

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



## CÔNG NGHỆ MÁY TÍNH HIỆN ĐẠI

---

# AIOT VÀ ỨNG DỤNG KHÓA CỦA THÔNG MINH NHẬN DIỆN KHUÔN MẶT

---

GVHD: TS. Nguyễn Tất Bảo Thiện  
HV: Đoàn Minh Trung - 220202016  
Nguyễn Hữu Quyền - 220202021

# Mục lục

<b>1 Giới thiệu</b>	<b>5</b>
1.1 Đặt vấn đề: . . . . .	5
1.2 Mục tiêu: . . . . .	6
<b>2 Cơ sở lý thuyết:</b>	<b>7</b>
2.1 Tổng quan về Raspberry Pi: . . . . .	7
2.1.1 Giới thiệu về Board Raspberry Pi: . . . . .	7
2.1.2 Cấu hình của Raspberry Pi: . . . . .	8
2.1.3 Kết nối của Raspberry Pi: . . . . .	9
2.1.4 Giới thiệu hệ điều hành cho Raspberry Pi: . . . . .	11
2.2 Giới thiệu về Relay . . . . .	12
2.3 Tổng quan về thư viện OpenCV: . . . . .	14
2.3.1 Ứng dụng của OpenCV: . . . . .	15
2.3.2 Tính năng và các module phổ biến của OpenCV: . . . . .	16
2.4 Tổng quan về Nextjs . . . . .	17
2.4.1 Nextjs là gì? . . . . .	17
2.4.2 Tính năng chính của Nextjs: . . . . .	18
2.4.3 Ưu và nhược điểm của Nextjs . . . . .	19
2.5 Firebase . . . . .	20
2.5.1 Firebase là gì? . . . . .	20
2.5.2 Cách thức hoạt động của Firebase . . . . .	21
2.5.3 Ưu và nhược điểm của Firebase . . . . .	22
<b>3 Tính toán và thiết kế hệ thống</b>	<b>23</b>
3.1 Tính toán và thiết kế mạch . . . . .	23
3.2 Mô hình đề xuất . . . . .	25
<b>4 Thực nghiệm và đánh giá</b>	<b>26</b>
4.1 Môi trường và ngôn ngữ cài đặt . . . . .	26
4.2 Dữ liệu . . . . .	27
4.3 Thang đo . . . . .	27
4.4 Nhận diện khuôn mặt sử dụng thư viện OpenCV . . . . .	28
4.4.1 Huấn luyện mô hình nhận diện khuôn mặt cho RPi . . . . .	28
4.4.2 Khóa cửa nhận diện khuôn mặt . . . . .	31

4.5	Nhận diện khuôn mặt đeo khẩu trang . . . . .	34
4.5.1	Cấu hình mạng học sâu để huấn luyện . . . . .	34
4.5.2	Kết quả huấn luyện mô hình nhận diện khuôn mặt có khẩu trang . . . . .	35
4.6	Ứng dụng web . . . . .	37
<b>5</b>	<b>Kết luận và hướng phát triển</b>	<b>41</b>
5.1	Kết luận . . . . .	41
5.2	Hướng phát triển . . . . .	41
	<b>Tài liệu tham khảo</b>	<b>41</b>

## Danh sách hình vẽ

1	Hình ảnh minh họa khóa cửa thông minh và khóa cửa truyền thống . . . . .	6
2	Board Raspberry Pi. . . . .	7
3	Những thông số sơ lược về RPi B v3. . . . .	8
4	Tìm hiểu về các chân GPIO của RPi. . . . .	10
5	RPi và các thiết bị kết nối với nó. . . . .	11
6	Module Relay. . . . .	13
7	Sơ đồ khối của relay điện từ. . . . .	14
8	Logo của OpenCV . . . . .	15
9	Hình ảnh minh họa của việc ứng dụng OpenCV vào nhận diện khuôn mặt. . . . .	17
10	Hình ảnh minh họa về framework Nextjs. . . . .	18
11	Hình ảnh minh họa về Firebase. . . . .	20
12	Sơ đồ nguyên lý Gateway. . . . .	24
13	Hình ảnh thực tế Relay và cấu tạo bên trong của Relay. . . . .	24
14	Cơ cấu tác động của Relay. . . . .	25
15	Mô hình đề xuất. . . . .	26
16	Kết nối giao diện RPi đến laptop sử dụng VNC viewer. . . . .	27
17	Code thực hiện chụp ảnh chân dung để huấn luyện cho mô hình. . . . .	29
18	Code huấn luyện mô hình. . . . .	30
19	Quá trình huấn luyện cho mô hình. . . . .	31
20	Một đoạn code minh họa mô hình khóa cửa nhận diện khuôn mặt. . . . .	32
21	Khóa cửa mở ra khi nhận diện được khuôn mặt của Trung. . . . .	33
22	Khóa cửa đóng lại khi không nhận diện được khuôn mặt của Trung. . . . .	33
23	Cấu hình mô hình nhận diện khuôn mặt có khẩu trang với Tensorflow. . . . .	34
24	Hiệu suất mô hình nhận diện khuôn mặt có khẩu trang. . . . .	35
25	Mô hình nhận diện khuôn mặt khi không đeo khẩu trang. . . . .	36
26	Mô hình nhận diện khuôn mặt khi không đeo khẩu trang kể cả khi dùng tay che mặt. . . . .	36
27	Mô hình nhận diện khuôn mặt khi đeo khẩu trang. . . . .	37
28	Tạo đăng nhập để quản trị viên truy cập dữ liệu người dùng. . . . .	37
29	Giao diện chính của trang web. . . . .	38
30	Thêm người dùng vào bộ dữ liệu. . . . .	38
31	Giao diện sau khi thêm người dùng. . . . .	39
32	Tùy chọn chỉnh sửa dữ liệu người dùng. . . . .	39

33	Cập nhật dữ liệu người dùng, cụ thể là hình ảnh đại diện và ghi chú. . . . .	40
34	Giao diện sau khi cập nhật dữ liệu người dùng. . . . .	40

## Danh sách bảng

1	Bảng chân kết nối giữa RPi và Relay. . . . .	25
2	Danh sách linh kiện sử dụng. . . . .	26
3	Số lượng mẫu dữ liệu trong tập Train và tập Test. . . . .	27

# 1 Giới thiệu

## 1.1 Đặt vấn đề:

Trong thời đại công nghệ hiện đại, việc bảo vệ an ninh cho các căn nhà, văn phòng và các không gian khác trở nên ngày càng quan trọng. Việc sử dụng khóa cửa truyền thống đã được thực hiện từ lâu và đóng vai trò quan trọng trong việc kiểm soát truy cập và bảo vệ an toàn.

Tuy nhiên, khóa cửa truyền thống cũng tồn tại một số hạn chế và vấn đề cần được xem xét. Một trong những hạn chế chính của khóa cửa truyền thống là tính bảo mật. Các khóa truyền thống thường dựa trên các chìa khóa vật lý hoặc các mã số và chúng có thể bị mất hoặc đánh cắp, dẫn đến nguy cơ truy cập trái phép. Ngoài ra, việc quản lý nhiều chìa khóa hoặc mã số cũng có thể trở nên phức tạp và dễ dẫn đến việc nhầm lẫn hoặc mất mật khẩu.

Trong khi đó, khóa cửa thông minh nhận diện khuôn mặt mang lại nhiều lợi ích vượt trội. Một lợi ích quan trọng của khóa cửa thông minh nhận diện khuôn mặt là tính an toàn cao hơn. Với công nghệ nhận diện khuôn mặt, khóa cửa thông minh có thể xác định và cho phép truy cập dựa trên thông tin khuôn mặt của người sử dụng. Khuôn mặt của mỗi người là duy nhất và khó sao chép lại, do đó tăng cường tính bảo mật và ngăn chặn truy cập trái phép.

Hơn nữa, khóa cửa thông minh nhận diện khuôn mặt còn mang lại tính tiện ích cao. Người dùng không cần mang theo chìa khóa hoặc nhớ mã số mỗi khi muốn truy cập. Chỉ cần đưa khuôn mặt vào phạm vi nhận dạng, hệ thống sẽ tự động mở cửa. Điều này tiết kiệm thời gian và thuận tiện cho người dùng, đặc biệt là trong các tình huống khẩn cấp hoặc khi đang mang nhiều đồ vật.

Trong đồ án này, chúng tôi xây dựng một hệ thống khóa cửa thông minh nhận diện khuôn mặt sử dụng thư viện OpenCV [3] và mô hình nhận diện đối tượng dựa trên deep learning (học sâu) với hiệu suất cao, giúp chúng tôi xác định và nhận dạng khuôn mặt trong thời gian thực. OpenCV cung cấp các công cụ và thuật toán mạnh mẽ để xử lý, phân tích và nhận diện hình ảnh. Từ mô hình học sâu được huấn luyện dựa trên các đối tượng trong hình ảnh và video, chúng tôi có thể nhận diện được khuôn mặt kể cả đang đeo khẩu trang.



Hình 1: Hình ảnh minh họa khóa cửa thông minh và khóa cửa truyền thống

## 1.2 Mục tiêu:

Đồ án nghiên cứu và xây dựng một hệ thống khóa cửa thông minh nhận diện khuôn mặt nhằm mang lại sự tiện nghi cho con người trong cuộc sống hiện đại ngày nay cũng như khắc phục các hạn chế của khóa cửa truyền thống.

- Đầu tiên, chúng tôi xây dựng hệ thống khóa cửa thông minh nhận diện khuôn mặt với OpenCV trên Raspberry Pi. Cụ thể đầu vào là một ảnh và hình ảnh thu được trực tiếp từ Camera.
- Xây dựng một mô hình tự huấn luyện nhận diện đối tượng sử dụng YOLOv5 dựa trên deep learning nhằm cung cấp khả năng phát hiện đa đối tượng trong một hình ảnh khuôn mặt đạt hiệu suất cao hơn và độ chính xác tốt hơn.
- Xây dựng trang web giúp quản lý dữ liệu người dùng sử dụng Nextjs (Front-end) [2] và Firebase (Back-end) [1].

## 2 Cơ sở lý thuyết:

### 2.1 Tổng quan về Raspberry Pi:

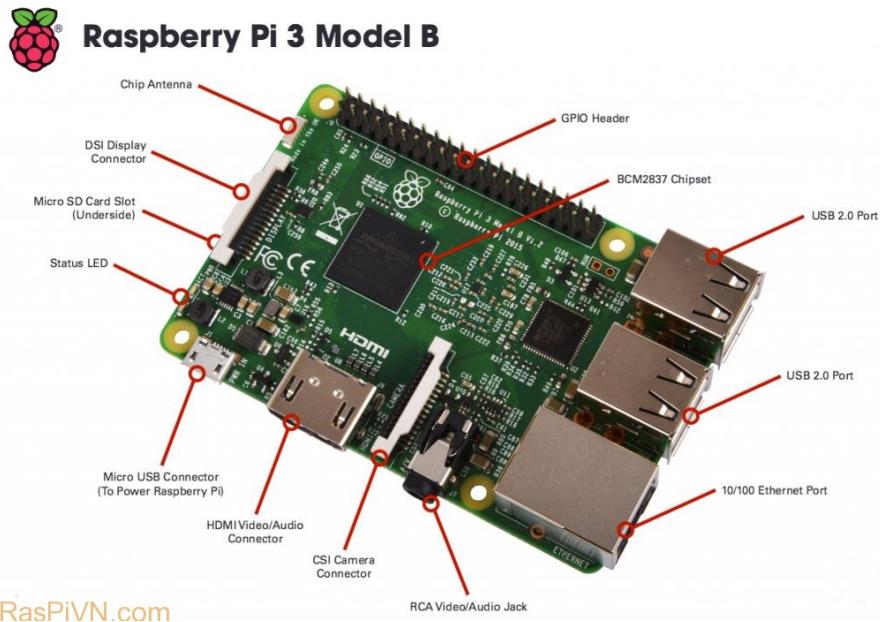
#### 2.1.1 Giới thiệu về Board Raspberry Pi:

Raspberry Pi (RPi) là một máy tính nhỏ gọn, có giá chỉ từ 35\$, và được ra mắt vào tháng 2 năm 2012 với hệ điều hành Linux. Ý tưởng ban đầu của RPi được phát triển bởi tiến sĩ Eben Upton tại đại học Cambridge, với mục tiêu tạo ra một máy tính giá rẻ để học sinh có thể tiếp cận và khám phá lĩnh vực tin học một cách dễ dàng. Ban đầu, ông Upton chỉ hy vọng bán được khoảng 1000 bo mạch cho các trường học. Tuy nhiên, thành công của RPi đã vượt xa sự mong đợi khi hơn một triệu bo mạch đã được bán ra chỉ trong một năm.



Hình 2: Board Raspberry Pi.

RPi có kích thước siêu nhỏ, chỉ như một chiếc thẻ ATM. Để sử dụng RPi, bạn chỉ cần chuẩn bị một bàn phím, một màn hình có cổng HDMI/DVI, một nguồn cấp 5V qua cổng USB, và một dây cáp micro USB. Với những phụ kiện này, bạn đã có thể sử dụng RPi như một máy tính thông thường. RPi cung cấp khả năng sử dụng các ứng dụng văn phòng, nghe nhạc và xem phim với độ nét cao lên tới 1024p.



Hình 3: Những thông số sơ lược về RPi B v3.

### 2.1.2 Cấu hình của Raspberry Pi:

RPi được trang bị vi xử lý Broadcom BCM2835 chạy ở tốc độ 700mHz. Đây là một vi xử lý SoC (system-on-chip), tức là hầu hết các thành phần của hệ thống như CPU, GPU, âm thanh và chip giao tiếp được tích hợp trong một chip duy nhất. Chip SoC này nằm bên dưới chip bộ nhớ Hynix 512 MB màu đen giữa bo mạch.

Tùy thuộc vào phiên bản, RPi sử dụng các chip khác nhau. Ví dụ, RPi Model B v1 sử dụng chip Broadcom BCM2835 với 256MB RAM, trong khi Model B v2 và Model A sử dụng chip Broadcom BCM2835 với 512MB RAM và 256MB RAM tương ứng. Vì RAM đã tích hợp sẵn trong chip, nên không thể nâng cấp RAM cho RPi.

CPU BMC2835 của RPi sử dụng nhân ARM1176JZFS (ARM11) cho hiệu năng cao và giá thành thấp. Khác với CPU trong máy tính PC thông thường sử dụng kiến trúc x86 của Intel, SoC của RPi sử dụng kiến trúc ARM, có ISA (Instruction Set Architecture) dạng rút gọn RISC và tiêu thụ điện năng rất thấp, phù hợp với các thiết bị di động.

ARM là một trong những nhà lãnh đạo trong lĩnh vực thiết bị di động. Ví dụ, chip ARM trên RPi hoạt động với nguồn 5V và chỉ tiêu hao khoảng 3.5W mỗi giờ, trong khi một laptop tiêu thụ ít nhất vài chục Watt. Thiết kế này giúp RPi hoạt động với hiệu suất tương đối cao trong kích thước nhỏ gọn mà không cần quạt tản nhiệt. Do đó, kiến trúc ARM được sử dụng rộng rãi trong điện thoại di động hiện nay.

Chip BCM2835 trên RPi thuộc thế hệ ARM11 của ARMv6. Mặc dù ARMv6 hoạt động hiệu quả và tiết kiệm năng lượng, nhưng không tương thích phần mềm với thế hệ ARMv7. Tuy nhiên, điều này có thể được giải quyết bằng cách chuyển đổi phần mềm để đạt được sự tương thích. RPi có cộng đồng phát triển mạnh mẽ trên toàn cầu, cung cấp nhiều dự án và phần mềm phong phú, giúp người dùng tận dụng tối đa tiềm năng của RPi.

Để sử dụng RPi, bạn cần cài đặt một hệ điều hành. Trong thế giới Linux mã nguồn mở, có nhiều phiên bản hệ điều hành tùy biến (distro) khác nhau để lựa chọn. Tùy thuộc vào nhu cầu và mục đích sử dụng, bạn có thể chọn distro phù hợp với khả năng học hỏi và sở thích của mình.

RPi có nhiều ứng dụng từ đơn giản đến phức tạp. Ví dụ, nó có thể được sử dụng làm trung tâm giải trí đa phương tiện, internet TV, ổ đĩa sao lưu dữ phòng trên mạng nội bộ, hệ thống phát hiện chuyển động và nhận diện khuôn mặt kết hợp với webcam, điều khiển robot, nhận và gửi tin nhắn GSM với USB 3G, điều khiển đèn trong nhà và còn rất nhiều ứng dụng khác.

### 2.1.3 Kết nối của Raspberry Pi:

**Khe cắm thẻ nhớ SD:** RPi sử dụng thẻ nhớ SD để lưu trữ hệ điều hành và dữ liệu hoạt động. Để RPi hoạt động tốt, cần sử dụng thẻ nhớ có dung lượng ít nhất 2GB.

**Cổng Micro USB Power:** RPi có thể sử dụng đến 700mA dòng điện ở mức 5V khi kết nối nhiều thiết bị USB và cổng LAN. Để đảm bảo RPi hoạt động ổn định, nên sử dụng bộ nguồn USB 5V 1A. Nguồn có thể được cấp qua cổng Micro USB hoặc cấp nguồn trực tiếp vào cổng GPIO (5V và GND). Tuy nhiên, việc cấp nguồn trực tiếp vào GPIO cần kiến thức về điện tử để tránh gây hỏng thiết bị.

**Màn hình cảm ứng TFT:** Kết nối màn hình cảm ứng TFT giúp hiển thị và tương tác với RPi một cách trực quan. Màn hình cảm ứng TFT cho phép thực hiện các thao tác tương tự như sử dụng chuột và bàn phím.

**Khe mở rộng camera:** Khe cắm này cho phép kết nối một camera module vào RPi. Người dùng có thể sử dụng camera để chụp ảnh, quay video và thực hiện các tác vụ khác như trên một camera thông thường.

**Cổng HDMI:** Cổng HDMI được sử dụng để kết nối RPi với các thiết bị hỗ trợ HDMI, như màn hình, TV hoặc máy chiếu. Nếu thiết bị chỉ hỗ trợ VGA, bạn cần sử dụng một cáp chuyển đổi từ HDMI sang VGA.

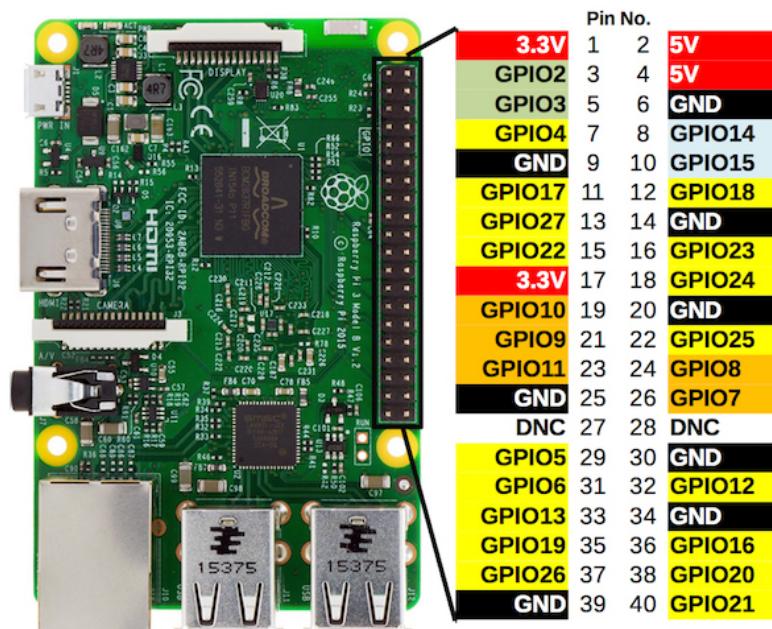
**Cổng Ethernet:** Cổng Ethernet được sử dụng để kết nối RPi với mạng LAN hoặc truy cập internet. Nó cho phép RPi truyền dữ liệu qua mạng và kết nối đến các thiết bị khác thông qua giao thức SSH.

**2 cổng USB:** Các cổng USB được sử dụng để kết nối các thiết bị như chuột, bàn phím, USB 3G, USB Wi-Fi và các thiết bị hỗ trợ USB khác.

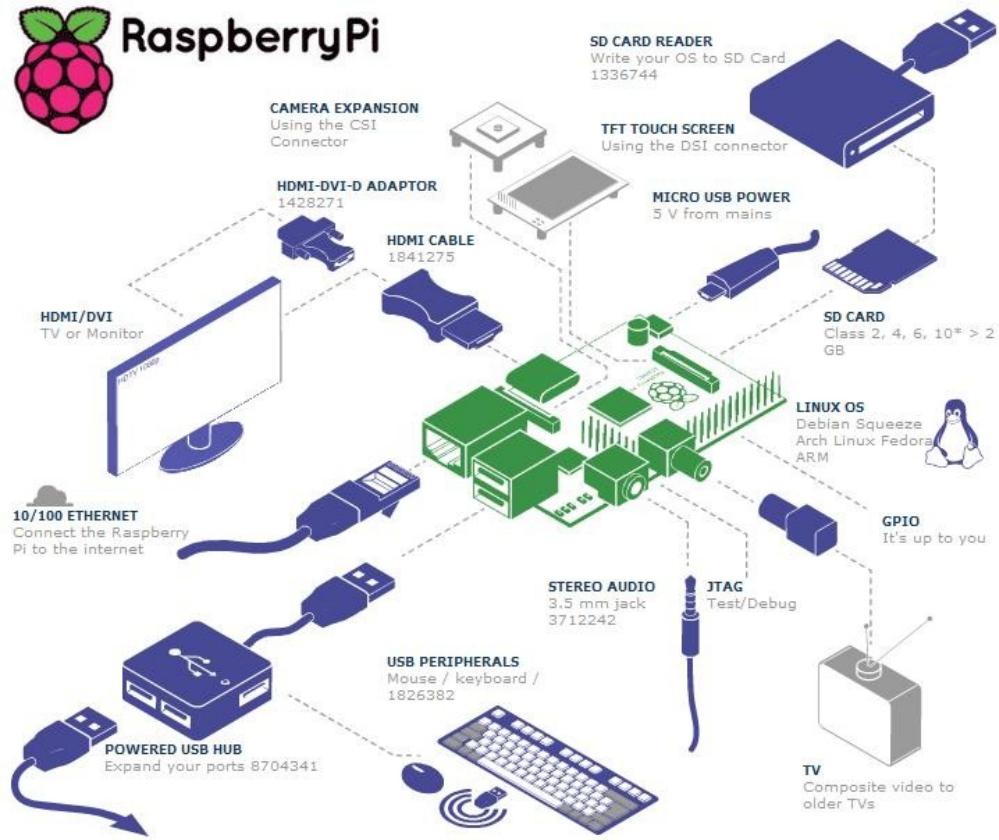
**Stereo Audio:** Cổng này được sử dụng để kết nối tai nghe, loa và các thiết bị âm thanh khác thông qua jack 3.5mm.

**Cổng TV:** Cổng này dùng để kết nối RPi với TV để phát tín hiệu hình ảnh lên màn hình lớn.

**GPIO:** RPi cung cấp nhiều cổng GPIO (General Purpose Input/Output) cho phép giao tiếp với các thiết bị ngoại vi bên ngoài. Các cổng GPIO có thể được sử dụng để gửi và nhận tín hiệu số (giá trị 0/1). Ngoài ra, RPi cũng hỗ trợ các giao tiếp như SPI, I2C và Serial, cho phép kết nối với các vi điều khiển và thiết bị khác.



Hình 4: Tìm hiểu về các chân GPIO của RPi.



Hình 5: RPi và các thiết bị kết nối với nó.

#### 2.1.4 Giới thiệu hệ điều hành cho Raspberry Pi:

RPi khác với vi điều khiển thông thường vì nó cần hệ điều hành để hoạt động. Điều này cung cấp một ưu điểm cho Raspberry Pi, cho phép người dùng tận dụng nhiều phần mềm và thiết bị ngoại vi để phát triển các ứng dụng phức tạp một cách nhanh chóng.

Dưới đây là các hệ điều hành hiện tại được hỗ trợ trên RPi:

- NOOBS (New Out Of Box Software): Đây là hệ điều hành cơ bản nhất cho RPi, không có giao diện đồ họa, chỉ sử dụng giao diện dòng lệnh.
- Raspbian: Được phát triển từ hệ điều hành Debian của Linux, Raspbian cung cấp giao diện đồ họa tương tự một máy tính thông thường. Nó hỗ trợ mạng và giao tiếp vào/ra.
- Pidora: Pidora được phát triển từ phiên bản Fedora của Linux. Nó cũng cung cấp giao diện đồ họa, hỗ trợ mạng và giao tiếp vào/ra.

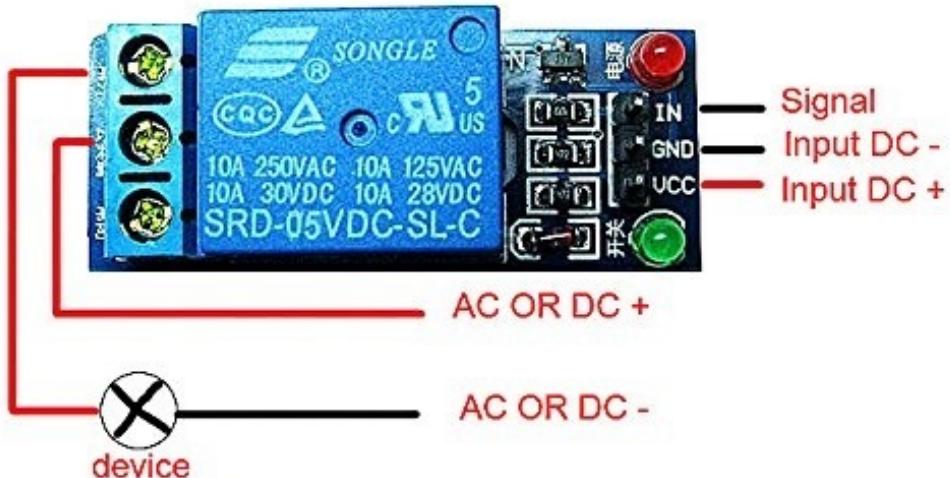
- RaspBMC: RaspBMC được thiết kế để sử dụng cho mục đích giải trí, với giao diện XBMC media center. Giao diện này thường được thấy trên các TV thông minh hiện nay.
- OpenELEC: Đây là một phiên bản khác của XBMC Media Center, nhằm cung cấp trải nghiệm nhanh và thân thiện cho người dùng.
- RISC OS: RISC OS là một phiên bản hệ điều hành rút gọn, được phát triển với mục tiêu hoạt động nhanh nhất trong các hệ điều hành trên RPi.
- Arch: Arch Linux là một phiên bản hệ điều hành phát triển đặc biệt cho các thiết bị sử dụng chip ARM.

Đây là một số hệ điều hành phổ biến mà RPi hỗ trợ. Mỗi hệ điều hành có các đặc điểm và tính năng riêng, cho phép người dùng lựa chọn theo mục đích sử dụng của họ.

## 2.2 Giới thiệu về Relay

Relay là một loại thiết bị điện tự động mà tín hiệu đầu ra thay đổi nhảy cấp khi tín hiệu đầu vào đạt những giá trị xác định. Relay là thiết bị điện dùng để đóng/cắt mạch điện điều khiển, bảo vệ và điều khiển sự làm việc của mạch điện động lực.

## Normal Open

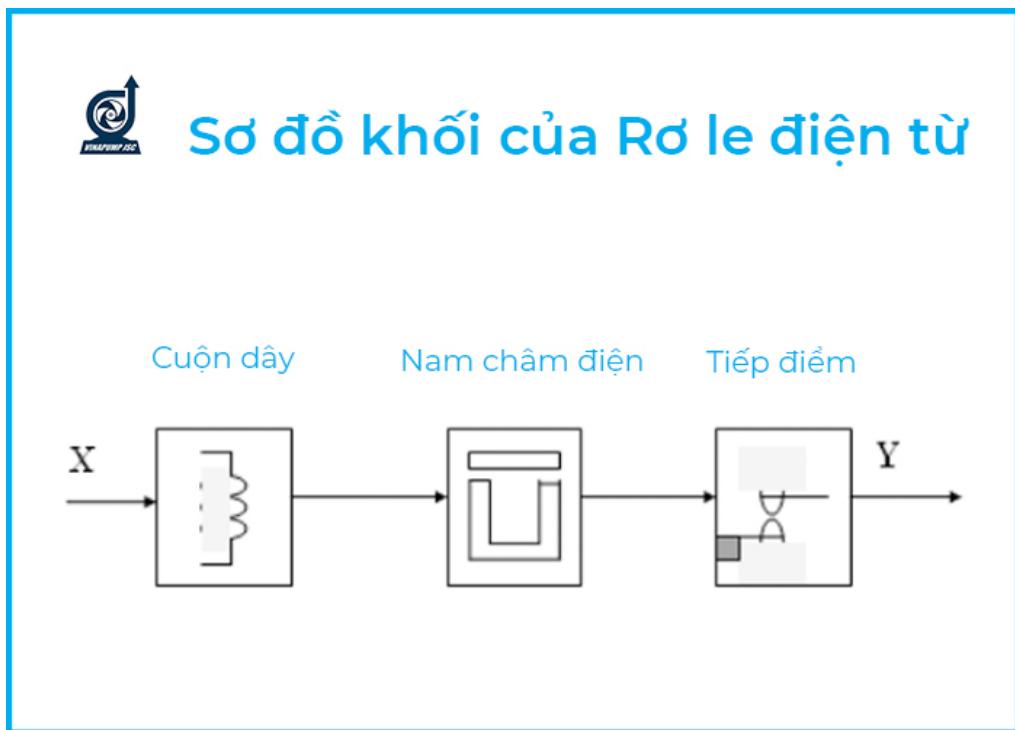


## Normal Close



Hình 6: Module Relay.

Các bộ phận chính của relay + Cơ cấu tiếp thu (khối tiếp thu): Có nhiệm vụ tiếp nhận những tín hiệu đầu vào và biến đổi nó thành đại lượng cần thiết cung cấp tín hiệu phù hợp cho khối trung gian. + Cơ cấu trung gian (khối trung gian): Làm nhiệm vụ tiếp nhận những tín hiệu đưa đến từ khối tiếp thu và biến đổi nó thành đại lượng cần thiết cho role tác động. + Cơ cấu chấp hành (khối chấp hành): Làm nhiệm vụ phát tín hiệu cho mạch điều khiển.



Hình 7: Sơ đồ khối của relay điện từ.

Cơ cấu tiếp thu ở đây là cuộn dây và cơ cấu trung gian là mạch từ nam châm điện còn lại là cơ cấu chấp hành là hệ thống tiếp điểm.

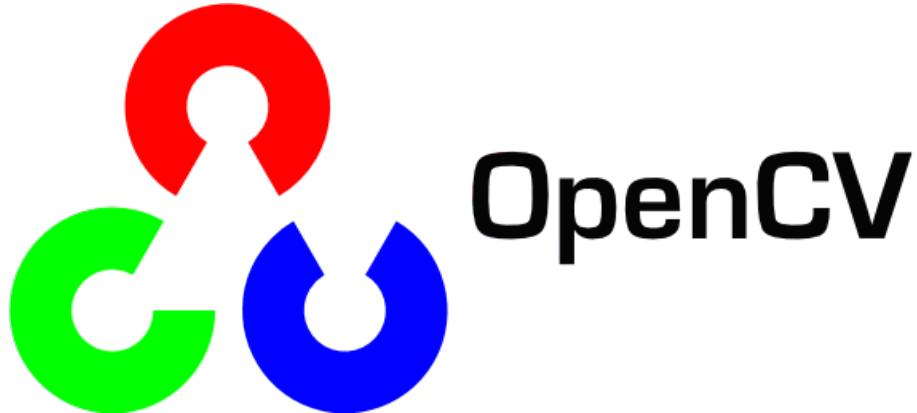
### 2.3 Tổng quan về thư viện OpenCV:

Dự án OpenCV được bắt đầu bởi Gary Bradsky của Intel vào năm 1999. OpenCV là viết tắt của Open Source Computer Vision Library, tức là Thư viện Mã nguồn mở cho Máy tính và Thị giác Máy. OpenCV là một thư viện hàng đầu trong lĩnh vực Thị giác máy tính và Học máy, và hiện đã hỗ trợ thêm khả năng tăng tốc GPU cho các tác vụ thời gian thực.

OpenCV được phát hành theo giấy phép BSD (một loại giấy phép phần mềm tự do được sử dụng trong lĩnh vực phát triển phần mềm mã nguồn mở), cho phép miễn phí sử dụng và học tập, bao gồm cả mục đích thương mại. Nó cung cấp các giao diện cho C++, C, Python và Java, và hỗ trợ trên các hệ điều hành Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để tính toán hiệu quả và đặc biệt tập trung vào ứng dụng thời gian thực. Khi được viết bằng C/C++ tối ưu, thư viện này có thể tận dụng được khả năng xử lý đa lõi (multi-core processing).

OpenCV có một cộng đồng người dùng đông đảo và hoạt động rộng khắp trên toàn cầu. Điều này bởi vì nhu cầu sử dụng Thị giác máy tính ngày càng tăng, đặc biệt trong cuộc đua của các công ty công nghệ. Hiện nay, OpenCV được áp dụng rộng rãi trên toàn cầu, với một cộng đồng

hơn 47.000 thành viên, và được sử dụng cho nhiều mục đích và tính năng khác nhau, từ nghệ thuật tương tác, khai thác mỏ, bản đồ web đến robot cao cấp.



Hình 8: Logo của OpenCV

### 2.3.1 Ứng dụng của OpenCV:

OpenCV được sử dụng trong nhiều mục đích và ứng dụng khác nhau, bao gồm:

- Hình ảnh street view: OpenCV có thể được sử dụng để xử lý và phân tích hình ảnh thu thập từ các hệ thống street view, giúp xác định vị trí và hiển thị thông tin địa lý.
- Kiểm tra và giám sát tự động: OpenCV có khả năng nhận dạng và phân loại đối tượng, giúp trong việc kiểm tra tự động, giám sát và phân tích dữ liệu.
- Robot và xe hơi tự lái: OpenCV cung cấp các công cụ xử lý hình ảnh và nhận dạng đối tượng, giúp robot và xe tự lái nhận biết và tương tác với môi trường xung quanh.
- Phân tích hình ảnh y học: OpenCV được sử dụng trong lĩnh vực y học để phân tích và xử lý hình ảnh y học, như phát hiện và phân loại khối u, phân tích hình ảnh y khoa và hỗ trợ trong quá trình chẩn đoán.
- Tìm kiếm và phục hồi hình ảnh/video: OpenCV cung cấp các thuật toán và công cụ cho việc tìm kiếm và phục hồi hình ảnh và video, bao gồm việc nhận dạng khuôn mặt, phát hiện vật thể, theo dõi chuyển động và tái tạo hình ảnh/video.
- Phim - cấu trúc 3D từ chuyển động: OpenCV hỗ trợ trong việc xây dựng cấu trúc 3D từ dữ liệu chuyển động, giúp trong việc tạo ra hiệu ứng đồ họa và xử lý hình ảnh/video trong ngành công nghiệp phim.

- Nghệ thuật sắp đặt tương tác: OpenCV có thể được sử dụng để tạo ra các ứng dụng nghệ thuật tương tác, như điều khiển bằng cử chỉ, nhận diện khuôn mặt và tương tác với các đối tượng trong môi trường thực tế.

OpenCV có khả năng linh hoạt và mở rộng, cho phép ứng dụng trong nhiều lĩnh vực và mục đích khác nhau, từ công nghiệp đến nghệ thuật và y học.

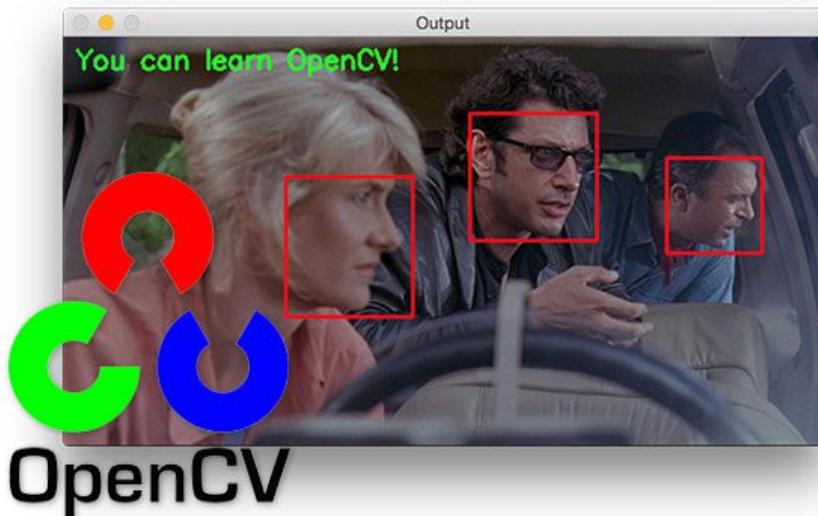
### **2.3.2 Tính năng và các module phổ biến của OpenCV:**

Dưới đây là danh sách các nhóm tính năng và module tương ứng trong OpenCV:

- Xử lý và hiển thị hình ảnh/video/I/O (core, imgproc, highgui): Nhóm này bao gồm các module cơ bản để xử lý, biến đổi và hiển thị hình ảnh và video. Core module chứa các cấu trúc dữ liệu cơ bản và chức năng chung. Imgproc module cung cấp các chức năng xử lý hình ảnh như lọc hình ảnh, biến đổi hình ảnh và chuyển đổi không gian màu. Highgui module cung cấp các chức năng giao diện người dùng đồ họa.
- Phát hiện các vật thể (objdetect, features2d, nonfree): Nhóm này chứa các module cho việc phát hiện và trích xuất đặc trưng của các vật thể trong hình ảnh. Objdetect module cung cấp các chức năng phát hiện các đối tượng được định nghĩa sẵn như khuôn mặt, mắt, con người, xe hơi, v.v. Features2d module cung cấp các thuật toán để phát hiện và trích xuất đặc trưng 2D trong hình ảnh. Nonfree module chứa các thuật toán trích xuất đặc trưng không miễn phí như SIFT và SURF.
- Geometry-based monocular hoặc stereo computer vision (calib3d, stitching, videostab): Nhóm này tập trung vào xử lý và phân tích hình ảnh dựa trên hình học đa chiều. Calib3d module cung cấp các thuật toán hiệu chuẩn máy ảnh đơn và stereo, ước tính kiểu dáng đối tượng và tái tạo 3D. Stitching module cho phép ghép các hình ảnh lại với nhau để tạo thành một hình ảnh toàn cảnh. Videostab module cung cấp các thuật toán ổn định hình ảnh/video để giảm rung và giữ ổn định hình ảnh khi quay video.
- Computational photography (photo, video, superres): Nhóm này tập trung vào các thuật toán và công nghệ xử lý hình ảnh để tạo ra các hiệu ứng và kỹ thuật nhiếp ảnh tính toán. Photo module cung cấp các chức năng như xử lý đồ họa, làm mịn hình ảnh, giảm nhiễu và nâng cao chất lượng hình ảnh. Video module chứa các thuật toán xử lý video như theo dõi chuyển động và trích xuất khung hình. Superres module cung cấp các thuật toán tăng cường độ phân giải hình ảnh/video.

- Machine learning & clustering (ml, flann): Nhóm này liên quan đến machine learning và các thuật toán phân cụm. ML module cung cấp các công cụ và thuật toán cho machine learning trong OpenCV. Flann module cung cấp một thư viện tìm kiếm nhanh để phục vụ cho các thuật toán phân cụm.
- CUDA acceleration (gpu): Nhóm này cung cấp các chức năng và thuật toán được tăng tốc bằng sử dụng GPU thông qua CUDA (Compute Unified Device Architecture). Các module khác của OpenCV có thể sử dụng GPU để tăng tốc xử lý hình ảnh và video.
- Ngoài ra, OpenCV còn có các module hỗ trợ khác như FLANN (Fast Library for Approximate Nearest Neighbors) và Google Test Wrapper (gtest) để thực hiện các kiểm thử và đánh giá chất lượng.

OpenCV cung cấp một cấu trúc module giúp tổ chức các tính năng và chức năng của thư viện một cách có tổ chức và dễ quản lý, cho phép người dùng chọn các module cần thiết để sử dụng trong ứng dụng của mình.



Hình 9: Hình ảnh minh họa của việc ứng dụng OpenCV vào nhận diện khuôn mặt.

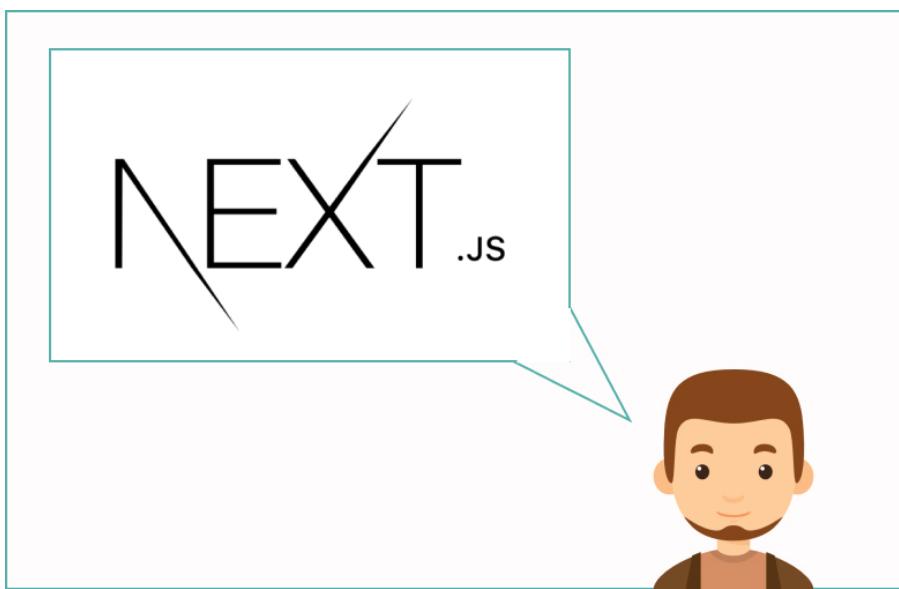
## 2.4 Tổng quan về Nextjs

### 2.4.1 Nextjs là gì?

Nextjs là một framework front-end dựa trên React, được phát triển như một dự án mã nguồn mở. Nó cung cấp những khả năng tối ưu hóa bổ sung như render phía máy chủ (SSR - Server-Side Rendering) và tạo trang web tĩnh (Static Site Generation). Next.js xây dựng trên nền tảng của

thư viện React, do đó ứng dụng Next.js sử dụng cốt lõi của React và bổ sung các tính năng mở rộng.

Việc triển khai SSR cho phép máy chủ xử lý yêu cầu và render toàn bộ nội dung trang web bằng JavaScript trước khi gửi về trình duyệt. Điều này giúp tăng tốc độ tải trang và cải thiện trải nghiệm người dùng với khả năng phản hồi nhanh hơn. Bên cạnh đó, việc sử dụng SSR cũng mang lại lợi ích cho việc tối ưu hóa SEO, giúp trang web xếp hạng cao hơn trên các công cụ tìm kiếm. SSR giúp tăng tốc độ tải trang và cho phép SEO trackers quét nhiều nội dung trang web hơn. Next.js cũng cung cấp khả năng chỉnh sửa thẻ <head> của một trang web, giúp tùy chỉnh metadata của trang và cải thiện xếp hạng SEO.



Hình 10: Hình ảnh minh họa về framework Nextjs.

#### 2.4.2 Tính năng chính của Nextjs:

Next.js cung cấp một loạt tính năng mạnh mẽ để phát triển ứng dụng web. Dưới đây là một số tính năng chính của Next.js:

- Render phía máy chủ (Server-Side Rendering - SSR): Next.js cho phép bạn xây dựng ứng dụng web với khả năng SSR, tức là máy chủ có thể render trang và gửi nội dung đã render cho trình duyệt. Điều này giúp tăng tốc độ tải trang, cải thiện trải nghiệm người dùng và hỗ trợ SEO tốt hơn.
- Tạo trang web tĩnh (Static Site Generation - SSG): Next.js hỗ trợ tạo trang web tĩnh, cho phép bạn tiền rendering trước nội dung của trang và lưu trữ chúng dưới dạng các file HTML

tĩnh. Điều này giúp tăng tốc độ tải trang và phù hợp cho các trang web không yêu cầu dữ liệu thay đổi thường xuyên.

- Hot Module Replacement (HMR): Next.js hỗ trợ HMR, cho phép bạn chỉnh sửa mã nguồn trong quá trình phát triển và ngay lập tức xem kết quả mà không cần tải lại trang hoàn toàn. Điều này giúp tăng hiệu suất phát triển và tiết kiệm thời gian.
- Routing tự động: Next.js có hệ thống routing tự động, giúp bạn xây dựng các đường dẫn và tạo ra các trang động dễ dàng. Bạn không cần cấu hình lại routing một cách tường minh, mà Next.js sẽ tự động xác định các đường dẫn dựa trên cấu trúc thư mục của dự án.
- Pre-fetching dữ liệu: Next.js cho phép bạn pre-fetch dữ liệu cần thiết cho mỗi trang trước khi render. Điều này giúp đảm bảo rằng dữ liệu đã sẵn sàng khi trang được hiển thị, giúp cải thiện hiệu suất và trải nghiệm người dùng.
- Hỗ trợ TypeScript: Next.js tích hợp tốt với TypeScript, cho phép bạn phát triển ứng dụng sử dụng TypeScript để có kiểm tra kiểu dữ liệu mạnh mẽ và mã nguồn dễ bảo trì.
- Tích hợp CSS Modules và CSS-in-JS: Next.js hỗ trợ CSS Modules và các thư viện CSS-in-JS như Styled Components và Emotion. Điều này giúp tạo ra các phạm vi CSS tương tự trong từng thành phần và giữ cho CSS của bạn được tách biệt và dễ quản lý.

#### 2.4.3 Ưu và nhược điểm của Nextjs

##### Ưu điểm:

- Mạng lại khả năng SEO tốt.
- Trải nghiệm người dùng tốt hơn.
- Làm việc với cơ chế SSG (Static Site Generation) , SSR (Server Side Rendering) và cả CSR (Client Side Rendering).
- Khởi tạo nhanh chóng.
- Hỗ trợ nền React cực kì tốt.
- Hỗ trợ cấu trúc và tổ hợp một cách tối ưu.
- Hỗ trợ phát triển tính năng nhanh chóng cho việc cấu hình như: Webpack, Babel,...
- Bảo mật về dữ liệu.

- Khả năng thích ứng và đáp ứng thay đổi.

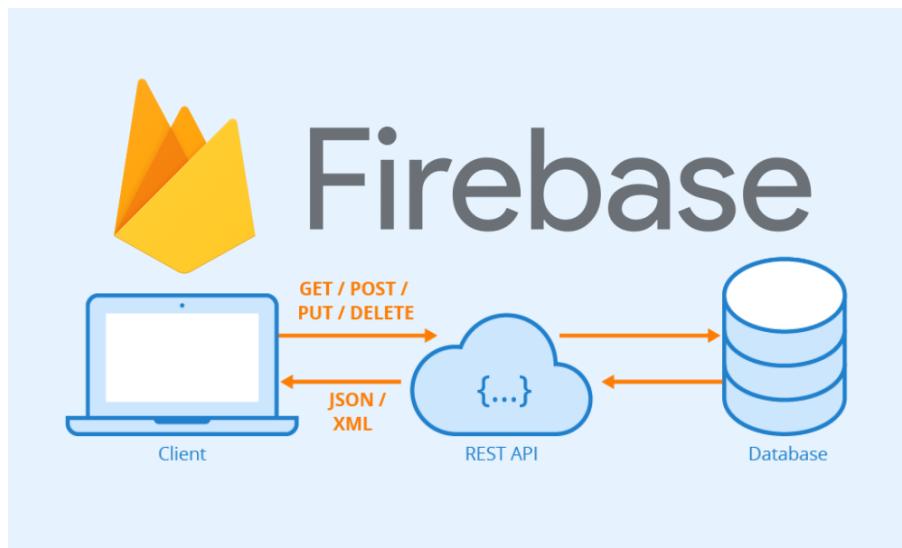
#### Nhược điểm:

- Ít plug-in thích ứng.
- Nextjs bị giới hạn bởi việc chỉ sử dụng bộ định tuyến trên tệp của nó, bạn không thể nào sửa đổi cách nó giao dịch với các tuyến. Vì vậy, để sử dụng tuyến động, bạn cần làm việc thêm với Node.js máy chủ.
- Nextjs không cung cấp nhiều trang nhất tích hợp, để làm việc bạn cần phải tạo toàn bộ front-end từ đầu lên.

## 2.5 Firebase

### 2.5.1 Firebase là gì?

Firebase là một nền tảng phát triển ứng dụng di động và web đám mây do Google phát triển. Nó cung cấp một tập hợp các dịch vụ đám mây được tích hợp để giúp phát triển, triển khai và quản lý ứng dụng một cách dễ dàng và hiệu quả.



Hình 11: Hình ảnh minh họa về Firebase.

Firebase được thành lập vào năm 2011 bởi James Tamplin và Andrew Lee. Ban đầu, Firebase được biết đến dưới tên "Envolve" và tập trung vào việc cung cấp dịch vụ chat thời gian thực cho các ứng dụng web. Tuy nhiên, sau khi nhận được sự quan tâm của các nhà phát triển và khách hàng, Firebase đã mở rộng phạm vi của mình để trở thành một nền tảng phát triển ứng dụng di động toàn diện.

Vào năm 2014, Firebase đã chính thức ra mắt và tạo ra sự chú ý lớn trong cộng đồng phát triển ứng dụng di động. Firebase được công nhận với khả năng cung cấp các dịch vụ đám mây mạnh mẽ và dễ sử dụng cho các nhà phát triển, như cơ sở dữ liệu thời gian thực, xác thực người dùng, lưu trữ, và các công cụ phân tích.

Sau đó, vào năm 2017, Google đã mua lại Firebase và nâng cấp nền tảng này thành một phần của hệ sinh thái Google Cloud. Việc hợp nhất với Google đã mang lại nhiều lợi ích cho Firebase, bao gồm sự tích hợp với các dịch vụ và công cụ của Google, quy mô lớn hơn và sự ổn định.

Từ đó, Firebase đã tiếp tục phát triển và mở rộng các dịch vụ của mình, bao gồm cải tiến cơ sở dữ liệu thời gian thực, tích hợp xác thực người dùng, phân tích, lưu trữ tệp tin, gửi thông báo đẩy, và nhiều tính năng khác. Firebase ngày nay là một trong những nền tảng phát triển ứng dụng di động hàng đầu, hỗ trợ hàng triệu nhà phát triển trên toàn thế giới.

### 2.5.2 Cách thức hoạt động của Firebase

Firebase hoạt động dựa trên mô hình đám mây (cloud) và cung cấp một loạt các dịch vụ backend và công cụ phát triển ứng dụng. Dưới đây là cách thức hoạt động của Firebase:

**Đăng ký và cấu hình dự án:** Bạn bắt đầu bằng việc đăng ký tài khoản Firebase và tạo một dự án mới. Sau đó, bạn cấu hình dự án của mình bằng cách thêm các thông tin cần thiết như tên dự án, giao diện người dùng, cấu hình xác thực, và các thiết lập khác.

**Tích hợp SDK:** Firebase cung cấp các SDK cho các nền tảng phát triển phổ biến như Android, iOS, JavaScript, và nhiều nền tảng khác. Bạn tích hợp SDK tương ứng vào ứng dụng của mình để sử dụng các dịch vụ Firebase.

Sử dụng các dịch vụ Firebase: Firebase cung cấp nhiều dịch vụ khác nhau, bao gồm:

- Cơ sở dữ liệu thời gian thực (Realtime Database): Đây là một cơ sở dữ liệu NoSQL dựa trên JSON, cho phép bạn lưu trữ và đồng bộ dữ liệu thời gian thực giữa các thiết bị và người dùng.
- Lưu trữ (Storage): Firebase cung cấp một hệ thống lưu trữ đám mây để lưu trữ và quản lý các tệp tin như hình ảnh, video, tài liệu, và nhiều loại tệp tin khác.
- Xác thực (Authentication): Firebase cung cấp các công cụ để xác thực người dùng, bao gồm đăng ký, đăng nhập, và quản lý thông tin người dùng.
- Phân tích (Analytics): Firebase cung cấp các công cụ phân tích hiệu suất ứng dụng và hành vi người dùng, giúp bạn hiểu rõ hơn về việc sử dụng ứng dụng và tối ưu hóa trải nghiệm người dùng.

- Cloud Messaging: Firebase cho phép bạn gửi thông báo đẩy tới các thiết bị di động để tương tác và thông báo cho người dùng.
- và nhiều dịch vụ khác như Hosting, Cloud Functions, Test Lab, và Crashlytics.

**Quản lý và giám sát:** Firebase cung cấp một giao diện quản lý để bạn có thể quản lý dự án, theo dõi hoạt động và hiệu suất của ứng dụng, và xem báo cáo phân tích.

**Tích hợp với các công cụ và dịch vụ khác:** Firebase cho phép tích hợp với các công cụ và dịch vụ khác của Google và bên thứ ba, như Google Cloud Platform, Google Analytics, và các dịch vụ lưu trữ đám mây khác.

### 2.5.3 Ưu và nhược điểm của Firebase

#### Ưu điểm:

- Dễ sử dụng và triển khai: Firebase cung cấp một giao diện người dùng thân thiện và các công cụ phát triển dễ sử dụng, giúp người dùng nhanh chóng triển khai và quản lý ứng dụng.
- Tích hợp dịch vụ đa dạng: Firebase cung cấp một loạt các dịch vụ backend như cơ sở dữ liệu thời gian thực, lưu trữ, xác thực, phân tích, thông báo đẩy và nhiều dịch vụ khác. Điều này giúp giảm công sức tích hợp và quản lý các dịch vụ từ các nhà cung cấp khác nhau.
- Tích hợp tốt với các nền tảng phát triển: Firebase cung cấp SDK cho các nền tảng phổ biến như Android, iOS, JavaScript và nhiều nền tảng khác, giúp việc tích hợp và phát triển ứng dụng trở nên dễ dàng.
- Hỗ trợ phân tích và tối ưu hóa: Firebase cung cấp công cụ phân tích và giám sát mạnh mẽ, giúp bạn theo dõi hiệu suất ứng dụng, hành vi người dùng và tối ưu hóa trải nghiệm người dùng.
- Độ tin cậy và khả năng mở rộng: Firebase được xây dựng trên cơ sở hạ tầng đám mây của Google, đảm bảo độ tin cậy cao và khả năng mở rộng linh hoạt cho ứng dụng của bạn.

#### Nhược điểm:

- Giới hạn chức năng: Mặc dù Firebase cung cấp nhiều dịch vụ phổ biến, nhưng nó có thể không đáp ứng được tất cả các yêu cầu đặc thù của mọi ứng dụng. Có một số hạn chế về chức năng và tùy chỉnh trong một số trường hợp.
- Chi phí: Dịch vụ Firebase có một mô hình giá cả, và việc sử dụng nhiều tính năng hoặc lưu lượng truy cập lớn có thể dẫn đến chi phí cao hơn. Việc đánh giá và quản lý chi phí là một yếu tố quan trọng cần xem xét.

- Phụ thuộc vào nền tảng Google: Firebase là một sản phẩm của Google, điều này có thể làm nảy sinh một số rủi ro liên quan đến sự phụ thuộc và ổn định của dịch vụ trong trường hợp Google thay đổi chiến lược hoặc chính sách của mình.
- Hạn chế địa lý: Một số tính năng của Firebase có thể bị hạn chế địa lý, đặc biệt là trong một số quốc gia hoặc vùng lãnh thổ nhất định.

Tuy nhiên, những ưu điểm của Firebase thường vượt trội hơn so với nhược điểm, và nó là một lựa chọn phổ biến cho việc xây dựng và quản lý ứng dụng di động và web hiệu quả.

### 3 Tính toán và thiết kế hệ thống

#### 3.1 Tính toán và thiết kế mạch

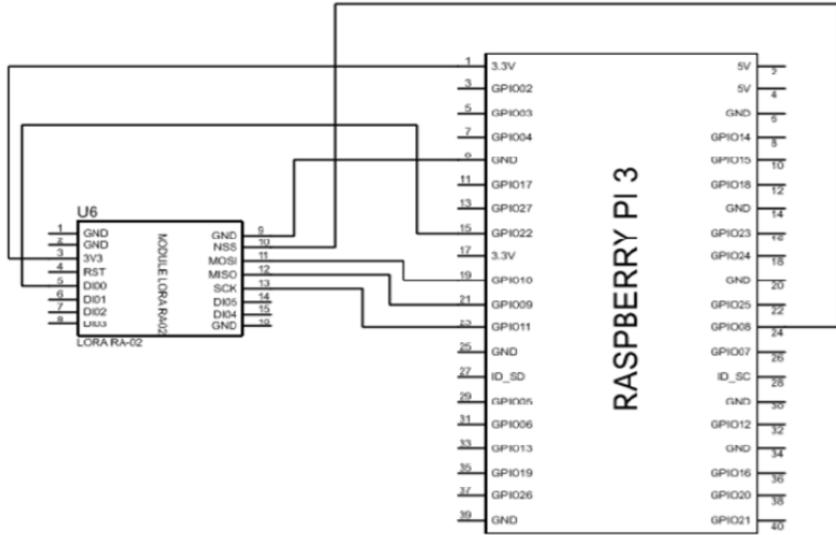
##### **Khối xử lý trung tâm:**

Khối xử lý trung tâm đóng vai trò trung tâm quan trọng của hệ thống, nơi tiếp nhận và xử lý tất cả các tín hiệu đến từ cảm biến, các cơ cấu tác động, và tín hiệu điều khiển từ web. Nó cũng đảm nhiệm nhiệm vụ truyền nhận dữ liệu giữa web và phần cứng để thực hiện quá trình xử lý. Sau đó, các thông số đo được và xử lý được sẽ được hiển thị cho người dùng để theo dõi.

Gateway (Khối điều khiển trung tâm) là cổng liên lạc giúp chuyển đổi các giao thức giữa device và network.

*Thông số kỹ thuật của Raspberry:*

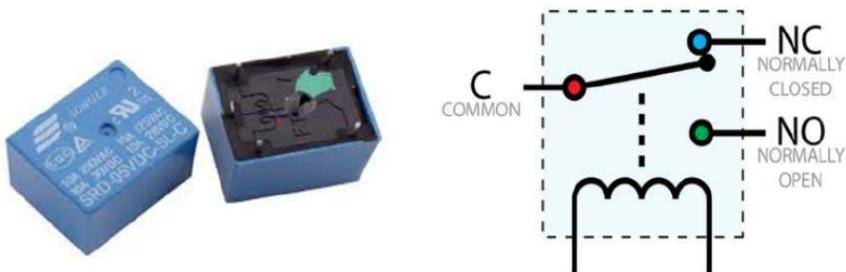
- Vi xử lý: Broadcom BCM2836, ARMv7 (32bit) quad core
- GPU: Broadcom VideoCore IV, OpenVG 1080p30, 250 MHz
- Tốc độ xử lý: 900 MHz
- Power Ratings: 800mA @5V
- RAM (chia sẻ với GPU): 1 GB SDRAM (400MHz)
- Bộ nhớ: Micro SD
- Kết nối: 4xUSB 2.0; 10/100mb Ethernet; CSI, DSI
- Mở rộng: 40-pin GPIO
- Video và âm thanh: 1080p HDMI, stereo audio 3.5mm jack
- Kích thước: 85x56mm



Hình 12: Sơ đồ nguyên lý Gateway.

### Relay:

Để điều khiển các thiết bị hoạt động ở mức điện áp 12V bằng tín hiệu điều khiển 5V từ khối xử lý trung tâm, chúng ta sử dụng một thiết bị trung gian gọi là Relay. Relay có khả năng đóng ngắt tiếp điểm và tải điện với điện áp 5V, giúp điều khiển các thiết bị 12V. Đặc biệt, Relay cũng có khả năng cách ly, bảo vệ khỏi xử lý trung tâm khỏi các sự cố như cháy nổ hoặc chập cháy. Với Relay, ta có thể đảm bảo sự an toàn và hiệu quả trong việc điều khiển các thiết bị 12V từ tín hiệu 5V của khối xử lý trung tâm.

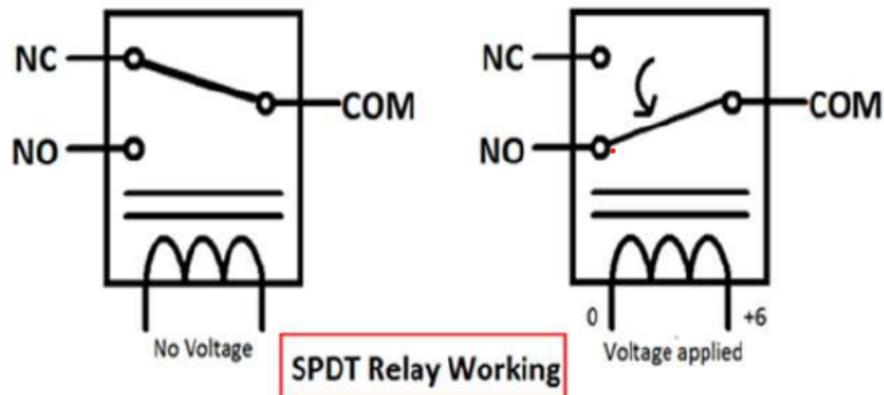


Hình 13: Hình ảnh thực tế Relay và cấu tạo bên trong của Relay.

Relay là một loại linh kiện điện tử thụ động rất hay gặp trong các ứng dụng thực tế khi gặp các vấn đề liên quan đến công suất và cần sự ổn định cao, ngoài ra có thể dễ dàng bảo trì.

Relay là một công tắc (khóa K). Nhưng khác với công tắc ở một chỗ cơ bản, Relay được kích hoạt bằng điện thay vì dùng tay người. Chính vì lẽ đó, Relay được dùng làm công tắc điện tử [4].

Vì relay là một công tắc nên nó có 2 trạng thái: đóng và mở.



Hình 14: Cơ cấu tác động của Relay.

Để sử dụng được Relay, ta phải cấp nguồn vào 2 chân + và - của cuộn dây của Relay, khi cuộn dây chưa có điện thì tiếp điểm của Relay ở vị trí NC, khi cuộn dây có điện, nó sẽ hút tiếp điểm của Relay từ vị trí NC sang vị trí NO, ta nối dây của thiết bị cần điều khiển vào 2 chân COM và NO để điều khiển đóng ngắt thiết bị đó.

Dưới đây bảng ?? là bảng sơ đồ kết nối chân giữa RPi và Relay, và bảng

Raspberry Pi 3	Relay
Chân số 2 (5V PWR)	VCC
Chân số 6 (GND)	GND
Chân số 16 (GPIO 23)	IN

Bảng 1: Bảng chân kết nối giữa RPi và Relay.

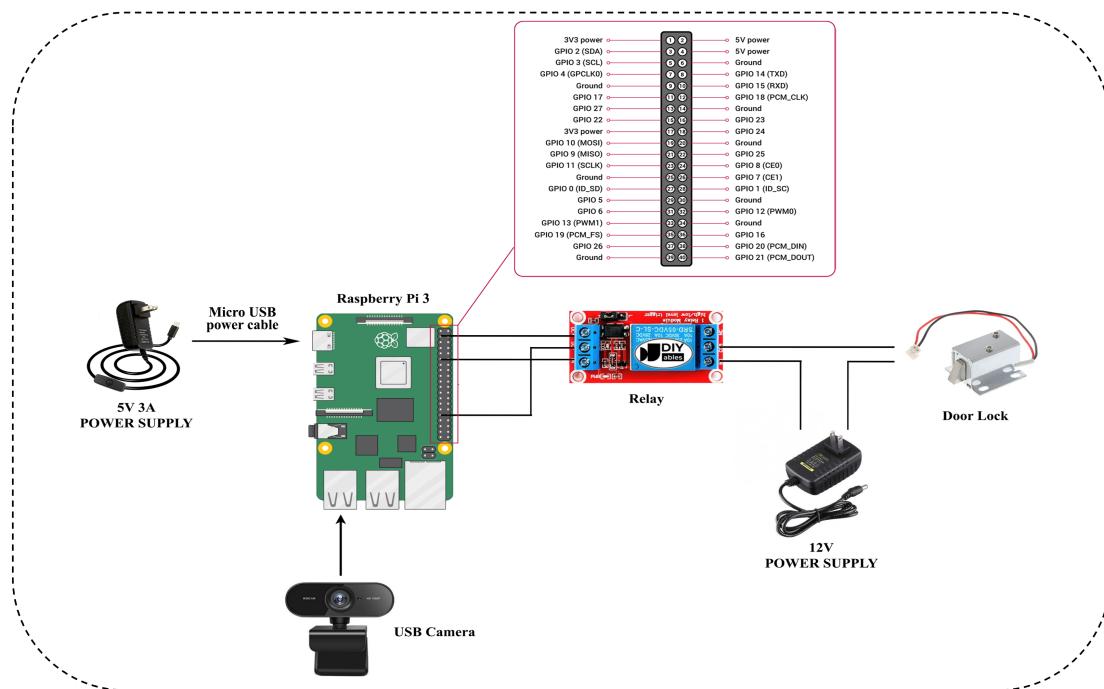
### 3.2 Mô hình đề xuất

Sau quá trình tính toán, lựa chọn thiết bị cho hợp lý thì tiếp theo chúng tôi sẽ tiến hành thiết kế và lắp ráp hệ thống. Dưới đây là danh sách toàn bộ các linh kiện mà chúng tôi sẽ dùng để xây dựng hệ thống.

STT	Tên linh kiện	Số lượng	Giá trị
1	Raspberry Pi 3	1	5V
2	Relay 5V 1 kênh	1	5V
3	Nguồn cho RPi	1	5V-3A
4	Nguồn cho lock và relay	1	12V
5	Khóa cửa	1	12V
6	Webcam USB	1	

Bảng 2: Danh sách linh kiện sử dụng.

Sau khi chuẩn bị đầy đủ các linh kiện, chúng tôi tiến hành triển khai lắp đặt theo mô hình đề xuất dưới đây (Hình 15).



Hình 15: Mô hình đề xuất.

## 4 Thực nghiệm và đánh giá

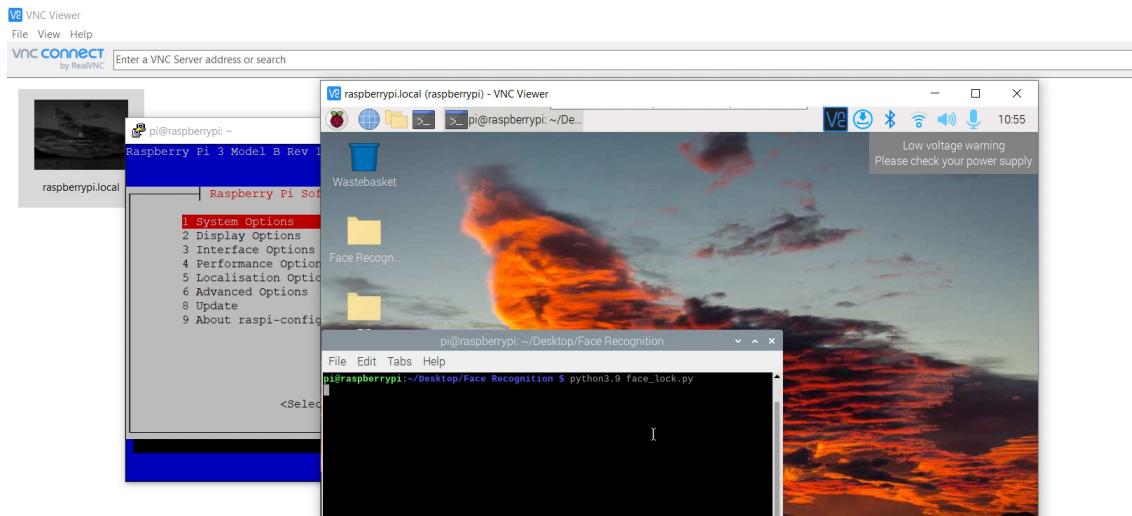
### 4.1 Môi trường và ngôn ngữ cài đặt

Để tiến hành thực nghiệm tôi sử dụng ngôn ngữ lập trình Python3 cùng với các thư viện: opencv, face\_recognition, imutils, tensorflow, numpy, pandas,...

- Chúng tôi sẽ sử dụng package face\_recognition của Python để tính toán bounding box xung quanh mỗi khuôn mặt, tính toán nhúng khuôn mặt và so sánh các khuôn mặt trong bộ dữ liệu mã hóa.
- Imutils là một loạt các chức năng tiện lợi để tăng tốc độ tính toán của OpenCV trên RPi.

Cấu hình RPi sử dụng để thực nghiệm trên Raspian OS 32 bit: 1GB SDRAM, Vi xử lý Broadcom BCM2836, ARMv7 (32bit) quad core và bộ nhớ Micro SD 16GB.

Dễ dàng triển khai cài đặt hệ thống khóa cửa nhận diện khuôn mặt vào RPi, chúng tôi kết nối giao diện RPi đến laptop thông qua mạng LAN sử dụng VNC viewer.



Hình 16: Kết nối giao diện RPi đến laptop sử dụng VNC viewer.

## 4.2 Dữ liệu

Bảng 3: Số lượng mẫu dữ liệu trong tập Train và tập Test.

Train		Test		Total
NoMask	Mask	NoMask	Mask	
1,544	1,732	386	433	4,095

## 4.3 Thang đo

Dể đánh giá được sự hiệu quả của quá trình huấn luyện bằng phương pháp học cộng tác, tôi đánh giá dựa trên 4 thang đo:

- **accuracy:** đây là tỉ số giữa số lượng mẫu phân lớp đúng  $n_{true}$  và tổng số mẫu  $n_{total}$ .

$$accuracy = \frac{n_{true}}{n_{total}} \quad (1)$$

- **loss:** Binary cross entropy là một số liệu mô hình theo dõi việc ghi nhãn lớp dữ liệu không chính xác theo một mô hình. Giá trị mất mát thấp tương đương với giá trị độ chính xác cao.

$$loss = -\frac{1}{N} \sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i)) \quad (2)$$

## 4.4 Nhận diện khuôn mặt sử dụng thư viện OpenCV

### 4.4.1 Huấn luyện mô hình nhận diện khuôn mặt cho RPi

Dầu tiên, chúng ta sẽ thực thi file code *face\_shot.py* (Hình 17) để tiến hành chụp ảnh chân dung của ta và sau đó dùng những hình ảnh này để huấn luyện cho mô hình. Hướng webcam vào mặt chúng ta và nhấn phím cách để chụp ảnh. Mỗi lần nhấn phím cách, chúng ta sẽ chụp một bức ảnh khác. Chúng tôi sẽ chụp khoảng 10 bức ảnh khuôn mặt của mình ở các góc độ khác nhau (hãy quay nhẹ đầu trong mỗi bức ảnh). Nếu đeo kính, chúng ta có thể chụp một vài bức ảnh khi đeo kính và không đeo kính. Nếu đội mũ thì không được khuyến khích cho hình ảnh huấn luyện. Nhấn Esc khi bạn đã hoàn tất việc chụp ảnh chính mình.

```

        face_shot.py

1 import cv2
2
3 name = 'Trung'
4
5 cam = cv2.VideoCapture(0)
6
7 cv2.namedWindow("press space to take a photo", cv2.WINDOW_NORMAL)
8 cv2.resizeWindow("press space to take a photo", 500, 300)
9
10 img_counter = 0
11
12 while True:
13     ret, frame = cam.read()
14     if not ret:
15         print("failed to grab frame")
16         break
17     cv2.imshow("press space to take a photo", frame)
18
19 k = cv2.waitKey(1)
20 if k%256 == 27:
21     # ESC pressed
22     print("Escape hit, closing...")
23     break
24 elif k%256 == 32:
25     # SPACE pressed
26     img_name = "dataset/" + name + "/image_{}.jpg".format(img_counter)
27     cv2.imwrite(img_name, frame)
28     print("{} written!".format(img_name))
29     img_counter += 1
30
31 cam.release()
32
33 cv2.destroyAllWindows()

```

Hình 17: Code thực hiện chụp ảnh chân dung để huấn luyện cho mô hình.

Mất khoảng 3-4 giây để RPi phân tích từng ảnh trong tập dữ liệu của chúng ta. Đối với tập dữ liệu có 10 ảnh, RPi sẽ mất khoảng gần 1 phút để phân tích ảnh và tạo file **encoding.pickle**. Tiếp theo, chúng ta sẽ thực thi file code *train\_model.py* (Hình 18) để huấn luyện cho mô hình bằng lệnh sau:

*python train\_model.py*

```

train_model.py

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

```

Hình 18: Code huấn luyện mô hình.

```
pi@raspberrypi:~ $ cd facial_recognition
pi@raspberrypi:~/facial_recognition $ python train_model.py
[INFO] start processing faces...
[INFO] processing image 1/11
[INFO] processing image 2/11
[INFO] processing image 3/11
[INFO] processing image 4/11
[INFO] processing image 5/11
[INFO] processing image 6/11
[INFO] processing image 7/11
[INFO] processing image 8/11
[INFO] processing image 9/11
[INFO] processing image 10/11
[INFO] processing image 11/11
[INFO] serializing encodings...
pi@raspberrypi:~/facial_recognition $
```

Hình 19: Quá trình huấn luyện cho mô hình.

#### 4.4.2 Khóa cửa nhận diện khuôn mặt

Sau quá trình huấn luyện cho mô hình nhận diện khuôn mặt, chúng ta tiến hành thực thi file code *face\_lock.py* (Hình 20). Trong vài giây, chế độ xem webcam sẽ mở ra. Hướng webcam vào mặt chúng ta. Nếu có một hộp màu xanh xung quanh khuôn mặt với tên của chúng ta, mô hình đã được huấn luyện chính xác để nhận dạng khuôn mặt. Đồng thời, khi nhận diện được khuôn mặt của chúng ta, khóa cửa sẽ tự động mở ra. Quan sát kết quả ở Hình 21 và Hình 22.

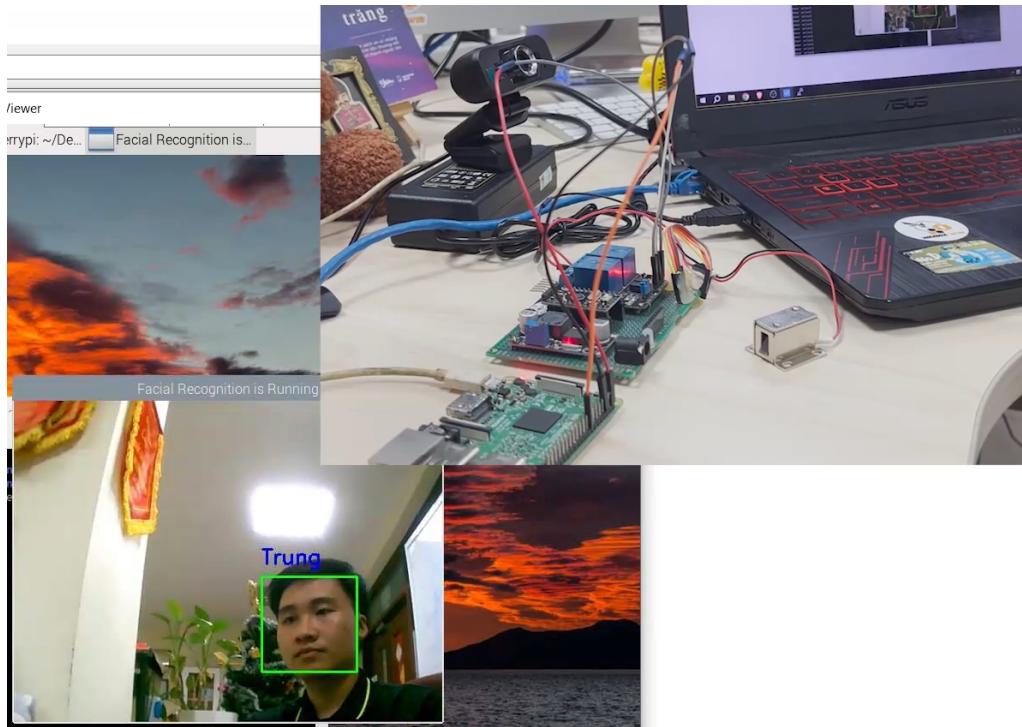
```

face_lock.py

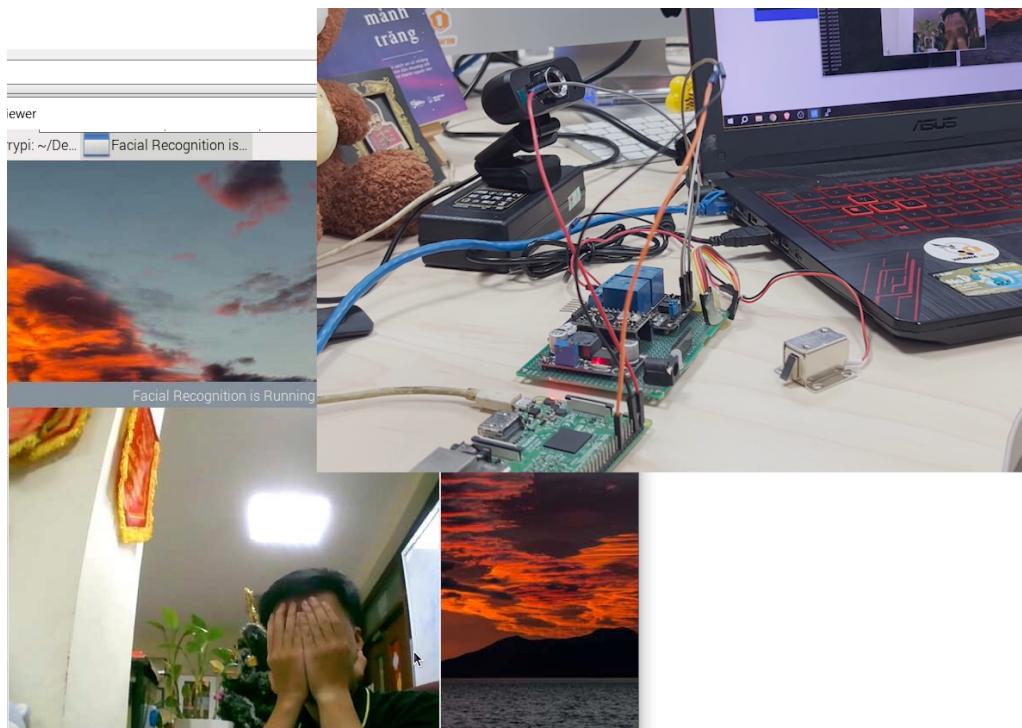
3 # import the necessary packages
4 from imutils.video import VideoStream
5 from imutils.video import FPS
6 import face_recognition
7 import imutils
8 import pickle
9 import time
10 import cv2
11 import RPi.GPIO as GPIO
12
13 RELAY = 23
14 GPIO.setwarnings(False)
15 GPIO.setmode(GPIO.BCM)
16 GPIO.setup(RELAY, GPIO.OUT)
17 GPIO.output(RELAY,GPIO.LOW)
18
19 #Initialize 'currentname' to trigger only when a new person is identified.
20 currentname = "unknown"
21 #Determine faces from encodings.pickle file model created from train_model.py
22 encodingsP = "encodings.pickle"
23 #use this xml file
24 #https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml
25 cascade = "haarcascade_frontalface_default.xml"
26
27 # Load the known faces and embeddings along with OpenCV's Haar
28 # cascade for face detection
29 print("[INFO] loading encodings + face detector...")
30 data = pickle.loads(open(encodingsP, "rb").read())
31 detector = cv2.CascadeClassifier(cascade)
32
33 # initialize the video stream and allow the camera sensor to warm up
34 print("[INFO] starting video stream...")
35 vs = VideoStream(src=0).start()
36 #vs = VideoStream(usePiCamera=True).start()
37 time.sleep(2.0)
38
39 # start the FPS counter
40 fps = FPS().start()
41
42 prevTime = 0
43 doorUnlock = False
44
45 # Loop over frames from the video file stream
46 while True:
47     # grab the frame from the threaded video stream and resize it
48     # to 500px (to speedup processing)
49     frame = vs.read()
50     frame = imutils.resize(frame, width=500)
51
52     # convert the input frame from (1) BGR to grayscale (for face

```

Hình 20: Một đoạn code minh họa mô hình khóa cửa nhận diện khuôn mặt.



Hình 21: Khóa cửa mở ra khi nhận diện được khuôn mặt của Trung.

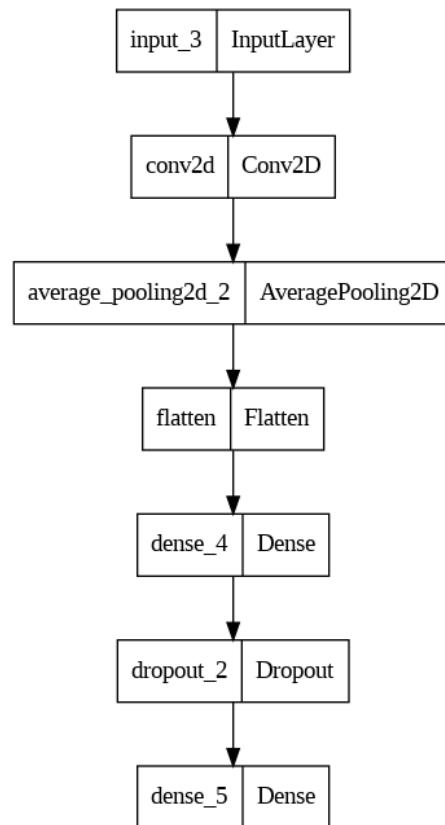


Hình 22: Khóa cửa đóng lại khi không nhận diện được khuôn mặt của Trung.

## 4.5 Nhận diện khuôn mặt đeo khẩu trang

### 4.5.1 Cấu hình mạng học sâu để huấn luyện

Mô hình mạng học sâu được cấu hình như mô tả ở **Hình 23** với Tensorflow.



Hình 23: Cấu hình mô hình nhận diện khuôn mặt có khẩu trang với Tensorflow.

Cấu hình khối tích chập:

- Conv2D
  - filters: 32
  - kernel\_size: 3
  - padding: same
  - stride: 1
  - activation: relu
- AveragePooling2D: pool\_size: 7x7
- Flatten

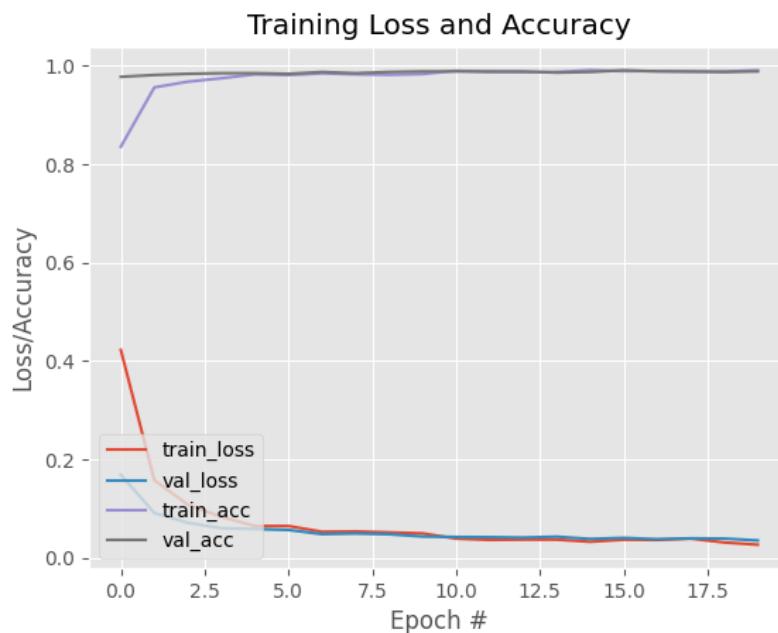
- Dense: units: 128
- Dropout: rate: 0.5
- Dense: units: 2

Mô hình được huấn luyện với các thông số như sau:

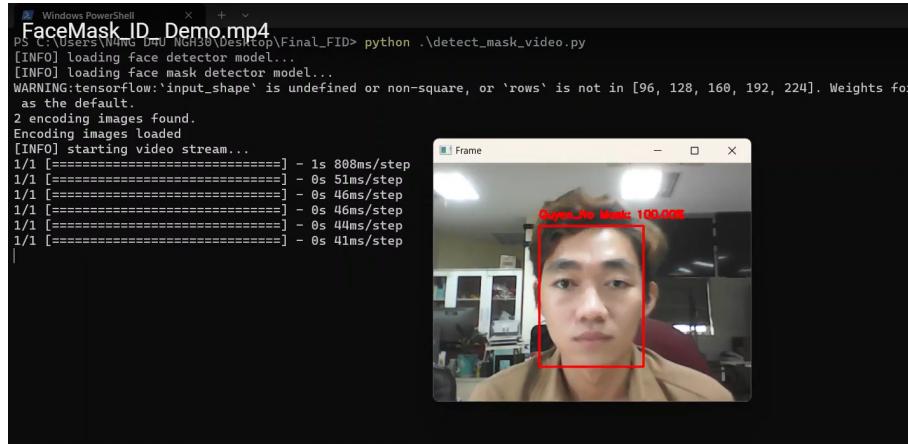
- Số epochs: 20
- Batch size: 32
- Thuật toán tối ưu: Adam
- Learning rate: 0.0001
- Hàm loss: Binary Cross Entropy

#### 4.5.2 Kết quả huấn luyện mô hình nhận diện khuôn mặt có khẩu trang

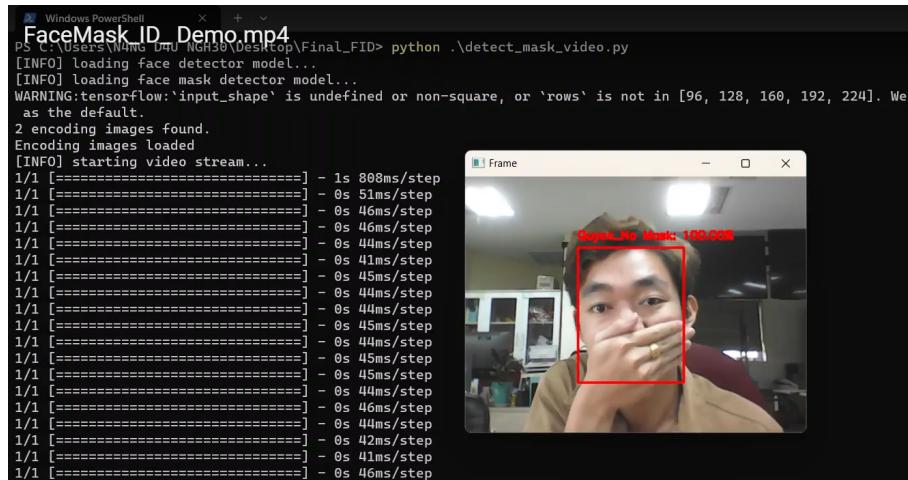
Quan sát Hình 24 có thể thấy độ chính xác đạt rất cao và giá trị mất mát giảm nhanh dần về 0 qua mỗi epochs. Từ những giá trị này, mô hình có khả năng đạt hiệu suất cao trong việc nhận diện khuôn mặt và đối tượng, cụ thể là kể cả khi chúng ta dùng tay che mặt hoặc đeo khẩu trang.



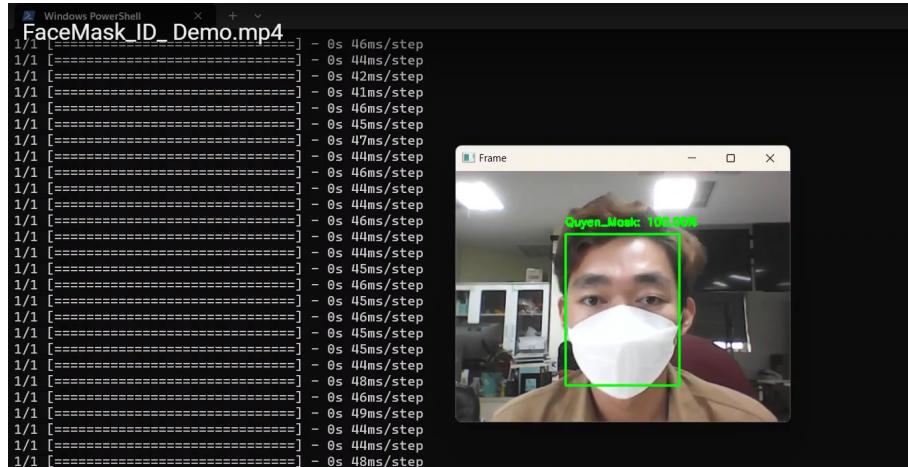
Hình 24: Hiệu suất mô hình nhận diện khuôn mặt có khẩu trang.



Hình 25: Mô hình nhận diện khuôn mặt khi không đeo khẩu trang.



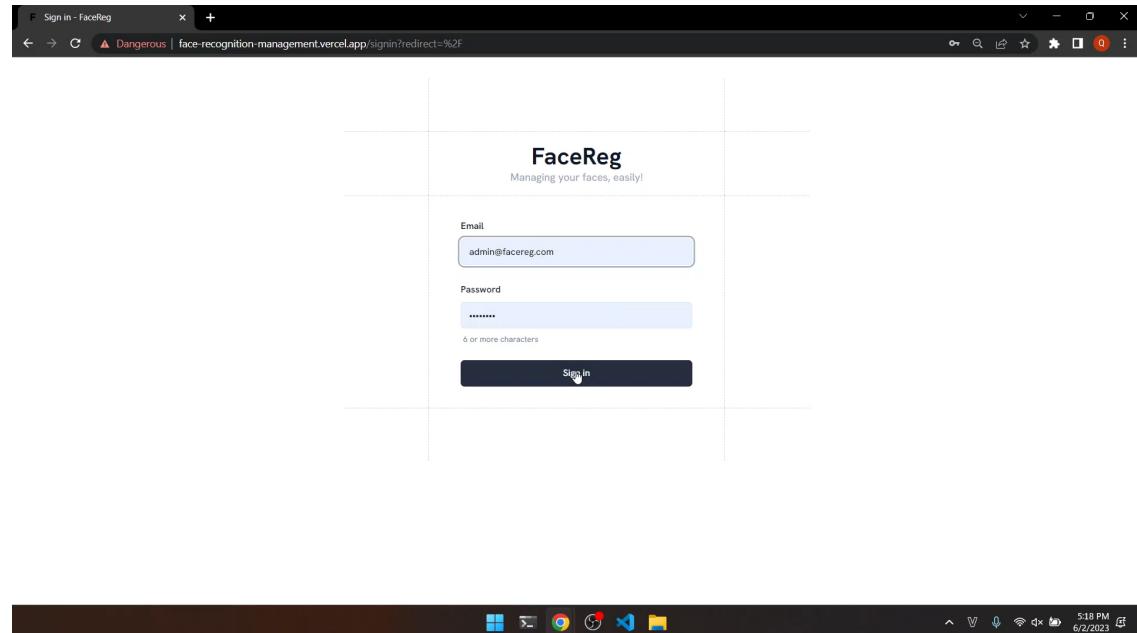
Hình 26: Mô hình nhận diện khuôn mặt khi không đeo khẩu trang kẽ cả khi dùng tay che mặt.



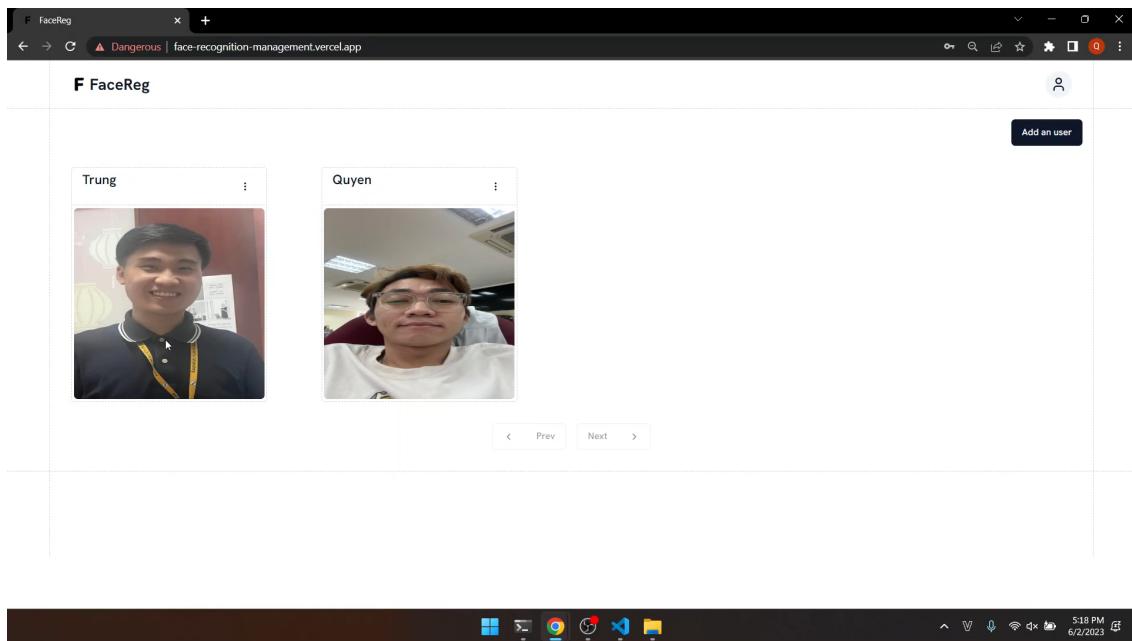
Hình 27: Mô hình nhận diện khuôn mặt khi đeo khẩu trang.

## 4.6 Ứng dụng web

Trong trang web này, chúng tôi sẽ cung cấp quyền quản lý cho các thành viên của quản trị viên. Để đảm bảo an toàn cho dữ liệu của người dùng mà quản trị viên đang quản lý, chúng tôi tạo tính năng đăng nhập cho trang web.

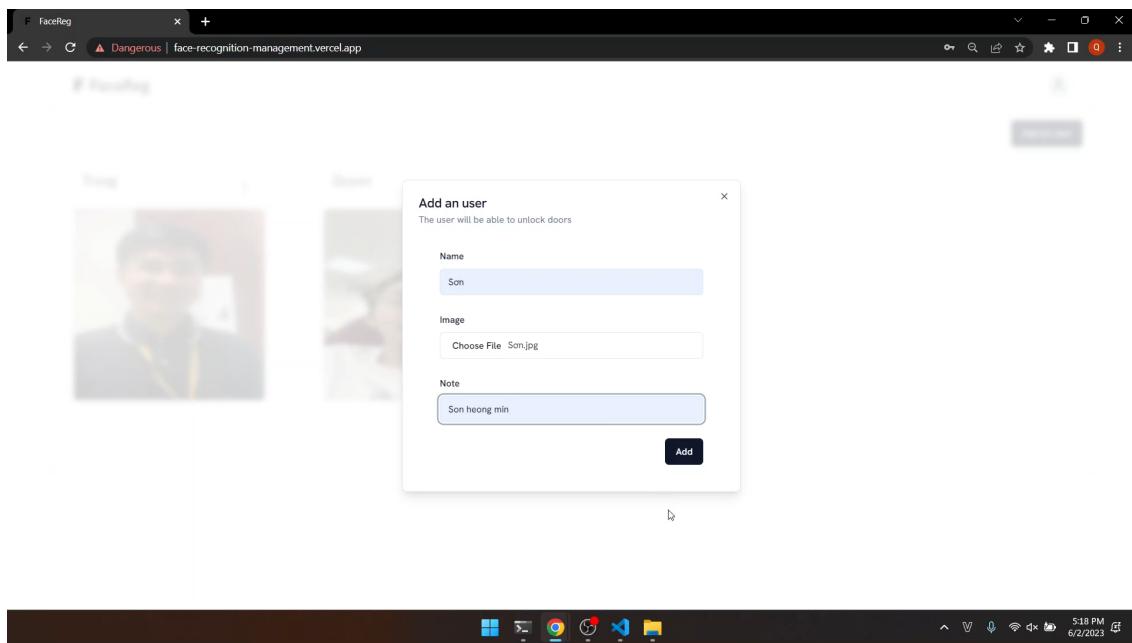


Hình 28: Tạo đăng nhập để quản trị viên truy cập dữ liệu người dùng.

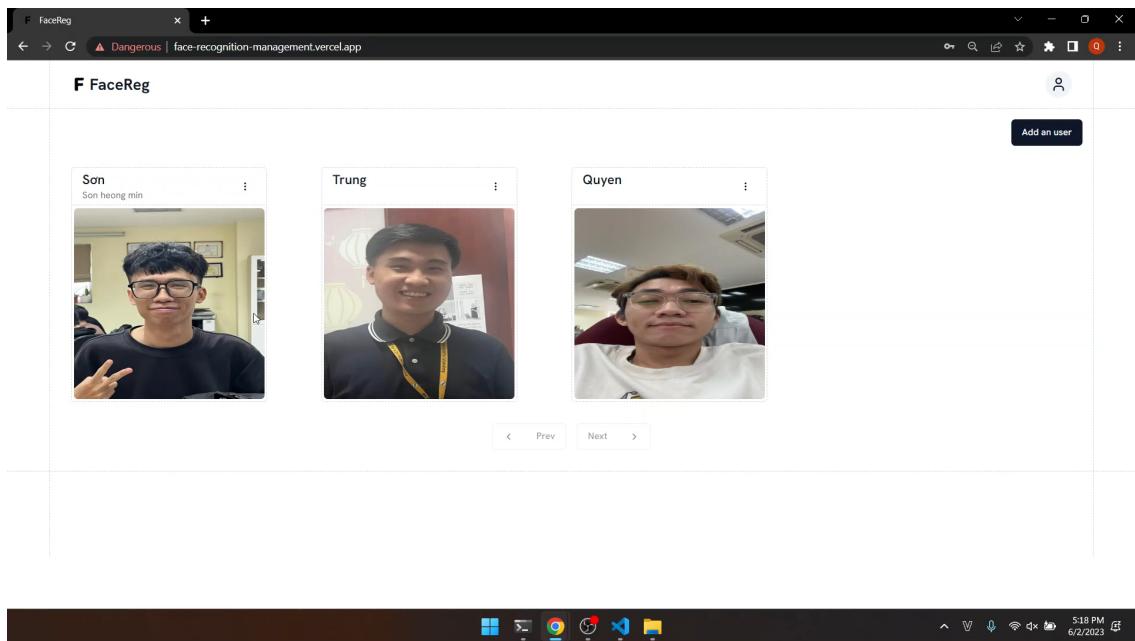


Hình 29: Giao diện chính của trang web.

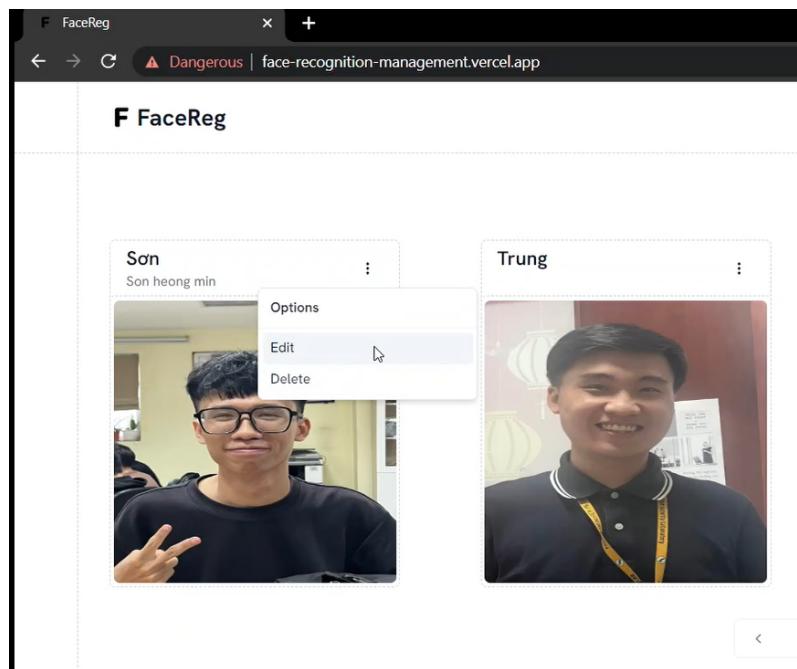
Sau khi đăng nhập thành công, quản trị viên có thể quản lý dữ liệu của người dùng. Quản trị còn có thể thực hiện các hành động như thêm, xóa, cập nhật dữ liệu của người dùng.



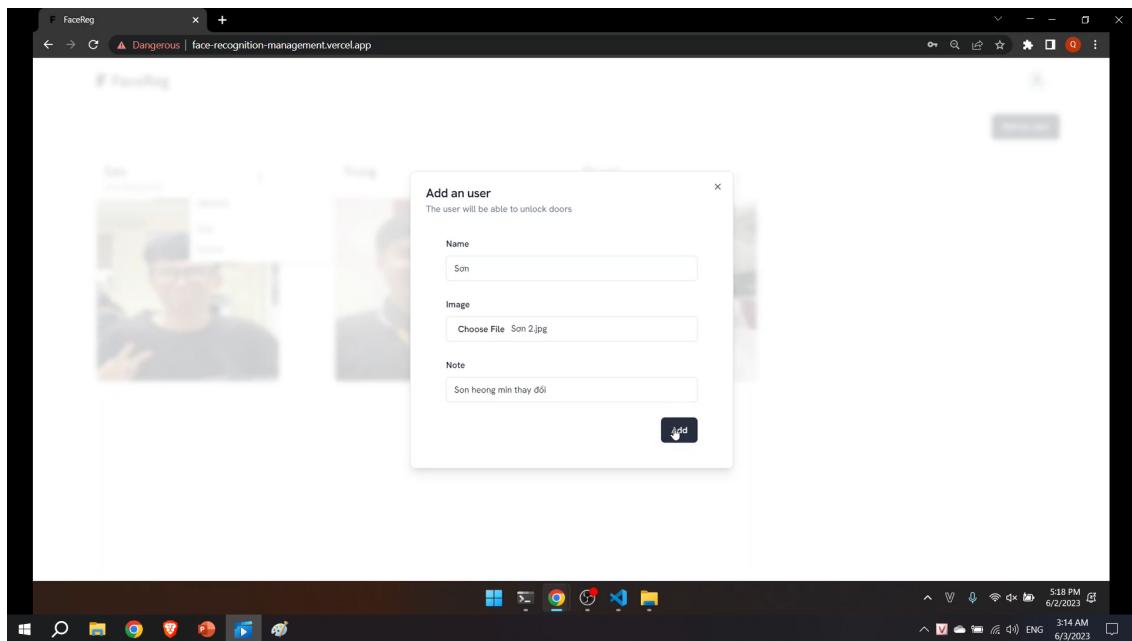
Hình 30: Thêm người dùng vào bộ dữ liệu.



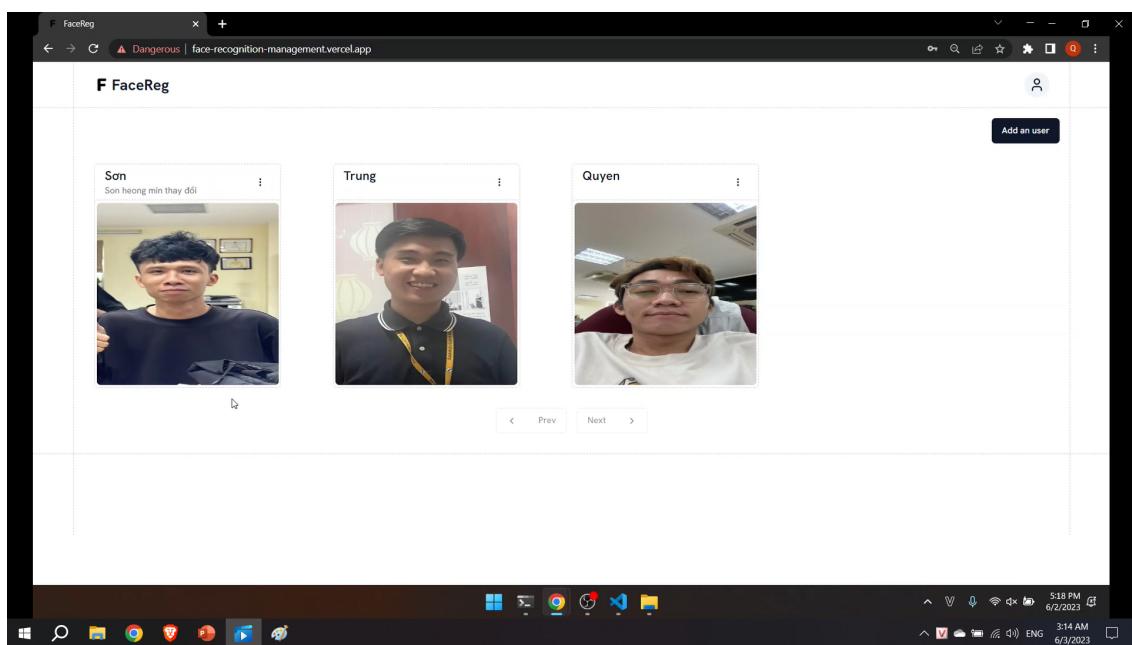
Hình 31: Giao diện sau khi thêm người dùng.



Hình 32: Tùy chọn chỉnh sửa dữ liệu người dùng.



Hình 33: Cập nhật dữ liệu người dùng, cụ thể là hình ảnh đại diện và ghi chú.



Hình 34: Giao diện sau khi cập nhật dữ liệu người dùng.

## 5 Kết luận và hướng phát triển

### 5.1 Kết luận

Chúng tôi đã xây dựng thành công hệ thống khóa cửa nhận diện khuôn mặt và xây dựng được mô hình tự huấn luyện có khả năng nhận diện đối tượng, cụ thể là khuôn mặt đang đeo khẩu trang. Đồng thời cũng cung cấp được cho quản trị viên một ứng dụng web để họ dễ dàng hơn trong việc quản lý dữ liệu người dùng. Tuy nhiên, chúng tôi gặp phải một vấn đề đó là chưa tích hợp được mô hình tự huấn luyện có khả năng nhận diện khuôn mặt đang đeo khẩu trang vào hệ thống khóa cửa. Nguyên nhân vì chúng tôi sử dụng RPi3, máy vi tính này không hỗ trợ hệ điều hành Raspian 64 bit. Một số thư viện trong ngôn ngữ Python, cụ thể là tensorflow chỉ hỗ trợ trên Python 64 bit. Do đó, chúng tôi không thể cài đặt thư viện này trên RPi3.

### 5.2 Hướng phát triển

Trong tương lai, chúng tôi sẽ chuyển RPi3 sang RPi4 để có thể cài đặt các thư viện hỗ trợ cho việc tích hợp. Đồng thời, chúng tôi cũng nâng cấp mô hình tự huấn luyện có khả năng nhận diện đối tượng để có thể đạt hiệu suất cao hơn nữa.

## Tài liệu tham khảo

- [1] "Firebase - Make your app the best it can be". URL: <https://firebase.google.com/docs/hosting/frameworks/nextjs?hl=vi>.
- [2] "Nextjs - The React Framework for the Web". URL: <https://nextjs.org/docs>.
- [3] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).
- [4] Lê Khải Nguyên Ngô Thành Đạt. *Thiết kế và thi công hệ thống IOT chăm sóc vườn cây ăn quả sử dụng pin năng lượng mặt trời*. Đồ Án Tốt Nghiệp DH, Trường ĐH Sư Phạm Kỹ Thuật TP.HCM, 2018.