồ nhận diện khuôn mặt

Quá trình nhận diện khuôn mặt bắt đầu với việc lấy thông tin từ tập huấn luyện lên, dữ liệu mẫu chứa các giá trị đã tính LBP của các ảnh mẫu có trong cơ sở dữ liệu.

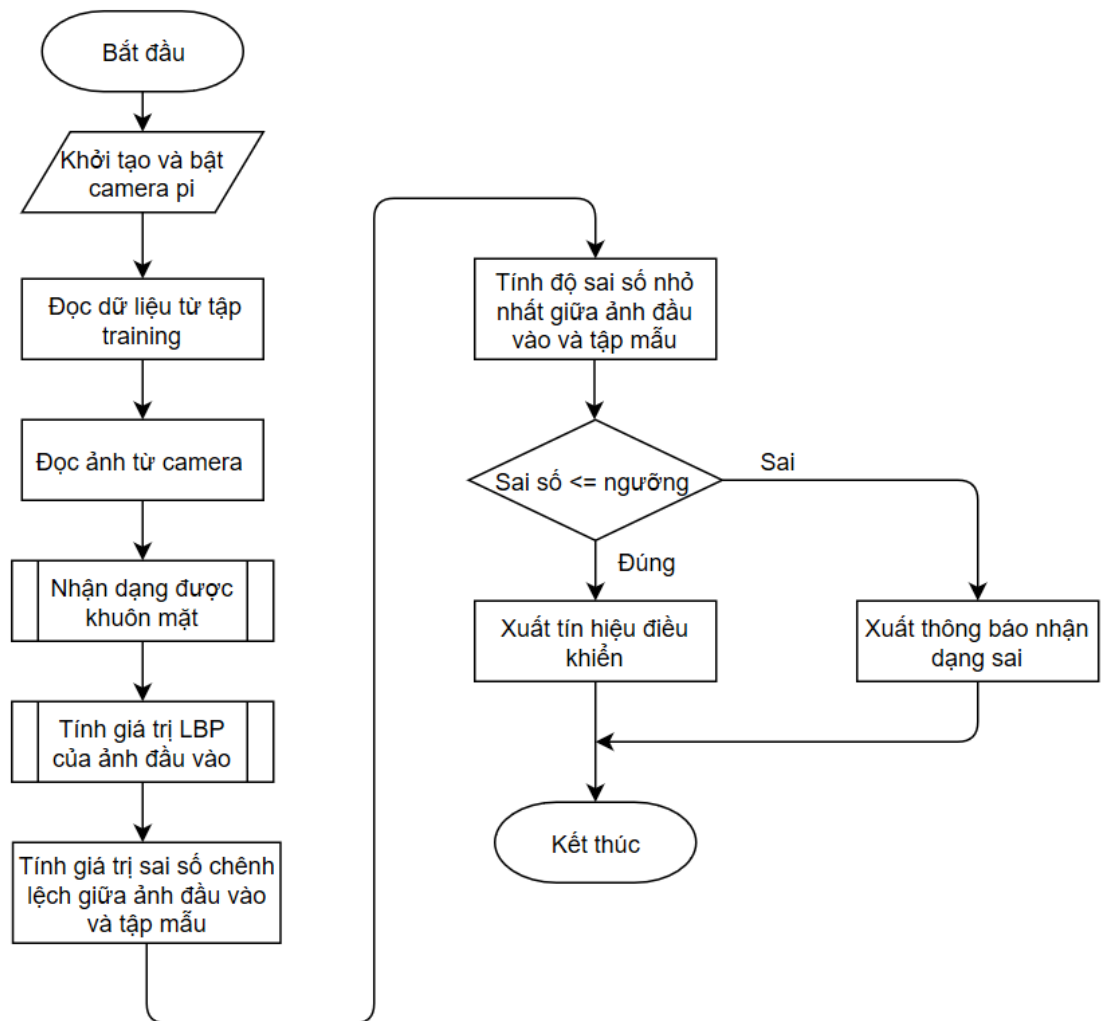
Sau đó sẽ đọc ảnh từ camera để xác định khuôn mặt và lấy ảnh người cần nhận diện.

Tiếp tục tìm khuôn mặt có trong ảnh, sau đó căn chỉnh và tách khuôn mặt. Sau khi tách khuôn mặt sẽ được lấy và xử lý để tính giá trị LBP.

Giá trị LBP vừa tính ra sẽ được mang đi tính độ sai lệch so với các giá trị LBP có trong tập cơ sở dữ liệu. Khi có được tất cả các giá trị độ lệch sai số, ta sẽ tìm giá trị sai số nhỏ nhất giữa ảnh mẫu và ảnh trong CSDL. Độ sai số càng nhỏ chứng tỏ cho độ giống với ảnh trong dữ liệu mẫu càng lớn, và ngược lại nếu sai số càng lớn thì độ giống càng thấp.

Mang giá trị sai số nhỏ nhất so sánh với giá trị ngưỡng, ở đây nhóm đã chọn mức ngưỡng bằng 0.4 (Sẽ được nêu lý do ở bảng bên dưới). Nếu sai số nhỏ hơn mức ngưỡng thì ta kết luận nhận diện ảnh đúng với người trong tập dữ liệu và thực hiện mở cửa nhà. Ngược lại, nếu sai số lớn hơn ngưỡng, ta kết luận nhận diện không đúng người trong cơ sở dữ liệu và không thực hiện mở cửa.

3.3.3.4 Lưu đồ giải thuật trên khối xử lý



Hình 3.13: Lưu đồ giải thuật khối xử lý

Khi thực hiện chương trình sẽ đọc giá trị mẫu đã được lưu, tiếp theo sẽ liên tiếp lấy hình ảnh từ camera, sau đó tính tiến hành tìm khuôn mặt trong ảnh, nếu có khuôn mặt sẽ tiến hành cắt khuôn mặt từ ảnh, sau đó tính các giá trị nhị phân cục bộ của ảnh rồi tìm sai số giữa ảnh đó với những giá trị được lưu.

Nếu giá trị sai số \leq giá trị ngưỡng hệ thống sẽ tiến hành tra dữ liệu có sẵn trong cơ sở dữ liệu rồi ra tín hiệu điều khiển từ vi xử lý xuống các thiết bị nhận tín hiệu.

Ngược lại, hệ thống sẽ báo động không đúng người.

Sau đây là bảng thực nghiệm việc chọn lựa mức ngưỡng để so sánh với sai số giữa hình ảnh đầu vào và ảnh trong tập CSDL :

Giá trị ngưỡng = 0.3	Số lần nhận điện	Nhận diện đúng	Nhận diện sai	Tỉ lệ chính xác
Mặt người lạ	50	50	0	100%
Mặt người quen	50	35	15	70%

Giá trị ngưỡng = 0.35	Số lần nhận điện	Nhận diện đúng	Nhận diện sai	Tỉ lệ chính xác
Mặt người lạ	50	50	0	100%
Mặt người quen	50	41	9	82%

Giá trị ngưỡng = 0.4	Số lần nhận điện	Nhận diện đúng	Nhận diện sai	Tỉ lệ chính xác
Mặt người lạ	50	50	0	100%
Mặt người quen	50	45	5	90%

Giá trị ngưỡng = 0.5	Số lần nhận điện	Nhận diện đúng	Nhận diện sai	Tỉ lệ chính xác
Mặt người lạ	50	46	4	92%
Mặt người quen	50	50	0	100%

Từ bảng dữ liệu thống kê ở trên, nhóm nhận thấy rằng với giá trị ngưỡng ở mức 0.4 thì tỉ lệ nhận dạng chính xác ở mặt người lạ và người quen là phù hợp nhất. Nên nhóm đã chọn mức giá trị ngưỡng là 0.4 để đưa vào chương trình nhận dạng cho mô hình này

Chương 4

KẾT QUẢ

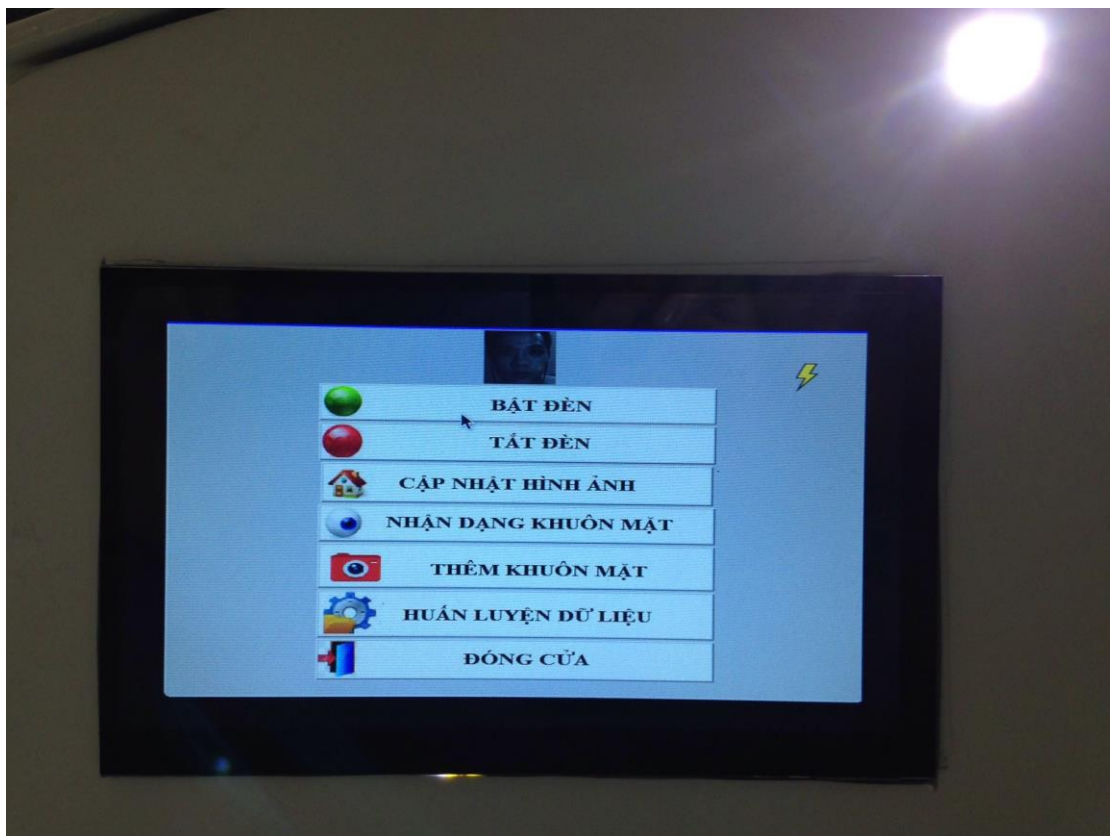
4.1 MÔ HÌNH CỦA HỆ THỐNG

Mô hình của hệ thống gồm các thành phần sau:

- Camera : được gắn ở phía trước mô hình.
- Màn hình được lắp cạnh cửa ra vào.
- Bộ xử lý (nằm bên trong hộp).
- Đèn được lắp phía trên khác phía so với camera.
- Relay và servo mở cửa được đặt phía trong mô hình.
- Mô hình có chức năng: Xử lý hình ảnh ở môi trường để nhận biết khuôn mặt có trong CSDL.
- Hai nút nhấn vật lý được mô phỏng bên ngoài, có chức năng đóng mở cửa từ phía trong nhà.

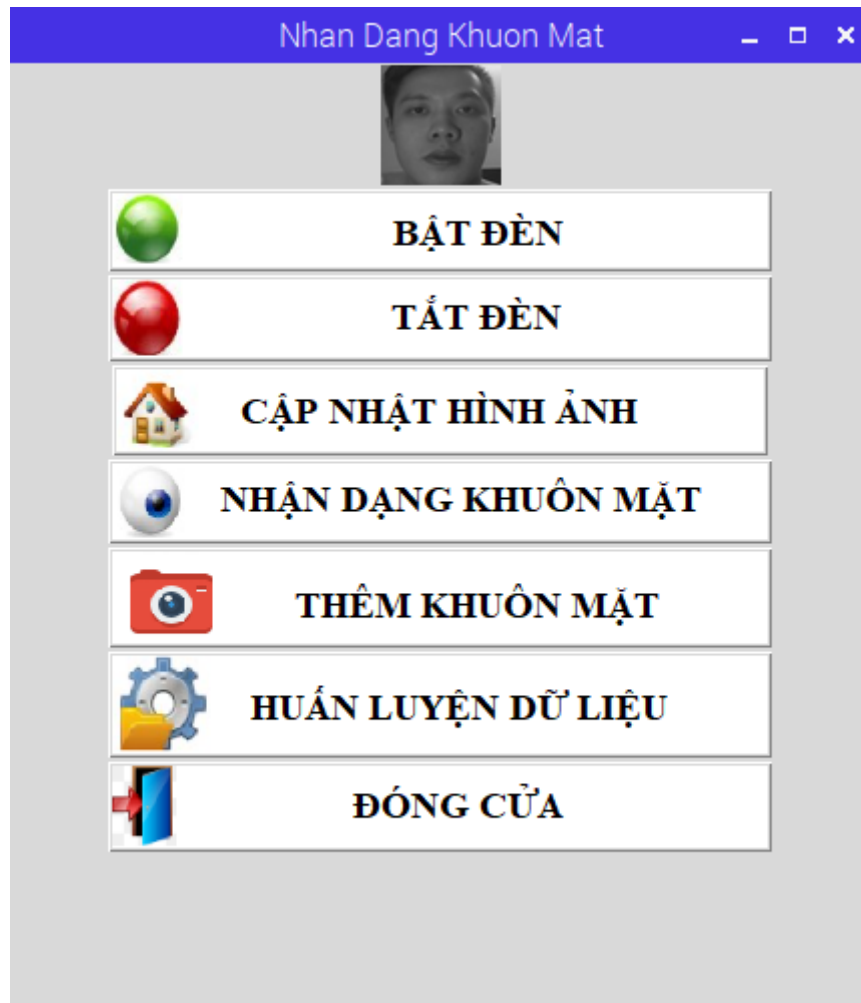


Hình 4.1: Mô hình của hệ thống



Hình 4.2: Màn hình hiển thị

4.2 HOẠT ĐỘNG CỦA HỆ THỐNG



Hình 4.3 : App hiển thị điều khiển.

App hiển thị trên màn hình cảm ứng điện dung bao gồm các chức năng tương tác với người dùng :

- **Chức năng nút bật tắt đèn :** Dùng để bật tắt đèn cung cấp ánh sáng. Khi môi trường thiếu ánh sáng sẽ không hoạt động hiệu quả quá trình nhận dạng khuôn mặt.
- **Chức năng cập nhật hình ảnh :** Dùng để hiển thị ra khuôn mặt của người vừa mới nhận dạng trước đó (bao gồm cả khuôn mặt người có trong CSDL và người lạ).
- **Chức năng nhận dạng khuôn mặt :** Dùng để bắt đầu quá trình khởi động camera, truy cập CSDL và tiến hành nhận dạng khuôn mặt trực

tiếp từ người đối diện camera để mở cửa nhà. Nếu khuôn mặt không nằm trong CSDL thì chuông báo hiệu sẽ reo lên.

- **Chức năng thêm khuôn mặt :** Dùng để thêm khuôn mặt của các thành viên trong gia đình vào CSDL để nhận dạng. (Theo tiến hành khảo sát số khuôn mặt người có thể chứa trong tập CSDL khoảng 9 người. Càng nhiều dữ liệu quá trình khởi động càng lấy nhiều thời gian hơn).
- **Chức năng huấn luyện dữ liệu :** Dùng để bắt đầu lấy hình ảnh chụp được trong dataset bắt đầu chuyển đổi khuôn mặt ra dữ liệu số và đánh dấu số thứ tự ID cho từng tập khuôn mặt trong CSDL và lưu vào tập trainingData.yml).
- **Chức năng đóng cửa :** Dùng để kích hoạt khóa trái cửa bên trong lại khi chủ nhà ra ngoài.

4.3 ĐÁNH GIÁ HOẠT ĐỘNG

Hệ thống đã hoạt động bình thường, tuy nhiên trong một số điều kiện ánh sáng thiếu hoặc quá lớn từ đèn có thể gây ra một số lần nhận diện sai khuôn mặt.

Dữ liệu khuôn mặt có trong cơ sở dữ liệu lớn làm cho quá trình khởi động camera, đọc file dữ liệu khuôn mặt từ CSDL tốn thời gian khá lâu.

Dưới đây là bảng thống kê số lần nhận diện đúng sai với khuôn mặt chụp được là 50 lần và giá trị ngưỡng nhóm chọn là 0.4 :

Giá trị ngưỡng = 0.4	Số lần nhận diện	Nhận diện đúng	Nhận diện sai	Tỉ lệ chính xác
Mặt người lạ	50	50	0	100%
Mặt người quen	50	45	5	90%

Chương 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 KẾT LUẬN

Sau quá trình tìm hiểu, nghiên cứu lý thuyết về phương pháp phát hiện khuôn mặt, các giải thuật để xử lý nhận diện khuôn mặt, sử dụng thư viện mã nguồn mở OpenCV, thư viện để tạo app điều khiển Tkinter, ngôn ngữ lập trình Python, sau đó tiến hành thực nghiệm, đã xây dựng đề tài “Ứng dụng khóa cửa thông minh nhận diện khuôn mặt người.” dựa theo mục tiêu đã thực hiện được các chức năng sau:

- App hiển thị điều khiển chương trình được sử dụng để chọn các mục nhận dạng, thêm mặt người, training, bật, tắt đèn trên màn hình cảm ứng điện dung.
- Xử lý ảnh thu được từ camera sau cho xuất tín hiệu điều khiển khi hình ảnh từ môi trường đúng với hình ảnh nằm trong cơ sở dữ liệu.
- Chương trình xử lý ảnh có khả năng nhận diện được khuôn mặt người trong CSDL.
- Mô hình nhận diện được khuôn mặt có trong CSDL và phát tín hiệu điều khiển đến khối nhận tín hiệu.

Độ chính xác của mô hình:

- Mô hình chưa có độ chính xác cao khi khuôn mặt nằm ở góc nghiêng rộng quá 20 độ, trong điều kiện thiếu sáng.
- Khi có hai người cùng xuất hiện thì sẽ xuất kết quả có giá trị sai số thấp nhất nếu hai người đều có trong CSDL. Nếu một người có trong CSDL còn một người không có thì sẽ xuất kết quả của người có trong CSDL.
- Độ chính xác cũng do sự điều chỉnh mức ngưỡng tác động vào.

Kết quả đạt được một số ưu điểm như sau:

- Nhận dạng có độ chính xác rất cao với ánh sáng đầy đủ, phù hợp.
- Có app hiển thị giao diện người dùng trên màn hình cảm ứng.

Bên cạnh đó vẫn còn những nhược điểm:

- Quá trình nhận diện bị ảnh hưởng lớn bởi môi trường bên ngoài đặc biệt là độ sáng. Khi độ sáng quá cao hoặc quá thấp dẫn đến khả năng nhận diện sai.
- Chất lượng camera cũng ảnh hưởng đến kết quả nhận diện.
- Chương trình chưa có chức năng xác nhận bảo mật khi thêm khuôn mặt.

5.2 HƯỚNG PHÁT TRIỂN

Với thời gian thực hiện có hạn và kiến thức vẫn còn ít nên nghiên cứu chỉ dừng lại ở các chức năng cơ bản và bám theo mục tiêu đã đề ra. Dựa vào những công việc đã thực hiện, nghiên cứu nhận thấy hệ thống có thể phát triển thêm các nội dung sau:

Hoàn thiện được chức năng xử lý ảnh người lạ không nhận diện được và báo lên hệ thống email của người dùng để cảnh báo người lạ xâm nhập hệ thống.

Nâng cấp bộ vi xử lý đời mới hơn để có thể xử lý nhận dạng nhanh hơn và camera độ phân giải cao hơn để có thể nhận dạng chính xác hơn.

Phát triển app điều khiển thêm các chức năng bảo mật với mật khẩu khi thêm khuôn mặt.

Mở cửa thủ công với nút nhấn và mật khẩu bảo mật khi camera có trực trực.

TÀI LIỆU THAM KHẢO

- [1] P. Viola and M. Jones, “Robust real-time face detection” International Journal of Computer Vision, vol. 57, no. 2, pp.137-154, 2004. 1, 2
- [2] Florian Schroff, Dmitry Kalenichenko, James Philbin. “*FaceNet: A Unified Embedding for Face Recognition and Clustering*”, 12 Mar 2015.
- [3] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, “*Deep Face Recognition*”, 2015
- [4] Kelvin Salton. “*Face Recognition: Understanding LBPH Algorithm*”, Nov 11, 2017.
- [5] Viola. P, Jones. M, Rapid object detection using a boosted cascade of simple features, May 2004.
- [6] Viola. P, Jones. M, Robust Real-time Object Detection, Second international workshop on statistical and computational theories of vision - modeling, learning, computing, and sampling. Vancouver, Canada, July 13, 2001.
- [7] Viola. P, Jones. M, Robust real-time face detection, International Journal of Computer Vision 57(2), 2004.
- [8] Jason Corso. “*AdaBoost for Face Detection*”, 2014
- [9] Ming-Hsuan Yang, Narendra Ahuja, and David Kriegman. “A Survey on Face Detection Methods”, March 6 1999.
- [10] S.A.A.M Faudzi and N.Yahya, “Evaluation of LBP – Based Face Recognition Techniques”, International Conference of Intelligent and Advanced System (ICIAS), 2014, pp. 1-6, ISBN:978-1-4799-4653-2.
- [11] www.sensetime.com, <https://www.sensetime.com/core>, 2014 - 2018

