

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN TỬ - VIỄN THÔNG**

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: ĐIỆN TỬ - VIỄN THÔNG**  
**CHUYÊN NGÀNH: KỸ THUẬT MÁY TÍNH**

**ĐỀ TÀI:**

## **NHẬN DIỆN KHUÔN MẶT BẰNG RASPBERRY PI4**

Sinh viên thực hiện: <b>LÊ VĂN MÃN</b>	<b>106180094</b>
<b>NGUYỄN ANH QUÂN</b>	<b>106180106</b>
<b>LÊ TUẤN ANH</b>	<b>106180064</b>
Người hướng dẫn: <b>ThS. THÁI VĂN TIẾN</b>	

Lớp: **18DT2**

**Đà Nẵng, 12/2022**

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA ĐIỆN TỬ - VIỄN THÔNG**

# **ĐỒ ÁN TỐT NGHIỆP**

**NGÀNH: ĐIỆN TỬ - VIỄN THÔNG**  
**CHUYÊN NGÀNH: KỸ THUẬT MÁY TÍNH**

**ĐỀ TÀI:**

## **NHẬN DIỆN KHUÔN MẶT BẰNG RASPBERRY PI4**

Sinh viên thực hiện: **LÊ VĂN MÃN** **106180094**

**NGUYỄN ANH QUÂN** **106180106**

**LÊ TUẤN ANH** **106180064**

Người hướng dẫn: **ThS. THÁI VĂN TIẾN**

Lớp: **18DT2**

**Đà Nẵng, 12/2022**

## LỜI NÓI ĐẦU

Đầu tiên, nhóm xin chân thành cảm ơn quý thầy cô khoa Điện tử - Viễn thông, trường Đại học Bách Khoa – Đại Học Đà Nẵng đã tận tình giảng dạy, truyền đạt kiến thức trong suốt thời gian học tập tại trường. Thầy cô không chỉ truyền cho chúng tôi kiến thức qua sách vở mà còn là kiến thức về kỹ năng sống, kỹ năng giao tiếp giúp chúng tôi trưởng thành hơn rất nhiều.

Nhóm xin chân thành cảm ơn thầy Ths.Thái Văn Tiến đã luôn tận tình hướng dẫn, giúp đỡ và động viên chúng tôi rất nhiều trong suốt quá trình thực hiện và hoàn thành đồ án. Những điều thầy chỉ dạy giúp chúng tôi củng cố thêm rất nhiều về kiến thức lý thuyết và kiến thức thực tiễn.

Bên cạnh đó, chúng tôi cũng dành lời cảm ơn đến các bạn trong nhóm đồ án do thầy Thái Văn Tiến hướng dẫn đã cùng nhau giúp đỡ, chia sẻ kiến thức trong quá trình thực hiện đồ án.

Trong quá trình thực hiện đồ án, chúng tôi đã cố gắng tìm tòi, nghiên cứu các tài liệu, làm việc một cách nghiêm túc, với mong muốn hoàn thành đồ án tốt nhất. Tuy nhiên, trong thời gian thực hiện đồ án, dù đã có nhiều cố gắng nhưng do còn nhiều hạn chế về kiến thức chuyên môn cũng như kiến thức thực tiễn nên đề tài của nhóm không tránh khỏi những sai sót. Mong quý thầy cô và các bạn đọc góp ý, xây dựng để đề tài này được hoàn thiện hơn.

Cuối cùng, chúng tôi xin chân thành cảm ơn đến các thầy đã tận tình truyền đạt lại những kiến thức hữu ích cho chúng tôi. Đặc biệt chúng tôi xin cảm ơn đến thầy Thái Văn Tiến đã quan tâm cung cấp tài liệu, nhiệt tình hướng dẫn trong quá trình làm đồ án.

Chúng tôi xin chân thành cảm ơn!

Đà Nẵng, ngày      tháng      năm 2023

Sinh viên thực hiện

## TÓM TẮT

Tên đề tài: **Nhận diện khuôn mặt bằng Raspberry pi 4**

Sinh viên thực hiện: Lê Văn Mẫn - Nguyễn Anh Quân - Lê Tuấn Anh

Số thẻ SV: 106180094-106180106-106180064

Lớp: 18DT2

Công nghệ khoa học ngày càng phát triển và có vai trò hết sức quan trọng không thể thiếu trong cuộc sống hiện đại. Con người ngày càng tạo ra những cỗ máy thông minh có khả năng tự nhận biết và xử lý được các công việc một cách tự động, phục vụ cho lợi ích của con người. Thừa hưởng những thành tựu của nền khoa học và kỹ thuật phát triển trong thời đại công nghệ 4.0, trong những năm gần đây, một trong những bài toán nhận được nhiều sự quan tâm và tốn nhiều công sức nhất của lĩnh vực công nghệ thông tin, đó chính là bài toán nhận dạng. Tuy mới xuất hiện chưa lâu nhưng nó đã rất được quan tâm vì tính ứng dụng thực tế của bài toán cũng như sự phức tạp của nó. Bài toán nhận dạng có rất nhiều lĩnh vực như: nhận dạng vật chất (nước, lửa, đất, đá, gỗ...) nhận dạng chữ viết, nhận dạng giọng nói, nhận dạng hình dáng, nhận dạng khuôn mặt... trong đó phổ biến và được ứng dụng nhiều hơn cả là bài toán nhận dạng khuôn mặt.

Nhận dạng khuôn mặt là một công nghệ hữu ích được ứng dụng nhiều trong đời sống hằng ngày của con người như các hệ thống giám sát tại các tòa nhà, sân bay, trạm ATM, hệ thống chấm công, camera chống trộm, xác thực danh tính,... Ngoài việc được ứng dụng trong quản lý nhân sự, thì nó còn là sự lựa chọn của rất nhiều đơn vị hoạt động trong lĩnh vực an ninh, bảo mật.

Nhóm chọn đề tài “**NHẬN DIỆN KHUÔN MẶT BẰNG RASPBERRY PI 4**” với mạng MTCNN và FaceNet làm đề án tốt nghiệp.

**Mục tiêu đề tài:** Nội dung đề tài là tìm hiểu các phương pháp, công nghệ, giải pháp phần cứng, phần mềm và công cụ sử dụng để thực hiện đề tài. Lựa chọn những công nghệ tối ưu và hiện đại để bắt kịp xu thế phát triển không ngừng của công nghệ hiện nay. Sau khi đã nắm bắt được công nghệ và hướng đi, nhóm bắt tay vào kế hoạch triển khai đề tài, đưa ra những công việc cụ thể cần thực hiện và tiến hành tìm hiểu các thuật toán, phương pháp, tính khả thi cũng như bất khả thi của những công việc cần thực hiện.

Nội dung báo cáo đề án gồm có 4 chương:

- ❖ **Chương 1:** Tổng quan.
- ❖ **Chương 2:** Cơ sở lý thuyết.

- ❖ **Chương 3:** Xây dựng hệ thống.
- ❖ **Chương 4:** Thực nghiệm, phân tích và đánh giá kết quả.

## LỜI CAM ĐOAN

Kính gửi: Hội đồng bảo vệ đồ án tốt nghiệp khoa Điện tử - Viễn thông, Trường Đại học Bách Khoa – Đại học Đà Nẵng.

Chúng tôi: Lê Tuấn Anh, Lê Văn Mẫn, Nguyễn Anh Quân, Sinh viên lớp 18DT2, Khoa Điện tử - Viễn thông, Trường Đại học Bách Khoa – Đại học Đà Nẵng.

Trong quá trình thực hiện đề tài, nhóm chúng tôi đã tham khảo những nguồn tài liệu sách giáo khoa chính thống, các giáo trình của các phần đã học, các nghiên cứu và các bài báo uy tín. Chúng tôi xin cam đoan nội dung của đồ án này không phải là bản sao chép của bất cứ đồ án hoặc công trình nào đã có từ trước. Nếu vi phạm chúng tôi xin chịu mọi hình thức kỷ luật của Khoa.

*Đà Nẵng, ngày      tháng      năm 2023*

**Sinh viên thực hiện**

Lê Văn Mẫn

Lê Tuấn Anh

Nguyễn Anh Quân

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN TỬ - VIỄN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
Độc lập - Tự do - Hạnh phúc

### NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ tên sinh viên	Số thẻ SV	Lớp	Ngành
1	Lê Văn Mẫn	106180094	18DT2	Kỹ thuật máy tính
2	Nguyễn Anh Quân	106180106	18DT2	Kỹ thuật máy tính
3	Lê Tuấn Anh	106180064	18DT2	Kỹ thuật máy tính

1. Tên đề tài đồ án:

.....

2. Đề tài thuộc diện: ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

.....

4. Nội dung các phần thuyết minh và tính toán:

a. Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Lê Văn Mẫn	Tìm hiểu tổng quan về đề tài. Thu thập, chuẩn bị cơ sở dữ liệu cho các chương trình đào tạo và đánh giá kết quả.
2	Nguyễn Anh Quân	Tiến hành xây dựng mô hình hệ thống mở cửa tự động.
3	Lê Tuấn Anh	Viết báo cáo.

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung
01	Lê Văn Mẫn	Cơ sở lý thuyết: - Tìm hiểu mạng nơ-ron CNN, mạng FaceNet - Tìm hiểu thuật toán MTCNN
02	Nguyễn Anh Quân	- Tìm hiểu nguyên lý hoạt động và cách kết nối các thiết bị phần cứng với Raspberry Pi - Tìm hiểu và cài đặt Raspberry Pi
03	Lê Tuấn Anh	Tìm hiểu về phần mềm: - Tìm hiểu code python để triển khai thuật toán

5. Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):

a. Phần chung:

TT	Họ tên sinh viên	Nội dung

b. Phần riêng:

TT	Họ tên sinh viên	Nội dung

6. Họ tên người hướng dẫn:	Phần/ Nội dung:
Ths. Thái Văn Tiến	

7. Ngày giao nhiệm vụ đồ án: ...../...../.....

8. Ngày hoàn thành đồ án: ...../...../.....

Đà Nẵng, ngày tháng năm 2023

**Trưởng Bộ môn.....**

**Người hướng dẫn**



## MỤC LỤC

<b>LỜI NÓI ĐẦU .....</b>	<b>i</b>
<b>LỜI CAM ĐOAN .....</b>	<b>iv</b>
<b>MỤC LỤC .....</b>	<b>vii</b>
<b>MỞ ĐẦU .....</b>	<b>ix</b>
<b>DANH SÁCH HÌNH ẢNH .....</b>	<b>xi</b>
<b>DANH SÁCH CHỮ VIẾT TẮT .....</b>	<b>xiv</b>
<b>CHƯƠNG 1: TỔNG QUAN .....</b>	<b>1</b>
1.1 Mở đầu chương .....	1
1.2 Tổng quan hệ thống .....	1
1.3. Máy học là gì? .....	1
1.4 Phân nhánh máy học .....	2
1.5 Tổng quan các phương pháp nhận dạng .....	4
1.6 Phương pháp tiếp cận .....	5
1.7 Một số thư viện học máy hiện nay .....	5
1.7.1 TensorFlow .....	5
1.7.2 Scikit-learn .....	5
1.7.3 Keras .....	6
1.7.5 PyTorch .....	6
1.8 Quy trình thiết kế .....	7
1.9 Công cụ nguyên liệu cần triển khai .....	7
1.10 Kết luận chương .....	11
<b>CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....</b>	<b>12</b>
2.1 Mở đầu chương .....	12
2.2 Giới thiệu về phần cứng của hệ thống .....	12
2.2.1 Sơ đồ kết nối giữa Raspberry pi và các thiết bị ngoại vi .....	12
2.2.2 Giới thiệu về các cổng GPIO của Raspberry Pi .....	13
2.2.3 Giới thiệu về relay module .....	15
2.2.4 Giới thiệu về cảm biến HC-SR04. ....	17
2.2.5 Giới thiệu về bàn phím ma trận keypad 4x3 .....	19
2.3 Tìm hiểu về Open CV .....	21
2.4.1 Lớp tích chập (Convolutional layer) .....	23

2.4.1.1 Tầng tích chập .....	24
2.4.1.2 Các tham số của tầng tích chập .....	25
2.4.2 Lớp phi tuyến (Nonlinear Layer) .....	26
2.4.3 Lớp tổng hợp (Pooling layer) .....	27
2.4.4 Lớp kết nối toàn bộ (Fully Connected) .....	27
2.5 Phát hiện khuôn mặt MTCNN( Face Detector) .....	27
2.6 Tìm hiểu mô hình Pre-trained model - FaceNet .....	34
2.7 Thuật toán Triplet loss trong trích xuất đặc trưng FaceNet .....	35
2.8 Kỹ thuật căn chỉnh khuôn mặt (Face alignment): .....	37
2.9 Tìm hiểu Support Vector Machine .....	38
2.10 Kết luận chương .....	41
<b>CHƯƠNG 3: XÂY DỰNG HỆ THỐNG .....</b>	<b>42</b>
3.1 Mở đầu chương .....	42
3.2 Sơ đồ khối của hệ thống .....	42
3.3 Lưu đồ hệ thống mở cửa tự động bằng phương pháp nhận dạng khuôn mặt .....	44
3.4 Cơ sở dữ liệu ảnh .....	46
3.5 Phát hiện khuôn mặt .....	47
3.6 Trích xuất đặc trưng cho khuôn mặt .....	48
3.7 Nhận dạng khuôn mặt .....	48
3.8 Kết luận chương .....	49
<b>CHƯƠNG 4: THỰC NGHIỆM, PHÂN TÍCH VÀ ĐÁNH GIÁ KẾT QUẢ .....</b>	<b>50</b>
4.1 Mở đầu chương .....	50
4.2 Mô hình hệ thống .....	50
4.3 Thực nghiệm trên Raspberry .....	50
4.4 Các phương pháp tối ưu hệ thống .....	54
4.4.1 Train lại model tăng số lượng ảnh. ....	54
4.4.2 Thay đổi các trọng số trong thuật toán MTCNN .....	55
4.4.3 Phân luồng trên Raspberry Pi 4 Model B 4GB (Threading) .....	57
4.6 Đánh giá hệ thống .....	58
4.7 Kết luận chương .....	61
<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI .....</b>	<b>62</b>
<b>PHỤ LỤC .....</b>	<b>64</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>65</b>

## MỞ ĐẦU

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

Nhận dạng khuôn mặt là một công nghệ hữu ích được ứng dụng nhiều trong đời sống hằng ngày của con người như các hệ thống giám sát tại các tòa nhà, sân bay, trạm ATM, hệ thống chấm công, camera chống trộm, xác thực danh tính,... Ngoài được ứng dụng trong việc quản lý nhân sự ra thì nó còn là sự lựa chọn của rất nhiều đơn vị hoạt động trong lĩnh vực an ninh, bảo mật. Hệ thống mở cửa tự động ngày nay rất là phổ biến và ngày càng được sử dụng rộng rãi ngoài xã hội. Việc sử dụng những cửa tự động không chỉ góp phần tạo nên sự sang trọng mà còn giúp tiết kiệm thời gian, đem đến sự tiện lợi cho người sử dụng. Chính vì vậy mà từ khi xuất hiện, cửa tự động ngày càng được phổ biến và được cải tiến hiện đại để đáp ứng nhu cầu của người sử dụng.

Nhằm tăng khả năng bảo mật và áp dụng những công nghệ tiên bộ vào hệ thống mở cửa tự động, nhóm chọn đề tài **“NHẬN DIỆN KHUÔN MẶT BẰNG RASPBERRY PI4”** với mạng MTCNN và FaceNet làm đồ án tốt nghiệp.

**Mục tiêu đề tài:** Nội dung đề tài là tìm hiểu các phương pháp, công nghệ, giải pháp phần cứng, phần mềm và công cụ sử dụng để thực hiện đề tài. Lựa chọn những công nghệ tối ưu và hiện đại để bắt kịp xu thế phát triển không ngừng của công nghệ hiện nay. Sau khi đã nắm bắt được công nghệ và hướng đi, nhóm bắt tay vào kế hoạch triển khai dự án, đưa ra những công việc cụ thể cần thực hiện và tiến hành tìm hiểu các thuật toán, phương pháp, tính khả thi cũng như bất khả thi của những công việc cần thực hiện. Tiếp theo, nhóm sẽ phân tích, tiến hành xây dựng hệ thống, từng bước hoàn chỉnh, liên kết các khối với nhau thành một mô hình hoàn chỉnh. Khi đã xây dựng hoàn chỉnh phần mềm, phần cứng, nhóm tiến hành thử nghiệm, chỉnh sửa, cập nhật sai sót để cho ra mô hình hoàn thiện nhất.

Nội dung báo cáo đồ án gồm có 4 chương:

- ❖ **Chương 1:** Tổng quan.
- ❖ **Chương 2:** Cơ sở lý thuyết.
- ❖ **Chương 3:** Phương pháp thực hiện.
- ❖ **Chương 4:** Thực nghiệm, phân tích và đánh giá kết quả.

## DANH SÁCH HÌNH ẢNH

Hình 1. 1: Mối quan hệ của các lĩnh vực AI, ML và DL.....	2
Hình 1. 2: Cấu trúc phân nhánh của Machine Learning .....	3
Hình 1. 3: Sơ đồ hoạt động của ONNX.....	6
Hình 1. 4: Raspberry Pi 4 model B ram 4gb.....	7
Hình 1. 5: Màn hình 5 inch Waveshare.....	8
Hình 1. 6: Webcam Gucce.....	8
Hình 1. 7: Cảm biến HC-SR04 .....	9
Hình 1. 8: Khóa điện từ ly .....	9
Hình 1. 9: Mô-đun relay chuyển tiếp 5V .....	10
Hình 1. 10: Adapter 12V .....	10
Hình 1. 11: Keypad 4x3.....	10
Hình 2. 1: Sơ đồ kết nối giữa Raspberry pi và các thiết bị ngoại vi .....	12
Hình 2. 2: Các chân của GPIO trên Raspberry Pi .....	13
Hình 2. 3: Sơ đồ chân GPIO của Raspberry Pi .....	13
Hình 2. 4: Các thông số trong các chuẩn giao tiếp của Raspberry Pi .....	15
Hình 2. 5: Giới thiệu relay module[4] .....	16
Hình 2. 6: Sơ đồ nối chân giữa Raspberry Pi với relay module .....	17
Hình 2. 7: Giới thiệu về cảm biến HC-SR04[7].....	18
Hình 2. 8: Sơ đồ nối chân cảm biến HC-SR04.....	18
Hình 2. 9: Giới thiệu về bàn phím ma trận 4x3[9] .....	19
Hình 2. 10: Cấu tạo của bàn phím ma trận 4x3 [10].....	20
Hình 2. 11: Sơ đồ nối chân giữa Raspberry Pi với keypad.....	20
Hình 2. 12: Cấu trúc các phần của OpenCV [12] .....	22
Hình 2. 13: Mô hình mạng CNN [13] .....	23
Hình 2. 14: Feature map [14].....	24
Hình 2. 15: Bộ lọc phát hiện cạnh.....	25
Hình 2. 16: Ví dụ về bộ lọc phát hiện cạnh.....	25

Hình 2. 17: Cách hoạt động của bộ lọc.....	26
Hình 2. 18: Quá trình tổng hợp.....	27
Hình 2. 19: Sơ đồ phát hiện khuôn mặt MTCNN .....	28
Hình 2. 20: Image Pyramid.....	28
Hình 2. 21: Mạng neural P-Net.....	29
Hình 2. 22: Quá trình phát hiện khuôn mặt trong mạng đề xuất P-Net.....	30
Hình 2. 23: NMS loại bỏ những box trùng nhau.....	30
Hình 2. 24: Quá trình phát hiện khuôn mặt trong mạng lọc R-Net .....	31
Hình 2. 25: Mạng neural R-Net.....	31
Hình 2. 26: Mạng neural O-Net .....	32
Hình 2. 27: Quá trình phát hiện khuôn mặt trong mạng đầu ra O-Net. ....	33
Hình 2. 28: Ảnh sau khi qua O-Net.....	33
Hình 2. 29: FaceNet lấy hình ảnh khuôn mặt làm đầu vào và xuất ra vector embedding .....	34
Hình 2.30 :Triplet loss trên hai positive faces-mặt tích cực và một negative face-mặt tiêu cực .....	35
Hình 2. 31: Lựa chọn ảnh thực hiện Triplet loss .....	36
Hình 2. 32: Triplet loss trước và sau khi được đào tạo[18] .....	37
Hình 2. 33: Mặt phẳng phân cách 2 lớp .....	39
Hình 2. 34: Phân tích bài toán SVM .....	40
Hình 3. 1: Sơ đồ khối thiết bị. ....	42
Hình 3. 2: Sơ đồ khối của hệ thống mở cửa.....	43
Hình 3. 3: Lưu đồ hệ thống nhận dạng khuôn mặt mở cửa.....	45
Hình 3. 4: Dữ liệu đã thu thập .....	47
Hình 3. 5 : Phát hiện khuôn mặt bằng MTCNN.....	48
Hình 3. 6: Trích xuất vector bằng FaceNet.....	48
Hình 3. 7: Đối tượng sau khi đã nhận dạng.....	49
Hình 4. 1: Mô hình hệ thống .....	50
Hình 4. 2: Hình minh họa quá trình căn chỉnh ảnh .....	51

Hình 4. 3: Ảnh trước khi được căn chỉnh.....	52
Hình 4. 4: Ảnh sau khi được căn chỉnh.....	52
Hình 4. 5: Hình ảnh minh họa chạy chương trình train model.....	53
Hình 4. 6: Hình ảnh minh họa chạy chương trình phân loại model.....	53
Hình 4. 7: Hình ảnh sử dụng camera lấy nguồn đầu vào.....	54
Hình 4. 8: Tỷ lệ thời gian của ba phần của mạng MTCNN.[21].....	55
Hình 4. 9: Biểu đồ so sánh hiệu suất thuật toán dưới các tham số minsize khác nhau.....	56
Hình 4. 10: Biểu đồ so sánh hiệu suất thuật toán dưới các ngưỡng khác nhau.....	57
Hình 4. 11: Các biến trọng số của thuật toán MTCNN.....	57
Hình 4. 12: Kết quả của việc tối ưu trên Raspberry Pi4.....	58

## DANH SÁCH BẢNG BIỂU

Bảng 4. 2: So sánh hiệu suất thuật toán dưới các tham số minsize khác nhau.....	55
Bảng 4. 3: So sánh hiệu suất thuật toán dưới các ngưỡng khác nhau.....	56
Bảng 4. 1: Đánh giá hệ thống.....	59

## DANH SÁCH CHỮ VIẾT TẮT

Từ viết tắt	Tên
Open Cv	Open Source Computer Vision Library (Thư viện nguồn mở cho Computer Vision và Machine Learning)
PCA	Principle Components Analysis (Phép phân tích thành phần chính)
LDA	Linear Discriminant Analysis (Phân tích biệt thức tuyến tính)
CNN	Convolution Neural Network (Mạng thần kinh tích chập)
GPIO	General Purpose Input/Output
MTCNN	Multi-task cascaded convolutional neural network (Mạng nơron tích chập xếp chồng)
SVM	Support vector machine (Máy vector hỗ trợ)
RELU	Rectified Linear Unit
SPI	Serial peripheral Interface (Giao diện ngoại vi nối tiếp)
IoTs	Internet of Things (Internet vạn vật)
TF	TensorFlow
ML	Machine Learning ( Học máy)
GPU	Graphics Processing Unit (Bộ phận xử lý đồ họa)



## CHƯƠNG 1: TỔNG QUAN

### 1.1 Mở đầu chương

Trong chương này, nhóm sẽ giới thiệu tổng quát về hệ thống mở cửa tự động bằng phương pháp nhận dạng khuôn mặt. Để có cái nhìn tổng quan về hệ thống, nhóm sẽ bắt đầu đi vào việc giới thiệu đề tài. Tiếp theo, nhóm sẽ đi vào giới thiệu sơ lược về các phương pháp nhận dạng hiện có và giải pháp sơ bộ sử dụng trong hệ thống.

### 1.2 Tổng quan hệ thống

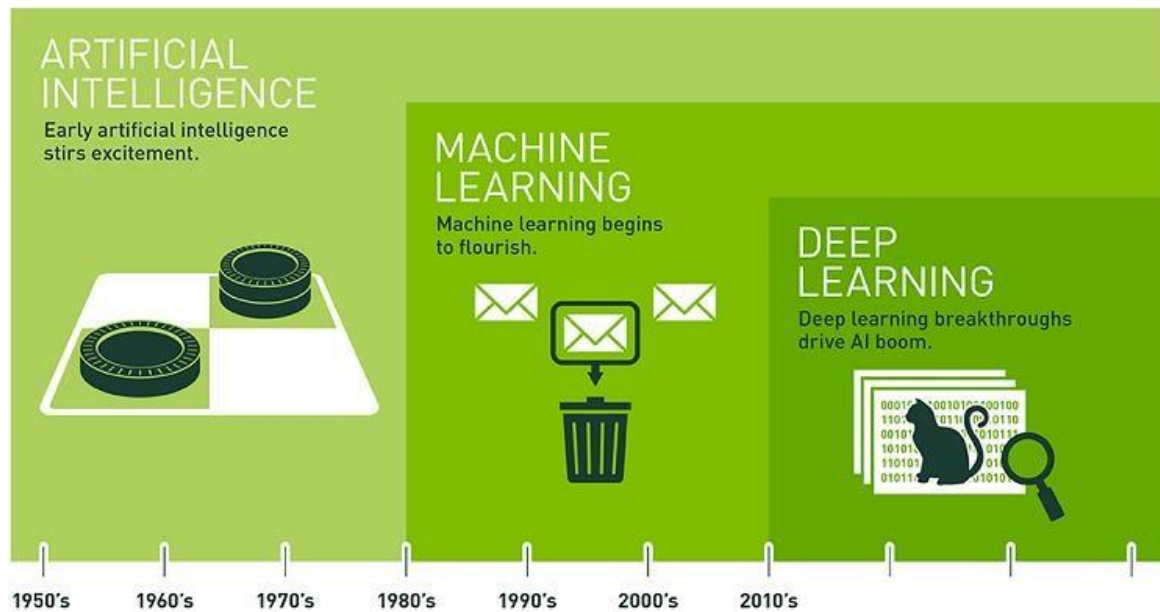
Nhờ vào sự phát triển của khoa học công nghệ, nhận dạng khuôn mặt đang được ứng dụng rộng rãi trong đời sống hàng ngày của con người như các hệ thống giám sát tại các tòa nhà, sân bay, trạm ATM..., hệ thống chấm công, camera chống trộm, xác thực danh tính ở các công ty, nhà máy, xí nghiệp. Một trong số đó là hệ thống mở cửa tự động bằng phương pháp nhận dạng khuôn mặt mà nhóm đang nghiên cứu và thực hiện.

Hệ thống sử dụng Raspberry Pi 4B và các thiết bị ngoại vi như relay module, cảm biến khoảng cách,... Khi hệ thống phát hiện và nhận dạng được khuôn mặt thì sẽ điều chỉnh relay module đóng/mở, từ đó làm khóa điện từ đóng/mở theo. Khi hệ thống không nhận dạng được khuôn mặt, người dùng tiến hành nhập mật khẩu bằng bàn phím để đóng mở khóa điện từ.

### 1.3. Máy học là gì?

Ngày nay, *Học máy (Machine Learning)* đang là một làn sóng mới, được ứng dụng rộng rãi vào đời sống của con người và mọi mặt của xã hội, như trong nhận diện khuôn mặt, xe hơi tự lái, đề xuất mua hàng trên Lazada, Tiki, Amazon, và hàng loạt các ứng dụng Trí tuệ nhân tạo tiên tiến khác. Machine learning tập trung vào việc tạo ra các hệ thống, cỗ máy có khả năng tự mình học hỏi mà không cần phải được lập trình một cách cụ thể. Đây là việc tạo ra các cỗ máy có khả năng tự học hỏi dựa trên các kinh nghiệm mà chúng thu thập được trong suốt quá trình hoạt động, điều này mô phỏng gần giống với hoạt động trí tuệ của con người.

Về định nghĩa khái niệm, máy học là một lĩnh vực thuộc Khoa học máy tính, đồng thời là một lĩnh vực con của *Trí tuệ nhân tạo (AI)*. Mối quan hệ giữa *Trí tuệ nhân tạo*, *Học máy* và *Học sâu* được thể hiện trong Hình 1.1:



Hình 1. 1: Mối quan hệ của các lĩnh vực AI, ML và DL

Arthur Samuel đã định nghĩa vào năm 1959, “ML là một lĩnh vực nghiên cứu cung cấp cho máy vi tính khả năng học mà không cần phải lập trình rõ ràng”. Một định nghĩa thể hiện rõ ràng hơn vào năm 1997: “Một chương trình máy tính được cho là học hỏi từ kinh nghiệm E có liên quan với một vài nhiệm vụ T và hiệu suất đo lường P, nếu hiệu suất của nó trên T được đo bằng P cải thiện sau khi trải qua kinh nghiệm E”.

Ví dụ: AlphaGo là một chương trình máy tính do Google DeepMind phát triển đã học được cách chơi cờ vây và đánh bại cờ thủ nổi tiếng Lee Se-dol.

Theo định nghĩa của Tom Mitchell, ta có:

- E: Kinh nghiệm có được khi chơi với các đối thủ và tự tăng cường.
- T: Nhiệm vụ là chơi cờ vây.
- P: Khả năng mà AlphaGo sẽ thắng trong trận đấu tiếp theo.

Nhìn chung, để xác định được một vấn đề học tập rõ ràng cần phải nhận biết được ba đặc trưng sau: các nhiệm vụ, hiệu suất cần được cải thiện và nguồn gốc kinh nghiệm.

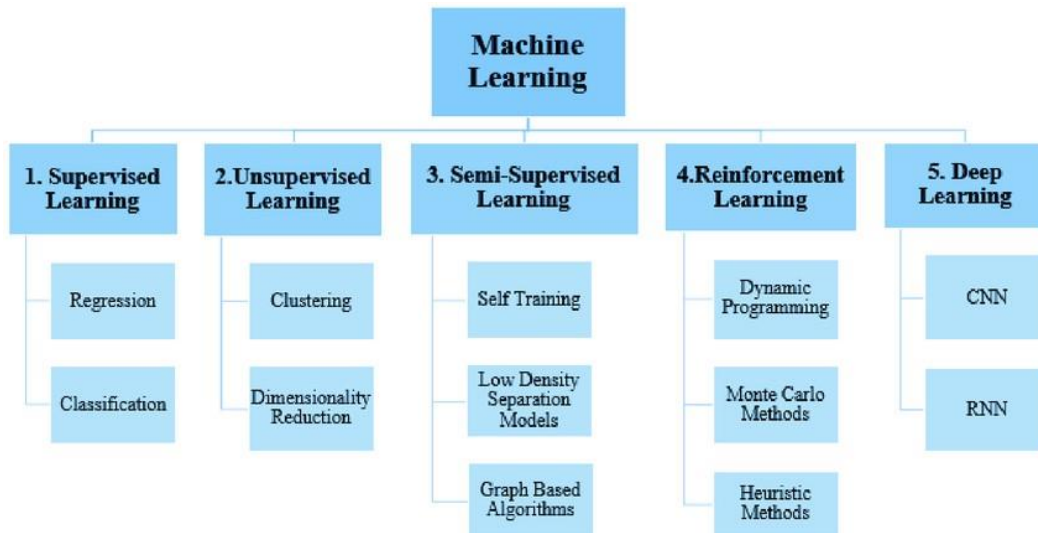
#### 1.4 Phân nhánh máy học

*Máy học (Machine learning)* là một thuật ngữ với khái niệm rộng, trong đó, nó chia ra các nhánh nhỏ, mỗi nhánh như vậy là một lĩnh vực khác nhau. Việc huấn luyện mô hình để đáp ứng yêu cầu đưa ra là rất phức tạp. Mỗi công việc khác nhau sẽ có từng cách giải quyết tương ứng. Để thỏa mãn những thách thức này, ML sinh ra nhiều phương pháp học máy khác nhau. Hình 1.2 là sơ đồ cấu trúc của ML và các phân nhánh của nó:

**Học có giám sát (Supervised learning):** Một tập dữ liệu chính xác bao gồm các cặp (data, label), đưa vào một thuật toán để huấn luyện, sau quá trình huấn luyện, nếu

đưa một đầu vào mới chưa có trong tập dữ liệu, thuật toán sẽ dự đoán đầu ra chính xác nhất có thể. Quá trình này gọi là học từ ví dụ hay học có giám sát.

**Học không giám sát (Unsupervised learning):** Khác với phương pháp trên, dữ liệu trong tập dữ liệu không có đầu ra, thay vì thuật toán gán nhãn thì thuật toán sẽ phải *phân cụm (cluster)* chúng. Cách tiếp cận theo thống kê của phương pháp này là *ước tính mật độ (density estimation)*.



Hình 1. 2: Cấu trúc phân nhánh của Machine Learning

**Học bán giám sát (Semi-supervised learning):** Học bán giám sát là sự kết hợp giữa hai phương pháp kể trên, chính vì vậy mà trong tập dữ liệu chia thành hai phần: một phần nhỏ dữ liệu đã được gán nhãn và phần còn lại dữ liệu chưa được gán nhãn. Đa số các bài toán thuộc về loại này vì chi phí để gán nhãn dữ liệu là khá lớn so với dữ liệu có được từ Internet.

**Học tăng cường (Reinforcement learning):** Phương pháp này cũng là sự kết hợp giữa học có giám sát và học không giám sát. Tuy nhiên, điểm khác biệt so với bán giám sát là thuật toán chỉ cần được biết nhiệm vụ này có làm sai hay không mà không cần biết cách để làm đúng nhiệm vụ. Thuật toán sẽ thám hiểm và thử các khả năng có thể (giống như thuật toán vét cạn) cho đến khi nó được biết là nó đang làm đúng.

**Học sâu (Deep Learning):** Là một lĩnh vực con của ML. Về bản chất, DL cố gắng bắt chước cách thức hoạt động của bộ não con người. Mọi người, nhất là giới khác chuyên môn, thường hay nhầm lẫn về hai thuật ngữ này. Trong khi machine learning là việc dạy cho máy tính biết cách giải quyết một nhiệm vụ nào đó mà không cần lập trình chi tiết, ML là một phương pháp học nông, chỉ giải quyết được các vấn đề hay bài toán đơn giản. Thì DL lại giúp chúng ta giải quyết được các vấn đề phức tạp trong thế giới thật, những bài toán mà đôi khi con người cũng có thể bị bối rối. Học sâu sử dụng

mạng *noron nhân tạo* (*Artificial Neural Networks*), để giải quyết những bài toán với dữ liệu trừu tượng, dữ liệu không rõ ràng, dữ liệu mờ,...

### 1.5 Tổng quan các phương pháp nhận dạng

Với phương pháp nhận dạng dựa trên các đặc trưng khuôn mặt sử dụng việc xác định các đặc trưng hình học trên các chi tiết của khuôn mặt (vị trí, diện tích, hình dạng mắt, mũi, miệng . . .) và mối quan hệ giữa chúng (khoảng cách của hai mắt, khoảng cách của hai lông mày. . .).

- Ưu điểm của phương pháp này là nó gần với cách mà con người sử dụng để nhận biết khuôn mặt và cho kết quả tốt trong các trường hợp ảnh hưởng tác động từ bên ngoài.
- Nhược điểm là thuật toán phức tạp do việc xác định mối quan hệ giữa các khuôn mặt và các ảnh kích thước nhỏ thì các đặc tính sẽ khó phân biệt do đó khó tính toán được các đặc trưng trên khuôn mặt.

Còn với phương pháp nhận dạng dựa trên toàn diện khuôn mặt, chúng ta sẽ không đi xét từng thành phần đặc trưng trên khuôn mặt mà sẽ xem khuôn mặt là một chiều không gian cụ thể và tìm đặc điểm đặc trưng trên không gian đó. Hai phương pháp thường được sử dụng trong hướng tiếp cận này là:

- Phương pháp truyền thống Principal Components Analysis(PCA)[1] là phương pháp trích xuất đặc trưng nhằm giảm số chiều của ảnh. Tuy đơn giản, nhưng nó mang lại hiệu quả khá tốt. Hệ thống ổn định và có thể thích nghi cao đối với dữ liệu đầu vào thay đổi nhiều.
- Linear Discriminant Analysis(LDA)[2] là phương pháp nhận dạng dựa trên một phép chiếu tuyến tính từ không gian hình ảnh vào một chiều không gian thấp hơn bằng cách tối đa giữa các lớp tán xạ và giảm nhiễu phân tán trong lớp.

Mạng Convolution Neural Network(CNN)[3] là một trong những mô hình Deep learning tiên tiến giúp xây dựng hệ thống thông minh với độ chính xác cao trong xử lý ảnh. CNN được sử dụng nhiều trong các bài toán nhận dạng đối tượng trong ảnh. Tuy nhiên phương pháp này lại có nhược điểm lớn là phải xây dựng một tập cơ sở dữ liệu lớn (mỗi tập dữ liệu huấn luyện phải sử dụng ít nhất 1000 nhãn mới mang lại hiệu quả cao) và đồng thời việc xử lý dữ liệu lớn đòi hỏi phải nhanh và chính xác. Mạng CNN là một tập hợp các lớp tích chập chồng lên nhau và sử dụng các hàm kích hoạt phi tuyến tính như ReLU để kích hoạt các trọng số trong các nút. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

## 1.6 Phương pháp tiếp cận

Để tìm hiểu thực hiện mô hình mở cửa tự động bằng phương pháp nhận dạng khuôn mặt, nhóm tiến hành thực hiện tìm hiểu các phương pháp sau đây:

Phương pháp nghiên cứu lý thuyết:

Tìm hiểu tổng quan về Deep Learning và bài toán nhận dạng khuôn mặt trong thị giác máy tính.

Tìm hiểu các thư viện Tensorflow, Keras, OpenCV và một số thư viện khác.

Phương pháp nghiên cứu thực nghiệm:

Tiến hành phân tích và cài đặt trên Python.

Tiến hành đào tạo bộ phân loại từ dữ liệu ban đầu.

## 1.7 Một số thư viện học máy hiện nay

### 1.7.1 TensorFlow

TF là một nền tảng Học máy mã nguồn mở, được thiết kế bởi đội ngũ Google Brain và tổ chức nghiên cứu trí tuệ máy của Google nhằm triển khai các ứng dụng của Học máy và Học sâu theo cách đơn giản. Nó là kết hợp giữa Đại số tính toán của các kỹ thuật tối ưu hoá để dễ dàng tính toán các biểu thức toán học. TF có một hệ sinh thái toàn diện, linh hoạt bao gồm các công cụ, thư viện và tài nguyên cộng đồng cho phép các nhà nghiên cứu xây dựng và triển khai các ứng dụng Học máy. Đây cũng là một trong những thư viện máy học lâu đời nhất.

Trang chủ của TF tại địa chỉ sau: <https://www.tensorflow.org/>. Trang mã nguồn Github của TF nằm tại: <https://github.com/tensorflow/tensorflow>.

TF có tài liệu đầy đủ và bao gồm nhiều thư viện Học máy nên rất phổ biến hiện nay. Vì là một sản phẩm của Google, nên hiện tại, TF đang được ứng dụng rất nhiều trong các sản phẩm của Google như phân loại chữ viết tay, nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên,...

### 1.7.2 Scikit-learn

Sklearn là một thư viện Học máy mã nguồn mở hữu ích và mạnh mẽ trong Python. Dự án của David Cournapeau bắt đầu vào năm 2007 với tư cách là một dự án của Google Summer of Code. Hiện tại, Sklearn đang được duy trì bởi một đội ngũ các tình nguyện viên. Sklearn cung cấp một sự lựa chọn các công cụ hiệu quả cho Học máy và mô hình thống kê, bao gồm phân loại, hồi quy, phân cụm và giảm chiều dữ liệu với giao diện nhất quán trong Python. Thư viện này phần lớn được viết bằng Python, được xây dựng dựa trên NumPy, SciPy và Matplotlib.

Trang chủ của Sklearn tại địa chỉ sau: <https://scikit-learn.org/>. Trang Github chứa mã nguồn Sklearn nằm tại: <https://github.com/scikit-learn/scikit-learn>.

Một số sản phẩm thương mại sử dụng Sklearn như Spotify, Evernote, Booking.com, J.P.Morgan, Hugging Face, Télécom ParisTech, Aweber,...

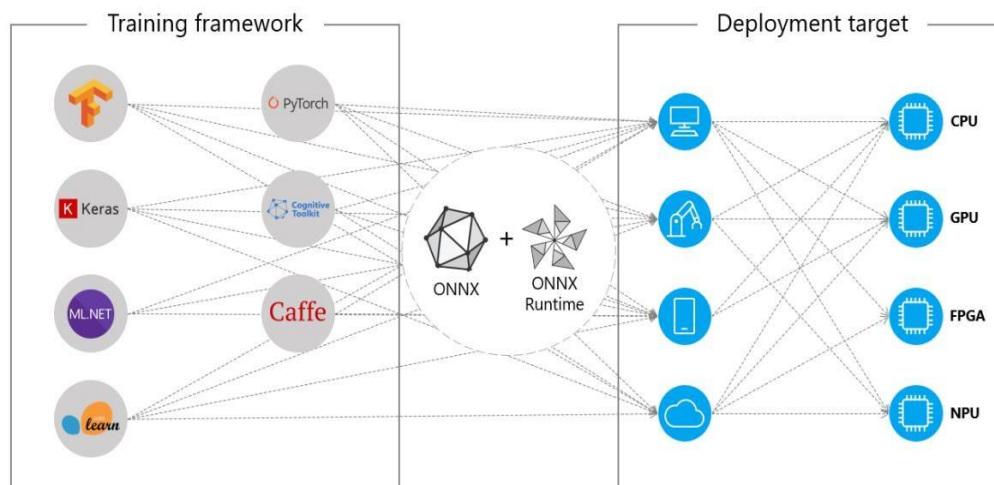
### 1.7.3 Keras

Keras là một thư viện Học sâu mã nguồn mở dành cho Python. Nó được phát triển bởi một nhà nghiên cứu trí tuệ nhân tạo của Google là Francois Chollet. Keras có thể chạy trên các thư viện mã nguồn mở như TensorFlow, Theano, R hay CognitiveToolkit (CNTK). Mục tiêu thiết kế của Keras là cho phép thử nghiệm các mạng Học sâu nhanh chóng. Các tổ chức hàng đầu như Google, Square, Netflix, Huawei và Uber hiện đang sử dụng Keras.

Trang chủ của Keras tại địa chỉ sau: <https://keras.io/>. Trang Github chứa mã nguồn của Keras nằm tại: <https://github.com/keras-team/keras>.

### 1.7.4 ONNX

ONNX (Open Neural Network Exchange) là một hệ sinh thái trí tuệ nhân tạo mã nguồn mở của các công ty công nghệ và tổ chức nghiên cứu. ONNX giúp thiết lập các tiêu chuẩn mở để đại diện cho các thuật toán ML và các phần mềm công cụ nhằm thúc đẩy đổi mới và hợp tác trong lĩnh vực AI. ONNX cung cấp framework bao gồm các mô hình đồ thị tính toán có thể mở rộng, các toán tử được tích hợp sẵn và các kiểu dữ liệu tiêu chuẩn, tập trung vào phân suy diễn (đánh giá).



Hình 1. 3: Sơ đồ hoạt động của ONNX

### 1.7.5 PyTorch

PyTorch được định nghĩa là một thư viện Học máy mã nguồn mở cho Python. Nó được sử dụng cho các ứng dụng như xử lý ngôn ngữ tự nhiên. Ban đầu nó được phát triển bởi nhóm nghiên cứu trí tuệ nhân tạo của Facebook và phần mềm Uber's Pyro để lập trình xác suất.

PyTorch được Hugh Perkins phát triển như một trình bao bọc Python cho LusJIT dựa trên khuôn khổ Torch.

Trang web thông tin chính thức của thư viện PyTorch nằm tại địa chỉ sau: <https://pytorch.org/>. Trang Github chứa mã nguồn của thư viện PyTorch nằm địa chỉ: <https://github.com/pytorch/pytorch>.

### 1.8 Quy trình thiết kế

- Cảm biến khoảng cách siêu âm Ultrasonic SR04, máy ảnh và khóa điện từ sẽ được kết nối với Raspberry Pi.
- Cảm biến siêu âm đo khoảng cách theo từng giây liên tục, bất cứ khi nào cảm biến khoảng cách siêu âm phát hiện vật thể ở cự ly 100 cm trở xuống, nó sẽ bật máy ảnh.
- Bất kỳ ai được cấp quyền truy cập vào hệ thống này sẽ phát hiện trên camera trong vòng 20 giây, khóa điện từ sẽ mở.
- Nếu trong vòng 20 giây cửa không mở, chúng ta sẽ bấm mật khẩu từ bàn phím keypad để mở khóa điện từ.
- Khóa chỉ mở trong 10 giây, sau đó nó sẽ đóng lại.

### 1.9 Công cụ nguyên liệu cần triển khai

Các công cụ thiết bị cần dùng để hoàn thành hệ thống nhận dạng khuôn mặt gồm có:

- Raspberry Pi 4 model B ram 4gb.



Hình 1. 4: Raspberry Pi 4 model B ram 4gb



- Màn hình 5 inch.



Hình 1. 5: Màn hình 5 inch Waveshare

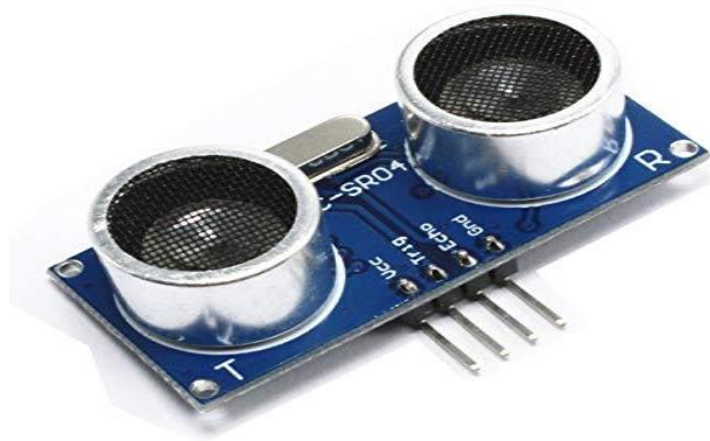
- Webcam.



Hình 1. 6: Webcam Gucee



- Cảm biến khoảng cách siêu âm ultrasonic HC-SR04.



Hình 1. 7: Cảm biến HC-SR04

- Khóa điện từ ly 12 VDC.



Hình 1. 8: Khóa điện từ ly

- Mô-đun relay chuyển tiếp 5V.



Hình 1. 9: Mô-đun relay chuyển tiếp 5V

- Bộ chuyển đổi 12V.



Hình 1. 10: Adapter 12V

- Keypad



Hình 1. 11: Keypad 4x3.

### **1.10 Kết luận chương**

Qua chương này, nhóm đã trình bày được tổng quan đề tài, các phương pháp nhận dạng, giải pháp kỹ thuật. Đồng thời, nhóm đã giới thiệu quy trình thiết kế, các công cụ nguyên liệu để áp dụng vào trong hệ thống. Nhóm sẽ trình bày cơ sở lý thuyết ở chương tiếp theo.

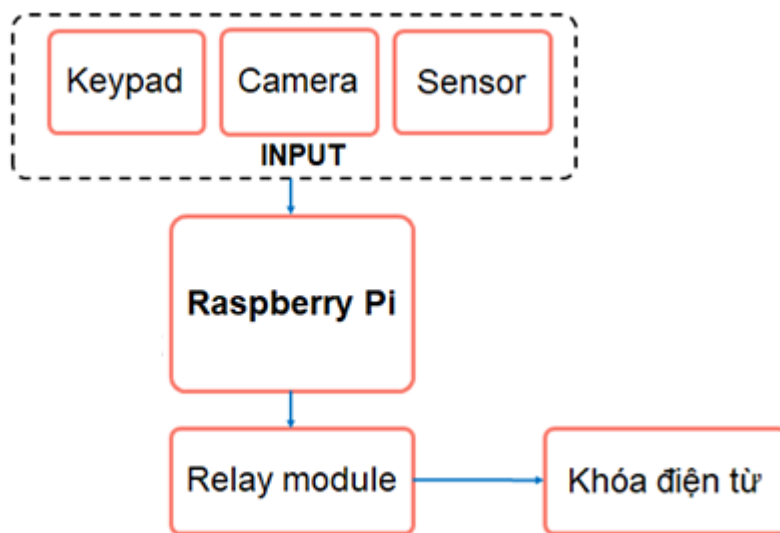
## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Mở đầu chương

Trong chương này, nhóm sẽ giới thiệu khái quát về phần cứng và thư viện Open CV trong môi trường Python để có cái nhìn cụ thể về cách thức hoạt động của hệ thống. Sau đó, nhóm sẽ giới thiệu về cơ sở lý thuyết, mạng CNN, tìm hiểu mô hình MTCNN, các đặc trưng của mạng FaceNet và thuật toán phân loại SVM (Support Vector Machine) dùng trong hệ thống.

### 2.2 Giới thiệu về phần cứng của hệ thống

#### 2.2.1 Sơ đồ kết nối giữa Raspberry pi và các thiết bị ngoại vi



Hình 2. 1: Sơ đồ kết nối giữa Raspberry pi và các thiết bị ngoại vi

**Nguyên lý hoạt động:** Khi người dùng đứng trước camera, cảm biến hoạt động đo khoảng cách từ cửa đến khuôn mặt. Nếu khuôn mặt nằm trong khoảng cách nhỏ hơn 100 cm thì camera sẽ bật lên và phát hiện khuôn mặt nhờ sử dụng thuật toán MTCNN. Mô hình sẽ nhận dạng khuôn mặt người dùng với cơ sở dữ liệu đã được đào tạo sẵn bằng thuật toán FaceNet và SVM. Sau khi nhận dạng được khuôn mặt thì Raspberry Pi giao tiếp với các thiết bị ngoại vi bên ngoài như là keypad và relay module. Sau khi nhận dạng được thì qua giao tiếp, relay module sẽ điều khiển khóa điện từ đóng mở khóa điện từ. Trong trường hợp không mở được khóa điện từ bằng nhận dạng khuôn mặt, thì người dùng sẽ sử dụng keypad để bấm mật khẩu mở khóa điện từ.

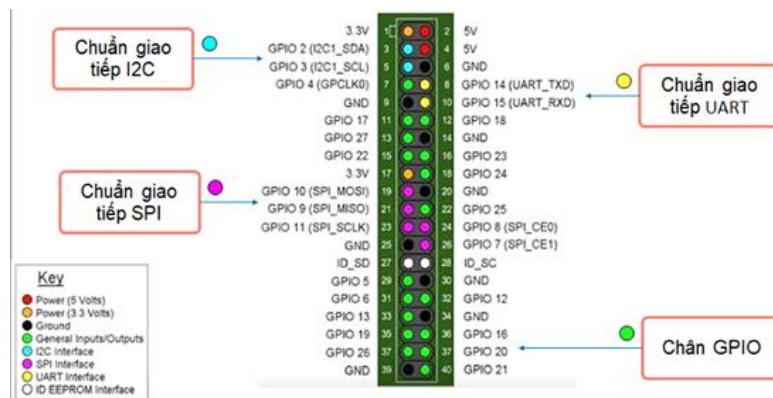
## 2.2.2 Giới thiệu về các cổng GPIO của Raspberry Pi.



Hình 2. 2: Các chân của GPIO trên Raspberry Pi

General purpose input/output (GPIO) trên Raspberry Pi có thể thiết lập dùng để làm ngõ vào hoặc ngõ ra tùy thuộc vào cách thiết lập của người điều khiển để tiến hành giao tiếp điều khiển với các thiết bị bên ngoài thông qua phần mềm. Trên bo mạch của Raspberry Pi gồm có 40 chân GPIO được chia thành 2 hàng mỗi hàng gồm có 20 chân. GPIO là nơi mà Raspberry Pi có thể giao tiếp với các thiết bị điện tử điều khiển được kết nối với chân GPIO.

### Sơ đồ mẫu của chân GPIO của Raspberry Pi.



Hình 2. 3: Sơ đồ chân GPIO của Raspberry Pi

**Chân cấp nguồn (Power):** Các chân cấp nguồn 5V được kết nối trực tiếp với ngõ vào nguồn của Raspberry Pi 4B và có khả năng sẽ cung cấp một dòng điện tối đa bằng với dòng điện định mức của adaptor nguồn ngoài.

Các chân cấp nguồn 3,3V trên những Raspberry Pi đời đầu có khả năng cung cấp một dòng điện tối đa khoảng 50mA. Khả năng này sẽ đủ để cung cấp năng lượng cho một vài đèn LED hoặc một bộ vi xử lý. Tất cả các Raspberry Pi kể từ loại B+ trở đi thì đều có thể cung cấp một dòng điện tối đa cao hơn một chút, có khả năng lên tới 500mA.

Tuy nhiên, thông thường bạn nên sử dụng nguồn cung cấp 5V, kết hợp với một bộ ổn áp 3,3V cho các thiết kế mạch điện dùng nguồn 3,3V.

Các chân nối đất (GND) trên Raspberry Pi đều được kết nối trực tiếp với nhau về mặt vật lý, vì vậy chúng ta không cần phải quan tâm đến việc phải sử dụng chân nối đất nào khi chúng ta cần nối dây cấp nguồn. Nói chung, khi cần sử dụng chân nối đất thì ta có thể dùng bất cứ chân GND nào trên Raspberry Pi miễn sao việc kết nối đó là thuận tiện đó là nhất cho chúng ta.

**Chân xuất/nhập dữ liệu đa mục đích (GPIO – General Purpose IO):** Trên Raspberry Pi 4B có 26 chân xuất/nhập dữ liệu đa mục đích. Các chân GPIO này có hai trạng thái hoạt động độc lập nhau, các trạng thái làm việc này sẽ được người sử dụng chủ động thiết lập trong mã nguồn điều khiển.

**Chân xuất dữ liệu (Output):** Các chân GPIO khi được thiết lập ở trạng thái xuất dữ liệu sẽ hoạt động ở mức 3,3V (cho trường hợp logic cao) và hoạt động ở mức 0V (cho trường hợp logic thấp). Dòng điện tối đa mỗi chân GPIO có thể cung cấp là 16mA.

**Chân nhập dữ liệu (Input):** Các chân GPIO khi được thiết lập ở trạng thái nhập dữ liệu thì có thể đọc mức 3,3V (ứng với trường hợp logic cao) và đọc mức 0V (ứng với trường hợp logic thấp). Các chân GPIO khi đóng vai trò là ngõ vào nhập dữ liệu thì chúng ta có thể thiết lập điện trở kéo lên hoặc kéo xuống bên trong bằng cách thiết lập trong mã nguồn điều khiển. Lưu ý, hai chân GPIO2 và GPIO3 luôn có điện trở kéo lên cố định, các chân còn lại thì người sử dụng phải chủ động thiết lập.

**Chân chức năng đặc biệt:** Trên Raspberry Pi 4B, một số chân GPIO sẽ có tích hợp thêm các tính năng đặc biệt như PWM, UART, SPI và I2C.

**Tính năng PWM (Pulse Width Modulation):** Trên mô-đun Raspberry có thể được thiết lập bằng hai phương pháp: thiết lập bằng phần cứng hoặc thiết lập bằng phần mềm. Trong trường hợp thiết lập bằng phần cứng thì tính năng PWM chỉ có thể thực hiện tại các chân GPIO12/Pin-32, GPIO13/Pin-33, GPIO18/Pin-12, GPIO19/Pin-35. Trong trường hợp thiết lập bằng phần mềm thì tính năng PWM có thể thực hiện tại tất cả các chân GPIO của mô-đun.

**Tính năng UART (Universal Asynchronous Receiver/Transmitter):** Trên mô-đun Raspberry được thiết lập bằng phần cứng tại các chân GPIO14/Pin-8 (TxD) và GPIO15/Pin-10 (RxD).

**Tính năng SPI (Serial Peripheral Interface):** Trên mô-đun Raspberry bao gồm hai mô-đun SPI riêng biệt, hai mô-đun này được thiết lập bằng phần cứng tại các chân sau:

Mô-đun SPI0: GPIO7/Pin-26 (CE1), GPIO8/Pin-24 (CE0), GPIO9/Pin-21 (MISO), GPIO10/Pin-19 (MOSI), GPIO11/Pin-23 (SCLK).

Mô-đun SPI1: GPIO16/Pin-36 (CE2), GPIO17/Pin-11 (CE1), GPIO18/Pin-12 (CE0), GPIO19/Pin-35 (MISO), GPIO20/Pin-38 (MOSI), GPIO21/Pin-40 (SCLK).

**Tính năng I2C (Inter-integrated Circuit):** Trên mô-đun Raspberry bao gồm hai mô-đun I2C riêng biệt, hai mô-đun này được thiết lập bằng phần cứng tại các chân sau:

Mô-đun I2C: GPIO2/Pin-3 (Data), GPIO3/Pin-5 (Clock).

Mô-đun EEPROM I2C: GPIO0/Pin-27 (Data), GPIO1/Pin-28 (Clock).

Một điểm rất quan trọng mà người sử dụng Raspberry Pi cần phải lưu ý là các tính năng đặc biệt đã nêu ra trên đây mặc định ban đầu sẽ chưa được phép sử dụng. Để cho phép hoặc không cho phép sử dụng các tính năng đặc biệt này thì người sử dụng cần phải tiến hành thiết lập cấu hình trong hệ thống.

Những giao tiếp ngoại vi trên Raspberry Pi: I2C, SPI, UART, USB, Ethernet. Tùy theo nhu cầu và mục đích sử dụng mà chúng ta có thể chọn loại giao tiếp phù hợp để sử dụng. Thông thường để giao tiếp với các dòng vi xử lý và các thiết bị ngoại vi khác chúng ta có thể chọn UART, I2C, SPI. Cả 3 giao thức đều rất phổ biến. Hai giao thức mới hơn là USB và Ethernet là chuẩn mới hơn dành cho giao tiếp tốc độ cao. Khi giao tiếp với các phần cứng khác như cảm biến thì đa phần phải chọn lựa theo giao tiếp mà thiết bị đó hỗ trợ. Khi muốn thành lập một mạng lưới giao tiếp với nhau có thể chọn I2C hay SPI. Giao tiếp cần tốc độ cao thì chúng ta cần sử dụng USB và Ethernet là phù hợp nhất.

	I2C	SPI	UART	USB	Ethernet
Tốc độ	Standard speed: 100kbit/s Full speed: 400kbit/s High speed: 3.4Mbit/s	Tùy thuộc vào tần số và vi xử lý, không giới hạn tối đa, Max thường tới mấy chục Mbit/s	Tùy thuộc vào tần số và vi xử lý, Max thường là vài Mbit/s	Low speed: 1.5Mb/s Full speed: 12Mb/s High speed: 480Mb/s	10-100 Mbit/s

Hình 2. 4: Các thông số trong các chuẩn giao tiếp của Raspberry Pi

### 2.2.3 Giới thiệu về relay module

Relay bao gồm 3 khối cơ bản:

Khối tiếp thu (cơ cấu tiếp thu): Có nhiệm vụ tiếp nhận tín hiệu đầu vào và sau đó biến nó thành đại lượng cần thiết cung cấp tín hiệu phù hợp cho khối trung gian.

Khối trung gian (cơ cấu trung gian): Tiếp nhận thông tin từ khối tiếp thu và biến đổi nó thành đại lượng cần thiết cho rơ le tác động

Khối chấp hành (cơ cấu chấp hành): làm nhiệm vụ phát tín hiệu cho mạch điều khiển.





Hình 2. 5: Giới thiệu relay module[4]

Các thông số của relay module[5]:

Thông số kỹ thuật của relay module:

- Kích thước: 50x30 mm.
- Điện áp vào khối điều khiển: 3.3– 5V.
- Điện áp vào khối relay: 12V.
- Relay: 220V– 10A

Các ký hiệu trong relay

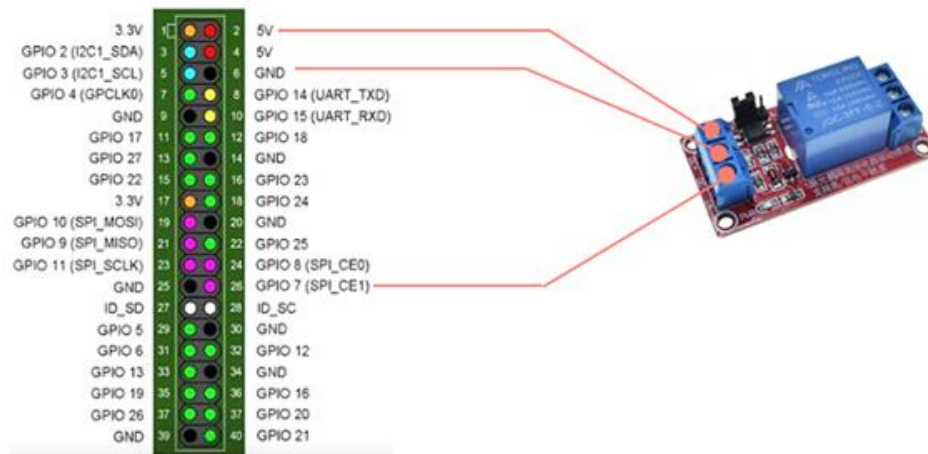
- VLR: Điện áp cho relay.
- VCC: Điện áp vào khối điều khiển.
- CTRL: Tín hiệu điều khiển.
- GND: Đất.
- NC: Tiếp điểm thường đóng.
- NO: Tiếp điểm thường mở.
- COM: Tiếp điểm này nối với dây nóng của tải điện xoay chiều hoặc cực dương của điện một chiều.

Thành phần chỉ thị

- POWER: Báo đèn nguồn.
- SIGNAL: Đèn báo tín hiệu điều khiển, tín hiệu ctrl ở mức cao thì signal tắt và ngược lại.

**Sơ đồ nối chân:**





Hình 2. 6: Sơ đồ nối chân giữa Raspberry Pi với relay module

**Cách nối chân:** Chân Vcc của module relay nối vào chân thứ 2 của Raspberry Pi. Chân GND của relay module nối vào chân thứ 6 của Raspberry Pi. Chân In của module nối vào chân thứ 26 (GPIO7) của Raspberry Pi.

#### Nguyên lý hoạt động:

Khi tín hiệu điều khiển CTRL ở mức cao, diode phát quang cho dòng điện đi qua, làm cho Transistor thông và cuộn hút nam châm hoạt động, relay chuyển mạch sang “NO”, khóa điện từ được mở. Khi tín hiệu điều khiển CTRL ở mức thấp, diode phát quang không cho dòng điện đi qua. Do đó, Transistor đóng, cuộn hút nam châm không hoạt động, relay ở chế độ “NC”, khóa điện từ đóng lại.

#### 2.2.4 Giới thiệu về cảm biến HC-SR04.

Cảm biến khoảng cách siêu âm HC-SR04[6] được sử dụng rất phổ biến để xác định khoảng cách vì rẻ và chính xác. Cảm biến sử dụng sóng siêu âm và có thể đo khoảng cách trong khoảng từ 2 -> 300 cm, với độ chính xác gần như chỉ phụ thuộc vào cách lập trình. Cảm biến HC-SR04 có 4 chân là: Vcc, Trig, Echo, GND.

- Vcc là nguồn cung cấp cho cảm biến khoảng cách siêu âm HC-SR04 mà chúng ta kết nối chân 5V trên Raspberry Pi.
- Chân Trig (Kích hoạt) được sử dụng để kích hoạt các xung âm thanh siêu âm.
- Chân Echo tạo ra một xung khi nhận được tín hiệu phản xạ. Độ dài của xung tỷ lệ với thời gian tín hiệu truyền được phát hiện.
- Chân GND được kết nối với GND của Raspberry Pi.



Hình 2. 7: Giới thiệu về cảm biến HC-SR04[7]

### Sơ đồ nối chân

VCC	5v
Trig	Pin8
Echo	Pin7
GND	Pin9

Hình 2. 8: Sơ đồ nối chân cảm biến HC-SR04

**Nguyên lý hoạt động:** Cảm biến khoảng cách siêu âm HC-SR04 bao gồm hai đầu dò siêu âm. Nó hoạt động như một máy phát chuyển đổi tín hiệu điện thành xung âm thanh siêu âm 40 KHz. Máy thu lắng nghe các xung được truyền đi. Nếu nó nhận được chúng, nó tạo ra một xung đầu ra có độ rộng có thể được sử dụng để xác định khoảng cách mà xung truyền đi.

Cảm biến khoảng cách HC-SR04 thường được sử dụng với cả nền tảng vi điều khiển và vi xử lý như Arduino, ARM, PIC, Raspberry Pie, ... Hướng dẫn sau sẽ dùng chung cho bất kể loại thiết bị tính toán nào được sử dụng.

Cấp nguồn cho cảm biến bằng cách sử dụng nguồn + 5V được điều chỉnh thông qua các chân nối đất và Vcc của cảm biến. Dòng điện được tiêu thụ bởi cảm biến nhỏ hơn 15mA và do đó có thể được cấp nguồn trực tiếp bằng các chân 5V trên bo mạch (nếu có). Các chân Trigger và Echo đều là các chân I / O và do đó có thể được kết nối với các chân I / O của vi điều khiển. Để bắt đầu đo, chân trigger phải được đặt ở mức cao 10uS và sau đó tắt. Hoạt động này sẽ kích hoạt một sóng siêu âm ở tần số 40Hz từ bộ phát và bộ thu sẽ đợi sóng quay trở lại. Khi sóng được trả lại sau khi nó bị phản xạ

bởi bất kỳ đối tượng nào, chân Echo sẽ tăng cao trong một khoảng thời gian cụ thể bằng với thời gian cần thiết để sóng quay trở lại cảm biến.

Khoảng thời gian mà chân Echo vẫn ở mức cao được đo bởi vi xử lý hoặc vi điều khiển vì nó cung cấp thông tin về thời gian cần thiết để sóng quay trở lại cảm biến. Sử dụng thông tin này để tính khoảng cách được đo.

### 2.2.5 Giới thiệu về bàn phím ma trận keypad 4x3.

Bàn phím ma trận keypad[8] là tổ hợp của các phím bấm đơn, được kết nối với nhau theo các hàng và các cột. Các thông số chi tiết:

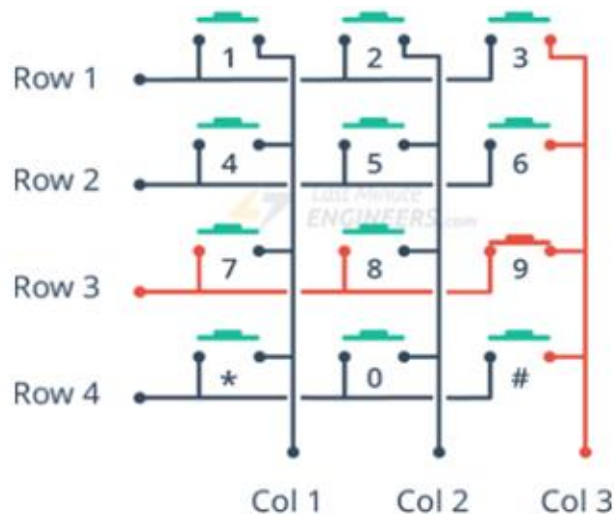
- Trọng lượng: 7,5 gam
- Kích thước bàn phím: 70mm x 77mm x 1mm (2,75" x 3" x 0,035")
- Chiều dài cáp + đầu nối: 85mm
- Đầu nối cao độ 7 chân 0,1"



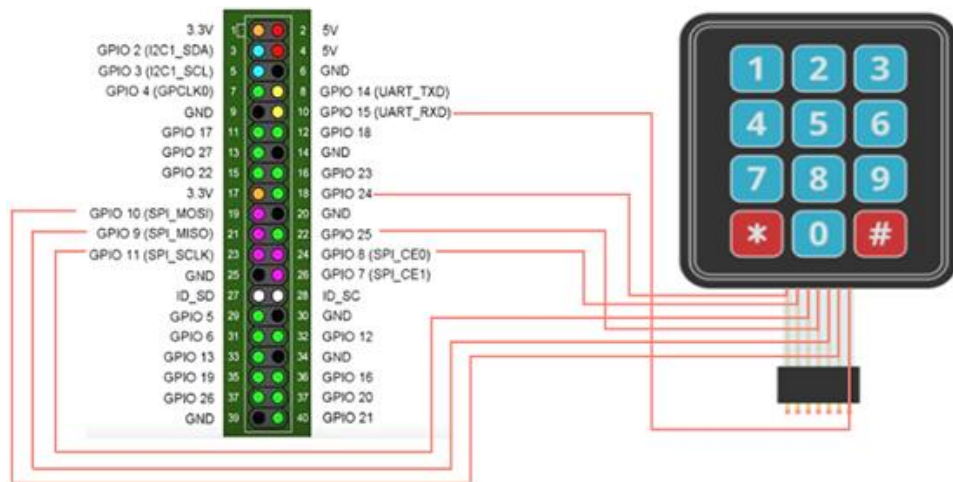
Hình 2. 9: Giới thiệu về bàn phím ma trận 4x3[9]

#### Cấu tạo của keypad 4x3:

Keypad 4x3 gồm có 7 đầu vào/ra và 12 phím bấm. Các phím bấm được chia thành 4 hàng và 3 cột, 1 đầu ra của phím bấm được nối với đầu cột, đầu còn lại được nối với đầu của hàng.



Hình 2. 10: Cấu tạo của bàn phím ma trận 4x3 [10]



Hình 2. 11: Sơ đồ nối chân giữa Raspberry Pi với keypad

**Cách nối chân:** Các đầu vào của keypad được nối theo thứ tự r1, r2, r3, r4 nối với pin 18,24,23,22. Các đầu ra của keypad được nối theo thứ tự c1, c2, c3 nối với pin 21,19,10.

**Nguyên lý hoạt động:** Để làm việc với Keypad, phải sử dụng giải thuật “quét phím”. Giải thuật này yêu cầu Raspberry Pi liên tục đưa các tín hiệu đầu ra ở hàng (hoặc cột) và thu lại đầu vào ở cột (hoặc hàng), nếu phím được bấm, đầu phát tín hiệu sẽ được kết nối với đầu thu, từ đó xác định được phím đã bấm. Việc lựa chọn đầu ra/vào hình thành 2 phương pháp quét phím: theo chiều dọc và theo chiều ngang. Trong báo cáo này, tín hiệu xuất ra ở các hàng và thu lại ở các cột. Giả sử một nút ‘2’ được nhấn, khi đó đường col 2 và row 1 được nối với nhau. Nếu đường col 2 được nối với GND, khi đó, điện áp ở chân số 2 sẽ mang điện áp 0V. Tương tự như thế với các phím cùng hàng col.

### **Thuật toán:**

Bước 1: Set các chân row1, row2, row3, row4 như các chân Output và giữ chúng ở mức cao, các chân col 1, col 2, col 3 như các chân input có điện trở kéo lên.

Bước 2: Đưa tín hiệu đầu ra ở các chân row1 = 1, row2 = 1, row3 = 1 và row4 = 1. Kiểm tra tín hiệu ở các chân col1, col2, col3 luôn bằng 1 dù có phím nào được nhấn hay không.

Bước 3: Đưa tín hiệu đầu ra ở các chân row1 = 0, row2 = 1, row3 = 1 và row4 = 1. Kiểm tra col1, 2, 3 nếu phím thuộc hàng 1 được nhấn sẽ có giá trị col nhận được bằng 0.

Bước 4: Tiếp tục đưa tín hiệu đầu ra ở các chân row1 = 1, row2 = 0, row3 = 1, row4 = 1 để xác định phím bấm được nhấn nếu nó ở hàng 2.

Bước 5: Thực hiện quá trình dịch chân đầu ra mang điện áp mức 0 một cách liên tục và xác định phím được bấm.

### **2.3 Tìm hiểu về Open CV**

OpenCV là tên viết tắt của open source computer vision library – có thể được hiểu là một thư viện nguồn mở cho máy tính. Cụ thể hơn OpenCV là kho lưu trữ các mã nguồn mở được dùng để xử lý hình ảnh, phát triển các ứng dụng đồ họa trong thời gian thực.

OpenCV cho phép cải thiện tốc độ của CPU khi thực hiện các hoạt động real time. Nó còn cung cấp một số lượng lớn các mã xử lý phục vụ cho quy trình của thị giác máy tính hay các learning machine khác.

Thư viện OpenCV được phát hành với giấy phép BDS. Do đó các dịch vụ nó cung cấp là hoàn toàn miễn phí và được hạn chế tối đa các rào cản thông thường. Cụ thể, bạn được phép sử dụng phần mềm này cho cả hoạt động thương mại lẫn phi thương mại. OpenCV sở hữu giao diện thiên thiện với mọi loại ngôn ngữ lập trình, ví dụ như C++, C, Python hay Java... Ngoài ra, nó cũng dễ dàng tương thích với các hệ điều hành khác nhau, bao gồm từ Windows, Linux, Mac OS, iOS cho đến cả Android.

Kể từ lần đầu xuất hiện từ năm 1999, giờ đây OpenCV đã sở hữu đội ngũ người dùng hùng hậu, con số ước tính có thể lên tới 47.000 người. Tất cả là nhờ những ưu điểm vượt trội của OpenCV. Cấu trúc tổng quan của OpenCV bao gồm 5 phần chính, 4 trong 5 phần đó được chỉ ra trong hình 2.12.



Hình 2. 12: Cấu trúc các phần của OpenCV [12]

- Phần CV bao gồm các thư viện cơ bản về xử lý ảnh và các giải thuật về thị giác máy tính.
- MLL là bộ thư viện về các thuật toán học máy, bao gồm rất nhiều bộ phân cụm và phân loại thống kê.
- HighGUI chứa đựng những thủ tục vào ra, các chức năng về lưu trữ cũng như đọc các file ảnh và video.
- Phần thứ 4, Cxcore chứa đựng các cấu trúc dữ liệu cơ bản (ví dụ như cấu trúc XML, các cây dữ liệu . . . ).
- Phần cuối cùng là CvAux, phần này bao gồm các thư viện cho việc phát hiện, theo dõi và nhận dạng đối tượng (khuôn mặt, mắt . . . ).

OpenCV - Python là một thư viện các ràng buộc Python được thiết kế để giải quyết các vấn đề về thị giác máy tính.

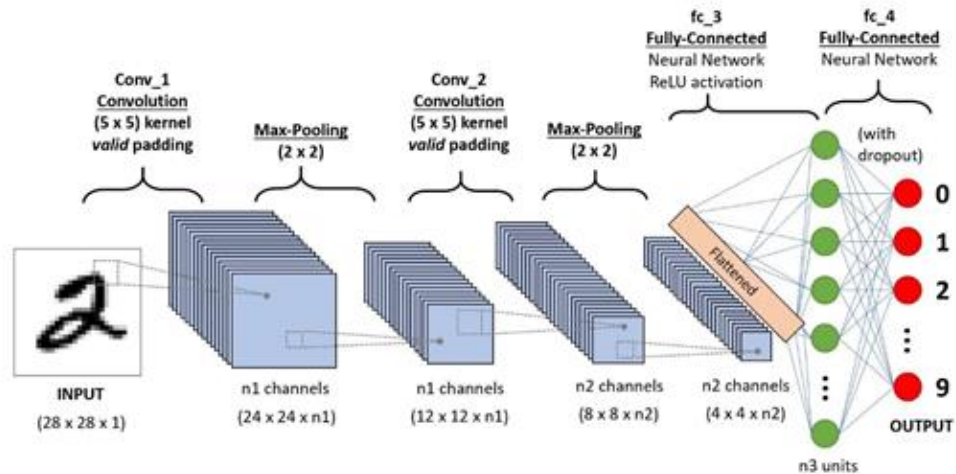
Python là ngôn ngữ lập trình có mục đích chung được bắt đầu bởi Guido van Rossum, nó trở nên rất phổ biến rất nhanh, chủ yếu vì tính đơn giản và khả năng đọc mã của nó. Nó cho phép lập trình viên thể hiện ý tưởng trong ít dòng mã hơn mà không làm giảm khả năng đọc.

So với các ngôn ngữ như C/C++, Python chậm hơn. Điều đó nói rằng, Python có thể dễ dàng được mở rộng với C/C++, cho phép chúng ta viết mã chuyên sâu tính toán trong C/C++ và tạo các trình bao bọc Python có thể được sử dụng làm mô-đun Python. Điều này mang lại cho chúng ta hai lợi thế: thứ nhất, mã nhanh như mã C/C++ gốc (vì đây là mã C++ thực tế hoạt động ở chế độ nền) và thứ hai, mã dễ dàng hơn trong Python so với C/C++. OpenCV - Python là một trình bao bọc Python để thực hiện OpenCV C++ ban đầu.

OpenCV - Python sử dụng Numpy, một thư viện được tối ưu hóa cao cho các hoạt động số với cú pháp kiểu MATLAB. Tất cả các cấu trúc mảng OpenCV được chuyển đổi sang và từ các mảng Numpy. Điều này cũng giúp tích hợp dễ dàng hơn với các thư viện khác sử dụng Numpy như SciPy và Matplotlib.

## 2.4 Mạng CNN - Convolutional Neural Network

Convolutional Neural Networks (CNN) là một trong những mô hình Deep Learning phổ biến nhất và có ảnh hưởng nhiều nhất trong cộng đồng thị giác máy tính. CNN được dùng trong nhiều bài toán như nhận dạng ảnh, phân tích video hoặc cho các bài toán trong lĩnh vực xử lý ngôn ngữ tự nhiên và hầu hết đều giải quyết tốt các bài toán này. CNN là một kiến trúc mạng neural rất thích hợp cho các bài toán mà dữ liệu là ảnh hoặc video. Có hai loại lớp chính trong CNN: Lớp tích chập (Convolutional layer) và lớp gộp (Pooling layer).



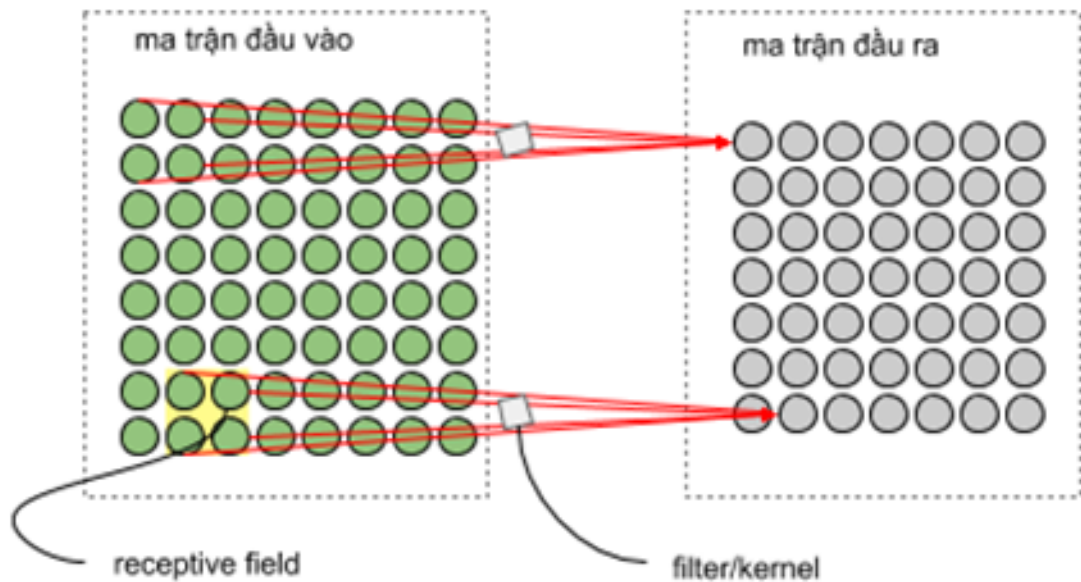
Hình 2. 13: Mô hình mạng CNN [13]

### 2.4.1 Lớp tích chập (Convolutional layer)

Lớp tích chập là lớp quan trọng nhất và cũng là lớp đầu tiên của mạng CNN. Lớp này có chức năng chính là phát hiện các đặc trưng có tính không gian hiệu quả. Trong tầng này có 4 đối tượng chính là: ma trận đầu vào, bộ lọc, receptive field, feature map. Lớp tích chập nhận đầu vào là một ma trận 3 chiều và một bộ lọc cần phải học. Bộ lọc này sẽ trượt qua từng vị trí trên bức ảnh để tính tích chập (convolution) giữa bộ lọc và phần tương ứng trên bức ảnh. Phần tương ứng này trên bức ảnh gọi là receptive field, tức là vùng mà một neural có thể nhìn thấy để đưa ra quyết định và ma trận cho ra bởi quá trình này được gọi là feature map. Để hình dung, chúng ta có thể tưởng tượng, bộ lọc giống như các tháp để quét người lần lượt qua không gian xung quanh. Khi phát hiện người, thì chuông báo động sẽ reo lên, giống như các bộ lọc tìm kiếm được đặc trưng nhất định thì tích chập đó sẽ cho giá trị lớn. Với ví dụ ở hình 2.14, dữ liệu đầu vào là ma



trận có kích thước  $8 \times 8 \times 1$ , một bộ lọc có kích thước  $2 \times 2 \times 1$ , feature map có kích thước  $7 \times 7 \times 1$ . Mỗi giá trị ở feature map được tính bằng tổng của tích các phần tử tương ứng của bộ lọc  $2 \times 2 \times 1$  với receptive field trên ảnh. Và để tính tất cả các giá trị cho feature map, cần trượt bộ lọc từ trái sang phải, từ trên xuống dưới. Do đó, có thể thấy rằng phép tích chập bảo toàn thứ tự không gian của các điểm ảnh. Ví dụ điểm góc trái của dữ liệu đầu vào sẽ tương ứng với một điểm bên góc trái của feature map.



Hình 2. 14: Feature map [14]

#### 2.4.1.1 Tầng tích chập

Tầng tích chập có chức năng chính là phát hiện đặc trưng cụ thể của bức ảnh. Những đặc trưng này bao gồm đặc trưng cơ bản là góc, cạnh, màu sắc. Vì bộ lọc quét qua toàn bộ bức ảnh nên những đặc trưng này có thể nằm ở vị trí bất kì trong bức ảnh, cho dù ảnh bị xoay trái, phải thì những đặc trưng này vẫn bị phát hiện. Ở minh họa hình 2.15, có một bộ lọc  $5 \times 5$  dùng để phát hiện góc cạnh, với bộ lọc này chỉ có một giá trị tại các điểm tương ứng với một góc cong



0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0

Hình 2. 15: Bộ lọc phát hiện cạnh

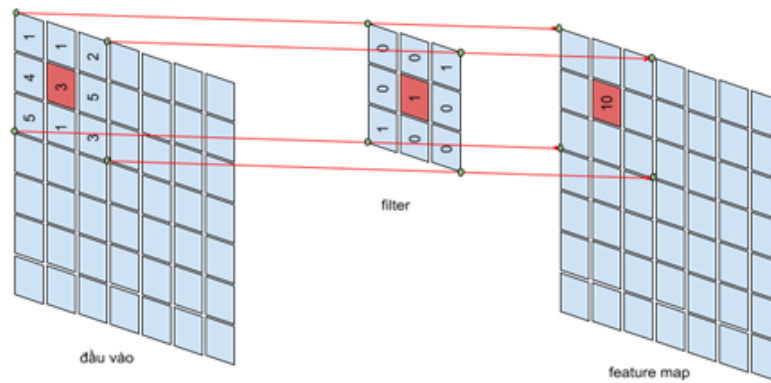
Ở những vị trí trên bức ảnh có dạng góc như đặc trưng ở bộ lọc thì mới có giá trị lớn trên feature map, những vị trí còn lại sẽ cho giá trị thấp hơn. Điều này có nghĩa là bộ lọc đã phát hiện thành công một dạng góc/cạnh trên dữ liệu đầu vào. Tập hợp của nhiều bộ lọc sẽ cho phép chúng ta phát hiện được nhiều loại đặc trưng khác nhau và giúp định danh được nhiều đối tượng.



Hình 2. 16: Ví dụ về bộ lọc phát hiện cạnh

#### 2.4.1.2 Các tham số của tầng tích chập

Kích thước bộ lọc là một trong những tham số quan trọng nhất của tầng tích chập. Kích thước này tỉ lệ thuận với số tham số cần học tại mỗi tầng tích chập và là tham số quyết định receptive field của tầng này. Kích thước phổ biến nhất của bộ lọc là 3x3.



Hình 2. 17: Cách hoạt động của bộ lọc

Kích thước bộ lọc nhỏ được ưu tiên lựa chọn thay kích thước lớn vì những lý do sau đây:

Bộ lọc nhỏ:

- Kích thước nhỏ thì mỗi lần nhìn được một vùng nhỏ các điểm ảnh.
- Rút trích được đặc trưng có tính cục bộ cao.
- Phát hiện được các đặc trưng nhỏ hơn.
- Đặc trưng rút trích được sẽ đa dạng, hữu ích hơn ở tầng sau.
- Giảm kích thước ảnh chậm hơn, do đó cho phép mạng sâu hơn.
- Ít trọng số hơn, chia sẻ trọng số tốt hơn.

Bộ lọc lớn:

- Receptive field lớn.
- Các đặc trưng có tính tổng quát hơn.
- Bắt được những phần cơ bản của bức ảnh.
- Thông tin ít, rút trích được ít đa dạng.
- Giảm kích thước ảnh nhanh, do đó chỉ cho phép mạng nông.
- Chia sẻ trọng số ít ý nghĩa hơn.

Kích thước filter của tầng convolution hầu hết đều là số lẻ, ví dụ như 3x3 hay 5x5. Với kích thước filter lẻ, các giá trị của feature map sẽ xác định một tâm điểm ở tầng phía trước. Nếu các chọn filter có kích thước 2x2, 4x4 thì chúng ta sẽ gặp khó khăn khi muốn tìm vị trí tương ứng của các giá trị feature map trên không gian ảnh.

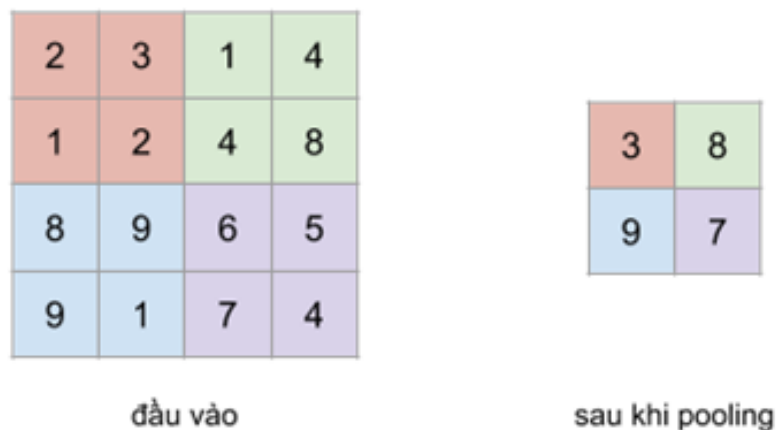
#### 2.4.2 Lớp phi tuyến (Nonlinear Layer)

ReLU (Rectified Linear Units,  $f = \max(0, x)$ ) là hàm kích hoạt phổ biến nhất cho CNN, được giới thiệu bởi Geoffrey E.Hinton năm 2010 [15]. Trước khi hàm ReLU được áp dụng thì những hàm như sigmoid hay tanh mới là những hàm được sử dụng phổ biến. Hàm ReLU được ưa chuộng vì tính toán đơn giản, giúp hạn chế tình trạng vanishing gradient và cũng cho kết quả tốt hơn. ReLU cũng như những hàm kích hoạt khác, được

đặt ngay sau tầng convolution, ReLU sẽ gán những giá trị âm bằng 0 và giữ nguyên giá trị của đầu vào khi lớn hơn 0. ReLU cũng có một số vấn đề tiềm ẩn như không có đạo hàm tại điểm 0, giá trị của hàm ReLU có thể lớn đến vô cùng và nếu chúng ta không khởi tạo trọng số cẩn thận hoặc khởi tạo learning rate quá lớn thì những neural ở tầng này sẽ rơi vào trạng thái chết, tức là luôn có giá trị nhỏ hơn 0.

### 2.4.3 Lớp tổng hợp (Pooling layer)

Sau hàm kích hoạt, chúng ta thường sử dụng tầng tổng hợp (pooling). Một số lớp tổng hợp phổ biến như là max-pooling, average pooling, với chức năng chính là giảm chiều của lớp đầu vào. Với một lớp tổng hợp có kích thước 2x2, cần phải trượt bộ lọc 2x2 này trên những vùng ảnh có kích thước tương tự rồi sau đó tính giá trị lớn nhất, trung bình cho vùng ảnh đó.



Hình 2. 18: Quá trình tổng hợp

Ý tưởng đằng sau của tầng pooling là vị trí tuyệt đối của những đặc trưng trong không gian ảnh không còn cần thiết, thay vào đó vị trí tương đối giữa các đặc trưng đã đủ để phân loại đối tượng. Tầng pooling có khả năng giảm chiều, làm hạn chế overfitting và giảm thời gian huấn luyện.

### 2.4.4 Lớp kết nối toàn bộ (Fully Connected)

Tầng cuối cùng của mô hình CNN trong bài toán phân loại ảnh là lớp kết nối toàn bộ (fully connected layer). Tầng này có chức năng chuyển feature map ở tầng trước thành vector chứa xác suất của các đối tượng cần được dự đoán. Quá trình huấn luyện mô hình CNN cho bài toán phân loại ảnh cũng tương tự như huấn luyện các mô hình khác, cần có hàm lỗi để tính sai số giữa dự đoán của mô hình và nhãn chính xác.

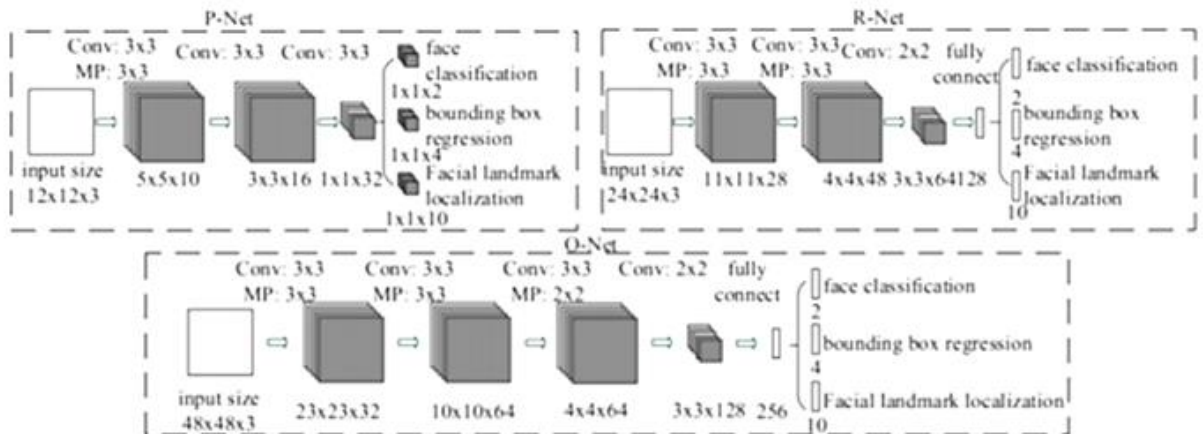
## 2.5 Phát hiện khuôn mặt MTCNN( Face Detector)

MTCNN[16] hoạt động theo ba bước, mỗi bước dùng một mạng neural riêng lần lượt là: mạng đề xuất P-Net (Proposal Network) nhằm dự đoán các vùng trong ảnh ví dụ là vùng chứa khuôn mặt, mạng tinh chế R-Net (Refine Network) sử dụng đầu ra của

P-Net để loại bỏ các vùng không phải khuôn mặt và mạng đầu ra O-Net (Output Network) sử dụng đầu ra R-Net để đưa ra kết quả cuối cùng với 5 điểm đánh dấu khuôn mặt: 2 điểm mắt, 1 điểm mũi và 2 điểm khóe miệng.

MTCNN hoạt động theo 3 bước, mỗi bước có một mạng neural riêng lần lượt là: P-Net, R-Net và O-Net.

### Sơ đồ quá trình phát hiện khuôn mặt



Hình 2. 19: Sơ đồ phát hiện khuôn mặt MTCNN

### Quy trình phát hiện khuôn mặt:

Với mỗi bức ảnh đầu vào, nó sẽ tạo ra nhiều bản sao của hình ảnh đó với các kích thước khác nhau.

#### Giai đoạn 1: P-Net

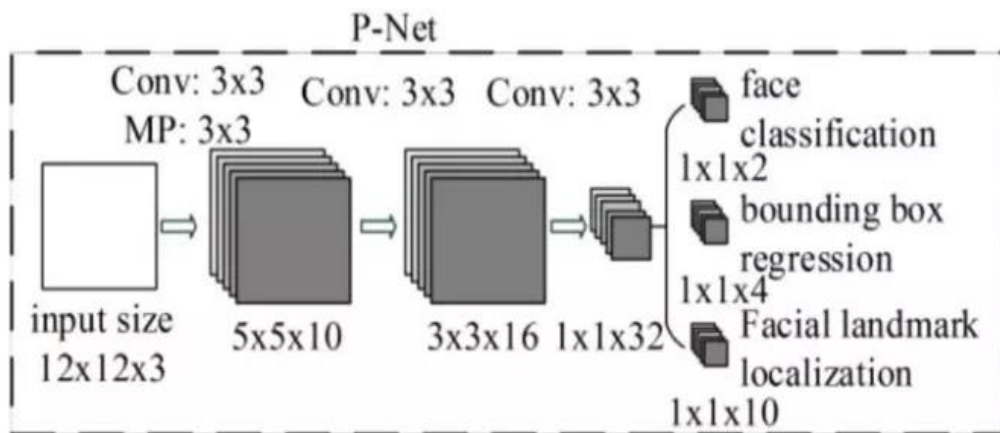
Trước hết, một bức ảnh thường sẽ có nhiều hơn một người - một khuôn mặt. Ngoài ra, những khuôn mặt thường sẽ có kích thước khác nhau. Ta cần một phương thức để có thể nhận dạng toàn bộ số khuôn mặt đó, ở các kích thước khác nhau. MTCNN đưa cho chúng ta một giải pháp, bằng cách sử dụng phép Resize ảnh, để tạo một loạt các bản copy từ ảnh gốc với kích cỡ khác nhau, từ to đến nhỏ, tạo thành 1 ảnh chồng nhau, gọi là Image Pyramid.



Hình 2. 20: Image Pyramid

Với mỗi một phiên bản copy-resize của ảnh gốc, ta sử dụng kernel  $12 \times 12$  pixel và  $\text{stride} = 2$  để đi qua toàn bộ bức ảnh, dò tìm khuôn mặt. Vì các bản sao của ảnh gốc có kích thước khác nhau, cho nên mạng có thể dễ dàng nhận biết được các khuôn mặt với kích thước khác nhau, mặc dù chỉ dùng 1 kernel với kích thước cố định (ảnh to hơn, mặt to hơn; ảnh nhỏ hơn, mặt nhỏ hơn).

Sau đó, ta sẽ đưa những kernels được cắt ra từ trên và truyền qua mạng P-Net (Proposal Network). Kết quả của mạng cho ra một loạt các bounding boxes nằm trong mỗi kernel, mỗi bounding boxes sẽ chứa tọa độ 4 góc để xác định vị trí trong kernel chứa nó.



Hình 2. 21: Mạng neural P-Net

Mạng P-Net sử dụng kiến trúc CNN gồm 3 lớp tích chập và 1 lớp co. Đầu vào của sơ trượt với kích thước  $12 \times 12 \times 3$  (với 3 tương đương với 3 màu: Đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường). Kết quả của P-Net gồm 3 cụm như sau:

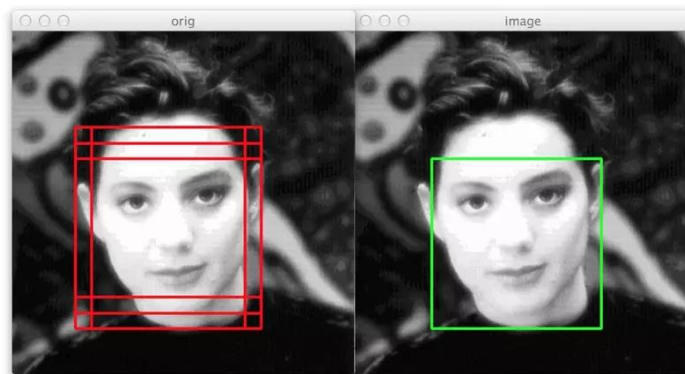
- Cụm thứ nhất có 2 bộ lọc kích thước  $1 \times 1$  nhận dạng khuôn mặt.
- Cụm thứ hai có 4 bộ lọc kích thước  $1 \times 1$  đóng khung 4 vị trí hộp giới hạn.
- Cụm thứ ba có 10 bộ lọc kích thước  $1 \times 1$  đóng khung 10 vị trí khuôn mặt.

Thuật toán sử dụng 1 kernel  $12 \times 12$  chạy qua mỗi bức hình để tìm kiếm khuôn mặt.



Hình 2. 22: Quá trình phát hiện khuôn mặt trong mạng đề xuất P-Net.

Để loại trừ bớt các bounding boxes trên các bức ảnh và các kernels, ta sử dụng 2 phương pháp chính là lập mức Threshold confident - nhằm xóa đi các box có mức confident thấp và sử dụng NMS (Non-Maximum Suppression) để xóa các box có tỷ lệ trùng nhau (Intersection Over Union) vượt qua 1 mức threshold tự đặt nào đó. Hình ảnh 2.23 là minh họa cho phép NMS, những box bị trùng nhau sẽ bị loại bỏ và giữ lại 1 box có mức confident cao nhất.



Hình 2. 23: NMS loại bỏ những box trùng nhau

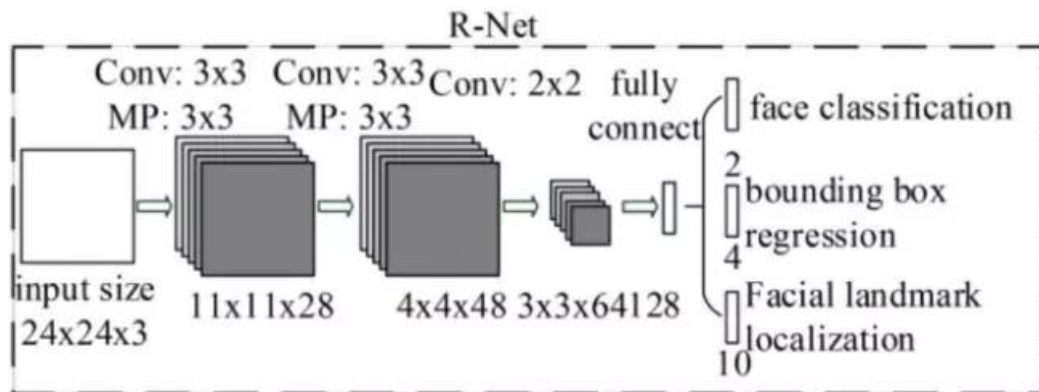
Sau khi đã xóa bớt các box không hợp lý, ta sẽ chuyển các tọa độ của các box về với tọa độ gốc của bức ảnh thật. Do tọa độ của box đã được normalize về khoảng (0,1) tương ứng như kernel, cho nên công việc lúc này chỉ là tính toán độ dài và rộng của kernel dựa theo ảnh gốc, sau đó nhân tọa độ đã được normalize của box với kích thước của kernel và cộng với tọa độ của các góc kernel tương ứng. Kết quả của quá trình trên sẽ là những tọa độ của box tương ứng ở trên ảnh kích thước ban đầu. Cuối cùng, ta sẽ

resize lại các box về dạng hình vuông, lấy tọa độ mới của các box và feed vào mạng tiếp theo, mạng R.

**Giai đoạn 2:** R-Net có cấu trúc tương tự với P-Net. Tuy nhiên sử dụng nhiều layer hơn. Tại đây, network sẽ sử dụng các bounding boxes được cung cấp từ P-Net và tính chỉnh là tọa độ.



Hình 2. 24: Quá trình phát hiện khuôn mặt trong mạng lọc R-Net



Hình 2. 25: Mạng neural R-Net

Trong bước R-net sử dụng kiến trúc CNN gồm: 3 lớp tích chập, 2 lớp co và 1 lớp kết nối đầy đủ. Đầu vào của sổ trượt với kích thước 24x24x3( 3 tương ứng với 3 màu: Đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường). Kết quả của R-Net phân được 3 cụm:

- Cụm thứ nhất có 2 lớp nhận dạng khuôn mặt.
- Cụm thứ hai có 4 lớp đánh dấu vị trí hộp giới hạn.
- Cụm thứ ba có 10 lớp vị trí khuôn mặt.



Mạng R (Refine Network) thực hiện các bước như mạng P. Tuy nhiên, mạng còn sử dụng một phương pháp tên là padding, nhằm thực hiện việc chèn thêm các zero-pixels vào các phần thiếu của bounding box nếu bounding box bị vượt quá biên của ảnh. Tất cả các bounding box lúc này sẽ được resize về kích thước 24x24, được coi như 1 kernel và feed vào mạng R.

Kết quả sau cũng là những tọa độ mới của các box còn lại và được đưa vào mạng tiếp theo, mạng O.

### Giai đoạn 3: O-Net

O-Net lấy các bounding boxes từ R-Net làm đầu vào và đánh dấu các tọa độ của các mốc trên khuôn mặt

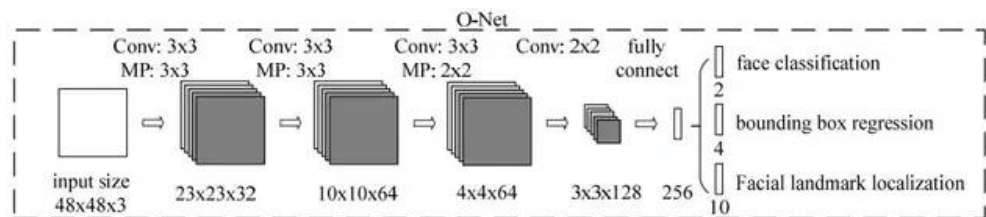


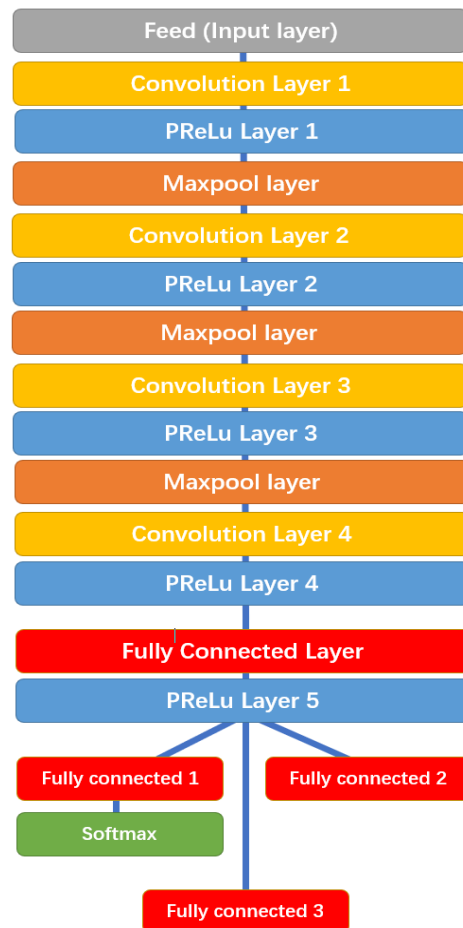
Fig. 2. The architectures of P-Net, R-Net, and O-Net, where "MP" means max pooling and "Conv" means convolution. The step size in convolution and pooling is 1 and 2, respectively.

Hình 2. 26: Mạng neural O-Net

Mạng O-Net sử dụng CNN gồm : 4 lớp tích chập, 2 lớp co, 1 lớp kết nối đầy đủ. Đầu vào của sơ trượt có kích thước 48x48x3( trong đó số 3 tương ứng với 3 màu: Đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường). Kết quả của O-Net phân được 3 cụm:

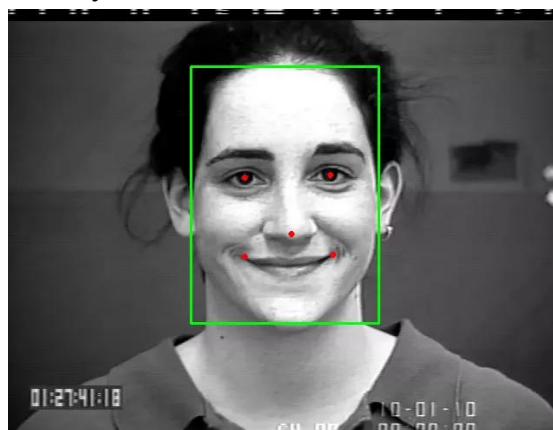
- Cụm thứ nhất có 2 lớp nhận dạng khuôn mặt.
- Cụm thứ hai có 4 lớp đánh dấu vị trí hợp giới hạn.
- Cụm thứ ba có 10 lớp vị trí khuôn mặt.





Hình 2. 27: Quá trình phát hiện khuôn mặt trong mạng đầu ra O-Net.

Cuối cùng là mạng O (Output Network), mạng cũng thực hiện tương tự như việc trong mạng R, thay đổi kích thước thành 48x48. Tuy nhiên, kết quả đầu ra của mạng lúc này không còn chỉ là các tọa độ của các box nữa, mà trả về 3 giá trị bao gồm: 4 tọa độ của bounding box (out[0]), tọa độ 5 điểm landmark trên mặt, bao gồm 2 mắt, 1 mũi, 2 bên cánh môi (out[1]) và điểm confident của mỗi box (out[2]). Tất cả sẽ được lưu vào thành 1 dictionary với 3 keys kể trên.

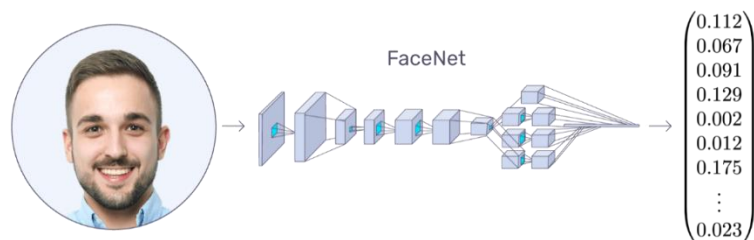


Hình 2. 28: Ảnh sau khi qua O-Net

## 2.6 Tìm hiểu mô hình Pre-trained model - FaceNet

Pre-trained model là mô hình đã được huấn luyện trước đó với một bộ dữ liệu lớn hoặc với các phương pháp tối ưu giúp giảm công sức huấn luyện mô hình từ đầu. Mô hình sau đó có thể được huấn luyện thêm để phù hợp với bộ dữ liệu thực tế hoặc sử dụng trực tiếp trong bài toán học máy.

FaceNet [17] là một mạng lưới thần kinh sâu được sử dụng để trích xuất các tính năng từ hình ảnh của một người mặt. Nó được xuất bản vào năm 2015 bởi các nhà nghiên cứu của Google.



Hình 2. 29: FaceNet lấy hình ảnh khuôn mặt làm đầu vào và xuất ra vector embedding

FaceNet lấy hình ảnh của mặt người làm đầu vào và xuất ra một vector 512 chiều, đại diện cho các tính năng quan trọng nhất của khuôn mặt. Trong học máy, vector này được gọi là nhúng (embeddings). Tại sao phải nhúng? Bởi vì tất cả các thông tin quan trọng từ một hình ảnh được nhúng vào vector này. Về cơ bản, FaceNet lấy một mặt người và nén nó thành một vector gồm 512 số. Khuôn mặt cần định danh cũng có nhúng tương tự.

Facenet chính là một dạng siam network có tác dụng biểu diễn các bức ảnh trong một không gian euclide n chiều (thường là 512) sao cho khoảng cách giữa các vector embedding càng nhỏ, mức độ tương đồng giữa chúng càng lớn.

Hầu hết các thuật toán nhận dạng khuôn mặt trước Facenet đều tìm cách biểu diễn khuôn mặt bằng một vector nhúng (embedding) thông qua một lớp có tác dụng giảm chiều dữ liệu:

- Tuy nhiên hạn chế của các thuật toán này đó là số lượng chiều embedding tương đối lớn (thường  $\geq 1000$ ) và ảnh hưởng tới tốc độ của thuật toán. Thường chúng ta phải áp dụng thêm thuật toán PCA để giảm chiều dữ liệu để giảm tốc độ tính toán.
- Hàm loss function chỉ đo lường khoảng cách giữa 2 bức ảnh. Như vậy trong một đầu vào huấn luyện chỉ học được một trong hai khả năng là sự giống nhau

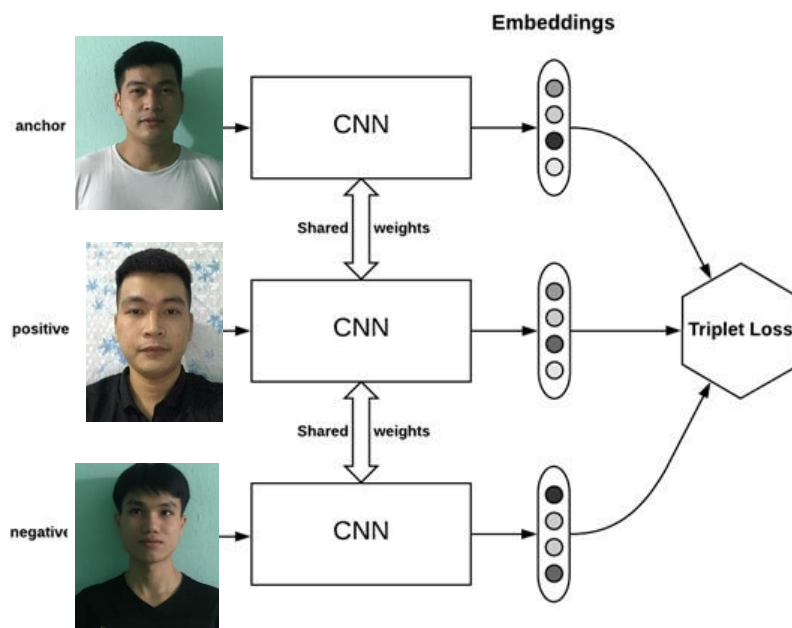
nếu chúng cùng 1 class hoặc sự khác nhau nếu chúng khác class mà không học được cùng lúc sự giống nhau và khác nhau trên cùng một lượt huấn luyện.

Facenet đã giải quyết cả 2 vấn đề trên bằng các hiệu chỉnh nhỏ nhưng mang lại hiệu quả lớn:

- Base network áp dụng một mạng convolutional neural network và giảm chiều dữ liệu xuống chỉ còn 512 chiều. Do đó quá trình suy diễn và dự báo nhanh hơn và đồng thời độ chính xác vẫn được đảm bảo.
- Sử dụng loss function là hàm triplet loss có khả năng học được đồng thời sự giống nhau giữa 2 bức ảnh cùng nhóm và phân biệt các bức ảnh không cùng nhóm. Do đó hiệu quả hơn rất nhiều so với các phương pháp trước đây.

## 2.7 Thuật toán Triplet loss trong trích xuất đặc trưng FaceNet

Trong facenet, quá trình encoding của mạng convolutional neural network (CNN) đã giúp ta mã hóa bức ảnh về 512 chiều. Sau đó những vector này sẽ làm đầu vào cho hàm triplet loss đánh giá khoảng cách giữa các vector.



Hình 2.30 :Triplet loss trên hai positive faces-mặt tích cực và một negative face-mặt tiêu cực

Mục tiêu của triplet loss là đảm bảo rằng:

- Hai ví dụ có cùng nhân có các vector nhúng của chúng gần nhau trong không gian nhúng.
- Hai ví dụ với các nhân khác nhau có các vector nhúng của chúng ở xa.

Trong trích xuất đặc trưng FaceNet, chúng ta cần lựa chọn 3 ảnh để thực hiện. Trong đó có 1 ảnh là Anchor và nó được cố định trước. Chúng ta sẽ lựa chọn 2 ảnh còn lại sao cho một ảnh là Negative (của một người khác với anchor) và một ảnh là positive (cùng một người với anchor).



Hình 2. 31: Lựa chọn ảnh thực hiện Triplet loss

Chúng ta cần lựa chọn các bộ 3 ảnh sao cho:

- Ảnh Anchor và Positive khác nhau nhất: Cần lựa chọn để khoảng cách  $d(A,P)$  lớn. Điều này cũng tương tự như bạn lựa chọn một ảnh của mình hồi nhỏ so với hiện tại để thuật toán học khó hơn. Nhưng nếu nhận biết được thì nó sẽ thông minh hơn.
- Ảnh Anchor và Negative giống nhau nhất: Cần lựa chọn để khoảng cách  $d(A,N)$  nhỏ. Điều này tương tự như việc thuật toán phân biệt được ảnh của một người anh em giống bạn.

**Hàm mất mát Triplet loss:** Luôn lấy 3 bức ảnh làm đầu vào và trong mọi trường hợp ta kì vọng:

$$d(A,P) < d(A,N) \quad (1)$$

Để làm cho khoảng cách giữa vế trái và vế phải lớn hơn, chúng ta sẽ cộng thêm vào vế trái một hệ số  $a$  không âm rất nhỏ. Khi đó công thức trở thành:

$$d(A,P) + a < d(A,N) \quad (2)$$

$$\|f(A) - f(P)\|_2^2 + a \leq \|f(A) - f(N)\|_2^2 \quad (3)$$

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + a \leq 0 \quad (4)$$

Như vậy hàm mất mát sẽ là:

$$L(A,P,N) = \sum_{i=0}^n \|f(A_i) - f(P_i)\|_2^2 - \|f(A_i) - f(N_i)\|_2^2 + a \leq 0 \quad (5)$$

Trong đó  $n$  là số lượng các bộ 3 hình ảnh được đưa vào huấn luyện. Sẽ không ảnh hưởng gì nếu ta nhận dạng đúng ảnh negative và positive là cùng cặp hay khác cặp với anchor. Mục tiêu của chúng ta là giảm thiểu các trường hợp mô hình nhận dạng sai ảnh negative thành positive nhất có thể. Do đó để loại bỏ ảnh hưởng của các trường hợp

nhận dạng đúng negative và positive lên hàm mất mát. Ta sẽ điều chỉnh giá trị đóng góp của nó vào hàm mất mát về 0.

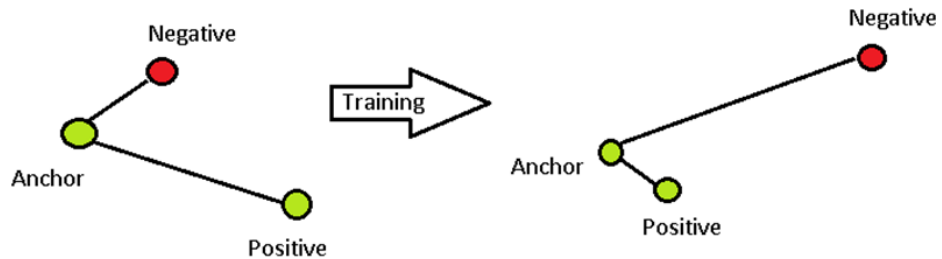
Tức là nếu:

$$\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + a \leq 0 \quad (6)$$

sẽ được điều chỉnh về 0. Hàm mất mát (loss function) trở thành:

$$L(A,P,N) = \sum_{i=0}^n \max\{\|f(A_i) - f(P_i)\|_2^2 - \|f(A_i) - f(N_i)\|_2^2 + a, 0\} \quad (7)$$

Như vậy khi áp dụng Triplet loss vào các mô hình convolutional neural network chúng ta có thể tạo ra các biểu diễn vector tốt nhất cho mỗi một bức ảnh. Những biểu diễn vector này sẽ phân biệt tốt các ảnh negative rất giống ảnh positive và đồng thời các bức ảnh thuộc cùng một nhãn sẽ trở nên gần nhau hơn trong không gian chiều euclidean.



Hình 2. 32: Triplet loss trước và sau khi được đào tạo[18]

## 2.8 Kỹ thuật căn chỉnh khuôn mặt (Face alignment):

Căn chỉnh khuôn mặt là nhiệm vụ xác định cấu trúc hình học của khuôn mặt trong hình ảnh kỹ thuật số và cố gắng có được sự căn chỉnh chính xác của khuôn mặt dựa trên sự biến dạng, tỷ lệ và xoay.

Có nhiều hình thức căn chỉnh khuôn mặt, một số phương pháp cố gắng áp đặt mô hình 3D (được xác định trước) sau đó áp dụng biến đổi cho hình ảnh đầu vào sao cho các mốc trên mặt đầu vào khớp với các mốc trên mô hình 3D.

Các phương pháp khác đơn giản hơn, chỉ dựa vào chính các mốc trên khuôn mặt (đặc biệt là vùng mắt) để thực hiện xoay, dịch chuyển và điều chỉnh tỷ lệ của khuôn mặt về cùng một kích thước.

Sau khi phát hiện các khuôn mặt có trên ảnh, các khuôn mặt có thể ở các trạng thái khác nhau, các góc độ khác nhau, có những khuôn mặt bị chéo và cũng có thể bị lệch do bước phát hiện chưa chính xác trong việc lấy ra khung hình chuẩn của mặt. Thì việc áp dụng căn chỉnh khuôn mặt ở đây là cần thiết, nó có thể hiểu như một hình thức

của chuẩn hóa dữ liệu, giúp tiêu chuẩn hoá lại dữ liệu trước khi đưa vào mô hình dự đoán. Điều này giúp cải thiện độ chính xác của mô hình nhận diện khuôn mặt.

## 2.9 Tìm hiểu Support Vector Machine

SVM[19] là phương pháp học có giám sát do Vladimir N. Vapnik đề xuất năm 1995 và ngày càng được sử dụng phổ biến trong nhiều lĩnh vực, đặc biệt là lĩnh vực phân loại mẫu và nhận dạng mẫu.

SVM sử dụng thuật toán học nhằm xây dựng một siêu phẳng làm cực tiểu hoá độ phân lớp sai của một đối tượng dữ liệu mới. Độ phân lớp sai của một siêu phẳng được đặc trưng bởi khoảng cách bé nhất tới siêu phẳng đấy. SVM có khả năng rất lớn cho các ứng dụng được thành công trong bài toán phân lớp văn bản. Như đã biết, phân lớp văn bản là một cách tiếp cận mới để tạo ra tập phân lớp văn bản từ các mẫu cho trước. Thông thường, hiệu quả ở mức độ cao không có các thành phần suy nghiệm. Phương pháp SVM có khả năng tính toán sẵn sàng và phân lớp, nó trở thành lý thuyết học mà có thể chỉ dẫn những ứng dụng thực tế trên toàn cầu. Đặc trưng cơ bản quyết định khả năng phân lớp là khả năng phân lớp những dữ liệu mới dựa vào những tri thức đã tích lũy được trong quá trình huấn luyện.

Sau quá trình huấn luyện nếu hiệu suất tổng quát hoá của bộ phân lớp cao thì thuật toán huấn luyện được đánh giá là tốt. Hiệu suất tổng quát hoá phụ thuộc vào hai tham số là sai số huấn luyện và năng lực của máy học. Trong đó sai số huấn luyện là tỷ lệ lỗi phân lớp trên tập dữ liệu huấn luyện. Kích thước VC là một khái niệm quan trọng đối với một họ hàm phân tách. Đại lượng này được xác định bằng số điểm cực đại mà họ hàm có thể phân tách hoàn toàn trong không gian đối tượng. Một tập phân lớp tốt là tập phân lớp có năng lực thấp nhất (có nghĩa là đơn giản nhất) và đảm bảo sai số huấn luyện nhỏ. Phương pháp SVM được xây dựng trên ý tưởng này.

### Khoảng cách từ 1 điểm đến một siêu mặt phẳng

Trong không gian 2 chiều khoảng cách từ một điểm có tọa độ  $(x_0; y_0)$  tới đường thẳng có phương trình  $ax + by + c = 0$  được xác định bởi:

$$\frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \quad (9)$$

mặt phẳng có phương trình  $ax + by + cz + d = 0$  được xác định bởi:

$$\frac{|ax_0 + by_0 + cz_0 + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (10)$$

Hơn nữa, nếu ta bỏ dấu trị tuyệt đối ở tử số, chúng ta có thể xác định được điểm đó nằm về phía nào của đường thẳng hay mặt phẳng đang xét. Những điểm làm cho biểu thức trong dấu giá trị tuyệt đối mang dấu dương nằm về cùng 1 phía. Những điểm nằm trên đường thẳng/mặt phẳng sẽ làm cho tử số có giá trị bằng 0, tức khoảng cách bằng 0.

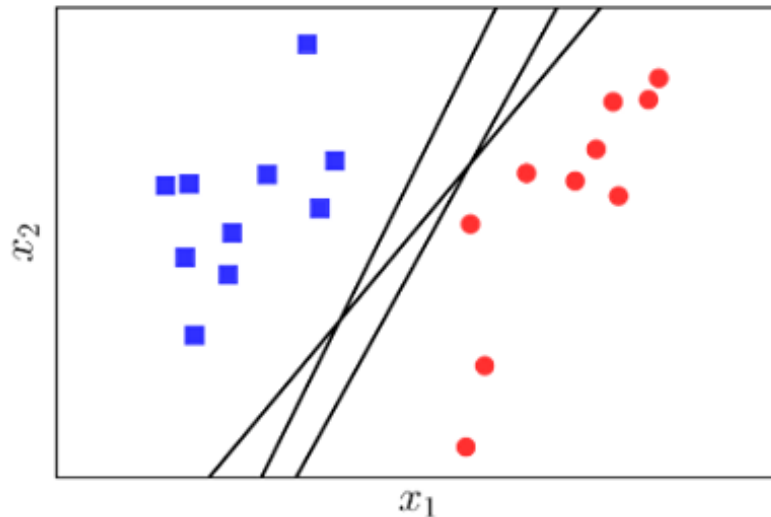
Việc này có thể tổng quát lên không gian nhiều chiều: Khoảng cách từ một điểm tọa độ  $x_0$  tới siêu mặt phẳng có phương trình  $a^T x + b = 0$  được xác định bởi:

$$\frac{|a^T x_0 + b|}{\|a\|_2} \quad (11)$$

Với  $\|a\|_2 = \sqrt{\sum_{i=1}^d a_i^2}$ ,  $i$  với  $d$  là số chiều của không gian

### Bài toán phân chia 2 lớp:

Giả sử rằng có hai lớp khác nhau được mô tả bởi các điểm trong không gian nhiều chiều. Hãy tìm một siêu mặt phẳng phân chia hai lớp đó, tức tất cả các điểm thuộc một lớp nằm về cùng một phía của siêu mặt phẳng đó và ngược phía với toàn bộ các điểm thuộc lớp còn lại.



Hình 2. 33: Mặt phẳng phân cách 2 lớp

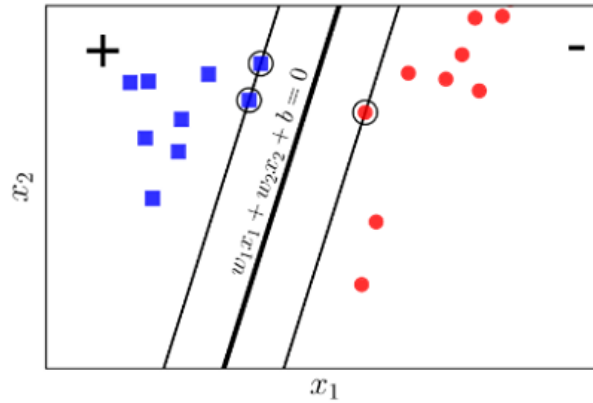
### Xây dựng bài toán tối ưu cho SVM

Giả sử rằng các cặp dữ liệu của training set là  $(x_1; y_1); (x_2; y_2), \dots, (x_n; y_n)$ , với vector  $X_i \in R^d$  thể hiện đầu vào của một điểm dữ liệu và  $y_i$  là nhãn của điểm dữ liệu đó.  $d$  là số chiều của dữ liệu và  $N$  là số điểm dữ liệu. Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi  $y_i = 1$  (class 1) hoặc  $y_i = -1$  (class 2). Chúng ta cùng xét trường hợp trong không gian hai chiều dưới đây.

Giả sử rằng các điểm vuông xanh thuộc class 1, các điểm vòng tròn thuộc class -1 và mặt  $a^T x + b = ax_1 + bx_2 + c = 0$  là mặt phân chia giữa 2 class. Hơn nữa class 1 nằm về phía dương, class -1 nằm về phía âm của mặt phân chia. Nếu ngược lại ta chỉ cần đổi dấu của "a và c", 1 điểm quan trọng sau đây: Với cặp dữ liệu  $(x_n; y_n)$  bất kỳ, khoảng cách từ điểm đó tới mặt phân chia là:

$$\frac{y_n(a^T x_n + b)}{\|a\|_2} \quad (12)$$

Điều này có thể dễ nhận thấy vì theo giả sử ở trên,  $y_n$  luôn cùng dấu với phía của  $x_n$ . Từ đó suy ra  $y_n$  cùng dấu với  $a^T x + b$  và tử số luôn là 1 số không âm.



Hình 2. 34: Phân tích bài toán SVM

Với mặt phân chia như hình 2.33, margin được tính là khoảng cách gần nhất từ 1 điểm tới mặt đó (bất kể điểm nào trong hai classes):

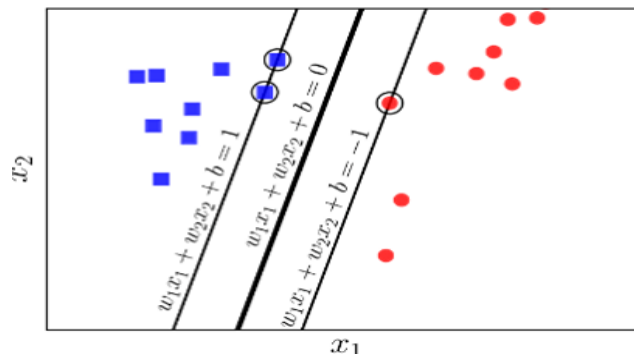
$$margin = \min \frac{y_n(a^T x_n + b)}{\|a\|_2} \quad (13)$$

Bài toán tối ưu SVM chính là bài toán tìm "a và c" sao cho margin này đạt giá trị lớn nhất:

Việc giải trực tiếp bài toán này sẽ rất phức tạp nhưng chúng ta sẽ thấy có cách để đưa nó về bài toán đơn giản hơn. Nhận xét quan trọng nhất là nếu thay vector hệ số  $a$  bởi  $k_a$  và  $k_c$  trong đó  $k$  là một hằng số dương thì mặt phân chia không thay đổi, tức khoảng cách từ từng điểm đến mặt phân chia không đổi, tức margin không đổi. Dựa trên tính chất này, có:

$$y_n(a^T x_n + b) = 1 \quad (14)$$

Với những điểm nằm gần mặt phẳng phân chia nhất, như hình 2.34.



Hình 2. 35: Các điểm gần 2 mặt phẳng phân chia nhất của 2 classes được khoanh tròn



## **2.10 Kết luận chương**

Qua chương này, nhóm đã tìm hiểu, nắm được các kiến thức, cách giao tiếp giữa Raspberry Pi với các thiết bị ngoại vi. Đồng thời, nhóm đã trình bày được cơ sở lý thuyết về mạng CNN, mô hình MTCNN, các đặc trưng của mạng FaceNet, thuật toán phân loại SVM được sử dụng trong hệ thống. Từ đó, nhóm sẽ đưa ra phương pháp, cách thức thực hiện mô hình hệ thống ở chương tiếp theo.

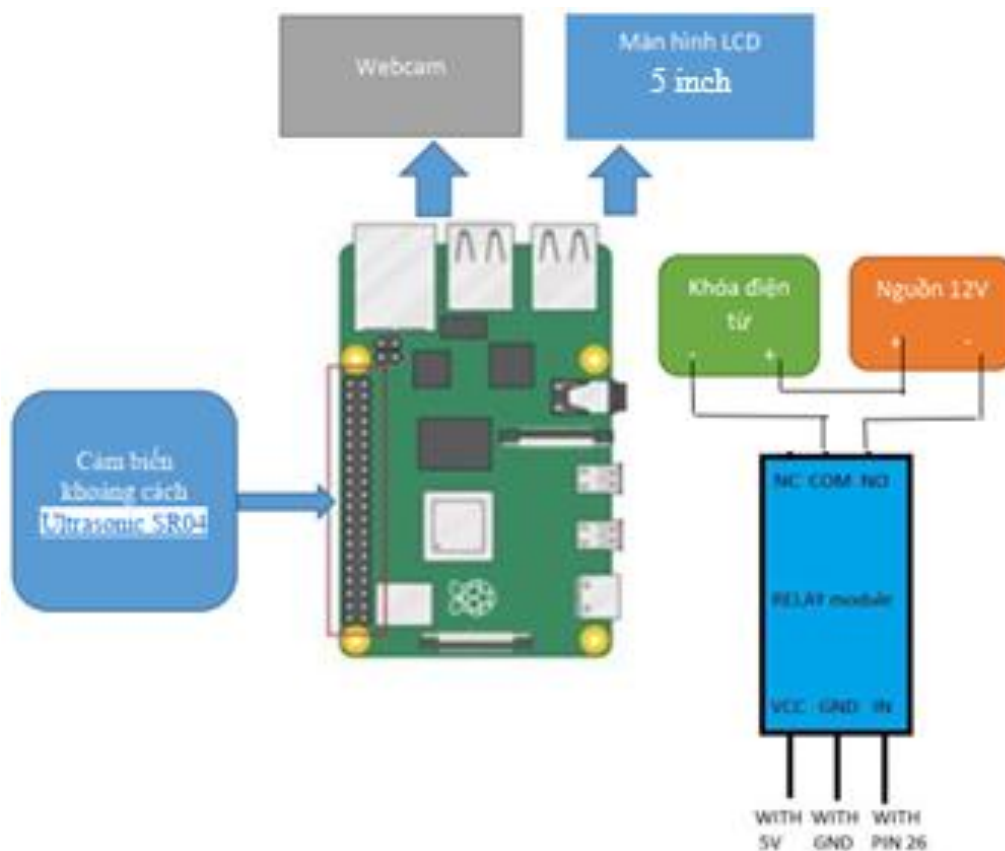
## CHƯƠNG 3: XÂY DỰNG HỆ THỐNG

### 3.1 Mở đầu chương

Trong chương này, nhóm sẽ trình bày về quá trình thực hiện, mô tả cách thức hoạt động, sơ đồ khối và lưu đồ của nhận diện khuôn mặt trên Raspberry Pi4.

### 3.2 Sơ đồ khối của hệ thống

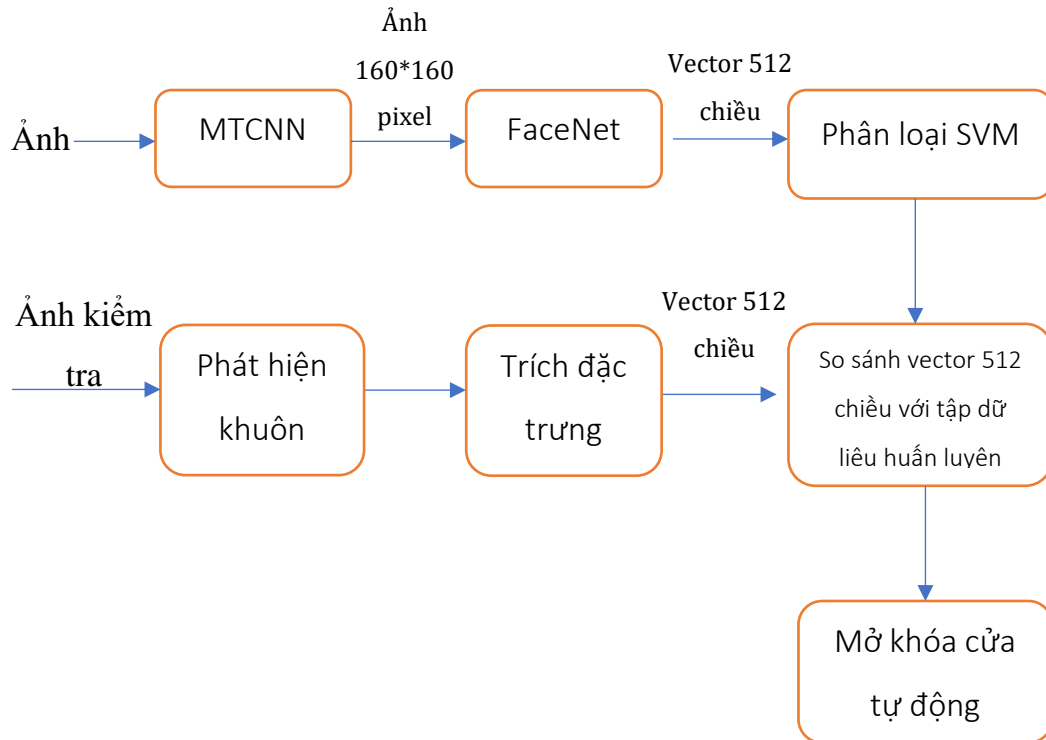
#### Sơ đồ khối thiết bị



Hình 3. 1: Sơ đồ khối thiết bị.

Hệ thống nhận dạng khuôn mặt gồm có các thiết bị ngoại vi relay module, cảm biến khoảng cách, bàn phím keypad... kết nối với Raspberry Pi. Khi hệ thống phát hiện và nhận dạng được khuôn mặt thì sẽ điều khiển relay module đóng/mở từ đó làm khóa điện từ đóng/mở theo, khi mà không nhận dạng được khuôn mặt, thì sẽ cho phép bấm mật khẩu bằng bàn phím keypad để đóng/mở khóa điện từ. Khi phát hiện chuyển động trước camera thì hệ thống sẽ ghi lại hình ảnh hoặc video gửi qua mail để cảnh báo cho người dùng.

Sơ đồ khối của hệ thống nhận dạng khuôn mặt:



Hình 3. 2: Sơ đồ khối của hệ thống mở cửa

**Mô tả:** Hệ thống mở cửa tự động bằng phương pháp nhận dạng khuôn mặt gồm 2 bước chính: Nhận dạng khuôn mặt và giao tiếp với relay module mở khóa điện từ.

Về nhận dạng khuôn mặt, hệ thống ban đầu cần phải tạo cơ sở dữ liệu. Cơ sở dữ liệu mà nhóm khởi tạo gồm 3 người, mỗi người hơn 2900 ảnh, ảnh lấy gần, nghiêng, đầy đủ ánh sáng để dễ nhận dạng khuôn mặt. Sau đó, dữ liệu sẽ được căn chỉnh, chỉ lấy mỗi khuôn mặt với kích thước ảnh 160\*160 pixel bằng mạng MTCNN. Sau khi cắt khuôn mặt thì sử dụng model FaceNet để trích xuất ra các vector 512 chiều và dùng SVM để phân loại rồi lưu vào mô hình.

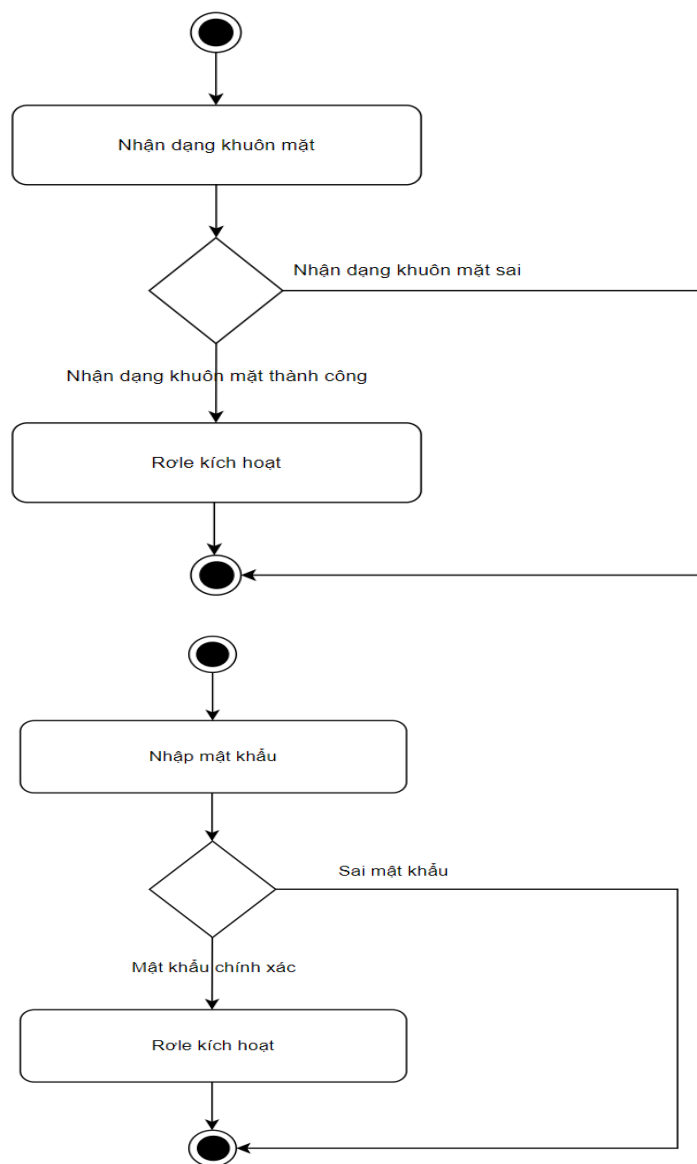
Để tiến hành nhận dạng khuôn mặt và mở cửa thì 1 người vào trong phạm vi <100 cm để cho cảm biến tiến hành quét và bật camera. Sau khi bật camera, thuật toán MTCNN được khởi chạy và phát hiện khuôn mặt người. FaceNet sẽ trích xuất đặc trưng ảnh thành vector 512 chiều, hệ thống sẽ tiến hành so sánh vector 512 chiều này với model được lưu vào trước đó bằng SVM. Vector nào trong cơ sở dữ liệu mà gần với

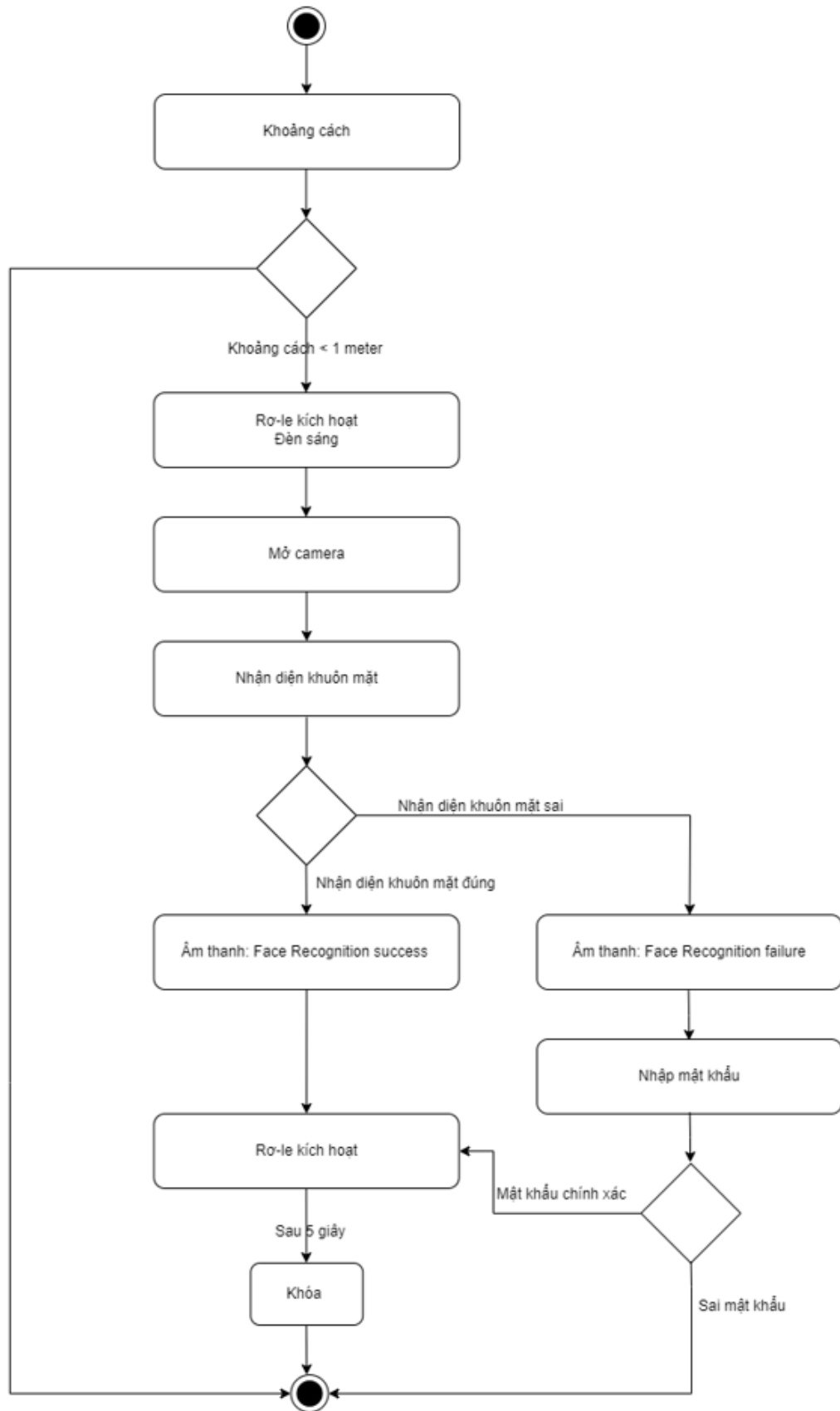
vector được kiểm tra thì tiến hành xuất ra tên và tỉ lệ chính xác, rồi tiếp tục giao tiếp với phần cứng mở khóa điện từ.

### 3.3 Lưu đồ hệ thống mở cửa tự động bằng phương pháp nhận dạng khuôn mặt

Hệ thống mở cửa tự động gồm các bước sau:

- Thu thập bộ dữ liệu khuôn mặt (dataset)
- Đọc ảnh từ camera
- Phát hiện khuôn mặt trên ảnh đầu vào (face detection)
- Trích xuất các vector đặc trưng của khuôn mặt (extract feature)
- Nhận dạng khuôn mặt (face recognition)
- Xử lý thông tin khuôn mặt vừa nhận dạng và tiến hành mở khóa.





Hình 3. 3: Lưu đồ hệ thống nhận dạng khuôn mặt mở cửa

Mô tả hệ thống:

Bắt đầu chương trình, nếu sensor bắt được khoảng cách (tức có người trong khoảng cách) Relay trigger => đèn sáng khi đó camera sẽ được bật nhận quá trình nhận diện bắt đầu. Input vào camera bắt khuôn mặt, dựa vào model đã train tính toán cho ra kết quả khi nhận diện.

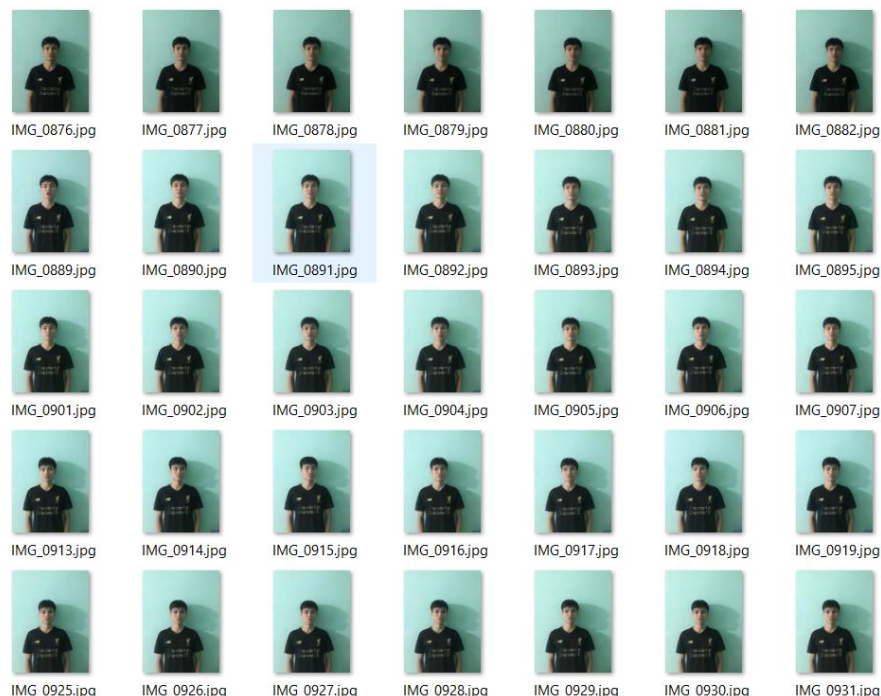
Nếu như nhận diện đúng xuất ra thông báo “Face Recognition success” relay trigger mở khóa, sau 5 giây khóa sẽ trở lại trạng thái ban đầu (Lock) => kết thúc chương trình.

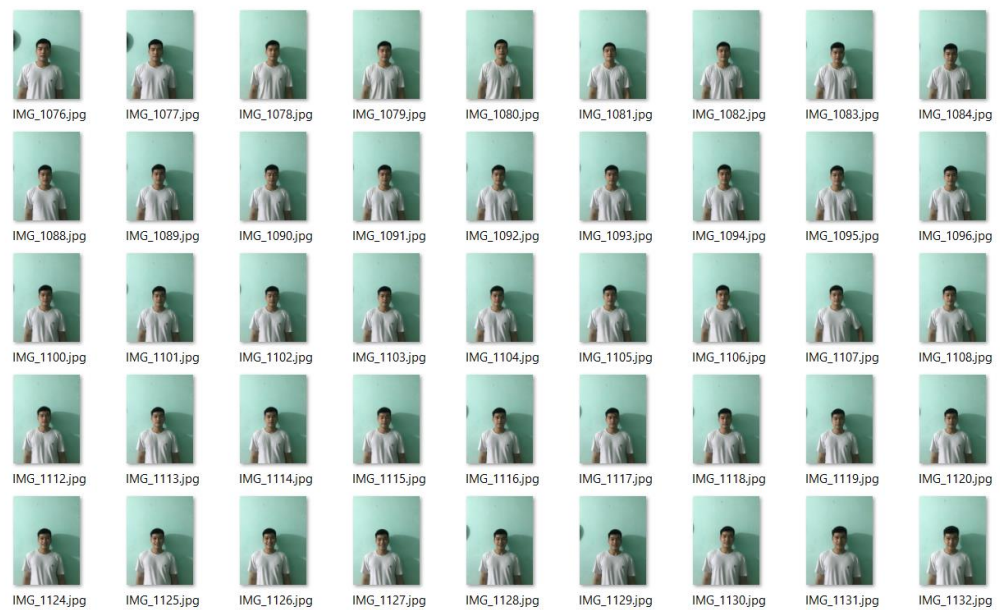
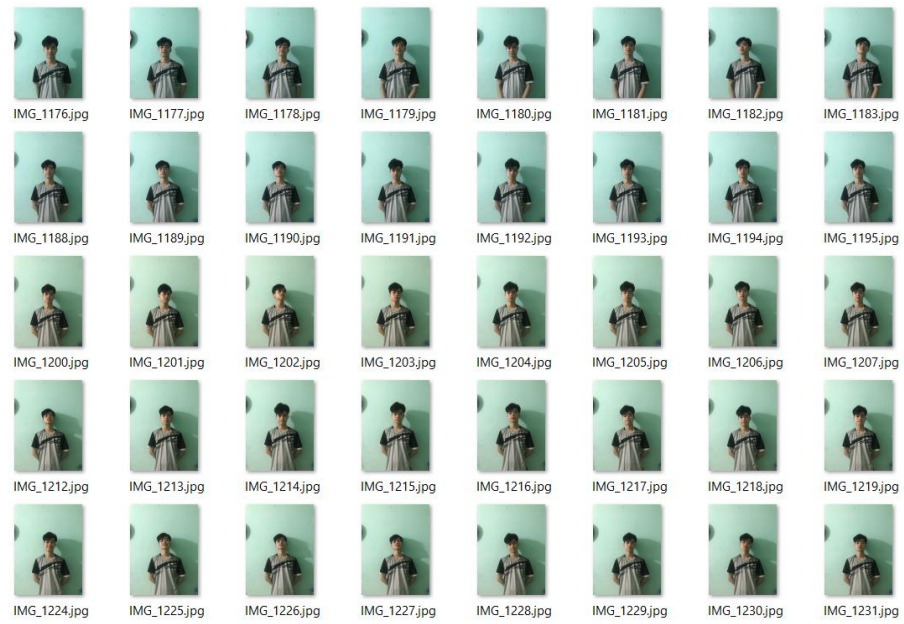
Nếu nhận diện sai xuất ra thông báo “Face Recognition failure” => Yêu cầu nhập mật khẩu. Nếu mật khẩu đúng Relay trigger, nếu sai thì kết thúc.

### 3.4 Cơ sở dữ liệu ảnh

Các thành viên trong nhóm đã thu thập ảnh bằng webcam của máy tính, ảnh chụp từ điện thoại. Các ảnh được thu thập cần đảm bảo các yếu tố như điều kiện ánh sáng, các góc độ khác nhau của khuôn mặt, . . . và khuôn mặt không nên có các vật cản như kính, khẩu trang,...

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





Hình 3. 4: Dữ liệu đã thu thập

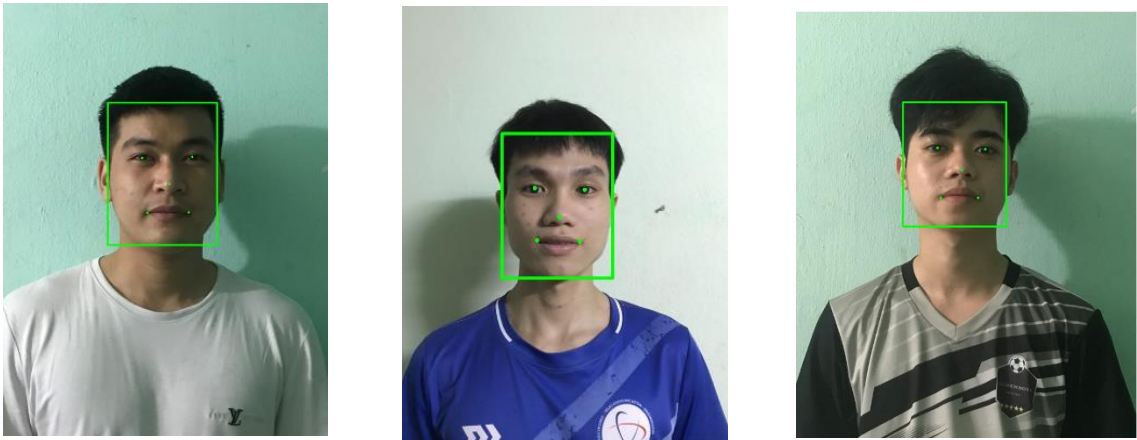
Nhóm đã tiến hành thu thập hơn 8700 bức ảnh của 3 người, với các góc chụp và biểu cảm, khung nền khác nhau để tăng độ chính xác.

### 3.5 Phát hiện khuôn mặt

Để trích chọn đặc trưng cho mỗi khuôn mặt, trước tiên chúng ta cần tìm ra vị trí khuôn mặt trong bức hình. Vì bộ dữ liệu sẽ bao gồm nhiều ảnh có điều kiện ánh sáng cũng như các góc độ của khuôn mặt khác nhau, cho nên việc lựa chọn cách phát hiện khuôn mặt cũng rất quan trọng để đảm hiệu quả cao nhất cho hệ thống.



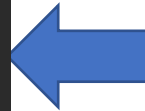
Kết quả cho thấy MTCNN hiệu quả với các khuôn mặt thẳng và những khuôn mặt bị che khuất ít. Kể cả những khuôn mặt nghiêng thì MTCNN cũng có thể phát hiện được, điều này thuận lợi cho việc trích xuất đặc trưng FaceNet.



Hình 3. 5 : Phát hiện khuôn mặt bằng MTCNN

### 3.6 Trích xuất đặc trưng cho khuôn mặt

-2.84519698e-02	-5.10276221e-02	-6.03105836e-02	4.44267057e-02
-8.37808996e-02	-1.35317240e-02	-6.61980435e-02	5.20075150e-02
7.28981430e-03	1.70165859e-02	-3.33273523e-02	-5.75600099e-03
9.50292125e-03	-5.65654412e-02	6.44711778e-02	2.12832727e-02
5.23021892e-02	4.94862981e-02	1.11566745e-02	3.13821458e-03
-4.79626022e-02	5.50328232e-02	5.00771515e-02	4.68698964e-02
5.22190146e-03	3.47724706e-02	-5.84698236e-03	2.39808857e-02
5.35608456e-02	1.06728133e-02	-1.57201365e-02	-1.04720153e-01
1.77023038e-02	5.28988801e-02	1.48668129e-03	1.54486941e-02
8.47174320e-03	-6.65551471e-03	-4.11354788e-02	4.72378395e-02
1.39460964e-02	-4.66311872e-02	7.90465102e-02	1.03075564e-01
5.73093519e-02	-1.90573670e-02	-7.07945377e-02	5.29652052e-02
1.55217648e-02	-4.78555225e-02	-6.40106201e-02	1.26074729e-02
6.35409132e-02	-7.21082985e-02	-5.12526557e-02	-7.20642358e-02
-4.42170352e-02	5.96023048e-04	7.90849850e-02	-7.86265582e-02
5.06585538e-02	9.99386758e-02	-2.25174911e-02	-4.92082871e-02



Hình 3. 6: Trích xuất vector bằng FaceNet

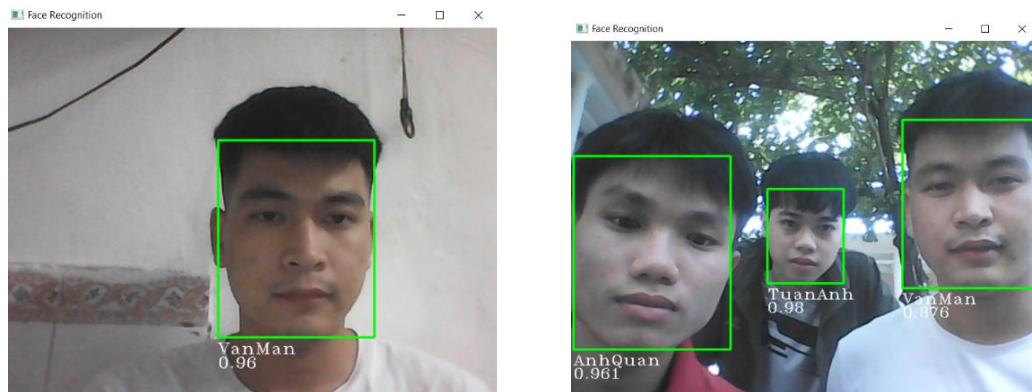
Mỗi ảnh trong cơ sở dữ liệu sau khi qua mô hình FaceNet sẽ tạo ra được 1 vector đặc trưng 512 chiều, vector này đại diện cho 1 ảnh. Sau khi trích xuất hết tất cả các vector trong cơ sở dữ liệu thì sẽ lưu lại trong tập dữ liệu các vector đặc trưng.

### 3.7 Nhận dạng khuôn mặt

Bước nhận dạng hay phân loại tức là xác định danh tính (identity) hay nhãn của ảnh (label) - đó là ảnh của ai. Ở bước nhận dạng / phân lớp, ta sử dụng phương pháp SVM (Support Vector Machine). SVM sẽ tiến hành phân lớp ảnh trong tập huấn luyện. Khi đưa ảnh vào nhận dạng, ảnh sẽ được so sánh tìm ra ảnh đó thuộc lớp nào.



Kết quả sau khi hệ thống nhận dạng được khuôn mặt như hình 3.7.



Hình 3. 7: Đối tượng sau khi đã nhận dạng

### 3.8 Kết luận chương

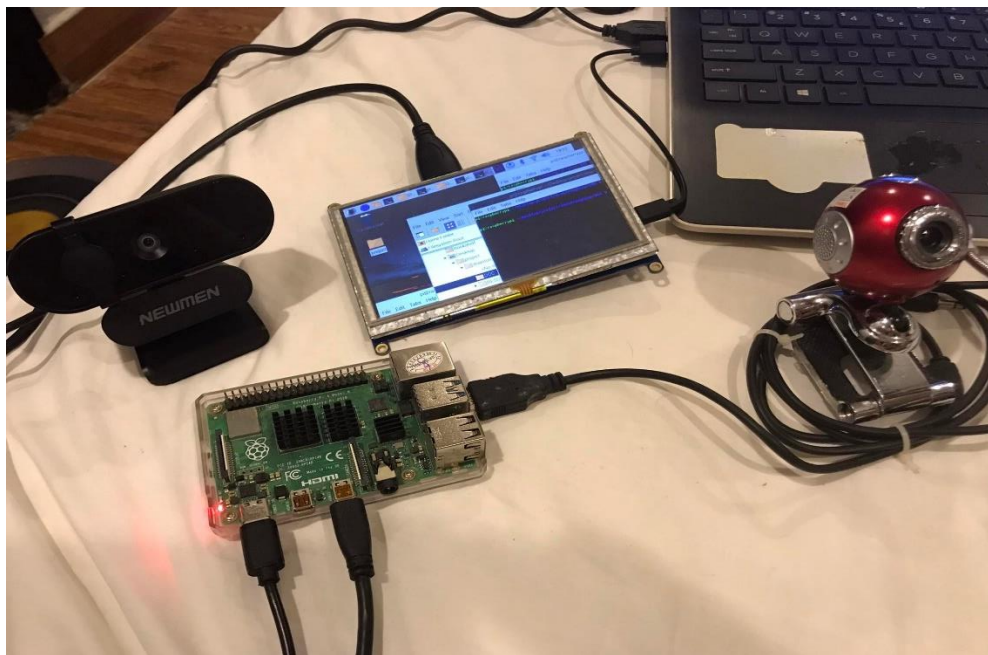
Trong chương này, nhóm đã mô tả sơ đồ khối, lưu đồ thuật toán của đề tài. Đồng thời, nhóm đã trình bày quá trình thực hiện, hoạt động của đề tài: Thu thập bộ dữ liệu khuôn mặt đưa vào dataset, tiến hành đọc hình ảnh từ camera, phát hiện khuôn mặt trên ảnh đầu vào (face detection) sau đó trích xuất các vector đặc trưng của khuôn mặt (extract feature) để nhận dạng khuôn mặt (face recognition), xử lý thông tin khuôn mặt vừa nhận dạng và tiến hành mở khóa

## CHƯƠNG 4: THỰC NGHIỆM, PHÂN TÍCH VÀ ĐÁNH GIÁ KẾT QUẢ

### 4.1 Mở đầu chương

Trong chương này, nhóm sẽ tiến hành thực nghiệm trên Raspberry pi4 phân tích kết quả đạt được, đánh giá tỉ lệ nhận dạng chính xác của mô hình. Từ đó, nhóm rút ra các kết luận, ưu, nhược điểm và hướng phát triển, cải tiến hệ thống.

### 4.2 Mô hình hệ thống



Hình 4. 1: Mô hình hệ thống

Bước 1: Hệ thống tự bật camera

Bước 2: Hệ thống nhận dạng khuôn mặt để mở khóa

Bước 3: Nếu quá trình nhận dạng thất bại, người dùng nhập mật khẩu để mở khóa

Mô hình hệ thống nhận dạng khuôn mặt gồm các quá trình: Khi có người bước vào khoảng cách dưới 1m thì camera tự động bật và tiến hành quét khuôn mặt để nhận dạng mở cửa. Nếu trong vòng 20s, hệ thống nhận dạng được khuôn mặt giống với cơ sở dữ liệu thì khóa cửa tự động mở, nếu không nhận dạng được thì hệ thống chuyển qua chế độ nhập mật khẩu để mở khóa.

### 4.3 Thực nghiệm trên Raspberry

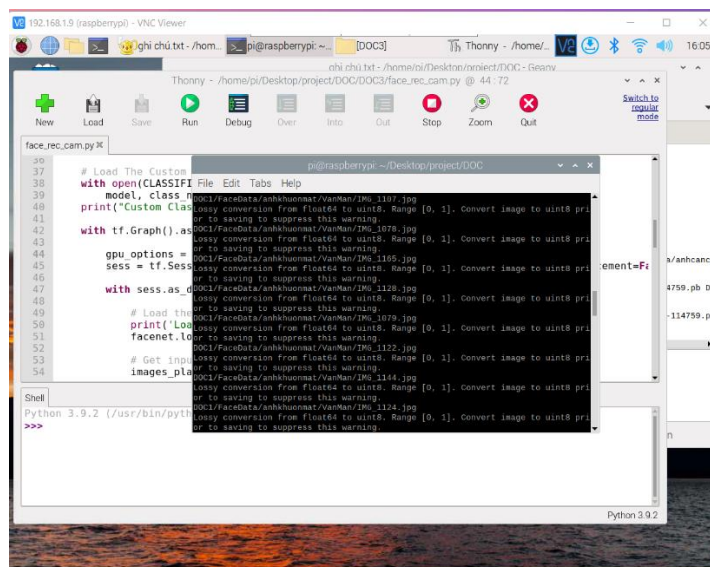
Tại thư mục Do\_An. (Sử Dụng “cd Desktop” “cd project” cd”Doc\_An”).

Tạo thư mục: lưu trữ dataset

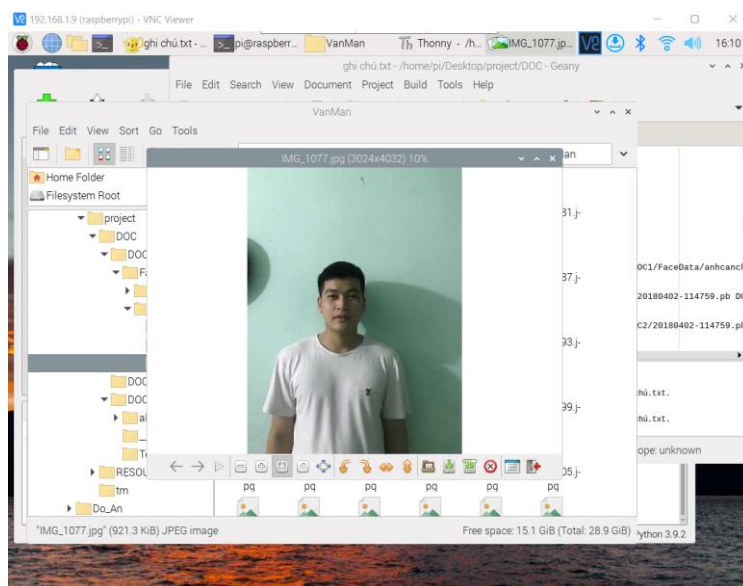
- FaceData
  - anhcanchinh
  - anhkhunmat
    - AnhQuan
    - TuanAnh (Dữ liệu vào)
    - VanMan

Sử dụng lệnh:

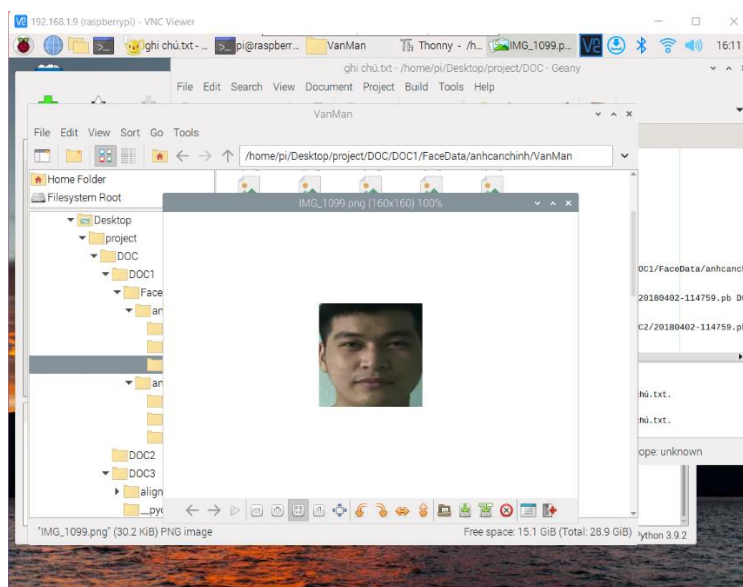
***python3 Do\_An3/align\_dataset\_mtcnn.py Do\_An1/FaceData/anhkhunmat Do\_An1/FaceData/anhcanchinh --image\_size 160 --margin 32 --random\_order --gpu\_memory\_fraction 0.25*** (Lệnh này dùng để cắt ảnh, căn chỉnh khuôn mặt – Tạo ra 3 thư mục mới ở thư mục anhcanchinh: AnhQuan, TuanAnh, VanMan lưu ảnh vừa crop và căn chỉnh vào đó)



Hình 4. 2: Hình minh họa quá trình căn chỉnh ảnh



Hình 4. 3: Ảnh trước khi được căn chỉnh

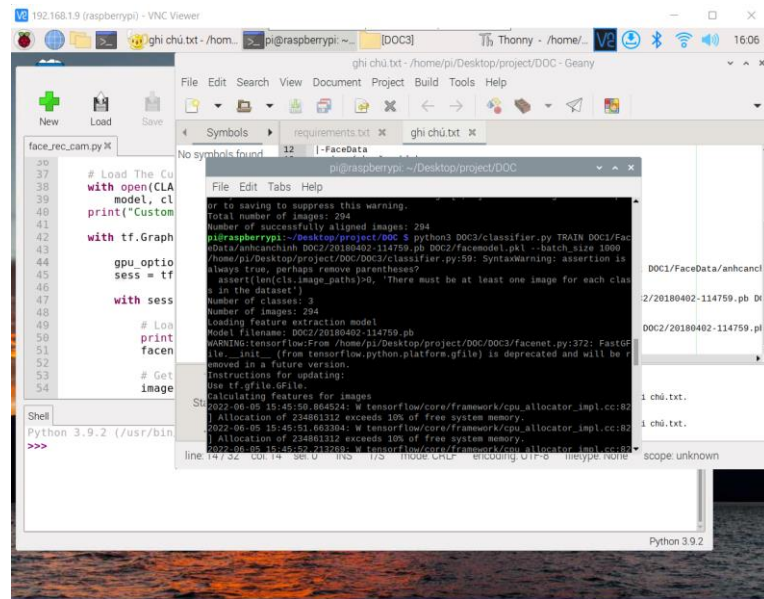


Hình 4. 4: Ảnh sau khi được căn chỉnh

Sử dụng lệnh:

*python3 Do\_An3/classifier.py TRAIN Do\_An1/FaceData/anhcanchinh*

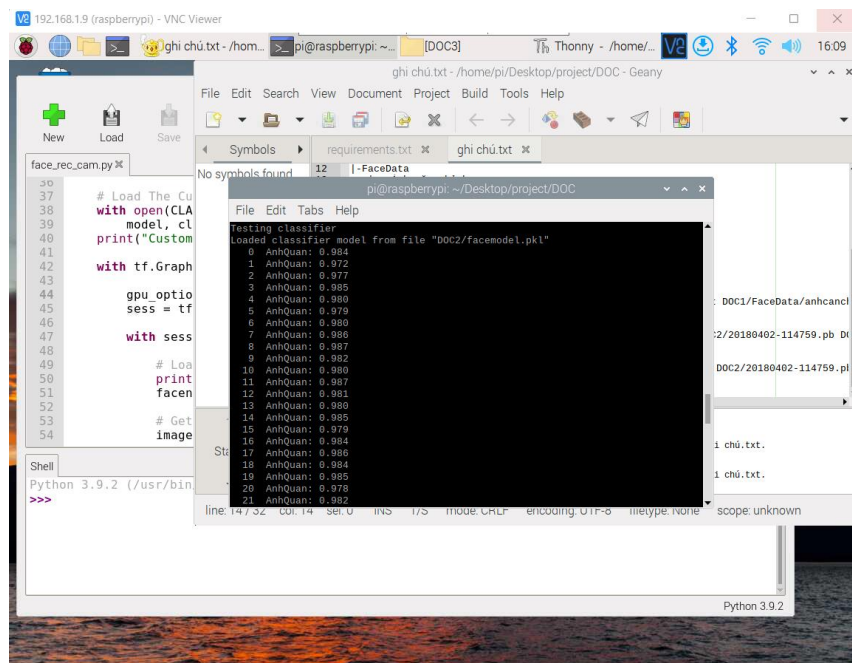
*Doc\_An2/20180402-114759.pb Do\_An2/facemodel.pkl --batch\_size 1000 (Dùng để train model).*



Hình 4. 5: Hình ảnh minh họa chạy chương trình train model

Sử dụng lệnh:

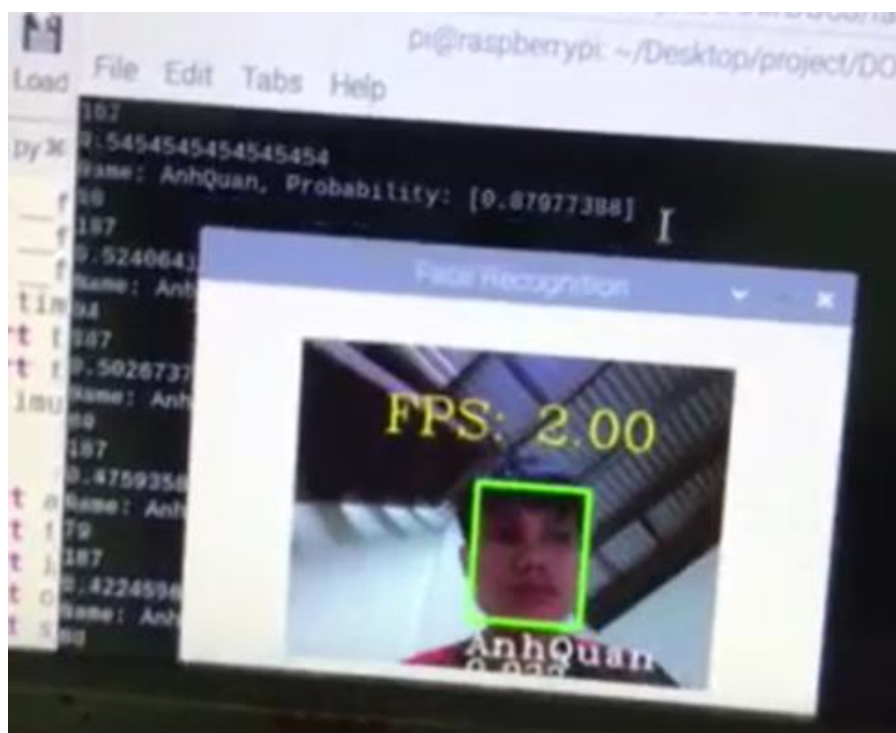
*python3 Do\_An3/classifier.py CLASSIFY Do\_An1/FaceData/anhcanchinh Do\_An2/20180402-114759.pb Do\_An2/facemodel.pkl --batch\_size 1000 (classifier model).*



Hình 4. 6: Hình ảnh minh họa chạy chương trình phân loại model

Sử dụng lệnh: *python3 Do\_An3/face\_rec\_cam.py* (Mở camera lấy nguồn đầu vào từ cam).





Hình 4. 7: Hình ảnh sử dụng camera lấy nguồn đầu vào

#### 4.4 Các phương pháp tối ưu hệ thống

Trong dự án này tôi sử dụng phần cứng Intel(R) Core(TM) i7-8550U, intel(R) UHD Graphics 620. Với bộ dữ liệu 8700 hình ảnh của 3 sinh viên. Thấy rằng kết số FPS đạt được chưa cao vì vậy chúng tôi đã cải thiện nó bằng cách phương pháp sau.

##### 4.4.1 Train lại model tăng số lượng ảnh.

Trong quá trình xử lý dữ liệu ảnh, cần quan tâm đến một số vấn đề sau:

Tư thế chụp, góc chụp: Ảnh chụp khuôn mặt có thể thay đổi rất nhiều bởi vì góc chụp giữa camera và khuôn mặt. Chẳng hạn như: chụp thẳng, chụp chéo bên trái hay chụp chéo bên phải, chụp từ trên xuống, chụp từ dưới lên, v.v... Với các tư thế khác nhau, các thành phần trên khuôn mặt như mắt, mũi, miệng có thể bị khuất một phần hoặc thậm chí khuất hết.

Sự xuất hiện hoặc thiếu một số thành phần của khuôn mặt: Các đặc trưng như: râu mép, râu hàm, mắt kính, v.v... có thể xuất hiện hoặc không. Vấn đề này làm cho bài toán càng trở nên khó hơn rất nhiều.

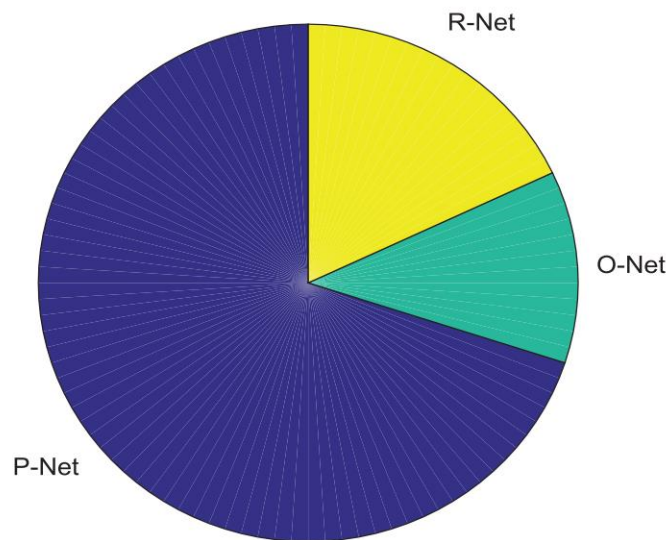
Sự biểu cảm của khuôn mặt: Biểu cảm của khuôn mặt con người có thể làm ảnh hưởng đáng kể lên các thông số của khuôn mặt. Chẳng hạn, cùng một khuôn mặt một người, nhưng có thể sẽ rất khác khi họ cười hoặc sợ hãi, v.v...

Sự che khuất: Khuôn mặt có thể bị che khuất bởi các đối tượng khác hoặc các khuôn mặt khác.

Hướng của ảnh (pose variations): Các ảnh khuôn mặt có thể biến đổi rất nhiều với các góc quay khác nhau của trục camera. Chẳng hạn chụp với trục máy ảnh nghiêng làm cho khuôn mặt bị nghiêng so với trục của ảnh.

Điều kiện của ảnh: Ảnh được chụp trong các điều kiện khác nhau về: chiều sáng, về tính chất camera (máy kỹ thuật số, máy hồng ngoại, v.v...), ảnh có chất lượng thấp ảnh hưởng rất nhiều đến chất lượng ảnh khuôn mặt.

#### 4.4.2 Thay đổi các trọng số trong thuật toán MTCNN

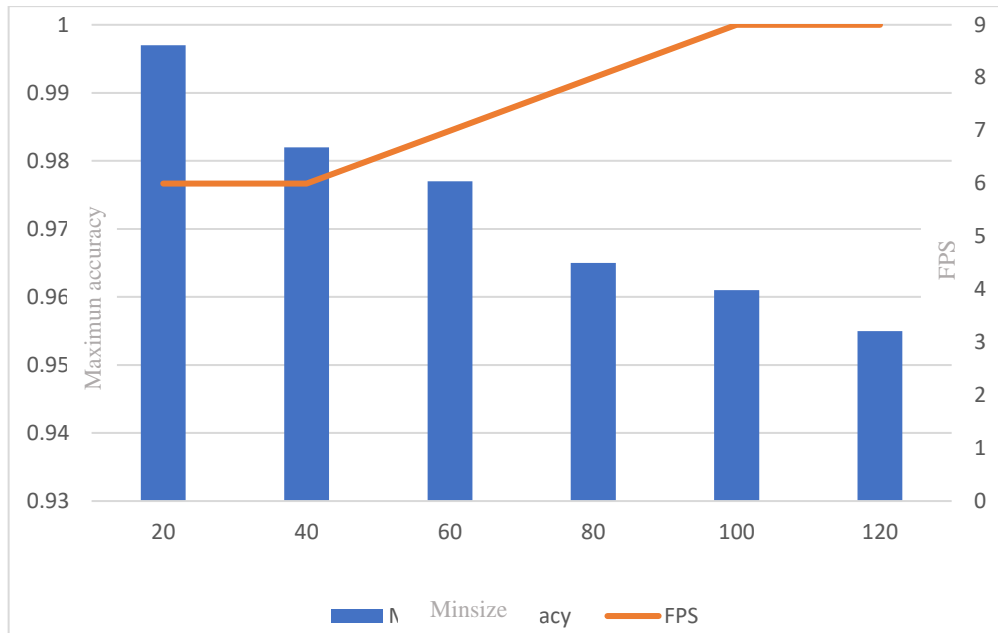


Hình 4. 8: Tỷ lệ thời gian của ba phần của mạng MTCNN.[21]

Theo phân tích dữ liệu trong Hình 4.6, kết luận rằng giai đoạn mô-đun PNet tiêu tốn nhiều thời gian nhất(70%), vì vậy để cải thiện tốc độ cần ưu tiên cho module Pnet.

Bảng 4. 1: So sánh hiệu suất thuật toán dưới các tham số minsize khác nhau

Minsize	Maximun accuracy	FPS
20	0.997	6
40	0.982	6
60	0.977	7
80	0.965	8
100	0.961	9
120	0.955	9



Hình 4. 9: Biểu đồ so sánh hiệu suất thuật toán dưới các tham số minsize khác nhau

Khi giá trị của kích thước tối thiểu(minsize) được đặt thành 20 và 100, tỷ lệ phát hiện đã giảm 3,6%, số FPS tăng đáng kể so với các minsize còn lại.

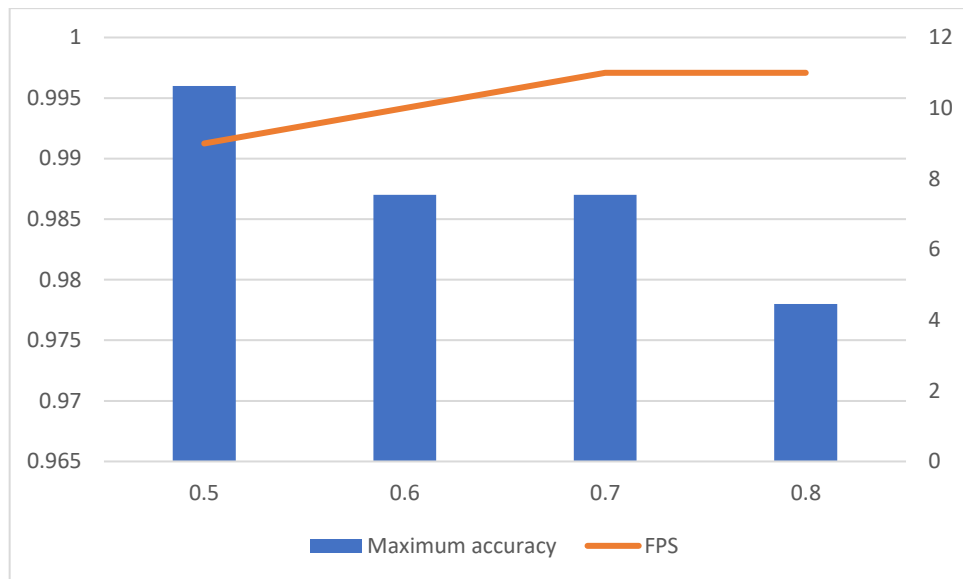
Vì kim tự tháp hình ảnh cần được tính toán một lần mỗi lần lặp lại P-Net, thời gian tính toán tổng thể của mô hình sẽ tăng lên. Kích thước tối thiểu có thể trực tiếp xác định tốc độ chạy tổng thể của MTCNN bằng cách kiểm soát số lần lặp lại của P-Net. Nếu giá trị của kích thước tối thiểu quá lớn, số lần lặp lại của kim tự tháp hình ảnh sẽ bị giảm đi, đồng thời giảm độ phức tạp về thời gian, nhưng dễ xảy ra hiện tượng bỏ sót các khuôn mặt có kích thước nhỏ. Nếu giá trị của kích thước tối thiểu quá nhỏ, việc nhận dạng các khuôn mặt có kích thước nhỏ sẽ chính xác hơn, nhưng số lần lặp lại của kim tự tháp hình ảnh sẽ tăng lên, và độ phức tạp về thời gian cũng vậy.

Trên cơ sở thay đổi kích thước nhỏ gọn từ 20 thành 100, chúng tôi sẽ tìm hiểu xem việc thay đổi ngưỡng mạng PNet có thể làm giảm đáng kể thời gian phát hiện hay không. Do đó, ngưỡng PNet được đặt thành 0,5, 0,6, 0,7, 0,8 và được thử nghiệm

Bảng 4. 2: So sánh hiệu suất thuật toán dưới các ngưỡng khác nhau.

Pnet Threshold	Maximum accuracy	FPS
0.5	0.996	9
0.6	0.987	10
0.7	0.987	11
0.8	0.978	11





Hình 4. 10: Biểu đồ so sánh hiệu suất thuật toán dưới các ngưỡng khác nhau

Sau khi phân tích dữ liệu trong Bảng 2, tỷ lệ phát hiện và thời gian phát hiện trung bình giảm nhẹ và thay đổi ít sau khi tăng giá trị ngưỡng PNet trong cài đặt ban đầu phát hiện trong ứng dụng thực tế. Do đó, ngưỡng PNet được chọn ở đây là 0,7. Cải tiến cuối cùng là điều chỉnh kích thước nhỏ gọn thành 100 và ngưỡng mạng PNet thành 0,7, đạt được phạm vi chấp nhận được.

```
def main():
    parser = argparse.ArgumentParser()
    parser.add_argument('--path', help='Path of the video you want to test on.', default=0)
    args = parser.parse_args()

    MINSIZE = 100
    THRESHOLD = [0.7, 0.7, 0.7]
    FACTOR = 0.709
    IMAGE_SIZE = 182
    INPUT_IMAGE_SIZE = 160
    CLASSIFIER_PATH = 'D:/project/DOC/DOC2/facemodel.pkl'
    VIDEO_PATH = args.path
    FACENET_MODEL_PATH = 'D:/project/DOC/DOC2/20180402-114759.pb'
```

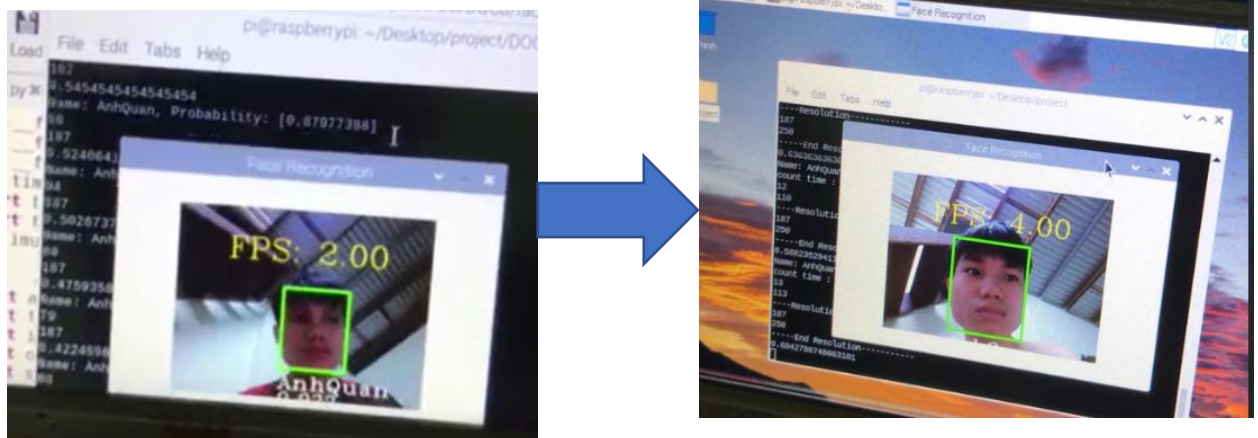
Hình 4. 11: Các biến trọng số của thuật toán MTCNN

#### 4.4.3 Phân luồng trên Raspberry Pi 4 Model B 4GB (Threading)

Sử dụng một luồng chuyên dụng (tách biệt với luồng chính) để đọc các khung hình từ cảm biến máy ảnh, chúng tôi có thể tăng đáng kể tốc độ xử lý FPS của quy trình. Sự tăng tốc này có được nhờ giảm độ trễ I/O và đảm bảo luồng chính không bao giờ bị chặn, cho phép chúng tôi lấy khung hình gần đây nhất mà máy ảnh đọc được vào bất kỳ

thời điểm nào. Sử dụng phương pháp đa luồng này, quy trình xử lý video của chúng tôi không bao giờ bị chặn, do đó cho phép chúng tôi tăng tốc độ xử lý FPS tổng thể của quy trình.

#### Kết quả phân luồng Raspberry Pi 4 Model B 4GB



Hình 4. 12: Kết quả của việc tối ưu trên Raspberry Pi4

Sau khi áp dụng các phương pháp trên thì số FPS đã tăng từ 2 lên đến 4 FPS.

#### 4.6 Đánh giá hệ thống

Sau khi hoàn thành quá trình mô phỏng với sự kết hợp giữa 2 thuật toán MTCNN và FaceNet chúng tôi đã tiến hành thử nghiệm để đánh giá độ chính xác của mô hình hệ thống dựa trên các điều kiện

*Bảng 4. 3: Đánh giá hệ thống*

Điều kiện		Số ảnh	Số ảnh nhận dạng đúng
Ánh sáng ổn định	Góc nhìn trực diện	200	200
	Lệch góc phải	200	190
	Lệch góc trái	200	195
	Hướng lên trên	200	180
	Hướng xuống dưới	200	180
	Tổng	1000	945
	Tỷ lệ nhận dạng chính xác	94.5%	
Môi trường thiếu sáng	Góc nhìn trực diện	200	185
	Lệch góc phải	200	169
	Lệch góc trái	200	160
	Hướng lên trên	200	160
	Hướng xuống dưới	200	151
	Tổng	1000	825
	Tỷ lệ nhận dạng chính xác	82.5%	
	Tổng	2000	88.5%

+ Từ kết quả thu được cho ta thấy rằng khi thực hiện với môi trường ánh sáng tốt, ổn định cho tỉ lệ nhận dạng khá cao so với điều kiện thiếu ánh sáng. Trong điều kiện ánh sáng tốt, hệ thống sẽ nhận dạng chính xác khuôn mặt trong phạm vi 100 cm. Sau đó khóa cửa lập tức được mở, hệ thống sẽ ở trạng thái mở cửa.

+ Với góc nhìn trực diện thì tỉ lệ nhận dạng đúng gần 100%, góc lệch khác cho tỉ lệ thấp hơn.

Qua kết quả cũng cho chúng ta thấy rằng với việc kết hợp giữa MTCNN và FaceNet thì tỉ lệ nhận dạng chính xác rất cao đặc biệt ở môi trường ánh sáng tốt, góc nhìn trực diện. Tuy vậy sự kết hợp giữa 2 thuật toán cũng có nhược điểm là điều kiện

ánh sáng yếu và góc lệnh cũng gây tỉ lệ nhận dạng không được chính xác như mong muốn.

**Kết luận:** Trên cơ sở tìm hiểu và nghiên cứu các thuật toán Deeplearning và kỹ thuật nhận dạng cụ thể, báo cáo đề án đạt kết quả sau đây:

Báo cáo đã giới thiệu khái quát được đề tài và các thuật toán được sử dụng. Trình bày được cách tiếp cận thuật toán phát hiện khuôn mặt MTCNN. Tiến hành thực nghiệm và phân loại bộ dữ liệu khuôn mặt đã thu thập (bộ dữ liệu gồm 3 người với hơn 8700 ảnh), kết quả thu được còn nhiều hạn chế do thời gian có hạn. Tuy nhiên, qua đó đã giới thiệu được quy trình phát hiện, nhận dạng khuôn mặt.

Tiến hành giao tiếp giữa các thiết bị ngoại vi với Raspberry Pi để thực hiện mở khóa điện từ. Các thiết bị ngoại vi như cảm biến đóng vai trò xác định được khoảng cách người dùng để bật camera, bàn phím keypad để nhập mật khẩu mở khóa, module relay để điều khiển đóng mở khóa điện từ.

Tiến hành xây dựng được mô hình tổng quan của hệ thống mở cửa tự động bằng phương pháp nhận dạng khuôn mặt.

**Ưu điểm:** Hệ thống áp dụng sử dụng các thuật toán Deeplearning, một trong những thuật toán nền tảng của công nghệ trí tuệ nhân tạo, nên tỉ lệ nhận dạng khuôn mặt cao, nhận dạng ở trong bất kỳ hoàn cảnh nào với những khuôn mặt góc nghiêng, thiếu ánh sáng...

#### **Nhược điểm:**

- Với việc sử dụng Raspberry trong hệ thống nhận dạng khuôn mặt thì tốc độ xử lý chậm hơn so với CPU của máy tính, số khung hình trên giây thấp, khả năng nhận dạng của hệ thống vẫn còn ít, chỉ nhận dạng được 3 người, chi phí làm hệ thống khá cao.
- Xử lý và sử dụng Deeplearning để huấn luyện sâu, cần sử dụng thiết bị có phần cứng có khả năng xử lý tính toán cao để có thể áp dụng cho một lượng dữ liệu khổng lồ.

**Hướng phát triển của hệ thống:** Với những nhược điểm của hệ thống thì nhóm cần phải cải thiện, tăng khả năng nhận dạng được số lượng người nhiều hơn để có thể áp dụng được nhiều lĩnh vực khác nhau. Hệ thống cần tích hợp module chống ảnh giả để tăng tính bảo mật cho hệ thống.

## 4.7 Kết luận chương

Trong chương này, nhóm chúng tôi đã tiến hành thực nghiệm trên phần cứng (bộ kit Raspberry Pi4) và đã thực hiện một số phương pháp tối ưu hóa để hệ thống nhận dạng tốt hơn.

.

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

Sau thời gian nghiên cứu đề tài đồ án "Mở cửa tự động bằng phương pháp nhận dạng khuôn mặt", nhóm đã có kết luận một số vấn đề: Những công việc về đề tài nhận dạng khuôn mặt gồm có 2 phần chính là tiến hành nhận dạng khuôn mặt và giao tiếp để điều khiển mở khóa. Bên cạnh những phần chính thì có những bước nhỏ như phát hiện khuôn mặt, trích xuất đặc trưng... và giao tiếp với các thiết bị ngoại vi. Về cơ bản, nhóm đã hoàn thành được mục tiêu đưa ra là tiến hành nhận dạng được 3 người và cho ra tỉ lệ nhận dạng khá cao. Bên cạnh đó, đồ án còn giúp các thành viên trong nhóm cải thiện những kỹ năng cơ bản như hoạt động nhóm, tính cẩn thận, kiên nhẫn, khả năng phân tích vấn đề và khả năng viết báo cáo. Mức độ hoàn thành đồ án như sau:

Mục tiêu 1: Xây dựng hướng đề tài và áp dụng những thuật toán deeplearning.

Mục tiêu 2: Áp dụng những thuật toán chạy trên Python.

Mục tiêu 3: Tiến hành phát hiện nhận dạng khuôn mặt với tỉ lệ chính xác trên 80 phần trăm.

Mục tiêu 4: Hoàn thành mô hình hệ thống

Trong thời đại phát triển vượt bậc của công nghệ hiện nay, nhất là trong giai đoạn của cuộc Cách mạng Công nghiệp 4.0, việc kết hợp IoTs, Machine Learning và các công nghệ tiên tiến hiện nay để đưa ra một hệ thống giải quyết vấn đề về nhận dạng là rất cần thiết. Bên cạnh đó, nhu cầu về việc sử dụng các thiết bị có tính bảo mật cao ngày càng cần thiết với các nhu cầu của xã hội, những thiết bị mang tính bảo mật ngày càng được áp dụng nhiều hơn. Đặc biệt là các thiết bị ứng dụng công nghệ liên quan đến "Nhận diện khuôn mặt bằng raspberry pi4".

Qua đồ án lần này, nhóm đã tìm hiểu được cách xây dựng bài toán nhận dạng đối tượng nói chung và nhận dạng khuôn mặt nói riêng. Cụ thể là đã tìm hiểu được các đặc trưng của phương pháp MTCNN và mạng FaceNet dùng để nhận dạng và phân loại khuôn mặt cũng như tìm hiểu được các ưu, nhược điểm của phương pháp trên. Dựa vào lý thuyết đã tìm hiểu, nhóm đã xây dựng được mô hình nhận dạng khuôn mặt chạy trên Raspberry Pi, sau đó mở khóa từ kết hợp với nhập mật khẩu từ bàn phím. Chương trình sử dụng các hàm được hỗ trợ trong thư viện OpenCV và viết bằng ngôn ngữ Python khá gọn nhẹ. Mô hình hoạt động ổn định và kết quả xử lý tương đối.

Bên cạnh những mặt mà đề tài đã đạt được thì vẫn còn những hạn chế chẳng hạn như khi số lượng đối tượng cần phân loại tăng lên thì tốc độ xử lý mô hình chậm, phần cứng chưa ổn định. Việc xây dựng CSDL còn nhiều hạn chế và đơn giản gây khó khăn cho quá trình nhận dạng. Nói chung, với việc áp dụng hệ thống nhận dạng khuôn mặt

để mở cửa tự động tạo những điều kiện thuận lợi cho việc đảm bảo tính bảo mật, tính an toàn khi ra vào các cơ quan. Các tính năng của hệ thống “Nhận diện khuôn mặt” đang áp dụng những công nghệ hiện đại hiện nay, đảm bảo được tính bảo mật, độ chính xác cao trong hệ thống.

Thông qua kết quả đạt được từ mô hình huấn luyện, nhằm đáp ứng tốt những nhu cầu thực tiễn phục vụ tốt cho người dùng, nhóm chúng tôi đã đưa ra một số hướng phát triển đề tài như sau:

- Nâng cao chất lượng CSDL để hệ thống nhận dạng chính xác hơn.
- Triển khai được việc tích hợp hệ thống chống ảnh giả vào trong hệ thống mở cửa tự động bằng phương pháp nhận dạng khuôn mặt. Chức năng có nhiệm vụ khi người khác sử dụng những bức ảnh có trong cơ sở dữ liệu mà không phải là khuôn mặt người thật thì hệ thống không nhận dạng được khuôn mặt và không tiến hành mở cửa, điều này giúp cho hệ thống nâng cao được tính bảo mật.
- Hiện tại thì hệ thống có tốc độ real-time không quá mượt với tốc độ khung hình thấp, cần tăng tốc độ quét khung hình để cho hệ thống chạy nhận dạng với tốc độ nhanh hơn.
- Triển khai mô hình hệ thống client-server để tăng tính bảo mật và tốc độ xử lý trên phần cứng cho việc nhận dạng.
- Phát triển hệ thống bằng việc kết nối mạng wifi, bluetooth để điều khiển từ xa.

## PHỤ LỤC

Source code:



## TÀI LIỆU THAM KHẢO

- [1]: Giới thiệu về PCA: [https://en.wikipedia.org/wiki/Principal\\_component\\_analysis](https://en.wikipedia.org/wiki/Principal_component_analysis)
- [2]: Giới thiệu về LDA: [https://en.wikipedia.org/wiki/Linear\\_discriminant\\_analysis](https://en.wikipedia.org/wiki/Linear_discriminant_analysis)
- [3]: Giới thiệu về CNN: <https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/>
- [4]: Relay module: <https://www.cytrontech.vn/p-single-channel-12v-relay-breakout-board?r=1>
- [5]: Giới thiệu về Relaymodule\_5V: <https://iotmaker.vn/relay-1-kenh-5v.html>
- [6]: Giới thiệu về cảm biến khoảng cách siêu âm HC-SR04: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [7]: Ultrasonic Ranging Module HC - SR04 <https://www.elabpeers.com/hc-ultrasonic-sensor.html>
- [8]: Giới thiệu về bàn phím ma trận keypad: <https://www.adafruit.com/product/419>
- [9]: <https://hshop.vn/products/ban-phim-mem-3x4>
- [10]: <https://khuenguyencreator.com/lap-trinh-stm32-quet-ma-tran-phim-keypad-3x4/>
- [11]: SPI-I2C Interface Design Example, Actel power matters
- [12]: Nguyễn Văn Châu, Báo cáo đồ án tốt nghiệp, "Xây dựng ứng dụng đếm danh qua hình ảnh",11-30,2020.
- [13]:<https://viblo.asia/p/deep-learning-tim-hieu-ve-mang-tich-chap-cnn-maGK73bOKj2>
- [14]: Nguyễn Văn Châu, Báo cáo đồ án tốt nghiệp, "Xây dựng ứng dụng đếm danh qua hình ảnh",16-30,2020.
- [15]: Geoffrey E.Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines”,University of Toronto, Toronto, ON M5S 2G4, Canada, 2010.
- [16]: Rosa Gradilla “Multi-task Cascaded Convolutional Networks for Face detection and Facial Landmark Alignment”, Target detection-MTCNN face recognition <https://www.programmingsought.com/article/62053721961/>
- [17]: Giới thiệu mô hình Facenet trong Face recognition\_Phạm Đình Khanh github :<https://phamdinhhkhanh.github.io/2020/03/12/faceNetAlgorithm.html>
- [18]: Triplet loss: <https://ichi.pro/vi/giai-thich-ve-siamese-net-triplet-loss-va-circle-loss-208539308200695>
- [19]: Support Vector Machine: <https://machinelearningcoban.com/2017/04/09/smv/>

- [20]: Sơ đồ khối, quy trình thiết kế, công cụ cần để thực hiện, triển khai mô hình:  
<https://github.com/ravirajsinh45/Face-recognition-home-door-lock-system/blob/master/final.py>
- [21]: <https://www.worldscientific.com/doi/pdf/10.1142/S0218001422540179>