

**BỘ GIÁO DỤC & ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**  
**KHOA ĐIỆN – ĐIỆN TỬ**  
**BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH**



**ĐÒ ÁN TỐT NGHIỆP**  
**NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG**

**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐIỂM DANH  
BẰNG NHẬN DIỆN KHUÔN MẶT KẾT HỢP MÃ QR  
VÀ MÃ VẠCH TRÊN RASPBERRY PI 4**

**GVHD: ThS. VÕ ĐỨC DŨNG**  
**SVTH: NGUYỄN ĐÔNG QUÝ**  
**MSSV: 16141251**  
**SVTH: LÊ PHÚ HIẾU**  
**MSSV: 16141153**



**Tp. Hồ Chí Minh, tháng 01/2021**

BỘ GIÁO DỤC & ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH



# ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG  
ĐỀ TÀI:

THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐIỂM DANH  
BẰNG NHẬN DIỆN KHUÔN MẶT KẾT HỢP MÃ QR  
VÀ MÃ VẠCH TRÊN RASPBERRY PI 4

GVHD: ThS. Võ Đức Dũng

SVTH: Nguyễn Đồng Quý 16141251

Lê Phú Hiếu 16141153

Tp. Hồ Chí Minh – 01/2021

BỘ GIÁO DỤC & ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

# ĐỒ ÁN TỐT NGHIỆP

NGÀNH KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐIỂM DANH  
BẰNG NHẬN DIỆN KHUÔN MẶT KẾT HỢP MÃ QR  
VÀ MÃ VẠCH TRÊN RASPBERRY PI 4

GVHD: ThS. Võ Đức Dũng

SVTH: Nguyễn Đồng Quý 16141251

Lê Phú Hiếu 16141153

Tp. Hồ Chí Minh – 01/2021

Tp. HCM, ngày 20 tháng 01 năm 2021

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên:	Lê Phú Hiếu	MSSV: 16141153
	Nguyễn Đồng Quý	MSSV: 15141251
Chuyên ngành:	Kỹ thuật Điện tử - Truyền thông	Mã ngành: 141
Hệ đào tạo:	Đại học chính quy	Mã hệ: 1
Khóa:	2016	Lớp: 16941DT

### I. TÊN ĐỀ TÀI:

#### THIẾT KẾ VÀ THI CÔNG HỆ THỐNG ĐIỂM DANH BẰNG NHẬN DIỆN KHUÔN MẶT KẾT HỢP MÃ QR VÀ MÃ VẠCH TRÊN RASPBERRY PI 4.

### II. NHIỆM VỤ

#### 1. Các số liệu ban đầu:

- Webcam Logitech C270p đóng vai trò thu nhận ảnh đầu vào.
- Raspberry Pi 4 đóng vai trò bộ xử lý trung tâm.
- Màn hình cảm ứng điện dung đóng vai trò giao tiếp với người dùng.

#### 2. Nội dung thực hiện:

- Tìm hiểu về các lý thuyết và các thuật toán sử dụng cho bài toán nhận dạng đối tượng.
- Nghiên cứu và xử lý hình ảnh từ camera.
- Nghiên cứu về trích xuất đặc trưng của khuôn mặt.
- Tìm hiểu cách lập trình giao tiếp với Raspberry Pi 4.
- Nghiên cứu thiết kế giao diện giao tiếp với người dùng.
- Nghiên cứu kết hợp với các phương thức điểm danh khác.
- Nghiên cứu về cách thức quản lý dữ liệu.
- Thực hiện quá trình huấn luyện cho mạng.

- Thiết kế hệ thống.
- Thi công hệ thống.
- Kiểm thử, chỉnh sửa và hoàn thiện hệ thống.
- Viết báo cáo thực hiện.
- Bảo vệ luận văn.

III. NGÀY GIAO NHIỆM VỤ: 05/10/2020

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 20/01/2021

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: ThS. Võ Đức Dũng

CÁN BỘ HƯỚNG DẪN BM. ĐIỆN TỬ CÔNG NGHIỆP – Y SINH

## LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

(Bản lịch trình này được đóng vào cuốn báo cáo)

Họ tên sinh viên: Lê Phú Hiếu

MSSV: 16141153

Họ tên sinh viên: Nguyễn Đồng Quý

MSSV: 16141251

Lớp: 16941DT (Sư phạm Điện tử công nghiệp)

Tên đề tài: “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện kết hợp mã QR và mã vạch trên Raspberry Pi 4**”.

Tuần/ngày	Nội dung	Xác nhận GVHD
Tuần 1 (5/10 – 11/10)	Gặp GVHD để nghe phổ biến yêu cầu làm đồ án, tiến hành chọn đồ án.	
Tuần 2 (12/10 – 18/10)	- Gặp GVHD trình bày và nộp đề cương. - Thiết kế và nộp lịch trình làm việc.	
Tuần 3 (19/10 – 25/10)	- Tìm hiểu công nghệ AI. - Tìm hiểu về các phương pháp nhận diện. - Tìm hiểu ngôn ngữ lập trình Python.	
Tuần 4,5 (26/10 – 08/11)	- Tìm hiểu cách trích xuất đặt trưng khuôn mặt. - Tìm hiểu xử lý ảnh, chụp hình đưa vào cơ sở dữ liệu từ camera.	
Tuần 6,7,8	- Xây dựng tập dữ liệu huấn luyện.	

(09/11 – 29/11)	- Viết chương trình Python nhận diện khuôn mặt.	
Tuần 9,10 (30/11 – 13/12)	Thiết kế giao diện giao tiếp người dùng.	
Tuần 11 (14/12 – 20/12)	Tìm hiểu kết hợp hệ thống điểm danh mã QR, Barcode	
Tuần 12 (21/12 – 27/12)	Thi công thiết kế mô hình	
Tuần 13 (28/12 – 03/01)	- Tiến hành thử nghiệm. - Đánh giá mô hình.	
Tuần 14 (04/01 – 10/01)	Viết Báo cáo	
Tuần 15+16 (11/01 – 24/01)	- Hoàn thành báo cáo - Hoàn thành mô hình	

### GV HƯỚNG DẪN

(Ký và ghi rõ họ và tên)

## LỜI CAM ĐOAN

Nhóm thực hiện – Lê Phú Hiếu và Nguyễn Đồng Quý cam đoan đề tài tốt nghiệp “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện khuôn mặt kết hợp mã QR và mã vạch trên Raspberry Pi 4**” là công trình nghiên cứu của nhóm thực hiện dưới sự hướng dẫn của thầy ThS.Võ Đức Dũng. Kết quả công bố trong khóa luận tốt nghiệp là trung thực và không sao chép từ bất kỳ công trình nào khác.

*Tp.HCM, ngày 20 tháng 01 năm 2021*

SV thực hiện đồ án

(Ký và ghi rõ họ tên)

Nguyễn Đồng Quý

Lê Phú Hiếu

## LỜI CẢM ƠN

Để hoàn thành đề tài nghiên cứu này, lời đầu tiên nhóm thực hiện xin chân thành cảm ơn các thầy cô giáo Trường Đại Học Sư Phạm Kỹ Thuật TP.HCM nói chung và các thầy cô trong Khoa Điện – Điện tử nói riêng, những người đã dạy dỗ, trang bị cho các thành viên nhóm thực hiện đề tài với những kiến thức nền tảng và chuyên ngành bối cảnh, giúp nhóm thực hiện có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng tôi trong quá trình học tập.

Đặc biệt, nhóm thực hiện xin chân thành cảm ơn thầy ThS. Võ Đức Dũng đã tận tình giúp đỡ, trực tiếp chỉ bảo, đưa ra những định hướng nghiên cứu cũng như hướng giải quyết một số vấn đề trong suốt thời gian nghiên cứu thực hiện đồ án. Trong thời gian làm việc với thầy, nhóm thực hiện không ngừng tiếp thu thêm nhiều kiến thức, thái độ nghiên cứu khoa học nghiêm túc, hiệu quả, đây là điều rất cần thiết trong quá trình học tập và công tác sau này.

Đồng thời nhóm thực hiện cũng xin gửi lời cảm ơn chân thành tới gia đình, bạn bè đặc biệt là tập thể lớp 16941DT, các anh chị em trong Khoa Điện – Điện tử đã động viên chúng tôi trong những lúc khó khăn, góp ý, chia sẻ kinh nghiệm, hỗ trợ trong quá trình nghiên cứu để nhóm chúng tôi có thể hoàn thành đề tài tốt nghiệp này.

Nhóm thực hiện xin chân thành cảm ơn!

Tp.HCM, ngày 20 tháng 01 năm 2021

SV thực hiện đồ án

(Ký và ghi rõ họ tên)

Nguyễn Đồng Quý

Lê Phú Hiếu

## MỤC LỤC

LỜI CAM ĐOAN .....	iii
LỜI CẢM ƠN .....	iv
MỤC LỤC .....	v
LIỆT KÊ HÌNH VẼ .....	ix
LIỆT KÊ BẢNG .....	xii
TÓM TẮT.....	xii
CHƯƠNG 1. TỔNG QUAN .....	1
1.1. Đặt vấn đề.....	1
1.2. Mục tiêu.....	2
1.3. Nội dung nghiên cứu .....	2
1.4. Giới hạn đề tài .....	3
1.5. Bố cục đề tài .....	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	5
2.1. Giới thiệu về xử lý ảnh .....	5
2.1.1. Thu nhận ảnh.....	6
2.1.2. Tiền xử lý ảnh .....	6
2.1.3. Phân đoạn ảnh .....	6
2.1.4. Biểu diễn và mô tả.....	6
2.1.5. Nhận dạng và nội suy.....	7
2.2. Giới thiệu về trí tuệ nhân tạo AI (Artificial Intelligent) .....	7
2.2.1. Trí tuệ nhân tạo AI .....	7
2.2.2. Học máy (Machine learning) .....	9
2.2.3. Học sâu (Deep Learning) .....	9
2.3. Thuật toán rút trích đặc trưng LBP (Local Binary Pattern) .....	10
2.3.1. Lý thuyết về LBP .....	10

2.3.2. Thuật toán LBP .....	12
2.3.3. Nguyên lí phân lớp không tham biến .....	13
2.3.4. Phép quay bất biến .....	14
2.3.5. Độ tương phản và kết cấu mẫu .....	15
2.4. Đặc trưng Haar – Like .....	16
2.5 Giải thuật AdaBoost .....	18
2.5. Ngôn ngữ lập trình Python .....	20
2.5.1. Giới thiệu về ngôn ngữ lập trình Python[4] .....	20
2.5.2. Giới thiệu thư viện OpenCV [4] .....	21
2.5.3. Giới thiệu thư viện Tkinter.....	22
2.5.4. Đọc mã Barcode, QR Code.....	23
2.5.4.1. <i>Tổng quan mã vạch và mã QR code</i> .....	23
2.5.4.2. <i>Giới thiệu thư viện Zbar</i> .....	24
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG.....	26
3.1. Giới thiệu .....	26
3.2. Chọn và giới thiệu từng loại linh kiện của từng khối .....	27
3.2.1. Khối xử lý ảnh.....	27
3.2.2. Khối xử lý trung tâm .....	28
3.2.3. Khối hiển thị.....	31
3.2.4. Khối nguồn.....	34
3.3. Cài đặt hệ điều hành cho Raspberry Pi 4.....	36
3.3.1. Thiết bị cần để cài đặt .....	36
3.3.2 Cài đặt hệ điều hành Raspian .....	36
3.4. Cài đặt thư viện cho Raspberry Pi 4.....	38
3.4.1. Cài đặt thư viện OpenCV .....	38
3.4.2. Cài đặt thư viện Pyzbar .....	43
3.4.3. Cài đặt thư viện Pandas .....	43

3.4.4. Thiết kế hộp chứa linh kiện hệ thống .....	44
<b>CHƯƠNG 4. THI CÔNG HỆ THỐNG .....</b>	<b>46</b>
4.1. Giới thiệu .....	46
4.2. Yêu cầu và mô tả hoạt động của hệ thống:.....	46
4.2.1. Yêu cầu hệ thống.....	46
4.2.2. Mô tả hoạt động hệ thống .....	46
4.3. Thi công phần cứng hệ thống .....	47
4.3.1. Chuẩn bị linh kiện phần cứng .....	47
4.3.2. Lắp ráp phần cứng.....	48
4.3.3. Lắp giá đỡ.....	51
4.4. Lập trình hệ thống .....	52
4.4.1. Thiết kế giao diện của hệ thống .....	52
4.4.1.1. <i>Tạo Label trong Tkinter:</i> .....	52
4.4.1.2. <i>Tạo Button trong Tkinter</i> .....	54
4.4.1.3. <i>Cấu trúc giao diện của hệ thống</i> .....	57
4.4.2. Lưu đồ lập trình hệ thống hệ thống .....	58
4.4.2.1. <i>Lưu đồ giải thuật tổng thể</i> .....	58
4.4.2.2. <i>Lưu đồ giải thuật lấy dữ liệu</i> .....	60
4.4.2.3. <i>Lưu đồ giải thuật Huấn luyện Dữ liệu</i> .....	62
4.4.2.4. <i>Lưu đồ giải thuật Nhận Diện Khuôn mặt</i> .....	64
4.4.2.5. <i>Lưu đồ giải thuật điểm danh bằng QR Code</i> .....	66
4.4.3. Hướng dẫn sử dụng .....	67
<b>CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT VÀ ĐÁNH GIÁ .....</b>	<b>70</b>
5.1. Kết quả.....	70
5.1.1. Kết quả phần cứng .....	70
5.1.2. Kết quả phần mềm .....	71
5.1.3. Kết quả thực hiện thử nghiệm hệ thống .....	72

5.2. Nhận xét và đánh giá .....	85
5.2.1. Nhận xét .....	85
5.2.2. Đánh giá .....	86
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>87</b>
6.1. Kết luận.....	87
6.1.1. Ưu điểm.....	87
6.1.2. Hạn chế.....	88
6.2. Hướng phát triển.....	88
6.2.1. Hướng khắc phục .....	88
6.2.2. Hướng phát triển .....	88
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>89</b>
<b>PHỤ LỤC .....</b>	<b>90</b>

## LIỆT KÊ HÌNH VẼ

Hình	Trang
Hình 2. 1: Các bước trong xử lý ảnh .....	5
Hình 2. 2: Mối quan hệ giữa AI, Machine Learning và Deep Learning. ....	8
Hình 2. 3: Ví dụ về LBP và độ tương phản cục bộ C.....	11
Hình 2. 4: Tập hợp các điểm xung quanh Ptt. ....	11
Hình 2. 5: Các biến thể của LBP LBP đồng dạng.....	12
Hình 2. 6: Bảng thống kê các mẫu của uniform LBP.....	13
Hình 2. 7: Minh họa các trường hợp của LBP khi quay với góc 15 độ .....	15
Hình 2. 8: Đặc trưng theo cạnh .....	16
Hình 2. 9: Đặc trưng theo đường .....	16
Hình 2. 10: Đặc trưng theo xung quanh tâm .....	16
Hình 2. 11: Đặc trưng theo đường chéo .....	16
Hình 2. 12: Cách tính Integral Image của ảnh.....	17
Hình 2. 13: Cách tính tổng giá trị pixel vùng cần tính .....	18
Hình 2. 14: Mô hình phân tầng kết hợp các bộ phân loại yếu.....	19
Hình 2. 15: Sơ đồ nhận diện khuôn mặt .....	19
Hình 2. 16: Ngôn ngữ lập trình Python .....	20
Hình 2. 17: Thư viện OpenCV .....	21
Hình 2. 18: Mã Barcode với MSSV: 16141251 .....	23
Hình 2. 19: Mã QR Code với MSSV: 16141251 .....	23
Hình 2. 20: Format định dạng cho mã QR Code .....	24
Hình 2. 21: Mô hình hệ thống Zbar .....	25
Hình 3. 1: Sơ đồ khái niệm .....	26
Hình 3. 2: Hình ảnh Webcam Logitech C270 .....	27
Hình 3. 3: Raspberry Pi 4 model B thực tế .....	29
Hình 3. 4: Các cổng giao tiếp ngoại vi Raspberry Pi 4 Model B, RAM 4GB. ....	31
Hình 3. 5: LCD 7 inch HDMI(B) hãng WaveShare .....	32
Hình 3. 6: Cấu tạo màn hình điện dung 7inch Waveshare .....	33
Hình 3. 7: Kết nối giữa màn hình cảm ứng và Raspberry Pi 4 Model B .....	34
Hình 3. 8: Adapter công tắc nguồn 5V/3A .....	35
Hình 3. 9: Sơ đồ cấp nguồn .....	36
Hình 3. 10: Tải phiên bản Raspbian Buster mới nhất. ....	37
Hình 3. 11: Cài đặt phần mềm SD Card Formatter .....	37
Hình 3. 12: Phần mềm Win32DiskImager. ....	38
Hình 3. 13: Mô phỏng vị trí Pi 4 và camera .....	44
Hình 3. 14: Thiết kế hộp chứa và kết nối thiết bị hoàn chỉnh .....	45
Hình 3. 15: Xuất file cắt mica .....	45
Hình 4. 1: Sơ đồ kết nối dây các thiết bị phần cứng .....	48
Hình 4. 2: Lắp quạt và Raspberry Pi 4 và hộp .....	49
Hình 4. 3: Màn hình được lắp vào khung .....	49
Hình 4. 4: Camera được lắp vào hộp .....	50
Hình 4. 5: Kết nối dây các thiết bị .....	50
Hình 4. 6: Kết nối và kiểm tra hoạt động phần cứng .....	51

Hình 4. 7: Giá đỡ hộp hệ thống .....	51
Hình 4. 8: Lắp khung kết nối với giá đỡ vào màn hình.....	52
Hình 4. 9: Giao diện được tạo bằng Tkinter với Lable .....	54
Hình 4. 10: Giao diện được tạo bằng Tkinter Vói Button.....	56
Hình 4. 11: Giao diện hoàn chỉnh.....	57
Hình 4. 12: Lưu đồ tổng thể hệ thống .....	58
Hình 4. 13: Lưu đồ giải thuật Lấy Dữ Liệu.....	60
Hình 4. 14: Lưu đồ giải thuật Huấn luyện Dữ liệu.....	62
Hình 4. 15: Lưu đồ giải thuật Nhận Diện .....	64
Hình 4. 16: Lưu đồ giải thuật Điểm danh bằng QR Code và Barcode.....	66
Hình 4. 17: Khởi động chương trình Pycharm .....	67
Hình 4. 18: Chạy file chương trình chính.....	67
Hình 4. 19: Giao diện chính của hệ thống .....	68
Hình 4. 20: Địa chỉ kết quả điểm danh .....	69
Hình 5. 1: Kết nối hoàn chỉnh phần cứng.....	70
Hình 5. 2: Lắp màn hình vào giá đỡ .....	71
Hình 5. 3: Màn hình giao tiếp người dùng hoàn chỉnh. ....	72
Hình 5. 4: Phòng nền phía sau khu vực điểm danh .....	73
Hình 5. 5: Thực hiện lấy dữ liệu sinh viên Nguyễn Thu Vy .....	74
Hình 5. 6: Ảnh sau khi lấy dữ liệu lưu trong data_face. ....	74
Hình 5. 7: Điểm danh sinh viên Lê Thị Thu Ba .....	75
Hình 5. 8: Điểm danh sinh viên Lê Phú Hiếu .....	76
Hình 5. 9: Điểm danh các sinh viên khác .....	76
Hình 5. 10: Trường hợp “ Không xác định” (Trần Thanh Nho). .....	77
Hình 5. 11: Kết quả điểm danh bằng nhận diện khuôn mặt. ....	77
Hình 5. 12: Thiết kế thẻ sinh viên .....	79
Hình 5. 13: Kết quả điểm danh bằng thẻ QR code và barcode .....	80
Hình 5. 14: Kết quả được lưu và file StudentDailBarQR.csv .....	81
Hình 5. 15: Thực nghiệm ở điều kiện không thuận lợi .....	82
Hình 5. 16: Sinh viên thử nghiệm mang khẩu trang khi điểm danh.....	84
Hình 5. 17: Các phần mã vạch bị che khuất .....	85
Hình 5. 18: Mã vạch bị đứt hoặc mất .....	85
Hình 5. 19: Mã thẻ bị phai màu và mờ.....	85

## **LIỆT KÊ BẢNG**

<b>Bảng</b>	<b>Trang</b>
Bảng 4. 1: Bảng linh kiện chuẩn bị lắp ráp hệ thống.	47
Bảng 4. 2: Các tùy chọn của Lable	53
Bảng 5. 1: Kết quả tiến hành thực nghiệm điểm danh bằng nhận diện khuôn mặt.....	78
Bảng 5. 2: Bảng thống kê chi tiết số lần đúng và số lần sai.....	78
Bảng 5. 3: Kết quả tiến hành thực nghiệm điểm danh bằng mã QR.....	79
Bảng 5. 4: Kết quả tiến hành thực nghiệm điểm danh bằng mã vạch (Barcode).....	79
Bảng 5. 5: Bảng tổng hợp kết quả thực nghiệm.....	81
Bảng 5. 6: Kết quả tiến hành thực nghiệm điểm danh bằng nhận diện khuôn mặt điều kiện không thuận lợi.....	82
Bảng 5. 7: Bảng ma trận nhầm lẫn điểm danh bằng nhận diện khuôn mặt điều kiện không thuận lợi.....	83
Bảng 5. 8: Kết quả tiến hành thực nghiệm điểm danh bằng mã QR và mã vạch trong môi trường không thuận lợi.....	83

## TÓM TẮT

Theo nhiều cách khác nhau AI đang dần trở nên phổ biến ở khắp mọi nơi trong cuộc sống hàng ngày. Không chỉ là những ứng dụng trong các lĩnh vực y tế, kinh doanh, tài chính ngân hàng... mà AI còn thể hiện vai trò quan trọng của mình trong lĩnh vực giáo dục. Duy trì việc đi học của học sinh và sinh viên có tính khách quan, công bằng và minh bạch là rất quan trọng trong tất cả các cơ sở giáo dục. Vì thế mỗi cơ sở giáo dục đều có các phương pháp điểm danh khác nhau. Trong đó có nơi điểm danh thủ công bằng cách đọc tên, một số khác đã áp dụng nhiều phương pháp công nghệ như vân tay, thẻ từ, mã vạch... Các phương pháp trên tốt hơn nhiều so với việc điểm danh truyền thống. Tuy nhiên vẫn còn nhiều hạn chế. Điểm danh bằng nhận diện khuôn mặt sẽ giải quyết được các khuyết điểm của các hệ thống trên như tránh tình trạng điểm danh hộ, lây lan dịch bệnh,... Nhóm chúng tôi quyết định thực hiện đề tài “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện khuôn mặt kết hợp mã QR và mã vạch trên Raspberry Pi 4**”.

Trong đề tài này nhóm thực hiện sẽ lập trình trên Raspberry pi 4, sử dụng màn hình cảm ứng điện dung để giao tiếp với người dùng. Nhóm thực hiện đã sử dụng phương pháp trích xuất đặc trưng trong việc nhận diện khuôn mặt so sánh đối chiếu đưa ra kết quả cuối cùng. Thu nhận ảnh từ camera, tự động cắt lấy khuôn mặt của từng sinh viên, sau đó đưa vào cơ sở dữ liệu, thực hiện quá trình huấn luyện, điểm danh bằng nhận diện khuôn mặt. Ngoài ra nhóm thực hiện còn kết hợp thêm việc điểm danh bằng mã QR và mã vạch tích hợp trên thẻ sinh viên, để đa dạng hóa cách thức điểm danh và tăng tính chính xác khi kết hợp điểm danh hai lớp cho việc phát triển đề tài sau này. Tất cả dữ liệu khi điểm danh sẽ được lưu vào file excel giúp giáo viên, người quản lý dễ dàng tổng hợp và lưu trữ.

Kết thúc nghiên cứu và thi công hệ thống, nhóm thực hiện đã đạt được mục tiêu ban đầu đề ra đó là nhận dạng điểm danh bằng khuôn mặt với độ chính xác trên 90%, tốc độ từ 10 giây đến 15 giây. Kết hợp với việc điểm danh bằng thẻ sinh viên mã QR và mã vạch độ chính xác 100% tốc độ từ 1 giây đến 2 giây, ở điều kiện ánh sáng vừa đủ, phông nền trắng (bức tường).

## CHƯƠNG 1. TỔNG QUAN

### 1.1. Đặt vấn đề

Theo nhiều cách khác nhau AI đang dần trở nên phổ biến ở khắp mọi nơi trong cuộc sống hằng ngày. Không chỉ là những ứng dụng trong các lĩnh vực y tế, kinh doanh, tài chính ngân hàng,... Mà AI còn thể hiện vai trò quan trọng của mình trong lĩnh vực giáo dục.

Duy trì việc đi học của học sinh và sinh viên có tính khách quan, công bằng và minh bạch là rất quan trọng trong tất cả các cơ sở giáo dục. Vì thế mỗi cơ sở giáo dục đều có các phương pháp điểm danh khác nhau. Trong đó có nơi điểm danh thủ công bằng cách đọc tên, một số khác đã áp dụng nhiều phương pháp công nghệ như vân tay, thẻ từ, mã vạch...

Tuy nhiên, công tác điểm danh bằng phương pháp thủ công hiện nay còn khá nhiều bất cập. Nếu điểm danh bằng cách đọc tên thì tốn rất nhiều thời gian, đôi khi có yếu tố tình cảm trong việc điểm danh hoặc điểm danh hộ. Nhóm thực hiện đã tham khảo đồ án tốt nghiệp của Trần Thị Linh Đa, Nguyễn Châu Ngân, “Thiết kế và thi công hệ thống điểm danh nhân viên sử dụng vi điều khiển ARM” [1], đã sử dụng quét thẻ RFID và vân tay. Nếu điểm danh bằng cách quét vân tay thì có thể lây dịch bệnh truyền nhiễm như COVID-19 hiện nay. Hay nếu điểm danh bằng quét thẻ thì có thể xảy ra tình trạng người này quét thẻ giúp người khác.

Nhóm thực hiện đã tham khảo đề tài của Lê Hoàng Thành, Hồ Đình Vương, “Thiết kế và thi công hệ thống bảo mật ứng dụng xử lý ảnh” [2]. Họ đã áp dụng việc nhận diện khuôn mặt để mở cửa, nhưng vẫn có khi hệ thống nhận diện sai dẫn đến việc bảo mật chưa cao. Cho nên cần phải kết hợp nhiều lớp bảo mật khác cùng với việc nhận diện khuôn mặt để tăng tính chính xác mà bảo mật cao.

Trong những năm gần đây, với sự phát triển của công nghệ, việc điểm danh bằng nhận diện khuôn mặt trở nên phổ biến, giúp các cơ sở giáo dục dễ thực hiện và tiết kiệm thời gian hơn. Máy học, phương pháp trích xuất đặc trưng để nhận dạng khuôn mặt. Đầu vào của hệ thống là một hình ảnh, sau đó sử dụng phương pháp trích đặc trưng để dự đoán những khuôn mặt.

## CHƯƠNG 1: TỔNG QUAN

Từ những khảo sát trên, cùng với các kiến thức đã được trang bị, chúng tôi làm đề tài thực hiện việc thiết kế một hệ thống điểm danh bằng nhận diện khuôn mặt. Từ đó tương tác với những thiết bị bên ngoài và đồng thời sử dụng thêm một camera và màn hình sẽ giúp cho người sử dụng dễ thực hiện hơn. Đề tài “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện khuôn mặt kết hợp mã QR và mã vạch trên Raspberry Pi 4**”. Sẽ khắc phục những khuyết điểm của các phương pháp điểm danh được nêu ở bên trên. Ngoài ra nhóm thực hiện sẽ kết hợp thêm một số phương pháp điểm danh khác như: QR code, Barcode để đa dạng hóa cách thức điểm danh trên cùng một hệ thống, tăng tính chính xác và bảo mật tuyệt đối.

### 1.2. Mục tiêu

Thiết kế hệ thống điểm danh bằng nhận diện khuôn mặt để giúp việc quản lý học sinh, sinh viên một cách chính xác, công bằng và minh bạch. Dễ dàng sử dụng và theo dõi đối tượng. Hệ thống chủ yếu sử dụng các phương thức, thuật toán trong máy học kỹ thuật xử lý hình ảnh, các phương pháp nhận diện khuôn mặt. Đồng thời ngoài việc điểm danh bằng công nghệ nhận diện khuôn mặt, hệ thống của nhóm thực hiện sẽ kết hợp thêm một số phương thức điểm danh khác như mã QR code, mã vạch (Barcode). Tăng tính chính xác, bảo mật và đa dạng cách thức điểm danh trên cùng một hệ thống.

### 1.3. Nội dung nghiên cứu

Trong quá trình thực hiện Đề án tốt nghiệp với đề tài Thiết kế hệ thống bằng công nghệ nhận diện khuôn mặt, nhóm thực hiện đã tập trung giải quyết và hoàn thành được những nội dung sau:

- **Nội dung 1:** Tìm hiểu về các lý thuyết các phương pháp và các thuật toán sử dụng cho bài toán nhận dạng đối tượng. Tìm hiểu về xử lý ảnh, máy học.
- **Nội dung 2:** Nghiên cứu và xử lý hình ảnh từ camera.
- **Nội dung 3:** Nghiên cứu về trích xuất đặc trưng của khuôn mặt.
- **Nội dung 4:** Nghiên cứu thiết kế giao diện giao tiếp với người dùng.
- **Nội dung 5:** Nghiên cứu kết hợp với các phương thức điểm danh khác.
- **Nội dung 6:** Nghiên cứu về cách thức quản lý dữ liệu.
- **Nội dung 7:** Thực hiện quá trình lập trình.
- **Nội dung 8:** Thiết kế hệ thống.
- **Nội dung 9:** Kiểm thử, chỉnh sửa và hoàn thiện hệ thống.

## CHƯƠNG 1: TỔNG QUAN

- **Nội dung 10:** Viết báo cáo thực hiện.
- **Nội dung 11:** Bảo vệ luận văn.

### 1.4. Giới hạn đề tài

Các thông số giới hạn của đề tài bao gồm:

- Dùng camera để phát hiện và nhận dạng khuôn mặt.
- Cách thức nhận dạng sử dụng các phương pháp, kỹ thuật như Haar cascade, Local Binary Patterns Histogram.
- Mô hình dừng lại ở việc điểm danh một lớp từ 20-30 học sinh/sinh viên/nhân viên.
- Camera nhận dạng kém chính xác trong điều kiện có ánh sáng cường độ mạnh hoặc quá yếu, bức xạ nhiệt lớn.
- Sử dụng framework, thư viện, cấu trúc đã được công bố.
- Camera nhận diện và phát hiện khuôn mặt tốt ở khoảng cách vừa phải (0,8m - 1m) không quá xa cũng như quá gần.
- Hệ thống điểm danh chỉ kết hợp tối đa 3 phương thức điểm danh là nhận diện khuôn mặt, QR code và mã vạch (Barcode).

### 1.5. Bộ cục đề tài

Gồm 6 chương:

#### ❖ **Chương 1:** Tổng quan.

Trong chương này nhóm tìm hiểu về các vấn đề hình thành nên đề tài. Kèm theo đó là nội dung thực hiện, mục tiêu và giới hạn của đề tài mà nhóm thực hiện đề tài đã đặt ra.

#### ❖ **Chương 2:** Cơ sở lý thuyết.

Trong chương này trình bày kiến thức về xử lý ảnh, máy học, các phương pháp nhận diện khuôn mặt, ngôn ngữ lập trình python và ứng dụng, trình thiết kế giao diện, một số kiến thức liên quan đến Barcode, QR code, camera, giới thiệu các thư viện được sử dụng trong đề tài.

#### ❖ **Chương 3:** Thiết kế hệ thống.

Chương này sẽ đi tính toán để lựa chọn các linh kiện phù hợp với hệ thống. Thiết kế sơ đồ khối, phân tích sơ đồ khối nêu chức năng của từng khối có trong hệ thống, các

## CHƯƠNG 1: TỔNG QUAN

giải thuật và hoàn thiện các khâu còn lại để có một hệ thống điểm danh, thiết kế giao diện giao tiếp người dùng, các phím chức năng có trong hệ thống.

### ❖ **Chương 4:** Thi công hệ thống.

Nội dung chương này là quá trình thiết kế phần cứng cho các phần của hệ thống. Viết chương trình điểm danh bằng nhận diện khuôn mặt kết hợp thêm QR code, Barcode, tạo cơ sở dữ liệu lưu hệ thống, viết chương trình tạo ra giao diện giao tiếp người dùng, tiến hành chế tạo giá đỡ cho màn hình.

### ❖ **Chương 5:** Kết quả, nhận xét và đánh giá.

Sau khi xây dựng hệ thống hoàn chỉnh, nhìn nhận lại kết quả đem lại có đúng với yêu cầu lúc đầu đã đề ra hay không? Nhận xét về hệ thống có hoạt động tốt không? Nêu lên ưu điểm và nhược điểm đã làm được. Đưa ra đánh giá về mọi phương diện thu được mà hệ thống này đem lại. Quan trọng nhất là phần trăm độ chính xác khi thực hiện điểm danh.

### ❖ **Chương 6:** Kết luận và hướng phát triển.

Trình bày những kết luận về hệ thống, những kết quả đạt được và những hạn chế, kết luận hiệu quả của hệ thống điểm danh trong thực tế, đồng thời nêu ra hướng phát triển cho hệ thống điểm danh.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Giới thiệu về xử lý ảnh

Xử lý ảnh là đối tượng nghiên cứu của lĩnh vực thị giác máy, là quá trình biến đổi từ một ảnh ban đầu sang một ảnh mới với các đặc tính và tuân theo ý muốn của người sử dụng. Xử lý ảnh có thể gồm quá trình phân tích, phân lớp các đối tượng, làm tăng chất lượng, phân đoạn và tách cạnh, gán nhãn cho vùng hay quá trình biên dịch các thông tin hình ảnh của ảnh. [3]

Cũng như xử lý dữ liệu bằng đồ họa, xử lý ảnh số là một lĩnh vực của tin học ứng dụng. Xử lý dữ liệu bằng đồ họa đề cập đến những ảnh nhân tạo, các ảnh này được xem xét như là một cấu trúc dữ liệu và được tạo bởi các chương trình. Xử lý ảnh số bao gồm các phương pháp và kỹ thuật biến đổi, để truyền tải hoặc mã hóa các ảnh tự nhiên. Mục đích của xử lý ảnh gồm: Biến đổi ảnh làm tăng chất lượng ảnh, tự động nhận dạng ảnh, đoán nhận ảnh, đánh giá các nội dung của ảnh.

Xử lý ảnh được áp dụng vào thực tế, đem lại nhiều hiệu quả trong các lĩnh vực như:

- Lĩnh vực quân sự: xử lý và nhận dạng ảnh quân sự.
- Lĩnh vực giao tiếp người máy: nhận dạng ảnh, xử lý âm thanh, đồ họa.
- Lĩnh vực an, bảo mật: nhận diện khuôn mặt người, nhận diện vân tay, ...
- Lĩnh vực giải trí: trò chơi điện tử.
- Lĩnh vực y tế: Xử lý ảnh y sinh, chụp X quang, MRI,...
- Trong công nghiệp: nhận dạng phân loại sản phẩm, trái cây,...

Các bước cần thiết trong xử lý ảnh. Trước đây, hình ảnh được thu từ camera là các ảnh tương tự. Gần đây, với sự phát triển không ngừng của công nghệ, ảnh màu hoặc ảnh đen trắng được lấy từ camera, sau đó được chuyển trực tiếp qua ảnh số để dễ dàng cho các bước xử lý tiếp theo. Dưới đây sẽ mô tả về các bước trong xử lý ảnh.



Hình 2. 1: Các bước trong xử lý ảnh

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1.1. Thu nhận ảnh

Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình xử lý ảnh. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, máy scan. Sau đó các tín hiệu này sẽ được số hóa. Các thông số quan trọng ở đây là độ phân giải, chất lượng. Màu và tốc độ thu nhận ảnh của các thiết bị. Chất lượng của một ảnh thu nhận được phụ thuộc vào thiết bị thu, vào môi trường (ánh sáng, phong cảnh). [4]

### 2.1.2. Tiền xử lý ảnh

Ở bước này, ảnh sẽ được xử lý để cải thiện về độ tương phản, khử nhiễu,... với mục đích làm cho chất lượng ảnh trở nên tốt hơn, rõ nét hơn. Quá trình này thường được thực hiện bởi các bộ lọc. Bộ lọc trung bình được sử dụng chính trong bước tiền xử lý ảnh này, nó là bộ lọc mà mỗi điểm ảnh được thay thế bằng trung bình trọng số của các điểm lân cận nhằm loại bỏ những điểm ảnh biến đổi lớn hơn so với điểm lân cận (hay còn gọi là nhiễu). [4]

### 2.1.3. Phân đoạn ảnh

Phân vùng ảnh là tách một ảnh đầu vào thành các vùng thành phần để biểu diễn phân tích, nhận dạng ảnh. Ví dụ: để nhận dạng chữ (hoặc mã vạch) trên phong bì thư cho mục đích phân loại bưu phẩm, cần chia các câu chữ về địa chỉ hoặc tên người thành các từ, các chữ, các số (hoặc các vạch) riêng biệt để nhận dạng. Đây là phần phức tạp khó khăn nhất trong xử lý ảnh và cũng dễ gây lỗi, làm mất độ chính xác của ảnh. Kết quả nhận dạng ảnh phụ thuộc rất nhiều vào công đoạn này. [4]

### 2.1.4. Biểu diễn và mô tả

Đây là phần sau phân đoạn chứa các điểm ảnh của vùng ảnh (ảnh đã phân đoạn) cộng với mã liên kết ở các vùng lân cận. Việc biến đổi các số liệu này thành dạng thích hợp là cần thiết cho xử lý tiếp theo bằng máy tính. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng( Feature Extraction) gắn với việc tách các đặc tính của ảnh dưới dạng các thông tin định lượng hoặc làm cơ sở để phân biệt lớp đối tượng này với đối tượng khác trong phạm vi ảnh nhận được. Ví dụ: trong nhận dạng ký tự trên phong bì thư, chúng ta miêu tả các đặc trưng của từng ký tự giúp nhận biết ký tự này với ký tự khác. [4]

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1.5. Nhận dạng và nội suy

Nhận dạng ảnh là quá trình xác định ảnh. Quá trình này thường thu được bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng. Ví dụ: một loạt nét gạch đứng gần nhau trên giấy có thể nội suy thành mã Barcode. Có nhiều cách phân loại ảnh khác nhau về ảnh. Theo lý thuyết về nhận dạng, các mô hình toán học về ảnh được phân theo hai loại nhận dạng ảnh cơ bản: Nhận dạng theo tham số, nhận dạng theo cấu trúc. Một số đối tượng nhận dạng khá phổ biến hiện nay được áp dụng trong khoa học và công nghệ là: nhận dạng ký tự (chữ in, chữ viết tay, chữ ký điện tử), nhận dạng văn bản (Text), nhận dạng vân tay, nhận dạng mã vạch, nhận dạng mặt người, ... [4]

## 2.2. Giới thiệu về trí tuệ nhân tạo AI (Artificial Intelligent)

### 2.2.1. Trí tuệ nhân tạo AI

Hiện nay công khoa học công nghệ phát triển rất nhanh. Một trong những ngành khoa học máy tính đang là xu hướng hiện nay chính là “trí tuệ nhân tạo AI” đây có thể là một bước đà của sự bùng nổ và khoa học công nghệ thế giới bằng cách chế tạo máy móc thông minh. AI là một trong những lĩnh vực hấp dẫn và phổ biến của khoa học máy tính có phạm vi rất lớn trong tương lai. AI nắm giữ xu hướng khiến một cỗ máy hoạt động như một con người.

Trí tuệ nhân tạo được xem như một nhánh của khoa học máy tính liên quan đến sự tự động hóa về hành vi thông minh. Trí tuệ nhân tạo là nghiên cứu về cách làm cho máy tính làm những việc mà hiện tại con người làm rất tốt. Vì vậy, chúng ta có thể định nghĩa AI là: "Đó là một nhánh của khoa học máy tính, qua đó chúng ta có thể tạo ra những cỗ máy thông minh có thể hành xử như con người, suy nghĩ như con người và có thể đưa ra quyết định." [6].

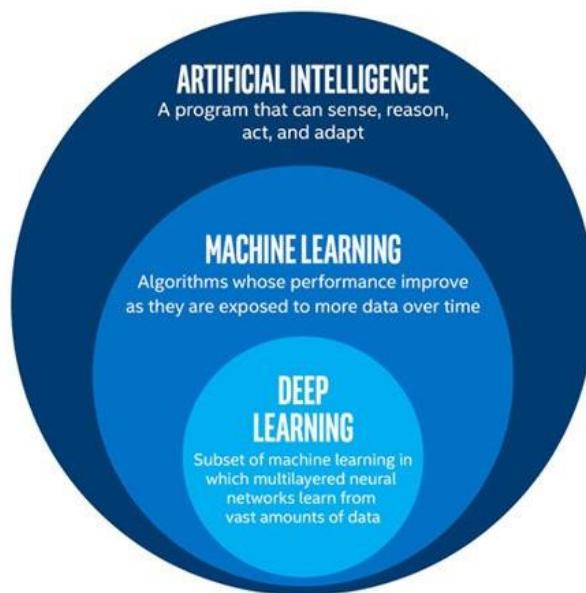
Giải thích thêm về mục tiêu của trí tuệ nhân tạo, các nhà khoa học đã chỉ ra bao gồm sáu mục tiêu sau:

- Lập luận logic: Cho phép máy tính thực hiện các nhiệm vụ đòi hỏi trí thông minh của con người như: chứng minh một định lý, lái xe trong giao thông, xác định đúng đối tượng.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- Biểu diễn tri thức: Cho phép máy tính mô tả các đối tượng, con người và ngôn ngữ.
- Lập kế hoạch và điều hướng: Cho phép máy tính đi từ điểm A đến điểm B.
- Xử lý ngôn ngữ: Cho phép máy tính hiểu và xử lý ngôn ngữ.
- Nhận thức: Cho phép máy tính tương tác với thế giới thông qua thị giác, thính giác, xúc giác và khứu giác.
- Tự học: Tạo một số hệ thống có thể thể hiện hành vi thông minh, tự học những điều mới, thể hiện, giải thích và có thể tư vấn cho người dùng.

Khi nói đến “trí tuệ nhân tạo” ta thường nghe một số thuật ngữ như AI, Machine learning, Deep learning những thuật ngữ trên có mối quan hệ được thể hiện ở hình sau:



Hình 2. 2: Mối quan hệ giữa AI, Machine Learning và Deep Learning.

Khi nói về 3 khái niệm trên ta có thể hiểu một cách đơn giản:

**Trí tuệ nhân tạo (AI):** một cỗ máy có thể bắt chước hành vi và tư duy của con người.

**Học máy (machine learning):** Một tính năng của AI, cho phép các chuyên gia đào tạo cho AI để nó nhận biết các mẫu dữ liệu và dự đoán.

**Học sâu (deep learning):** Một kỹ thuật nhỏ của machine learning, cho phép máy có thể tự đào tạo chính mình.

### 2.2.2. Học máy (Machine learning)

Học máy (Machine learning) là một tập con của trí tuệ nhân tạo (AI). Nếu trí tuệ nhân tạo AI nhằm mục đích là cho máy tính thông minh như con người. Thì Machine learning được định nghĩa "là khả năng để máy tự học mà không cần phải lập trình" theo Arthur Samuel vào năm 1959. Cụ thể hơn, machine learning đề cập tới bất kỳ hệ thống mà hiệu suất của máy tính khi thực hiện một nhiệm vụ sẽ trở nên tốt hơn sau khi hoàn thành nhiệm vụ đó nhiều lần. Hay nói cách khác, khả năng cơ bản nhất của machine learning là sử dụng thuật toán để phân tích những thông tin có sẵn, học hỏi từ nó rồi đưa ra quyết định hoặc dự đoán về một thứ gì đó có liên quan. Thay vì tạo ra một phần mềm với những hành động, hướng dẫn chi tiết để thực hiện một nhiệm vụ cụ thể, máy tính được "huấn luyện" bằng cách sử dụng lượng dữ liệu và các thuật toán để học cách thực hiện nhiệm vụ.

Nếu không có machine learning, AI hiện tại sẽ bị hạn chế khá nhiều bởi nó mang lại cho máy tính sức mạnh để tìm ra mọi thứ mà không được lập trình rõ ràng. Ví dụ về một loại machine learning, trong đề tài của nhóm thực hiện muốn chương trình xác định được khuôn mặt trong các bức ảnh. Đầu tiên, phải cung cấp cho AI một tập hợp các đặc điểm để máy tính nhận dạng, ví dụ như hình dáng, kích thước,...của khuôn mặt A. Tiếp theo bạn cung cấp một số hình ảnh cho AI trong đó các hình ảnh được dán nhãn là tên của khuôn mặt A. Sau khi đã nhận đầy đủ dữ liệu nó phải biết cách tìm ra khuôn mặt A trong bức ảnh.

### 2.2.3. Học sâu (Deep Learning)

Deep learning như là một loại machine learning với "mạng thần kinh - neural networks" sâu có thể xử lý dữ liệu theo cách tương tự như một bộ não con người có thể thực hiện. Điểm khác biệt chính ở đây là con người không sẽ phải dạy một chương trình deep learning biết một khuôn mặt người trông như thế nào, mà chỉ cần cung cấp cho nó đủ hình ảnh cần thiết về khuôn mặt A, và nó sẽ tự mình hình dung, tự học. Các bước cần làm như sau: cung cấp cho máy tính rất nhiều hình ảnh về khuôn mặt A. Thuật toán sẽ kiểm tra xem các đặc điểm chi tiết chung giữa các bức ảnh. Mỗi bức ảnh sẽ được giải mã chi tiết dưới nhiều cấp độ, từ các hình dạng lớn, chung đến các ô nhỏ và nhỏ hơn nữa. Nếu một hình dạng hoặc các đường được lặp lại nhiều lần, thuật toán sẽ gắn nhãn nó như là một đặc tính quan trọng. Sau khi phân tích đủ hình ảnh cần thiết, thuật toán

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

giờ đây sẽ biết được các mẫu nào cung cấp bằng chứng rõ ràng nhất về khuôn mặt A và tất cả những gì con người phải làm chỉ là cung cấp các dữ liệu thô. Deep learning đòi hỏi rất nhiều dữ liệu đầu vào và sức mạnh tính toán hơn là machine learning.

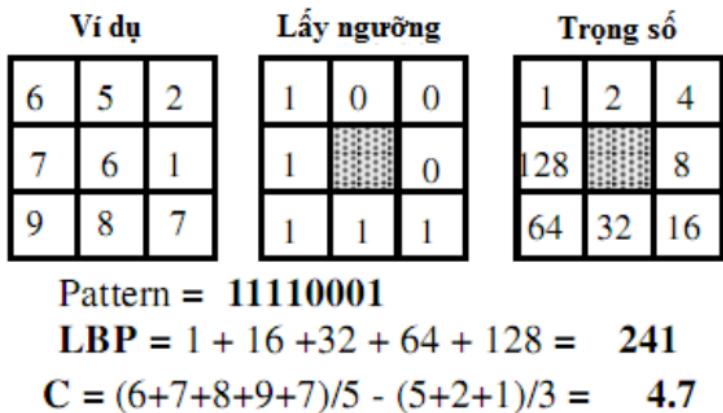
### 2.3. Thuật toán rút trích đặc trưng LBP (Local Binary Pattern)

#### 2.3.1. Lý thuyết về LBP

LBP là viết tắt của Local Binary Pattern hay là mẫu nhị phân địa phương được Ojala trình bày vào năm 1996 như là một cách đo độ tương phản cục bộ của ảnh. Phiên bản đầu tiên của LBP được dùng với 8 điểm ảnh xung quanh và sử dụng giá trị của điểm ảnh ở trung tâm làm ngưỡng. Giá trị LBP được xác định bằng cách nhân các giá trị ngưỡng với trọng số ứng với mỗi điểm ảnh sau đó cộng tổng lại. Hình dưới minh họa cách tính độ tương phản trực giao (C) là hiệu cấp độ xám trung bình của các điểm ảnh lớn hơn hoặc bằng ngưỡng với các điểm ảnh thấp hơn ngưỡng. [7]

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

Local binary pattern nó là một thuật toán mô tả texture (cấu trúc) của một image. Ý tưởng cơ bản của nó là mô phỏng lại cấu trúc cục bộ (local texture) của image bằng cách so sánh mỗi pixel với các pixel lân cận nó (neighbors). Ta sẽ đặt một pixel là trung tâm (center) và so sánh với các pixel lân cận với nó, nếu pixel trung tâm lớn hơn hoặc bằng pixel lân cận thì nó sẽ trả về giá trị 1, ngược lại 0. Ví dụ chúng ta lấy bán kính 8 pixel lân cận thì lbp sẽ có dạng 11001111, là một chuỗi nhị phân để đơn giản và dễ đọc hơn ta sẽ chuyển về dạng decimal 207.

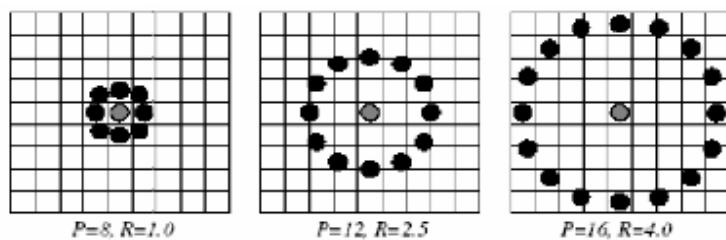


Hình 2. 3: Ví dụ về LBP và độ tương phản cục bộ C.

### Nguồn gốc:

Dãy LBP được Ojala trình bày vào năm 2002. Định nghĩa một cấu trúc điểm ảnh T là một phân phối đại số của cấp độ xám của  $P+1$  ( $P > 0$ ) điểm ảnh.  
 $T = t(gc, g0, \dots, gp-1)$

Với  $gc$  ứng với cấp độ xám của điểm ảnh trung tâm  $P_{tt}$ ,  $gp$  ( $p = 0, \dots, 1$ ) tương ứng với  $P$  điểm ảnh xung quanh,  $P$  điểm ảnh này nằm trên đường tròn bán kính  $R$  và tâm là  $P_{tt}$ .



Hình 2. 4: Tập hợp các điểm xung quanh  $P_{tt}$ .

Không mất thông tin, có thể trừ  $gp$  đi một lượng là  $gc$

$$T = t(gc, g0 - gc, \dots, gp - gc)$$

Giả sử sự sai số giữa  $gp$  và  $gc$  là độc lập với  $gc$ , ta có thể nhân tử hóa  $gc$  như sau:

$$T = t(gc)t(g0 - gc, \dots, gp - 1 - gc)$$

$t(gc)$  biểu thị xu hướng độ sáng tối của cả bức ảnh nên không liên quan đến kết cấu của ảnh cục bộ do đó có thể bỏ qua

$$T \sim t((g0 - gc), \dots, (gp - 1 - gc))$$

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Mặc dù tính bất biến ngược với độ thay đổi tỷ lệ xám của điểm ảnh, sự khác biệt ảnh hưởng bởi tỷ lệ. Để thu được đặc điểm bất biến với bất kỳ một sự thay đổi nào của ảnh đen trắng (gray scale) chỉ quan tâm đến dấu của độ lệch:

$$T \sim t(s(g_0 - g_c), \dots, s(g_{p-1} - g_c))$$

Với  $s$  là hàm dấu:  $s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$

Trọng số  $2^p$  được dùng cho các hàm dấu,  $s(g_p - g_c)$  để chuyển sự khác biệt giữa các điểm ảnh bên cạnh về một giá trị duy nhất.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) * 2^p$$

Với  $P$  pixel thì có  $2^p$  giá trị LBPP,R trong khoảng  $[0, 2^p - 1]$  nhưng để đơn giản ta có thể chọn một số giá trị trong  $2^p$  giá trị ký hiệu là:  $LBP_{P,R}^{u^2}$

### 2.3.2. Thuật toán LBP

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

Bước 1: Xác định bán kính làm việc.

Bước 2: Tính giá trị LBP cho pixel ở trung tâm  $(x_c, y_c)$  khối ảnh dựa trên thông tin của các pixel lân cận:

Trong đó,  $(g_p)$  là giá trị grayscale của các pixel lân cận,  $(g_c)$  là giá trị grayscale của các trung tâm và  $(s)$  là hàm nhị phân được xác định như sau:  $s(z) = 1$  nếu giá trị  $z \geq 0$ .

Ví dụ:

47	51	65
62	70	70
80	83	78

-23	-19	-5
-8	*	0
10	13	8

0	0	0
0	*	1
1	1	1

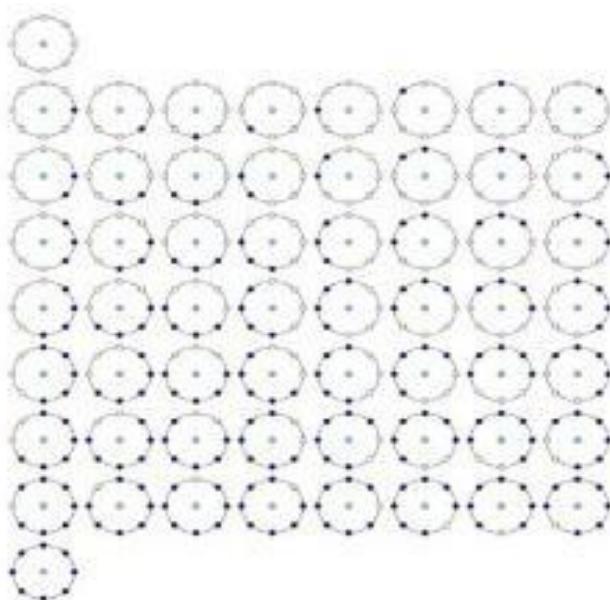
$$1*2^0 + 1*2^1 + 1*2^2 + 1*2^3 + 0*2^4 + 0*2^5 + 0*2^6 + 0*2^7 = 15.$$

Hình 2. 5: Các biến thể của LBP LBP đồng dạng.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Một mẫu nhị phân được gọi là đồng dạng khi xét chuỗi bit xoay vòng thì có nhiều nhất là 2 lần thay đổi (transitions) từ giá trị bit 0 sang 1 hoặc từ giá trị bit 1 sang 0. Ví dụ: 00000000 có 0 transitions, 01110000 có 2 transitions, 11001111 có 2 transitions nên đây là uniform LBP. 11001001 có 4 transitions, 01010011 có 6 transitions nên không phải là uniform LBP.

Dựa trên định nghĩa này, bảng ánh xạ cho bán kính làm việc P-neighbours sẽ có  $P(P-1) + 3$  nhãn. Có nghĩa là có 59 nhãn trong trường hợp làm việc với 8-neighbour. Hình vẽ sau đây thể hiện 59 nhãn (mẫu) và minh họa về histogram của đặc trưng LBP đồng dạng.



Hình 2. 6: Bảng thống kê các mẫu của uniform LBP

### 2.3.3. Nguyên lý phân lớp không tham biến

Trong phân lớp, sự khác biệt giữa mẫu và mô hình phân phối LBP được đánh giá bởi kiểm tra thống kê không tham biến. Phương pháp tiếp cận này có ưu điểm là không cần phải có những giả thiết về phân phối của các đặc trưng. Thông thường, những kiểm tra thống kê được chọn cho mục đích là nguyên lý crossentropy được giới thiệu bởi Kullback (1968). Sau đó, Sokal và Rohlf (1969) gọi cách đo này là thống kê G.

$$G(S, M) = 2 * \sum_{b=1}^B Sb \log \frac{Sb}{Mb} = 2 \sum_{b=1}^B [Sb * \log Sb - Sb * \log Mb]$$

Với  $S$ ,  $M$  kí hiệu phân phối mẫu và mô hình mong muốn.  $S_b$  và  $M_b$  là xác suất để  $b$  thuộc vào phân phối mẫu hoặc mô hình.  $B$  là số phần tử trong phân phối. Thống kê  $G$  sử dụng trong phân lớp có thể viết lại như sau:

$$L(S, M) = - \sum_{b=1}^B S_b \log M_b$$

Kiến trúc mô hình có thể xem như xử lý ngẫu nhiên có đặc tính có thể xác định bởi phân phối LBP. Trong một phân lớp đơn giản, mỗi lớp được biểu diễn bởi một mô hình phân phối đơn giản  $M_i$ . Tương tự, một kiến trúc mẫu không xác định có thể miêu tả bởi phân phối  $S$ .  $L$  là một giả ma trận đo khả năng mẫu  $S$  có thể thuộc lớp  $i$ .

Lớp  $C$  của một mẫu không xác định có thể được xác định bởi luật “hàng xóm gần nhất”:  $C = \operatorname{argmin}_i L(S, M_i)$

Bên cạnh đó, một thống kê log-likelihood có thể xem như đơn vị đo sự khác biệt và có thể sử dụng để liên kết nhiều bộ phân lớp giống như bộ phân lớp k-NN hoặc self-organizing map (SOM). Log-likelihood đúng trong một số trường hợp nhưng không ổn định khi mà cỡ mẫu nhỏ. Trong trường hợp này Chi-square-distance thường cho kết quả tốt hơn:

$$X^2(S, M) = \sum_{b=1}^B \frac{(S_b - M_b)^2}{S_b + M_b}$$

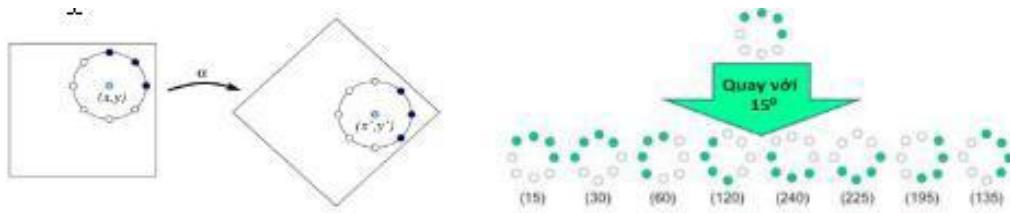
Để đạt được độ chính xác cao sử dụng giao histogram

$$H(S, M) = \sum_{b=1}^B \min\{S_b, M_b\}$$

#### 2.3.4. Phép quay bất biến

LBP bất biến với phép quay. Giả sử  $I_\alpha(x, y)$  là ảnh quay góc ( $\alpha$ ) của ảnh  $I(x, y)$ . Với phép quay này điểm ảnh  $(x, y)$  sẽ nằm tại vị trí  $(x', y')$  như hình vẽ sau đây (hình trái). Trong ví dụ này (hình phải): tất cả 8 mẫu LBP bên dưới được ánh xạ về mẫu LBP đầu tiên vì mẫu đầu tiên cho giá trị nhỏ nhất.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2. 7: Minh họa các trường hợp của LBP khi quay với góc 15 độ

LBP đồng dạng có khả năng bất biến với phép quay Kết hợp của mẫu LBP đồng dạng và LBP bất biến với phép quay có thể tạo nên một dạng biến thể khác của LBP (uniform rotation – invariant LBP). Dựa trên định nghĩa này, bảng ánh xạ cho bán kính làm việc P-neighbors sẽ có  $P + 2$  nhãn (label). Có nghĩa là có 10 nhãn trong trường hợp làm việc với 8-neighbour.

Để không bị ảnh hưởng bởi sự quay, mỗi giá trị LBP cần quay ngược lại về vị trí ban đầu, cách tốt nhất là tạo ra tất cả các trường hợp quay của một mẫu, sự quay có thể định nghĩa như sau:

$$LBP_{R,I}^{ri} = \min \{ ROR(LBP_{P,R}, i) \mid i=0,1,\dots,P-1 \}$$

Trong đó  $ri$  là viết tắt của rotation invariant (quay bất biến),  $ROR(x,i)$  dịch vòng tròn số nhị phân  $P$  - bit ( $x$ )  $i$  lần theo chiều kim đồng hồ.

### 2.3.5. Độ tương phản và kết cấu mẫu

Kết cấu có thể được coi là một hiện tượng hai chiều được đặc trưng bởi hai đặc tính trực giao: cấu trúc không gian (mô hình) và độ tương phản (độ mạnh của mô hình). Quay bất biến tương phản địa phương có thể được đo trong một hình tròn đối xứng xung quanh giống như LBP:

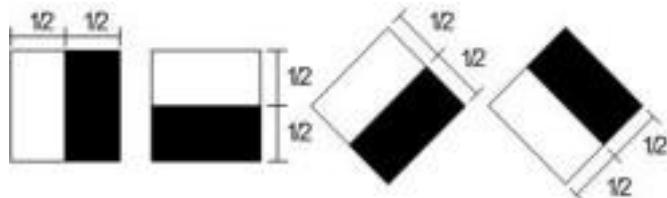
$$\text{VAR}_{P,R} = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - \mu)^2$$

$$\text{Trong đó: } \mu = \frac{1}{P} \sum_{p=0}^{P-1} g_p$$

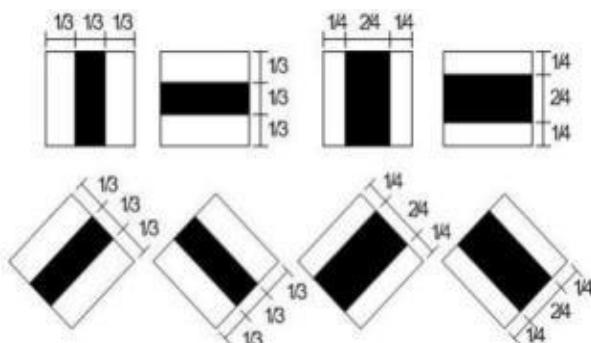
Tổng hợp lại ta có :  $LBP_{P1,R1}^{ri} / \text{VAR}_{P2,R2}$

## 2.4. Đặc trưng Haar – Like

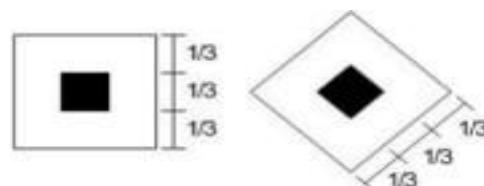
Đặc trưng Haar-like được sử dụng trong việc nhận dạng đối tượng trong ảnh số được phát biểu bởi Viola và Jones gồm 4 đặc trưng cơ bản để xác định một đối tượng trong ảnh. Mỗi đặc trưng Haar-like là sự kết hợp gồm 2 hoặc 3 khối chữ nhật mang giá trị “đen” hoặc “trắng”. Những khối chữ nhật này thể hiện sự liên hệ tương quan giữa các bộ phận trong ảnh mà bản thân từng giá trị pixel không thể diễn đạt được.[8]



Hình 2. 8: Đặc trưng theo cạnh



Hình 2. 9: Đặc trưng theo đường



Hình 2. 10: Đặc trưng theo xung quanh tâm



Hình 2. 11: Đặc trưng theo đường chéo

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Dựa theo các đặc trưng trên, giá trị của đặc trưng Haar-like được xây dựng bởi độ chênh lệch giữa tổng các pixel của các vùng đen so với tổng các pixel của các vùng trắng.

Dưới đây là công thức tính giá trị đặc trưng Haar-like:

$$f(x) = \sum_{\text{vùng đen}}(\text{pixel}) - \sum_{\text{vùng trắng}}(\text{pixel})$$

Như vậy để tính giá trị đặc trưng Haar-like cần phải thực hiện tính toán tổng các vùng pixel trên ảnh. Điều này làm cho chi phí bài toán lớn không thể đáp ứng các tính năng yêu cầu thời gian thực. Do vậy Viola và Jones đã đề xuất ra khái niệm “Integral Image” để giảm thiểu chi phí cho bài toán tính giá trị của đặc trưng Haar-like để bài toán có thể xử lý với thời gian thực. Tính “Integral Image” bằng cách sử dụng mảng 2 chiều với kích thước bằng kích thước của ảnh cần tính giá trị đặc trưng Haar-like. Ảnh chia nhỏ ở vị trí  $(x, y)$  được tính bằng tổng các giá trị pixel của vùng từ vị trí  $(0,0)$  đến vị trí  $(x-1, y-1)$ . Việc tính toán đơn giản là thực hiện phép cộng số nguyên nên tốc độ thực hiện được tối ưu hóa.



Hình 2. 12: Cách tính Integral Image của ảnh

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

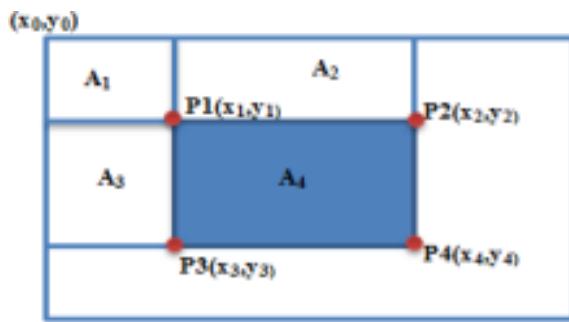
Kết quả có được sau khi tính Integral Image, việc tính tổng giá trị pixel trong vùng cần tính thực hiện như sau:

Gọi vùng cần tính tổng các giá trị pixel là vùng “ $A_4$ ”.

$$P_1(x_1, y_1) = A_1 \quad P_2(x_2, y_2) = A_1 + A_2$$

$$P_3(x_3, y_3) = A_1 + A_3 \quad P_4(x_4, y_4) = A_1 + A_2 + A_3 + A_4$$

$$\text{Ta được: } A_4 = P_4 + P_1 - P_2 - P_3$$



Hình 2. 13: Cách tính tổng giá trị pixel vùng cần tính

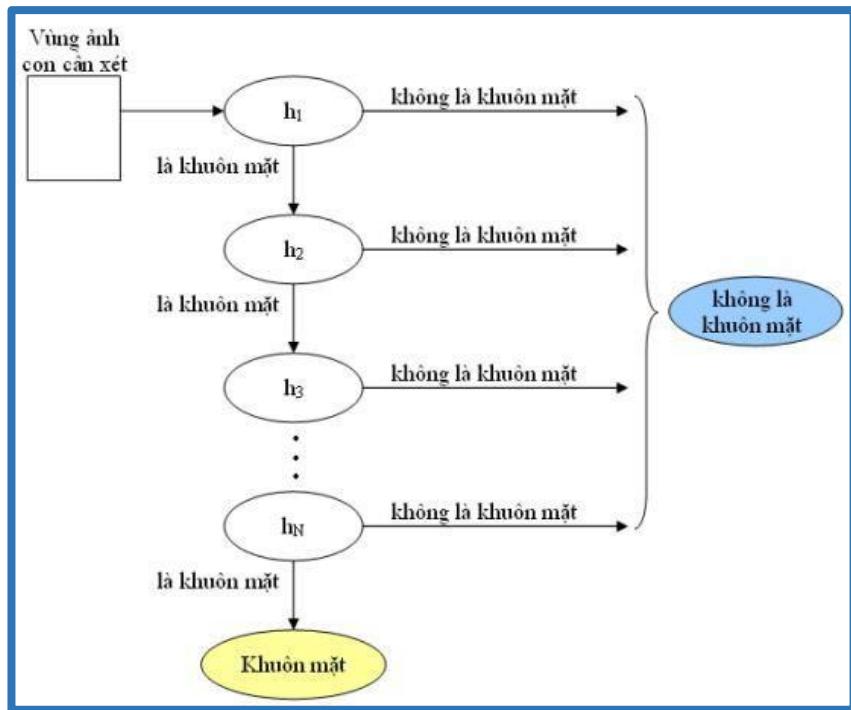
## 2.5 Giải thuật AdaBoost

AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1995. AdaBoost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các “weak classifiers” để hình thành một “strong classifier”.

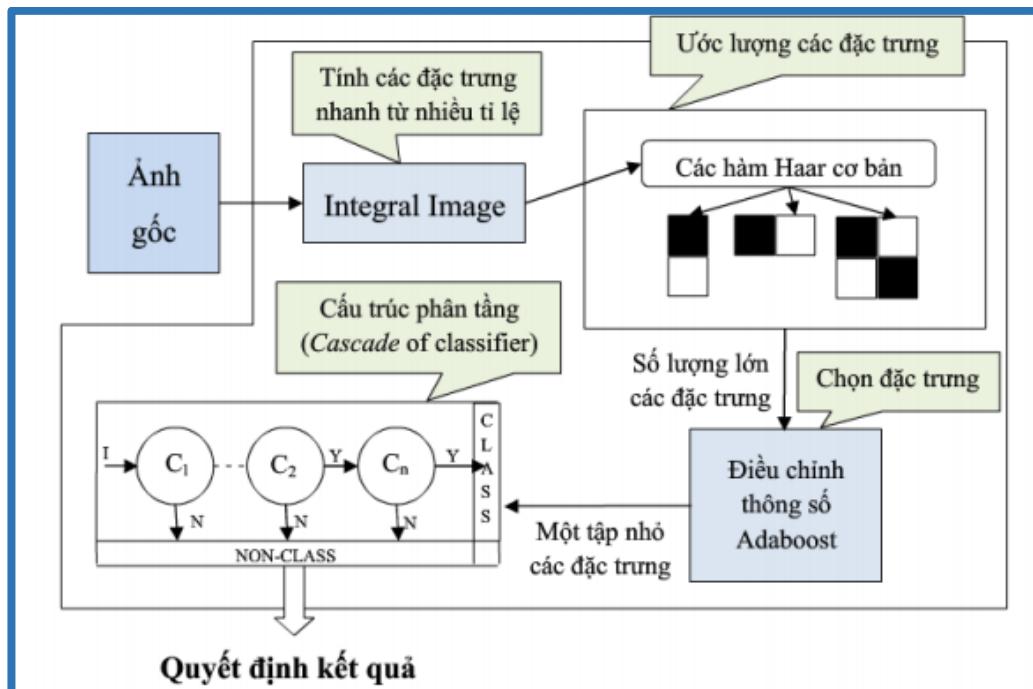
Là một cải tiến của tiếp cận boosting, AdaBoost sử dụng thêm khái niệm trọng số (weight) để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện, cứ mỗi “weak classifier” được xây dựng, thuật toán sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng “weak classifier” kế tiếp: tăng trọng số của các mẫu bị nhận dạng sai và giảm trọng số của các mẫu được nhận dạng đúng bởi “weak classifier” vừa xây dựng. Bằng cách này “weak classifier” sau có thể tập trung vào các mẫu mà các “weak classifiers” trước nó làm chưa tốt. Sau cùng, các “weak classifiers” sẽ được kết hợp tùy theo mức độ tốt của chúng để tạo nên “strong classifier”.

Viola và Jones dùng AdaBoost kết hợp các bộ phân loại yếu sử dụng các đặc trưng Haar-like theo mô hình phân tầng (cascade) như sau:

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2. 14: Mô hình phân tầng kết hợp các bộ phân loại yếu



Hình 2. 15: Sơ đồ nhận diện khuôn mặt

Từ ảnh gốc ban đầu, hệ thống sẽ chia ảnh thành nhiều tỉ lệ rất nhỏ để tính các đặc trưng qua các hàm đặc trưng cơ bản Haar-like. Sau khi tính toán được các hàm đặc trưng, ta nhận được một số lượng rất lớn các đặc trưng. Thì để lựa chọn và xác định có thể là khuôn mặt, ta dùng giải thuật Adaboost để lựa chọn và kết hợp các vùng trọng số yếu thành trọng số mạnh. Cuối cùng, các vùng đã được lựa chọn sẽ được đưa vào

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

bộ phân loại “Cascade of classifier” để loại bỏ các vùng không phải khuôn mặt và xác định vùng nào là khuôn mặt. Và đưa ra kết quả cuối cùng.

### 2.5. Ngôn ngữ lập trình Python

#### 2.5.1. Giới thiệu về ngôn ngữ lập trình Python[4]



Hình 2. 16: Ngôn ngữ lập trình Python

Python là ngôn ngữ lập trình hướng đối tượng bậc cao, dùng để phát triển website và nhiều ứng dụng khác nhau. Python được tạo ra bởi Guido van Rossum và được phát triển trong một dự án mã mở (open source).

Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

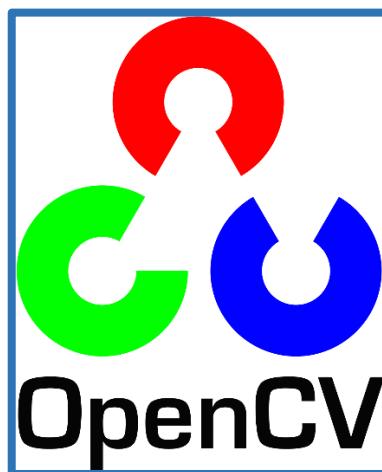
#### *Các tính năng chính của Python*

- ✓ **Ngôn ngữ lập trình đơn giản, dễ học:** Python có cú pháp đơn giản, rõ ràng. Dễ dàng đọc và viết hơn so với các ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, làm cho bạn tập trung hơn vào những giải pháp chứ không phải cú pháp.
- ✓ **Miễn phí, mã nguồn mở:** Bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

- ✓ **Khả năng di chuyển:** Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.
- ✓ **Khả năng mở rộng và có thể nhúng:** Bạn có thể kết hợp các phần code bằng C, C++ và những ngôn ngữ khác vào code Python
- ✓ **Ngôn ngữ thông dịch cấp cao:** Không giống như C/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.
- ✓ **Thư viện tiêu chuẩn lớn:** Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code
- ✓ **Hướng đối tượng:** Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

### 2.5.2. Giới thiệu thư viện OpenCV [4]



Hình 2. 17: Thư viện OpenCV

OpenCV (OpenSource Computer Vision) là một thư viện mã nguồn mở. OpenCV được phát hành theo giấy phép BSD, do đó nó hoàn toàn miễn phí cho cả học thuật và thương mại. Nó có các interface C++, C, Python, Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

OpenCV được thiết kế để tính toán hiệu quả và với sự tập trung nhiều vào các ứng dụng thời gian thực. Được viết bằng tối ưu hóa C/C++, thư viện có thể tận dụng lợi thế của xử lý đa lõi. Được sử dụng trên khắp thế giới, OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Phạm vi sử dụng từ nghệ thuật tương tác, cho đến lĩnh vực khai thác mỏ, bản đồ trên web hoặc công nghệ robot.

OpenCV đang được sử dụng rộng rãi trong các ứng dụng bao gồm:

- ✓ Nhận diện hình ảnh
- ✓ Kiểm tra và Giám sát tự động
- ✓ Robot và xe hơi tự lái
- ✓ Phân tích hình ảnh y tế
- ✓ Tìm kiếm và phục hồi Hình ảnh/Video
- ✓ Phim – cấu trúc 3D tự chuyển động
- ✓ Nghệ thuật sắp đặt tương tác

### 2.5.3. Giới thiệu thư viện Tkinter

Tkinter là một bộ công cụ widget đa nền tảng, mã nguồn mở cho phép các nhà phát triển tạo GUI cơ bản bằng cách sử dụng phát triển dựa trên widget. Tkinter ban đầu được phát hành vào năm 1991 với thư viện gốc được viết bằng C. Kể từ đó một số Language Binding đã được phát hành, bao gồm: Haskell, Ruby, Perl và Python.

Liên kết Python cho Tk được gọi là Tkinter và nó sẽ là thư viện mà nhóm thực hiện sẽ tập trung vào trong phần giới thiệu này. Tkinter kể từ đó đã trở thành tiêu chuẩn để tạo GUI bằng Python tuy nhiên có rất nhiều lựa chọn thay thế bao gồm: PyQt, PySide, PyGame, wxPython và PyGTK.

#### Các thuật ngữ:

- ✓ **Cửa sổ:** Khu vực hình chữ nhật chung được hiển thị trên màn hình của người dùng, tất cả các thành phần khác được đặt trong Cửa sổ.
- ✓ **Khung:** Một khu vực hình chữ nhật có thể được sử dụng để chia Cửa sổ cho các bộ cục phức tạp hơn, được coi là Widget cơ bản nhất.
- ✓ **Widget:** Một thuật ngữ chung cho bất kỳ khối xây dựng nào tạo nên GUI

#### 2.5.4. Đọc mã Barcode, QR Code

##### 2.5.4.1. Tổng quan mã vạch và mã QR code

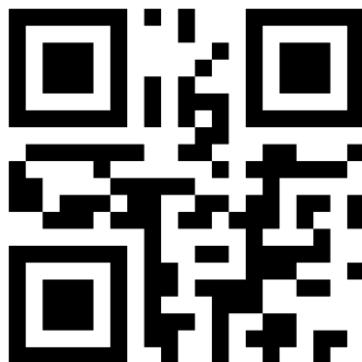
**Mã Barcode:** mã hóa, biểu diễn dữ liệu bằng các vạch đen và vạch trắng có độ rộng khác nhau tùy theo từng bảng mã.

Một số chuẩn phổ biến: UPC, Code 39, Code 128, EAN-8, EAN-13,...



Hình 2. 18: Mã Barcode với MSSV: 16141251

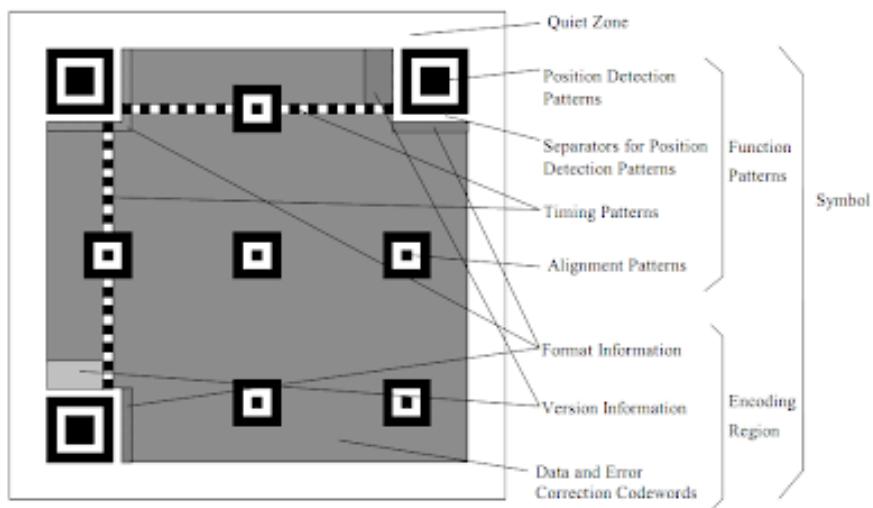
**Mã QR Code (Quick Response Code):** một dạng mã vạch hai chiều, có thể lưu trữ nhiều loại ký tự khác nhau (số, ký tự, ký tự trong bảng mã Kanji của tiếng Nhật...). Các bạn có thể xem đặc tả chi tiết của chuẩn mã QR code phổ biến nhất là chuẩn ISO18004. Mã QR code có nhiều ưu điểm vượt trội so với mã Barcode như khả năng lưu trữ được lượng dữ liệu lớn, khả năng tự sửa lỗi. [10]



Hình 2. 19: Mã QR Code với MSSV: 16141251

Format định dạng cho mã QR code với các phần thông tin quan trọng để dò tìm, nhận dạng, giải mã QR code. Ở đây có một số phần thông tin quan trọng:

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



Hình 2. 20: Format định dạng cho mã QR Code

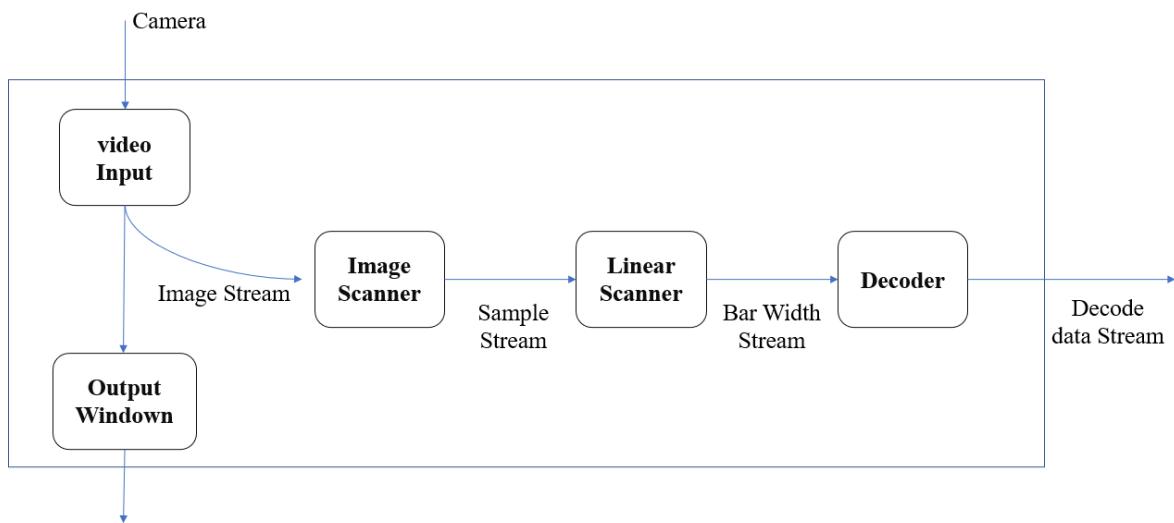
- ✓ **Position Detection Patterns:** giúp nhận diện vùng chứa mã QR code
- ✓ **Format Information & Version Information:** nhận diện phiên bản và format chuẩn để giải mã
- ✓ **Timing Patterns:** canh khung để tách các vùng dữ liệu và mã sửa lỗi
- ✓ **Alignment Patterns:** giúp căn chỉnh, hiệu chỉnh mã QR code trong các trường hợp bị xoay, méo...
- ✓ **Data and Error Correction Codewords:** chứa dữ liệu và các mã sửa lỗi

### 2.5.4.2. Giới thiệu thư viện Zbar

Để có thể giải mã Barcode và QR code chúng ta có thể sử dụng các thư viện mã nguồn mở, trong đó Zbar là một thư viện đa nền tảng, khá gọn nhẹ và cho kết quả nhận dạng tốt. Thư viện được xây dựng trên mã nguồn C/C++ nên dễ dàng porting lên trên các nền tảng khác nhau bao gồm cả iOS, Android và Linux.

Dưới đây là mô hình hệ thống của Zbar:

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT



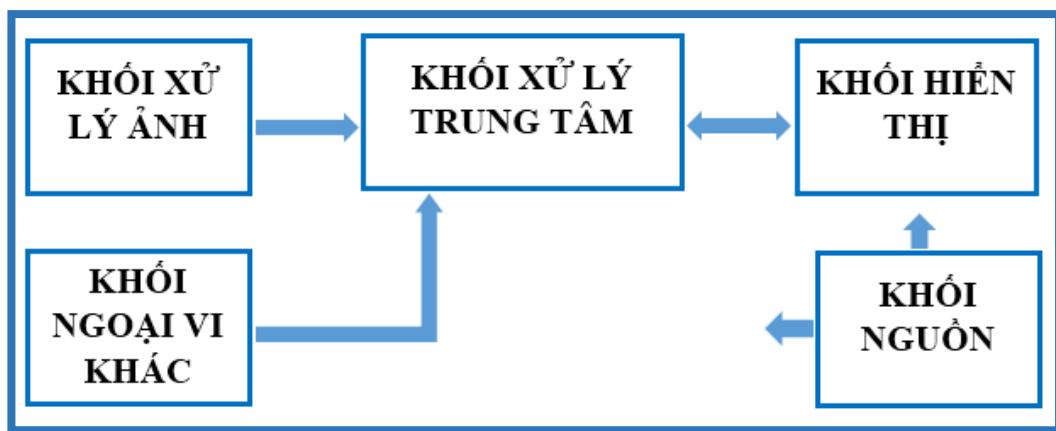
Hình 2. 21: Mô hình hệ thống Zbar

## CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

### 3.1. Giới thiệu

Để giảm thiểu sai sót và không xảy ra sự cố khi thi công mô hình theo như ý tưởng đã đề ra, ta bắt buộc phải tính toán và thiết kế. Với đề tài: “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện khuôn mặt kết hợp mã QR và mã vạch trên Raspberry Pi 4**” mục tiêu nhận dạng và điểm danh được khuôn mặt, thiết kế bao gồm những bước cụ thể như sau:

- Thiết kế sơ đồ khối của hệ thống.
- Tính toán, thiết kế từng khối của hệ thống:
  - + Khối nguồn.
  - + Khối Camera.
  - + Khối lập trình và xử lý dữ liệu đầu vào.
  - + Khối hiển thị.
  - + Các thiết bị ngoại vi khác: chuột, bàn phím,...



Hình 3. 1: Sơ đồ khối hệ thống

#### Chức năng từng khối:

**Khối nguồn:** Có nhiệm vụ cấp nguồn và dòng nuôi các khối còn lại, đảm bảo cho các khối hoạt động bình thường và không diễn ra hiện tượng sụt áp.

**Khối xử lý trung tâm:** Thực hiện nhiệm vụ nhận tín hiệu từ các khối khác, xử lý tín hiệu, hình ảnh và đưa ra các điều khiển theo chương trình được lập trình sẵn, truyền tới khối ngõ ra và hiển thị lên khối hiển thị.

**Khối hiển thị:** Hiển thị thông tin, giao tiếp với người sử dụng.

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

**Khối xử lý ảnh:** Có chức năng thu thập tín hiệu hình ảnh từ thực tế chuyển về tín hiệu và gửi dữ liệu cho khối xử lý trung tâm.

### 3.2. Chọn và giới thiệu từng loại linh kiện của từng khối

#### 3.2.1. Khối xử lý ảnh

- *Yêu cầu:*

Hệ thống nhận dạng khuôn mặt nên ta cần chọn camera nhận được ảnh màu và có độ phân giải cao.

Để đảm bảo chất lượng hình ảnh không thay đổi khi bị tác động bởi môi trường bên ngoài, bên trong khói xử lý ảnh cần có một nguồn sáng không đổi giúp việc thu ảnh và xử lý ảnh được hoạt động ổn định.

- *Lựa chọn thiết bị:*

Dựa vào yêu cầu trên ta chọn **Webcam Logitech C270**



Hình 3. 2: Hình ảnh Webcam Logitech C270

Webcam Logitech C270 giúp quay video với độ phân giải cao và sắc nét, phân giải video lên đến 1280 x 720 pixel. Ở tốc độ 30 khung hình/giây, chất lượng video trở nên mượt mà, trong khi hình ảnh rõ ràng, đầy màu sắc và có độ tương phản. Cảm biến hình ảnh với công nghệ RightLight cung cấp hình ảnh đạt chất lượng ngay cả khi quay trong bối cảnh mờ tối, C270 sẽ điều chỉnh với điều kiện ánh sáng để tạo ra hình ảnh có độ tương phản, tươi sáng hơn. Micrô tích hợp với công nghệ RightSound phục vụ mọi nhu cầu về chat voice. Webcam C270 nhỏ gọn, linh hoạt và có thể điều chỉnh góc quay dễ dàng giao tiếp với máy tính qua chuẩn giao tiếp USB với thiết lập đơn giản. [9]

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

- ❖ Thông số kỹ thuật:[10]
  - + Độ phân giải tối đa: 720p/30fps
  - + Loại tiêu cự: Lấy nét cố định
  - + Công nghệ thấu kính: tiêu chuẩn
  - + Micrô tích hợp: đơn âm
  - + Phạm vi quan sát: 60°
  - + Kẹp phổ dụng phù hợp với máy tính xách tay, LCD hoặc màn hình
  - + Độ dài dây cáp: 1.5m
- ❖ Phần mềm: Python 3.7 và thư viện OpenCV

### 3.2.2. Khối xử lý trung tâm

- *Yêu cầu:*

Hệ thống yêu cầu một bộ điều khiển đủ mạnh để vừa xử lý hình ảnh và chạy chương trình huấn luyện cho mạng nơ-ron, nhận dạng khuôn mặt một cách chính xác, lưu trữ cở sở dữ liệu. Yêu cầu trên đòi hỏi vi điều khiển phải có bộ nhớ lớn và tốc độ xử lý gần ngang ngửa với các dòng PC Laptop tầm trung hiện tại.

Hiện nay, trên thị trường Raspberry được khai thác rộng trong lĩnh vực điều khiển (giống một máy tính thu nhỏ) vì những ưu điểm sau:

- + Tốc độ xử lý nhanh, GPU mạnh.
- + Tích hợp với các module có sẵn như camera, màn hình.
- + Có thể hoạt động liên tục không ngừng nghỉ.
- + Dễ tiếp cận và làm việc.

- *Lựa chọn thiết bị:*

Dựa vào yêu cầu trên, ta chọn bộ điều khiển **Raspberry Pi 4 Model B** với những tính năng vượt trội hơn so với dòng raspberry cũ:

- + Tốc độ xử lý RAM nhanh gấp nhiều lần so với dòng cũ.
- + Tích hợp các module mới như: Ethernet, Audio, USB, HDMI,...
- + Hỗ trợ bộ nhớ cao giúp lưu trữ dữ liệu.
- + Giá thành rẻ và nhỏ gọn.
- + Tiêu thụ năng lượng thấp giúp tiết kiệm điện.

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

### \* **Tổng quan và Raspberry Pi:**

Raspberry Pi được phát triển đầu tiên vào năm 2012. Raspberry Pi ban đầu là như một thẻ card được cắm trên bo mạch máy tính được phát triển bởi các nhà phát triển ở Anh. Sau đó Raspberry Pi đã được phát triển thành một bo mạch đơn có chức năng như một máy tính mini dùng để giảng dạy ở các trường trung học. Được phát triển bởi Raspberry Pi Foundation – là tổ chức phi lợi nhuận với tiêu chí xây dựng hệ thống mà nhiều người có thể sử dụng được trong những công việc tùy biến khác nhau. Raspberry Pi sản xuất bởi 3 OEM: Sony, Qsida, Egoman. Và được phân phối chính bởi Element14, RS Components và Egoman.

Mặc dù chậm hơn so với các dòng laptop, máy tính hiện đại nhưng Raspberry Pi vẫn được xem là máy tính Linux hoàn chỉnh và có thể cung cấp tất cả các khả năng mà người dùng mong đợi, với mức tiêu thụ năng lượng thấp. Qua nhiều phiên bản, mỗi phiên bản đều có sự cải tiến hơn so với phiên bản trước. Đến nay Raspberry Pi Foundation đã ra mắt phiên bản mới nhất **Raspberry Pi 4 Model B** được tối ưu về mọi mặt với nhiều lựa chọn hấp dẫn cho người sử dụng.



Hình 3. 3: Raspberry Pi 4 model B thực tế

### \* **Giới thiệu về Raspberry Pi 4 Model B:**

Raspberry Pi 4 Model B là thế hệ thứ 4 và mới nhất tính đến thời điểm hiện tại của hệ thống Raspberry Pi, nó ra đời vào tháng 6 năm 2019. Cấu hình Raspberry Pi 4 có khá nhiều thay đổi. Raspberry Pi 4 gồm 10 phần chính:

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Chip SOC (System On Chip) Broadcom BCM 2711: chạy ở tốc độ 1.5 GHz, được nâng cấp hơn rất nhiều so với các thế hệ trước. Chip này tương đương với nhiều loại được sử dụng trong smartphone phổ thông hiện nay, và có thể chạy được hệ điều hành Linux. Tích hợp trên chip này là nhân đồ họa (GPU) Broadcom VideoCore VI 500MHz. GPU này đủ mạnh để có thể chơi một số game phổ thông và phát video chuẩn full HD hoặc 4K.

40 ngõ GPIO (General Purpose Input Output): 40 GPIO này tương tự như ở phiên bản Pi 3B. Các IO này của Raspberry Pi cũng được sử dụng để xuất tín hiệu ra led, thiết bị... hoặc đọc tín hiệu vào từ các nút nhấn, công tắc, cảm biến... Ngoài ra còn có các IO tích hợp các chuẩn truyền dữ liệu UART, I2C và SPI.

Cổng HDMI: Dùng để kết nối Pi với màn hình máy tính hay tivi. Pi 4 Model B trang bị hai cổng Micro HDMI.

Cổng DSI (Display Serial Interface): Cổng này dùng để kết nối với LCD hoặc màn hình OLED.

Ngõ audio 3.5mm: Kết nối dễ dàng với loa ngoài hay headphone. Đối với tivi có cổng HDMI, ngõ âm thanh được tích hợp theo đường tín hiệu HDMI nên không cần sử dụng ngõ audio này.

Cổng USB: Một khác biệt của Pi 4 là tích hợp 2 cổng USB 3.0 và 2 cổng USB 2.0. Để để cắm các ngoại vi cần thiết như chuột, bàn phím,...

Cổng Ethernet: Cho phép kết nối Internet dễ dàng. Cắm dây mạng vào Pi, kết nối với màn hình máy tính hay tivi và bàn phím, chuột là bạn có thể lướt web dễ dàng.

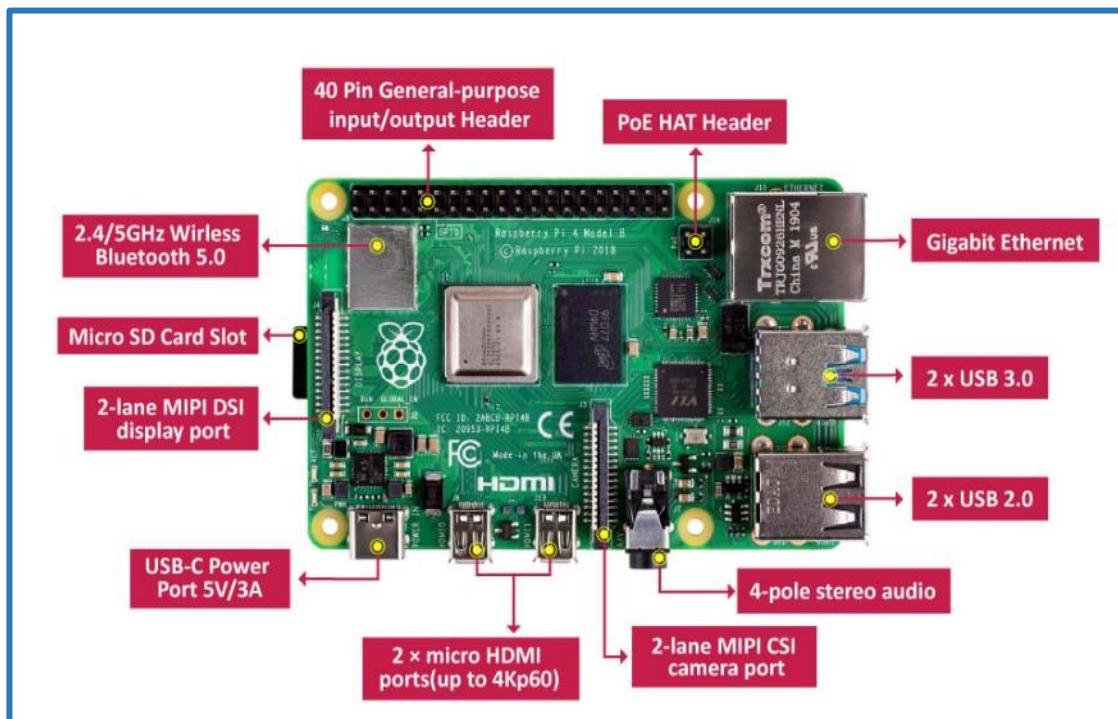
Khe cắm thẻ nhớ SD: Raspberry Pi không tích hợp ổ cứng. Thay vào đó nó có thể dùng thẻ SD để lưu trữ dữ liệu. Toàn bộ hệ điều hành Linux sẽ hoạt động trên thẻ SD này vì vậy nó cần kích thước thẻ nhớ tối thiểu 4GB.

Cổng CSI (Camera Serial Interface): Cổng này dùng để kết nối với module camera riêng của Raspberry Pi. Module này thu được hình ảnh chất lượng lên đến 1080p.

Jack nguồn micro USB 5V, 3A.

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Ngoài ra, Raspberry Pi 4 có đặc điểm nổi trội hơn các phiên bản trước là có tích hợp thêm wifi để có thể kết nối mạng internet không dây và bluetooth 5.0. RAM xử lý có thể lựa chọn 2GB, 4GB, 8GB, cấu hình RAM càng cao thì chi phí càng cao. Theo nhu cầu và chi phí của hệ thống nhóm thực hiện đã chọn dòng **Raspberry Pi 4 Model B, RAM 4GB.**



Hình 3. 4: Các cổng giao tiếp ngoại vi Raspberry Pi 4 Model B, RAM 4GB.

#### 3.2.3. Khởi hiển thị

- Yêu cầu:

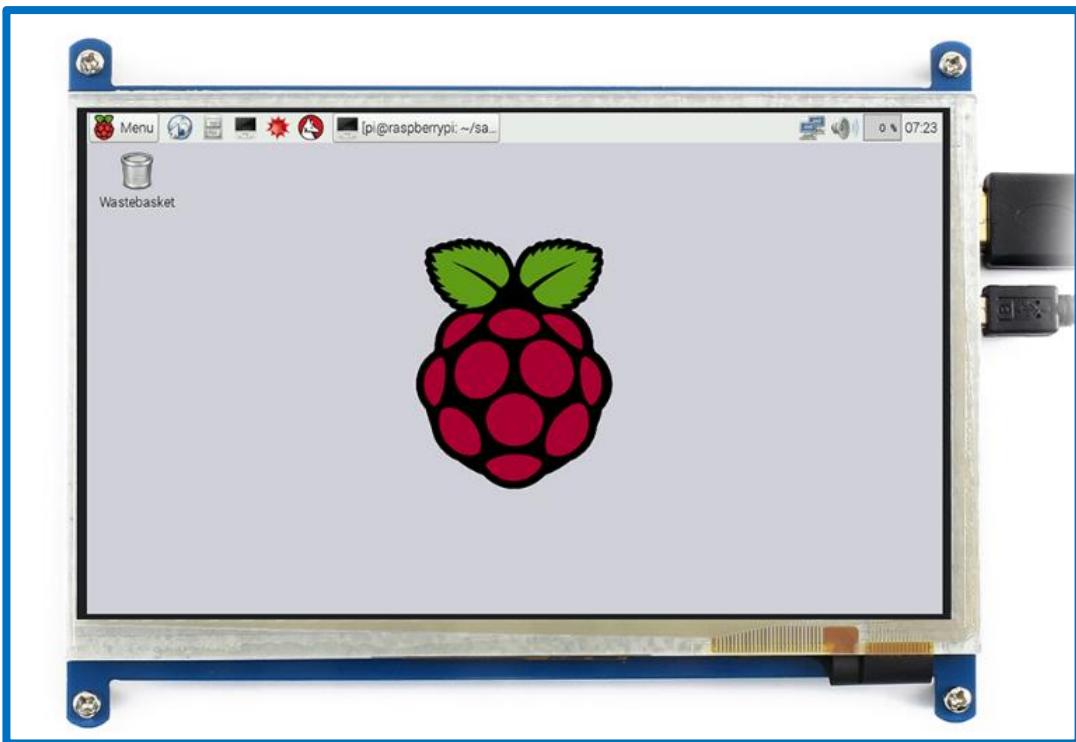
Đối với khói hiển thị, công việc nghiên cứu cần dễ dàng giao tiếp và lập trình với Raspberry, khói hiển thị màn hình đã tích hợp sẵn trong Raspberry. Để hệ thống trở nên tiện dụng và nhỏ gọn, khói hiển thị tích hợp thêm có thể thu nhận tín hiệu đưa và khói xử lý.

- Lựa chọn thiết bị:

Từ yêu cầu trên nhóm thực hiện đã quyết định chọn loại màn hình cảm ứng điện dung kích thước 7 inch. Màn hình LCD 7" HDMI công nghệ cảm ứng điện dung với tấm nền IPS nhập chính hãng từ nhà cung cấp linh kiện máy tính nổi tiếng thế giới

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

WaveShare. Phiên bản màn hình mới nhất với tấm nền IPS mang lại góc nhìn rộng hơn, rõ ràng hơn và tương thích với rất nhiều thiết bị phổ biến.



Hình 3. 5: LCD 7 inch HDMI(B) hảng WaveShare.

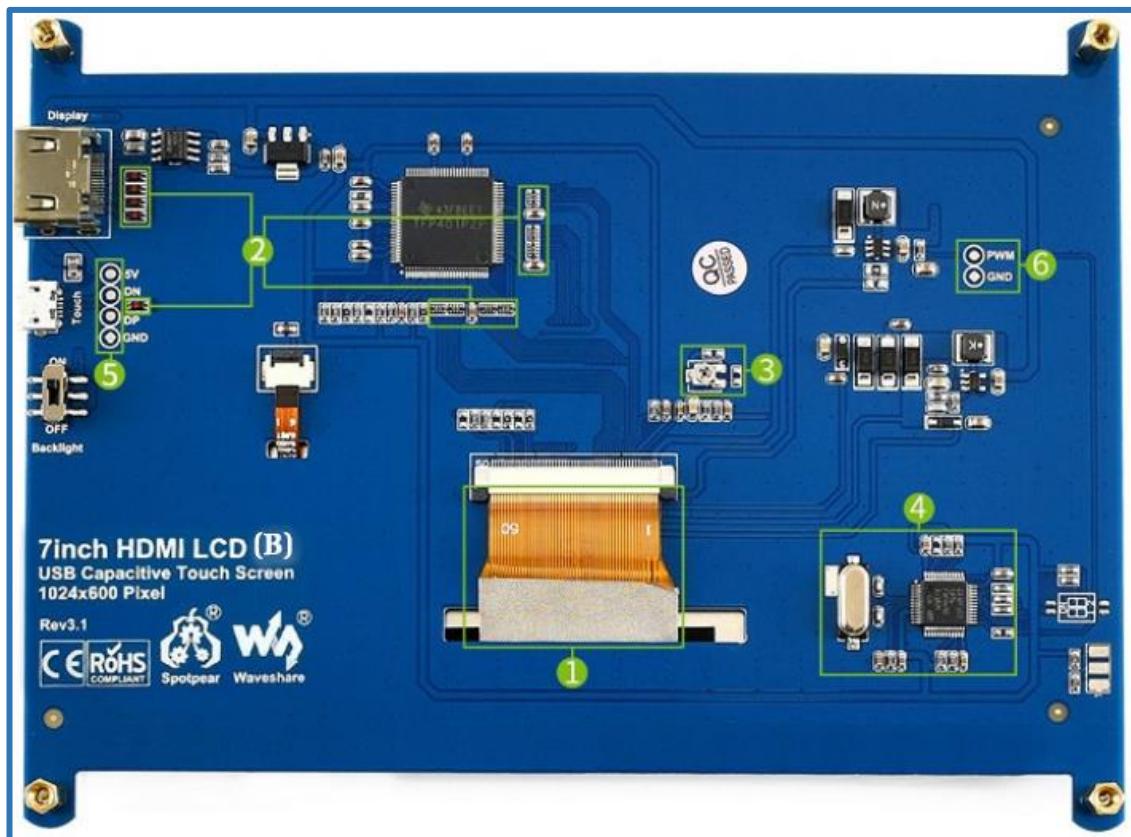
\* **Màn hình LCD 7" do WaveShare sản xuất có thông số kỹ thuật như sau:**

- Model: Waveshare 7 inch Capacitive Touch Screen LCD (B), 800×480, HDMI, Low Power.
- Kích thước màn hình: 7 inch.
- Độ phân giải:  $800 \times 480$  pixels.
- Cổng kết nối: HDMI, Touch.
- Điện áp sử dụng: 5VDC từ cổng Micro USB.
- Hỗ trợ cảm ứng điện dung và cấp nguồn 5VDC qua cổng Micro USB.
- Thiết kế Low Power tiết kiệm năng lượng.
- Raspberry Pi
  - Supports Raspbian, 5-points touch, driver free.
  - Supports Ubuntu / Kali / WIN10 IoT, single point touch, driver free.
  - Supports Retropie, driver free.
  - Supports all versions of Raspberry Pi.

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

- PC Supports : Windows 10 / 8.1 / 8 / 7, 5-points touch, driver free.
- Kích thước: 164.9 x 124.67mm

#### \* Cấu tạo chi tiết màn hình:



Hình 3. 6: Cấu tạo màn hình điện dung 7inch Waveshare

Số 1: Thiết kế chống nhiễu cáp LCD FFC, ổn định hơn cho ứng dụng công nghiệp.

Số 2: Thiết bị bảo vệ EMI và ESD, tuân thủ chứng nhận CE.

Số 3: Điều chỉnh điện áp VCOM để tối ưu hóa hiệu ứng hiển thị.

Số 4: Giao thức USB, chuyển đổi tín hiệu cảm ứng thành cảm ứng đa điểm để điều khiển cảm ứng đa điểm mượt mà .

Số 5: USB và đầu vào nguồn, dễ dàng kết nối nguồn điện bên ngoài.

Số 6: Miếng đệm điều chỉnh đèn nền cho đầu vào tín hiệu PWM, để kiểm soát độ sáng của đèn nền, tăng tuổi thọ của màn hình.

#### \* Kết nối giữa màn hình cảm ứng và Raspberry Pi 4 Model B:

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG



Hình 3. 7: Kết nối giữa màn hình cảm ứng và Raspberry Pi 4 Model B

Bước 1: Mở file config.txt trong phân vùng boot, có thể mở file này từ Windows thông qua đồ đọc thẻ nhớ.

Bước 2: Copy các dòng code sau dán vào cuối file config.txt:

```
max_usb_current=1  
hdmi_group=2  
hdmi_mode=87  
hdmi_cvt 800 480 60 6 0 0 0  
hdmi_drive=1
```

Bước 3: Save file config.txt, sau đó bỏ thẻ nhớ vào máy Raspberry Pi là đã có thể sử dụng.

Lưu ý: nên cấp nguồn cho Raspberry Pi bằng loại nguồn tốt, có dòng ra lớn hơn 2.5A khi sử dụng với màn hình, nguồn cấp vào màn hình được cấp thông qua cổng Micro USB chung với cảm ứng điện dung.

#### 3.2.4. Khởi nguồn

- Yêu cầu:

Cấp đủ nguồn và áp cho hệ thống. Khởi nguồn cần hoạt động ổn định để việc xử lý của các thiết bị được hoạt động chính xác, tránh sai sót không đáng có.

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Bảng 3.1: Bảng liệt kê dòng điện tiêu thụ và điện áp định mức của các linh kiện.

STT	Linh kiện	Số lượng	Dòng định mức	Điện áp định mức	Ghi chú
1	Raspberry Pi 4 Mode B	1	3 A	5V	Tổng dòng ngoại vi nhỏ hơn 1.2 (A)
2	Quạt tản nhiệt cho Pi 4	1	200 mA	5V	
3	Webcam Logitech C270	1	100 mA	5V	
4	Màn hình cảm ứng điện dung 7inch HDMI (B) LCD	1	500 mA	5V	

Như bảng trên ta thấy Raspberry Pi 4 Mode B cần cấp dòng 3A và tổng dòng ngoại vi kết nối phải nhỏ hơn 1.2A, 3 thiết bị kết nối trong hệ thống có tổng dòng là 0.8A thỏa điều kiện.

Nguồn Adapter Có Công Tắc Cho máy tính nhúng 5V/3A tích hợp công tắc nguồn trên dây cắm tiện dụng được sử dụng để cấp nguồn cho các board mạch máy tính nhúng, máy tính Mini có tốc độ xử lý và hiệu suất ngày càng cao hiện nay như: Raspberry Pi 3, Pi 4, Orange Pi, Nano Pi, ... nguồn có chất lượng linh kiện tốt, độ bền cao, là sự lựa chọn thích hợp cho các board mạch máy tính nhúng hiện nay.

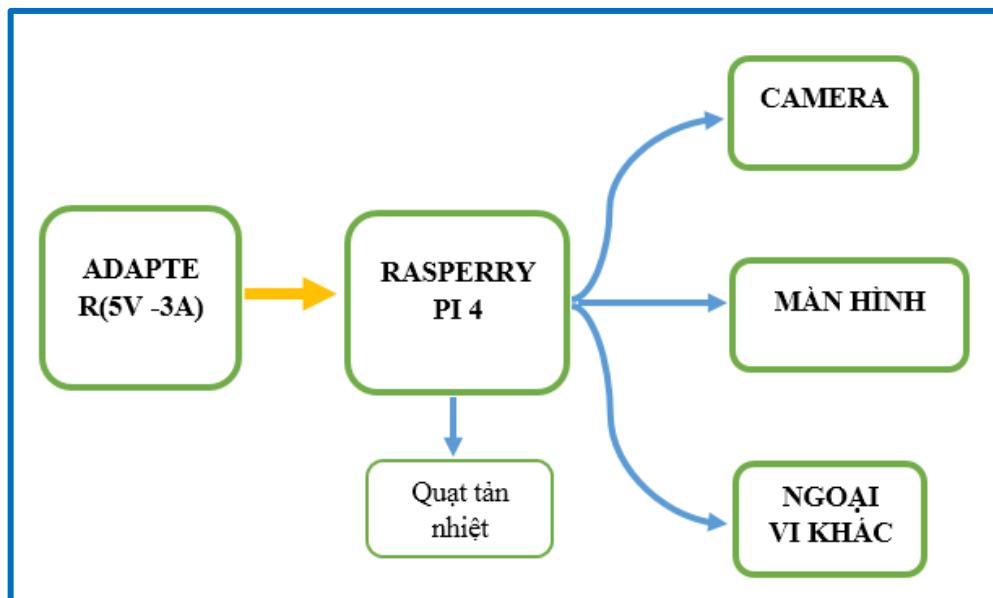


Hình 3.8: Adapter công tắc nguồn 5V/3A

**Thông số kỹ thuật:**

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

- Nguồn điện đầu vào: 100~240VAC 50/60Hz
- Nguồn điện đầu ra: 5VDC
- Dòng đầu ra tối đa: 3A
- Chuẩn đầu ra: Cáp Micro USB dài 1m.
- Tích hợp công tắc nguồn trên dây tiện dụng.



Hình 3. 9: Sơ đồ cấp nguồn hệ thống

Hệ thống mà nhóm thực hiện sẽ cấp nguồn trực tiếp cho Raspberry Pi 4 thông qua Adapter công tắc 5V/3A. Từ các cổng USB của PI sẽ cấp nguồn cho Camera, màn hình, các ngoại vi khác như chuột, bàn phím (nếu cần). Từ các port của PI sẽ cấp nguồn hoạt động cho quạt tản nhiệt.

### 3.3. Cài đặt hệ điều hành cho Raspberry Pi 4

#### 3.3.1. Thiết bị cần để cài đặt

- Thẻ nhớ Micro SD dung lượng 32GB, class 10.
- Máy tính cá nhân.
- Đầu đọc thẻ nhớ Micro SD.
- Nguồn adapter 5V/3A.

#### 3.3.2 Cài đặt hệ điều hành Raspbian

**Bước 1:** Tải bản cài đặt về máy tính và giải nén.

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Chúng ta vào trang [www.raspberrypi.org](http://www.raspberrypi.org) và tải phiên bản mới nhất đó là hệ điều hành Raspbian Buster.

### Raspberry Pi Desktop

Compatible with:  
PC and Mac



#### Debian Buster with Raspberry Pi Desktop

Release date: December 11th 2020  
Kernel version: 4.19  
Size: 3.021MB  
[Show SHA256 file integrity hash:](#)

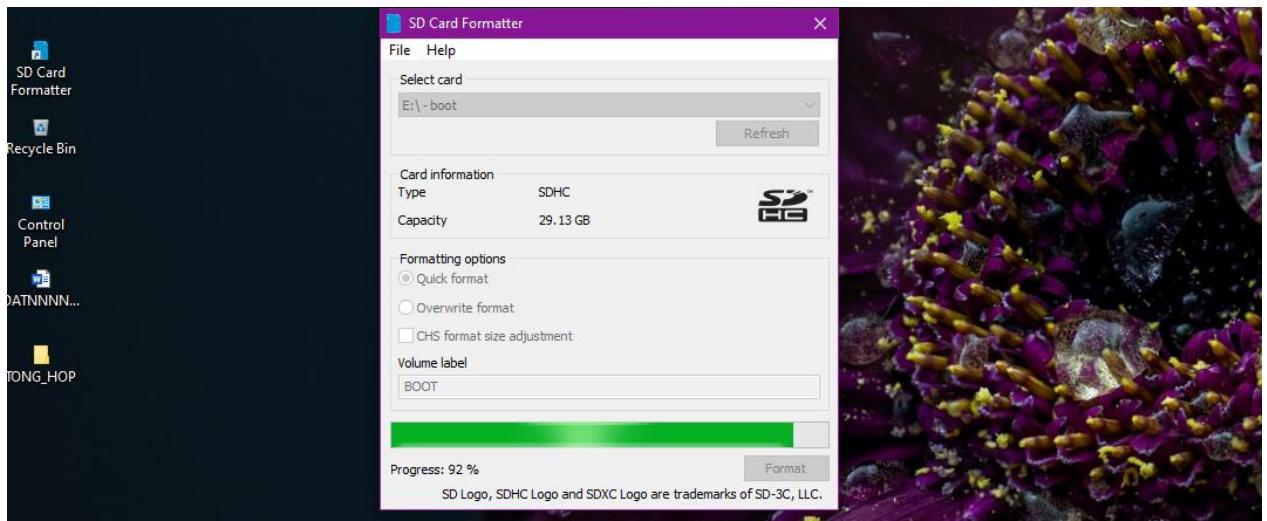
[Download](#)

[Download torrent](#)

Hình 3. 10: Tải phiên bản Raspbian Buster mới nhất.

**Bước 2:** Tải phần mềm SD Card Formatter và cài đặt.

Chúng ta dùng phần mềm SD Card Formatter để định dạng thẻ nhớ SD. Sau khi đã cài đặt hoàn thành, ta mở phần mềm và định dạng thẻ



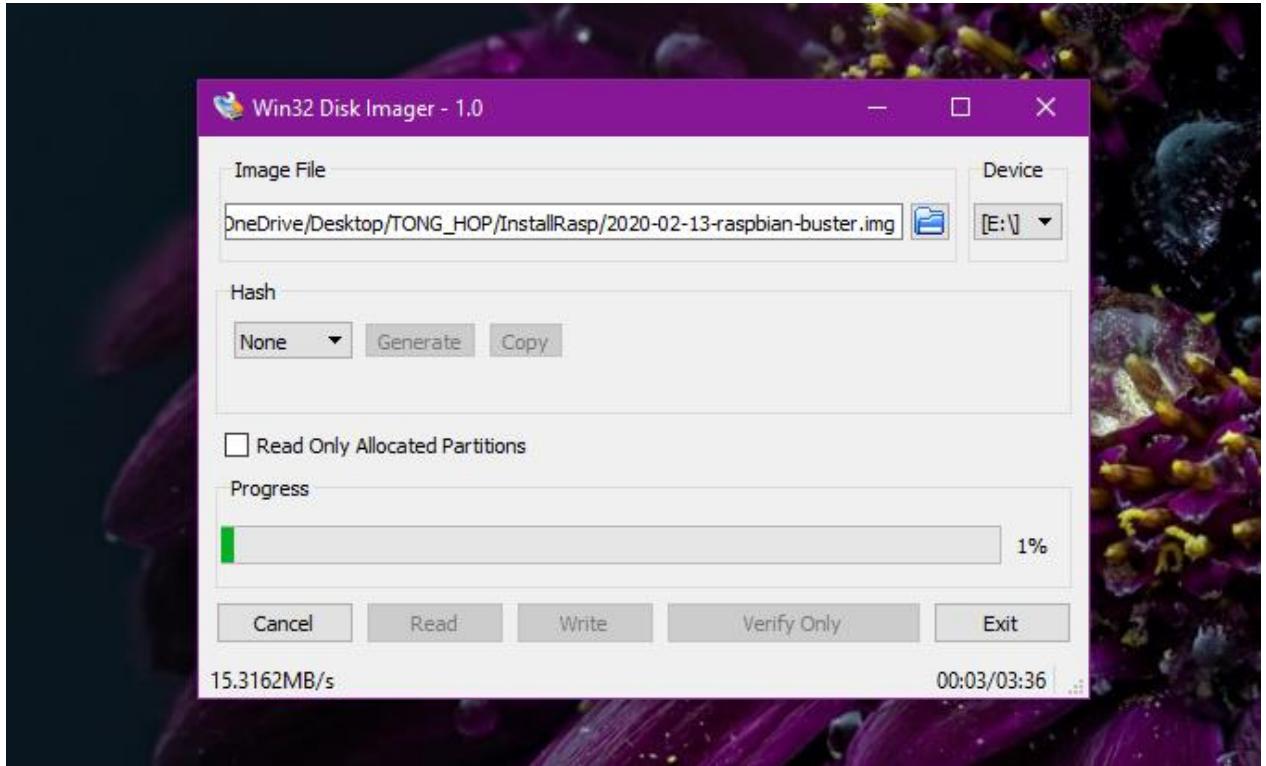
Hình 3. 11: Cài đặt phần mềm SD Card Formatter.

**Bước 3:** Tải phần mềm win32diskimager và cài đặt.

Sau khi tải và cài đặt thành công phần mềm win32diskimager. Chúng ta mở phần mềm, chọn đường dẫn file chứa hệ điều hành Raspbian Buster và chọn Write. Chờ báo

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

hoàn thành là đã cài đặt xong hệ điều hành vào thẻ nhớ. Sau đó gắn thẻ nhớ vào Raspberry Pi4 để khởi động hệ điều hành.



Hình 3. 12: Phần mềm Win32DiskImager.

#### 3.4. Cài đặt thư viện cho Raspberry Pi 4

##### 3.4.1. Cài đặt thư viện OpenCV

**Bước 1:** Mở rộng hệ thống tệp tin

```
$ sudo raspi-config
```

```
$ sudo reboot
```

Sau khi khởi động lại, hệ thống tệp tin cần được mở rộng bao gồm tất cả các không gian có sẵn trên thẻ nhớ micro-SD. Chúng ta xác minh rằng đĩa đã được mở rộng bằng cách thực hiện df-h và kiểm tra đầu ra.

OpenCV, cùng với tất cả các phụ thuộc của nó, sẽ cần một vài gigabyte trong quá trình biên dịch, vì vậy nên xóa các công cụ Wolfram để giải phóng một số không gian trên Pi:

```
$ Sudo apt-get purge wolfram-engine
```

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Sau khi gỡ bỏ Wolfram Engine, có thể phục hồi gần 700MB.

### Bước 2: Những cài đặt ràng buộc

Cập nhật hệ thống:

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

Sau đó chúng ta cần phải cài đặt một số công cụ phát triển, bao gồm cả CMake, giúp cấu hình quá trình xây dựng OpenCV:

```
$ sudo apt-get install build-essential cmake pkg-config
```

Tiếp theo, chúng ta cần phải cài đặt một số gói ảnh I/O cho phép chúng ta tải các định dạng tệp hình ảnh khác nhau từ đĩa. Ví dụ về định dạng tệp như JPEG, PNG, TIFF, vv...:

```
$ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
```

Cũng như các gói I/O hình ảnh, chúng ta cũng cần các gói I/O video. Các thư viện này cho phép đọc các định dạng tệp video khác nhau từ đĩa cũng như làm việc trực tiếp với các luồng video:

```
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

```
$ sudo apt-get install libxvidcore-dev libx264-dev
```

Thư viện OpenCV đi kèm với một mô đun phụ có tên highgui được sử dụng để hiển thị hình ảnh lên màn hình và xây dựng các GUI cơ bản. Để biên dịch mô đun highgui, cần phải cài đặt thư viện phát triển GTK:

```
$ sudo apt-get install libgtk2.0-dev
```

Nhiều hoạt động bên trong OpenCV (cụ thể là hoạt động ma trận) có thể được tối ưu hóa hơn nữa bằng cách cài đặt một vài phụ thuộc:

```
$ sudo apt-get install libatlas-base-dev gfortran
```

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Những thư viện tối ưu hóa này rất quan trọng đối với các thiết bị hạn chế nguồn lực như Raspberry Pi.

Cuối cùng, cài đặt Python 3.7 để biên dịch OpenCV với các ràng buộc Python:

```
$ sudo apt-get install python3.7-dev
```

#### Bước 3: Tải mã nguồn OpenCV

Bây giờ chúng ta cài đặt OpenCV 3.1.0 và giải nén:

```
$ cd ~
```

```
$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip
```

```
$ unzip opencv.zip
```

Chúng ta sẽ cài đặt đầy đủ OpenCV 3 (để có quyền truy cập vào các tính năng như SIFT và SURF), vì vậy chúng ta cũng cần lấy tập tin chứa opencv\_contrib:

```
$ wget -O opencv_contrib.zip
```

```
https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip
```

```
$ unzip opencv_contrib.zip
```

#### Bước 4: Python 3.7

Trước khi biên dịch OpenCV trên Raspberry Pi 4, chúng ta cần phải cài đặt pip, một trình quản lý gói Python:

```
$ wget https://bootstrap.pypa.io/get-pip.py
```

```
$ sudo python get-pip.py
```

Cài đặt môi trường ảo. Môi trường ảo là một công cụ đặc biệt được sử dụng để giữ các yêu cầu phụ thuộc vào các dự án khác nhau ở các vị trí riêng biệt bằng cách tạo ra các môi trường Python độc lập, độc lập cho mỗi môi trường ảo nên sử dụng môi trường ảo sẽ tiện lợi và cần thiết.

```
$ sudo pip install virtualenv virtualenvwrapper
```

```
$ sudo rm -rf ~/.cache /pip
```

## CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Mở một thiết bị đầu cuối mới

```
$ source ~ / .profile
```

Tiếp theo, chúng ta hãy tạo ra môi trường ảo Python mà chúng ta sẽ sử dụng để phát triển tầm nhìn máy tính:

```
$ mkvirtualenv cv -p python2
```

Lệnh này sẽ tạo một môi trường ảo Python mới có tên cv bằng Python 3.7.

Môi trường ảo CV Python hoàn toàn độc lập và bị cô lập từ phiên bản Python mặc định được tải xuống của Raspbian Buster. Bất kỳ gói Python nào trong thư mục các gói trang web toàn cầu sẽ không có sẵn cho môi trường ảo CV. Tương tự như vậy, bất kỳ gói Python nào được cài đặt trong gói trang web của cv sẽ không có sẵn cho việc cài đặt toàn cầu của Python. Điều này sẽ giúp chúng ta tránh nhầm lẫn.

Cài đặt NumPy trên Raspberry Pi

Cài đặt NumPy trong môi trường ảo cv:

```
$ pip install numpy
```

**Bước 5:** Biên dịch và cài đặt OpenCV

Một khi đảm bảo rằng đang ở trong môi trường ảo cv, chúng ta thiết lập xây dựng bằng cách sử dụng CMake:

```
$ cd ~ / opencv-3.1.0 /
```

```
$ mkdir build
```

```
$ cd build
```

```
$ cmake -D CMAKE_BUILD_TYPE = RELEASE \
-D CMAKE_INSTALL_PREFIX = /usr/local \
-D INSTALL_PYTHON_EXAMPLES = ON \
-D OPENCV_EXTRA_MODULES_PATH = ~ /opencv_contrib-3.1.0 /modules \
-D BUILD_EXAMPLES = ON ..
```

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Cuối cùng, chúng ta biên dịch OpenCV:

```
$ make -j4
```

- D CMAKE\_INSTALL\_PREFIX = /usr/local \
- D INSTALL\_PYTHON\_EXAMPLES = ON \
- D OPENCV\_EXTRA\_MODULES\_PATH = ~ /opencv\_contrib-3.1.0 /modules \
- D BUILD\_EXAMPLES = ON ..

Cuối cùng, chúng ta biên dịch OpenCV:

```
$ make -j4
```

Bước 5 kết thúc mà không có lỗi, OpenCV nên được cài đặt trong /usr/local/lib/python2.7/site-pacakges, xác minh điều này bằng lệnh ls:

```
$ ls -l /usr/local/lib/python2.7/site-packages/  
total 1852  
-rw-r--r-- 1 root staff 1895772 Mar 20 20:00 cv2.so
```

Cuối cùng là liên kết OpenCV vào môi trường ảo cv cho Python 3.7:

```
$ cd ~/.virtualenvs/cv/lib/python2.7/site-packages/  
$ ln -s /usr/local/lib/python2.7/site-packages/cv2.so cv2.so
```

**Bước 6:** Kiểm tra cài đặt OpenCV 3

Mở một thiết bị đầu cuối mới, thực hiện lệnh nguồn và workon, và cuối cùng là cố gắng nhập các ràng buộc Python + OpenCV:

```
$ workon cv
```

```
$ python
```

```
>>> import cv2
```

```
>>> version cv2 ._____
```

```
'3.1.0'
```

>>>

### 3.4.2. Cài đặt thư viện Pyzbar

Đầu tiên, chúng ta cần Download mã nguồn thư viện zbar tại mã nguồn:

<http://sourceforge.net/projects/zbar/files/zbar/0.10/zbar-0.10.tar.bz2/download>

(Phiên bản hiện tại là 0.10)

Tiếp theo, chúng ta tiến hành giải nén file mã nguồn, chuyển đường dẫn tới thư mục chứa mã nguồn và tiến hành biên dịch thư viện này:

- ✓ Biên dịch trên Desktop:

```
./configure --without-qt --without-gtk --without-python --without-
imagemagick
make
sudo make install
```

- ✓ Biên dịch trên Kit:

```
./configure --without-qt --without-gtk --without-python --without-
imagemagick CC=arm-linux-gcc --host=arm-linux --prefix=/opt/zbar.arm -
-with-x=no
make
sudo make install
```

Sau khi biên dịch, copy các file trong thư mục /opt/zbar.arm/include xuống thư mục /usr/include và /opt/zbar.arm/lib xuống thư mục /usr/lib ở trên KIT.

### 3.4.3. Cài đặt thư viện Pandas

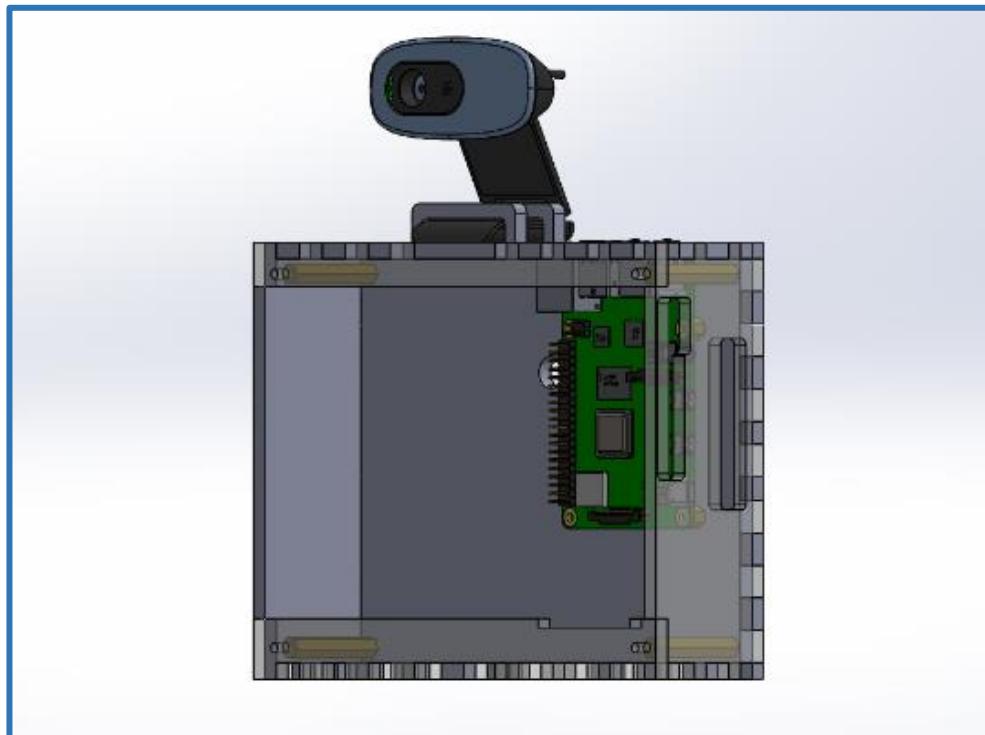
Để cài đặt thư viện Pandas:

- ✓ Đối với Window: Chúng ta vào Command gõ: `pip install pandas`
- ✓ Đối với Raspian: Chúng ta vào Terminal gõ: `sudo apt-get install pandas`

Sau khi cài đặt xong, trong Python, chúng ta cần khai báo `import pandas` để có thể bắt đầu sử dụng các hàm của pandas. Vì pandas là thư viện được sử dụng thường xuyên nên nó thường được khai báo gọn lại thành `pd` `import pandas as pd` `pd` có thể thay thế bằng các từ khác, tuy nhiên bạn nên đặt là `pd` vì các tài liệu hướng dẫn đều ngầm quy ước như thế.

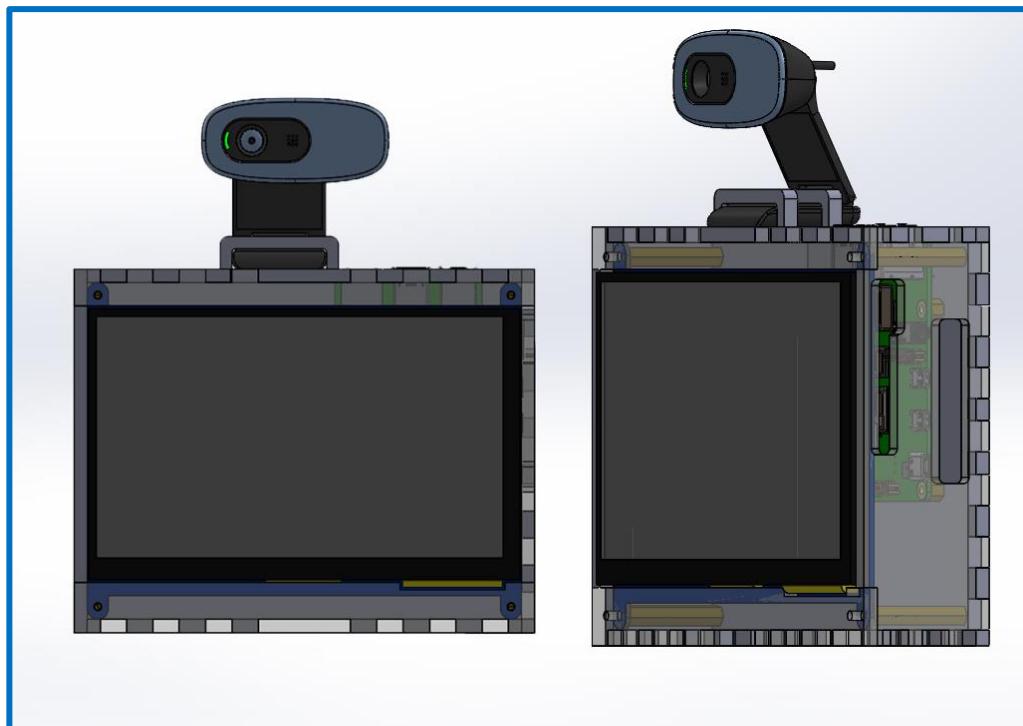
#### 3.4.4. Thiết kế hộp chứa linh kiện hệ thống

Hệ thống điểm danh sinh viên dùng Raspberry Pi 4 gồm các thiết bị chính như Pi 4, màn hình cảm ứng, camera, các thiết bị ngoại vi khác. Để các thiết bị cố định, dễ kết nối, hệ thống đạt tính thẩm mỹ. Nhóm thực hiện đã đưa ra ý tưởng thiết kế hộp chứa và kết nối các thiết bị kèm theo giá đỡ duy chuyển linh động, tiện lợi cho người dùng.

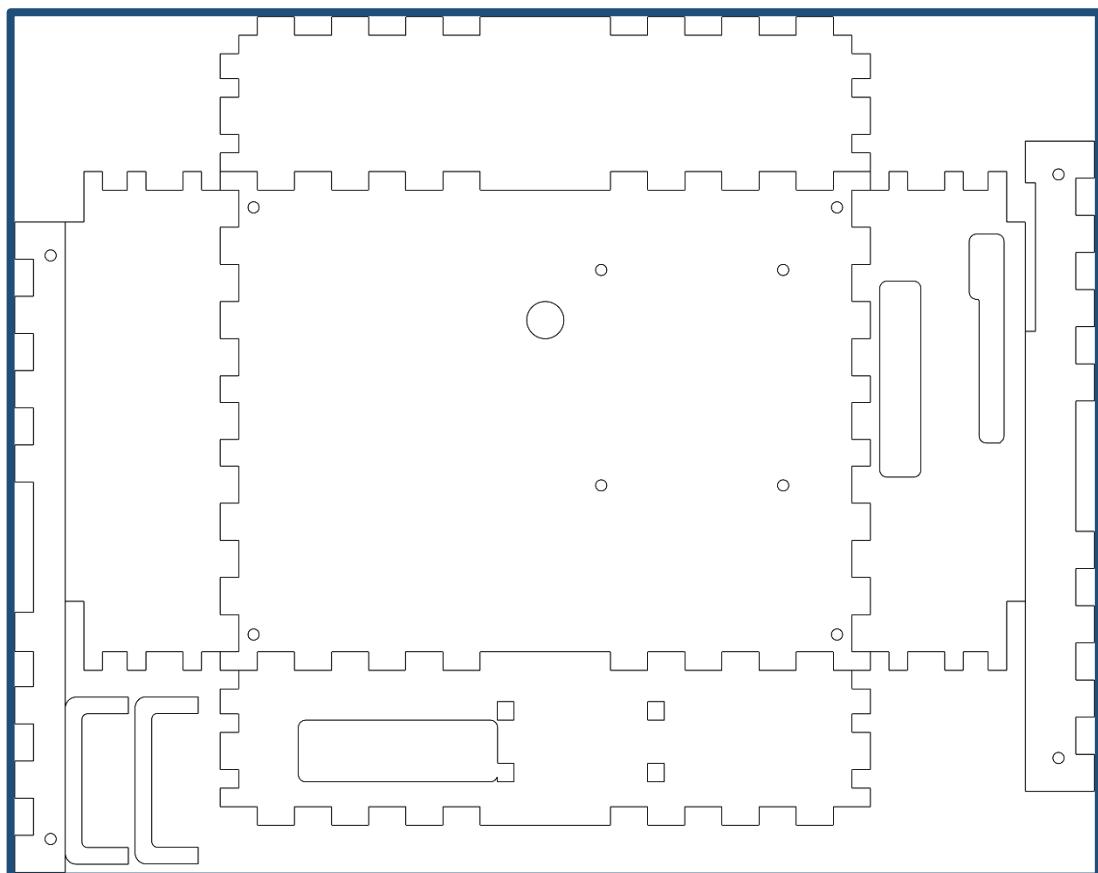


Hình 3. 13: Mô phỏng vị trí Pi 4 và camera

### CHƯƠNG 3: THIẾT KẾ HỆ THỐNG



Hình 3. 14: Thiết kế hộp chứa và kết nối thiết bị hoàn chỉnh



Hình 3. 15: Xuất file cắt mica

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### 4.1. Giới thiệu

Đề tài xử lý ảnh hiện nay đã và đang đi vào trong giảng đường đại học từ sơ cấp đến nâng cao. Vì vậy, đề tài “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện khuôn mặt kết hợp mã QR và mã vạch trên Raspberry Pi 4**” được ứng dụng thông qua việc xử lý hình ảnh, nhận dạng hình ảnh, điểm danh lớp học với độ chính xác cao nhất mà nhóm có thể làm hơn thế nữa là sự gọn gàng và tiện lợi khi sử dụng mô hình vào thực tế. Ở phần này nhóm sẽ trình bày quá trình thi công hệ thống. Hệ thống chỉ dừng lại ở dạng mô hình chưa thể đạt độ chính xác 100%, độ thẩm mỹ cũng như sự tinh tế như các sản phẩm thương mại hiện nay.

### 4.2. Yêu cầu và mô tả hoạt động của hệ thống:

#### 4.2.1. Yêu cầu hệ thống

Hệ thống được thiết kế với các yêu cầu sau:

- Giao diện giao tiếp người dùng đơn giản, dễ sử dụng.
- Nhận diện được khuôn mặt và tiến hành xử lý ảnh thông qua hình ảnh được ghi lại nhờ Webcam Logitech.
- Hệ thống hoạt động thời gian dài, ít bị lỗi trong khi đang hoạt động.
- Đảm bảo các phương thức điểm danh đạt độ chính xác cao.
- Xuất được file danh sách điểm danh đúng thời gian thực cho người quản lý.
- Hệ thống có thể chỉnh sửa và nâng cấp chương trình.

#### 4.2.2. Mô tả hoạt động hệ thống

Hệ thống điểm danh được giám sát và điều khiển bởi một giao diện trên màn hình cảm ứng.

Giao diện điều khiển có các nút nhấn “Lấy dữ liệu”, nút “xử lý dữ liệu”, “Điểm danh”, “Quét thẻ”, “Xác nhận” và nút “Giám sát”.

Đối với một sinh viên khi chưa có thông tin gì trong hệ thống. Bắt buộc nhấn nút “Lấy dữ liệu” nhập thông tin gồm mã số sinh viên và tên sinh viên, lúc đó phần chương trình lấy dữ liệu sẽ hoạt động. Hệ thống sẽ xác định khuôn mặt của sinh viên và chụp 200 tấm ảnh đưa vào file data\_faces. Tên và mã số sinh viên được lưu và cơ sở dữ liệu và dán nhãn cho 200 ảnh với tên [mã số sinh viên].[số thứ tự tấm ảnh]. Các sinh viên

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

khác thực hiện tương tự như vậy. (Trong khi lấy dữ liệu để đạt độ chính xác cao khi điểm danh, chỉ có 1 khuôn mặt xuất hiện vào khuôn hình và đồng thời xoay khuôn mặt nhiều góc cạnh trong khi hệ thống chụp hình).

Khi đã lấy tất cả dữ liệu của khuôn mặt của tất cả sinh viên trong lớp học. Nhấn phím “Xử lý dữ liệu” lúc đó hệ thống tiến hành huấn luyện.

Sau khi huấn luyện xong, hệ thống đã có thể điểm danh bằng nhận dạng khuôn mặt. Để điểm danh bằng nhận dạng khuôn mặt, sinh viên nhấn phím “Điểm danh” lúc đó hệ thống tiến hành bật camera xác định khuôn mặt sinh viên đang điểm danh, so sánh với dữ liệu đã huấn luyện. Xác định được bạn sinh viên đó là ai, thì hệ thống sẽ trích mã số sinh viên và tên sinh viên trong cơ sở dữ liệu ra màn hình. Sinh viên sẽ kiểm tra thông tin điểm danh đã chính xác chưa, nếu đúng nhấn phím “Xác nhận” thì thông tin đó sẽ được ghi vào file excel kèm với thời gian thực điểm danh. Nếu thông tin sai, sinh viên nhấn phím “Điểm danh” tiến hành điểm danh lại (điểm danh nếu xuất ra tên sai không nhấn phím “Xác nhận” thông tin sẽ không được ghi vào file excel. Đối với những khuôn mặt chưa có trong hệ thống điểm danh, hệ thống sẽ xuất ra màn hình dòng chữ “Không xác định”).

Để điểm danh bằng thẻ sinh viên chỉ cần nhấn phím “Quét thẻ” khi đó chương trình đọc QR code, Barcode sẽ hoạt động. Sinh viên đưa thẻ cá nhân của mình lên camera, hệ thống đọc được mã thẻ và xuất ra màn hình, sinh viên kiểm tra và nhấn phím “Xác nhận”.

Ngoài ra hệ thống có phím “GS” nghĩa là giám sát. Khi bấm vào phím này camera sẽ bật lên, sinh viên có mặt trong khung hình sẽ hiển thị.

### 4.3. Thi công phần cứng hệ thống

#### 4.3.1. Chuẩn bị linh kiện phần cứng

Sau khi tính toán và thiết kế, chúng ta chuẩn bị linh kiện để tiến hành lắp ráp.

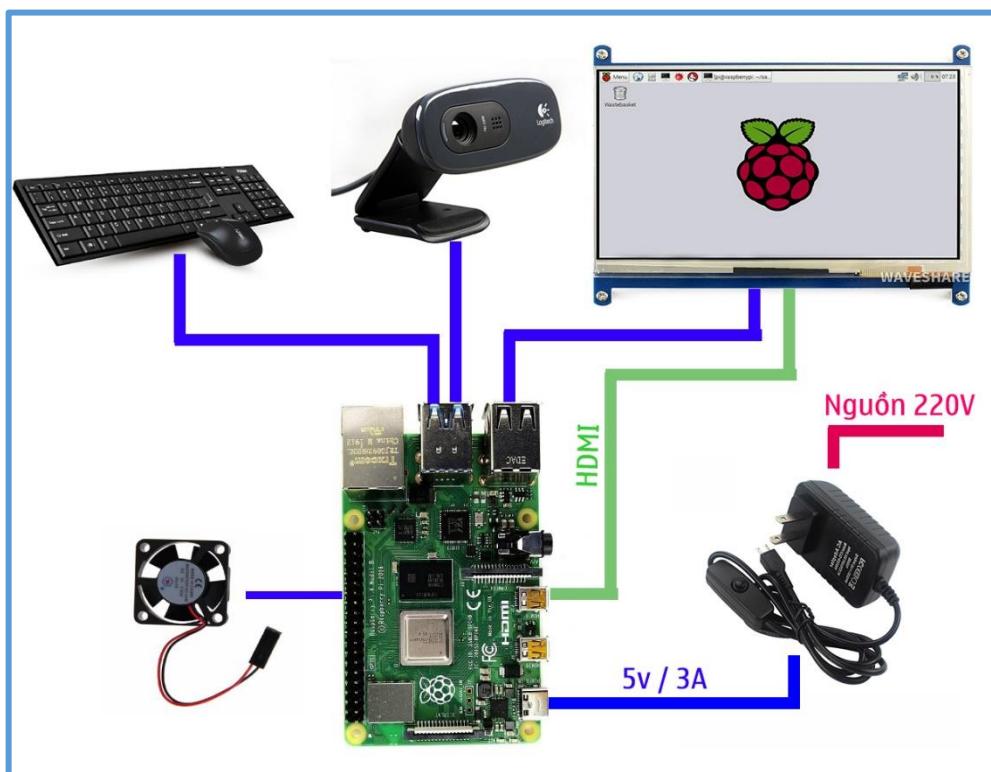
Bảng 4. 1: Bảng linh kiện chuẩn bị lắp ráp hệ thống.

STT	Tên linh kiện	Số lượng	Chú thích
1	Raspberry Pi 4 Mode B	1	
2	Quạt tản nhiệt cho Pi 4	1	

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

3	Webcam Logitech C270	1	
4	Màn hình cảm ứng điện dung 7inch HDMI (B) LCD	1	Kèm dây HDMI và dây cáp nguồn cho màn hình.
5	Adapter công tắc nguồn 5V/3A	1	
6	Óc, vít, trụ đồng cho Raspberry Pi 4 và màn hình	8 bộ	
7	Hộp lắp linh kiện	1	Thiết kế và cắt mica
8	Giá đỡ màn hình	1	Thiết kế

### 4.3.2. Lắp ráp phần cứng



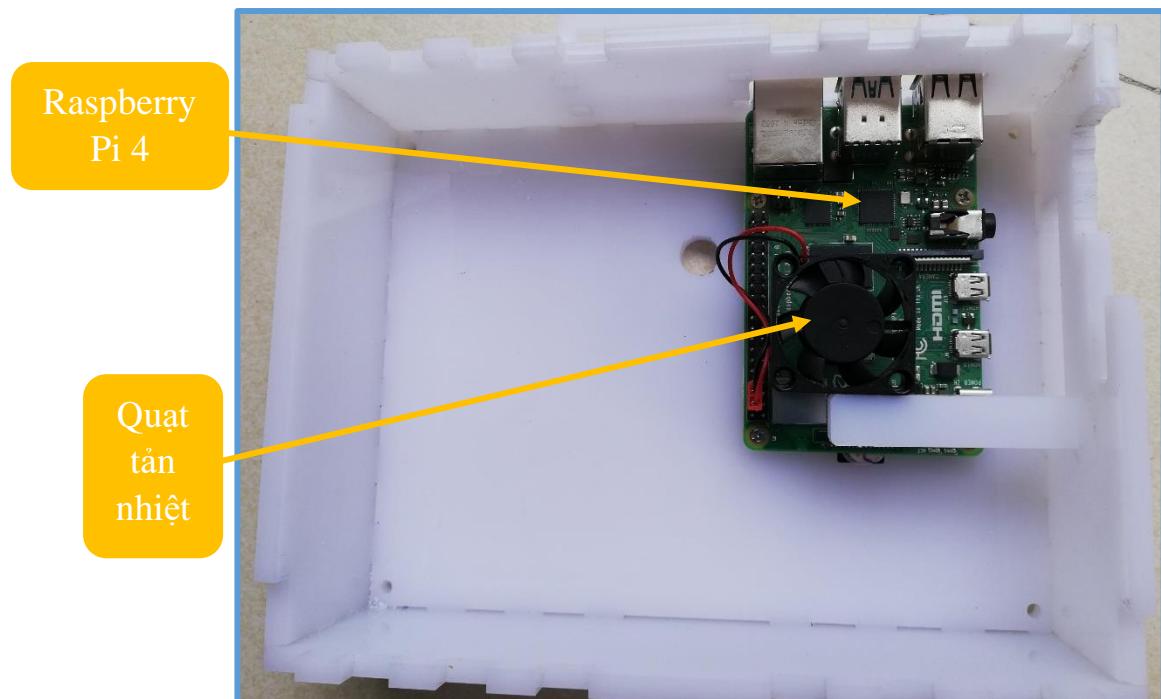
Hình 4. 1: Sơ đồ kết nối dây các thiết bị phần cứng.

Nguồn 220V qua Adapter công tắc chuyển thành nguồn 5V/3A cung cấp cho Raspberry Pi 4 hoạt động. Từ các cổng của Raspberry Pi 4 kết nối với các thiết bị khác. Màn hình cảm ứng điện dung 7 inch HDMI (B) LCD có hai dây, một dây HDMI và một dây cáp nguồn. Quạt tản nhiệt được lấy nguồn từ các port của Raspberry Pi 4. Các thiết

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

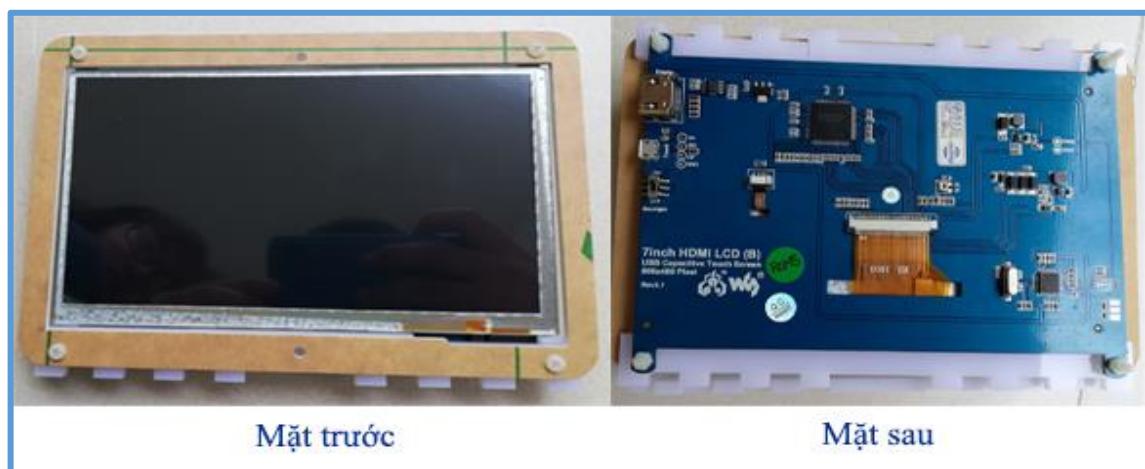
bị ngoại vi khác như chuột, bàn phím được sử dụng khi thi công và lập trình hệ thống để dễ thao tác. Trên thực tế màn hình cảm ứng điện dung 7 inch HDMI (B) LCD có thể cài bàn phím ảo và giao tiếp trực tiếp người dùng.

Lắp ráp linh kiện và kết nối dây các thiết bị. Đầu tiên lắp Raspberry Pi 4 và quạt tản nhiệt vào hộp. Khi lắp cần để ý khoảng cách của quạt phải cách Pi 4, ngoài ra cần xác định đúng chiều gió của quạt làm mát cho chip của Raspberry Pi 4.



Hình 4. 2: Lắp quạt và Raspberry Pi 4 và hộp

Tiếp theo, lắp màn hình cảm ứng điện dung 7 inch HDMI (B) LCD vào khung. Khi lắp cần chú ý các dây cáp riêng của màn hình bị cấn bởi khung.



Hình 4. 3: Màn hình được lắp vào khung.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

Lắp camera vào khung và đi dây gọn gàng vào hộp.



Hình 4. 4: Camera được lắp vào hộp

Kết nối các dây cáp cần thiết và các thiết bị.



Hình 4. 5: Kết nối dây các thiết bị.

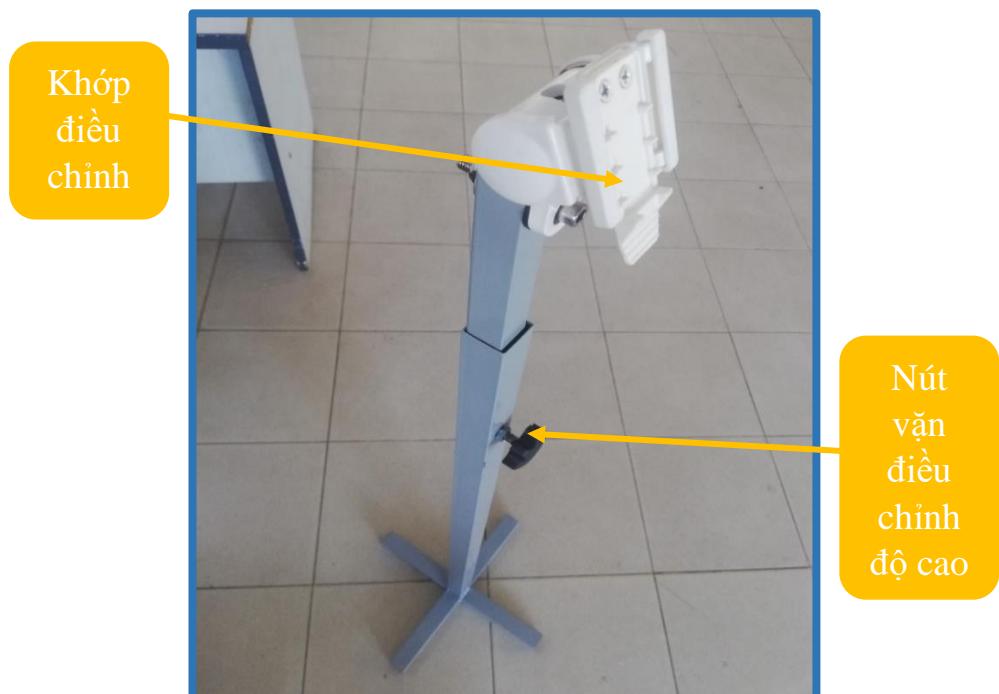
Cáp nguồn, kết nối và kiểm tra hoạt động của phần cứng.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG



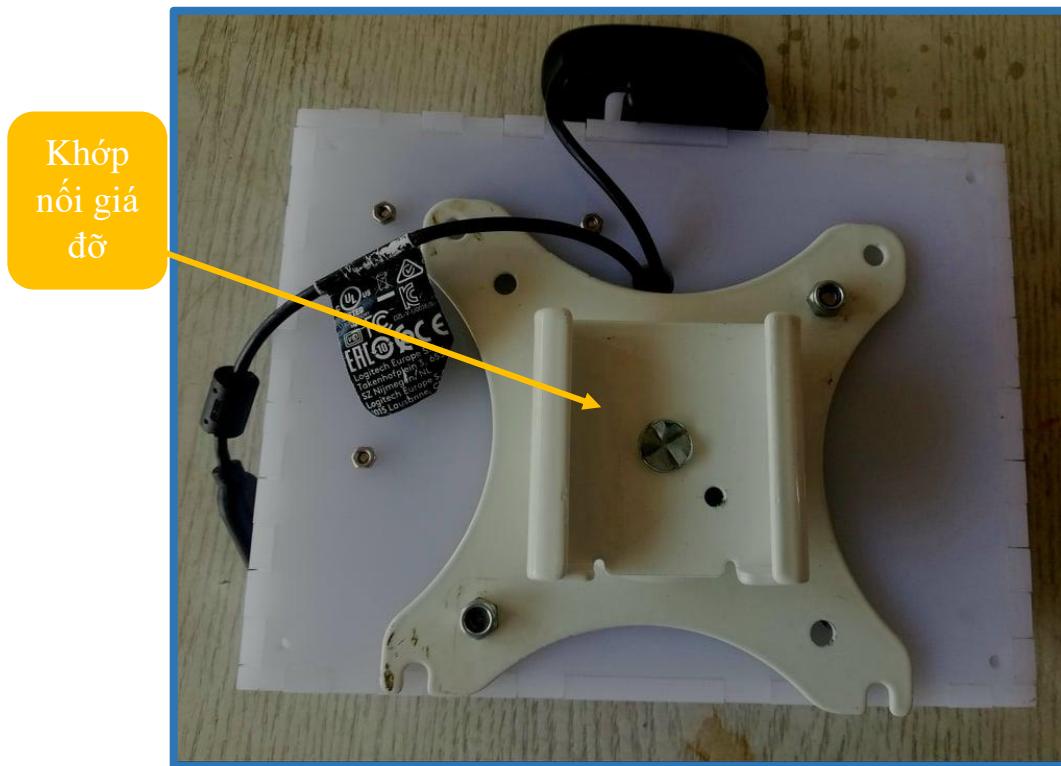
Hình 4. 6: Kết nối và kiểm tra hoạt động phần cứng

### 4.3.3. Lắp giá đỡ



Hình 4. 7: Giá đỡ hộp hệ thống.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG



Hình 4. 8: Lắp khung kết nối với giá đỡ vào màn hình

Giá đỡ đảm bảo dễ sử dụng, dễ di chuyển có thể điều chỉnh cao thấp, điều chỉnh góc nghiêng khi thực hiện điểm danh.

### 4.4. Lập trình hệ thống

#### 4.4.1. Thiết kế giao diện của hệ thống

Để giao tiếp với người dùng nhóm thực hiện đã dùng **Tkinter** để thiết kế giao diện. Giao diện hệ thống gồm các phím nhấn và vùng hiển thị thông tin. Tkinter nhóm thực hiện đã giới thiệu ở chương 2 – Cơ sở lý thuyết.

##### 4.4.1.1. Tạo Label trong Tkinter:

Label là nơi hiển thị văn bản và hình ảnh. Văn bản được hiển thị trong Label có thể thay đổi bất kì lúc nào bạn muốn.

- ✓ Cú pháp:

Đây là cú pháp đơn giản để tạo Label:

```
w = Label ( master, option, ... )
```

- ✓ Thông số:

+ master: Đại diện cho cửa sổ chính.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

+ option: là danh sách các tùy chọn thường được sử dụng cho Lable. Các tùy chọn này được biểu diễn bằng một từ khóa và giá trị, chúng ngăn cách nhau bằng dấu phẩy.

*Bảng 4. 2: Các tùy chọn của Lable*

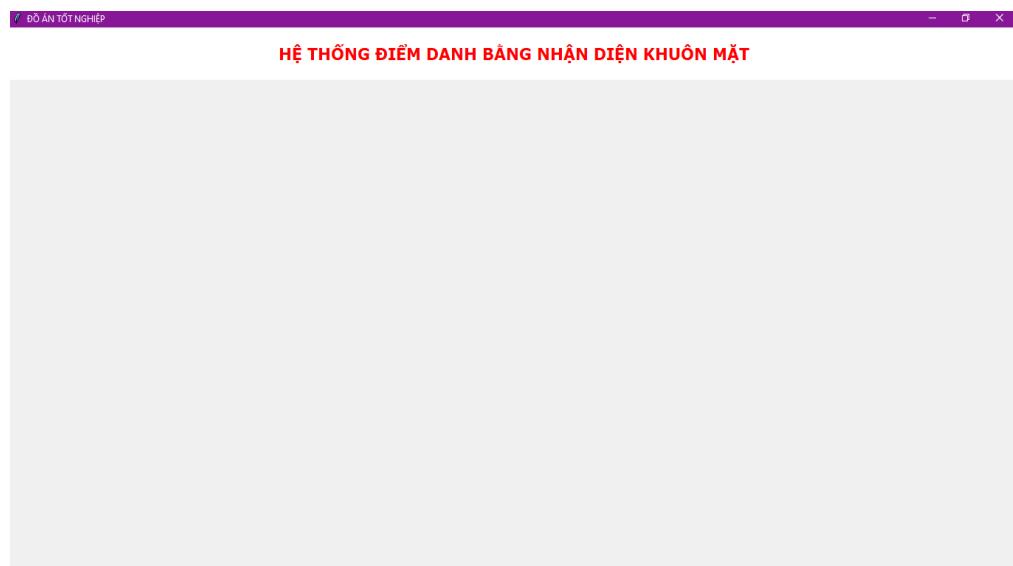
Chức năng	Mô tả
<b>anchor</b>	Tùy chỉnh vị trí của văn bản: Left, Center và Right
<b>bg</b>	Màu nền của Lable
<b>bitmap</b>	Tùy chọn một đối tượng ảnh và Lable sẽ hiển thị ảnh đó
<b>bd</b>	Kích thước đường viền Lable
<b>cursor</b>	Tùy chỉnh hiển thị của con trỏ (Mũi tên, dấu chấm,...), con trỏ sẽ thay đổi khi ở trên Lable đó.
<b>font</b>	Tùy chỉnh phông chữ của Lable
<b>fg</b>	Tùy chọn màu của văn bản
<b>height</b>	Kích thước dọc của khung
<b>image</b>	Hiển thị hình ảnh trong Lable
<b>justify</b>	Tùy chỉnh căn chỉnh các dòng văn bản: Left, Center và Right
<b>padx</b>	Thêm khoảng trắng vào bên trái, phải của văn bản. Mặc định là 1
<b>relief</b>	Tùy chỉnh các kiểu đường viền của Lable. Mặc định là FLAT
<b>text</b>	Hiển thị một hoặc nhiều dòng văn bản trong Lable
<b>underline</b>	Hiển thị gạch dưới ( _ ) bên dưới chữ cái thứ n của văn bản, tính từ 0. Mặc định là gạch dưới = -1, có nghĩa là không có gạch dưới
<b>width</b>	Tùy chỉnh chiều rộng của Lable. Được tính bằng kí tự. Nếu tùy chỉnh này không được thiết lập thì Lable sẽ có kích thước vừa khung với nó
<b>textvariable</b>	Để bổ sung nội dung văn bản được hiển thị trong Lable thành một biến điều khiển (stringvar)
<b>wraplength</b>	Giới hạn số lượng kí tự của một dòng

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

- ✓ Ví dụ minh họa:

```
from tkinter import *
win =Tk()
win.title('ĐỒ ÁN TỐT NGHIỆP')

LbMain = Label(win, text = "HỆ THỐNG ĐIỂM DANH BẰNG NHẬN  
DIỆN KHUÔN MẶT", bg = 'White', fg = 'red',
               font = 'verdana 16 bold ', width =100,
               padx = 450, pady = 20)
LbMain.pack()
win.mainloop()
```



Hình 4. 9: Giao diện được tạo bằng Tkinter với Lable

### 4.4.1.2. Tạo Button trong Tkinter

Button được sử dụng để thêm các nút ứng dụng trong Python. Nó có thể hiển thị văn bản hoặc hình ảnh. Bạn có thể đính kèm một chức năng vào một nút khi nhấp vào nó.

- ✓ Cú pháp:

Đây là cú pháp đơn giản cho Button:

```
w = Button ( master, option=value, ...)
```

- ✓ Thông số:

master: Cửa sổ chính.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

Option: là danh sách các tùy chọn thường được sử dụng cho Button. Các tùy chọn này được biểu diễn bằng một từ khóa và giá trị, chúng ngăn cách nhau bằng dấu phẩy.

Option	Mô tả
<b>activebackground</b>	Màu nền của nút khi đưa trỏ vào.
<b>activeforeground</b>	Màu nền của nút trước khi đưa con trỏ vào.
<b>bd</b>	Chiều rộng của đường viền được tính bằng pixel. Mặc định là 2.
<b>bg</b>	Màu nền của nút.
<b>command</b>	Hàm hoặc chức năng được gọi khi nút được nhấn
<b>fg</b>	Màu nền của văn bản.
<b>font</b>	Phông chữ văn bản của nút.
<b>height</b>	Chiều cao của nút.
<b>image</b>	Hình ảnh hiển thị trên nút.
<b>justify</b>	Hiển thị nhiều dòng văn bản: Left, center và right.
<b>padx</b>	Khoảng trắng bên trái và bên phải của văn bản.
<b>pady</b>	Khoảng trắng bên trên và dưới của văn bản.
<b>relief</b>	Thiết lập các dạng đường viền của button.
<b>state</b>	Thiết lập DISABLED làm mờ và tắt phản hồi của Button. Có giá trị ACTIVE khi di chuột qua nó. Mặc định là NORMAL.
<b>underline</b>	Hiển thị gạch dưới ( _ ) bên dưới chữ cái thứ n của văn bản, tính từ 0. Mặc định là gạch dưới = -1, có nghĩa là không có gạch dưới.
<b>width</b>	Chiều rộng của Button.
<b>wraplength</b>	Giới hạn số lượng kí tự của một dòng.

### ✓ Tạo các phím button trong hệ thống:

```
from tkinter import *
win =Tk()
```

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

```
win.title('ĐỒ ÁN TỐT NGHIỆP')

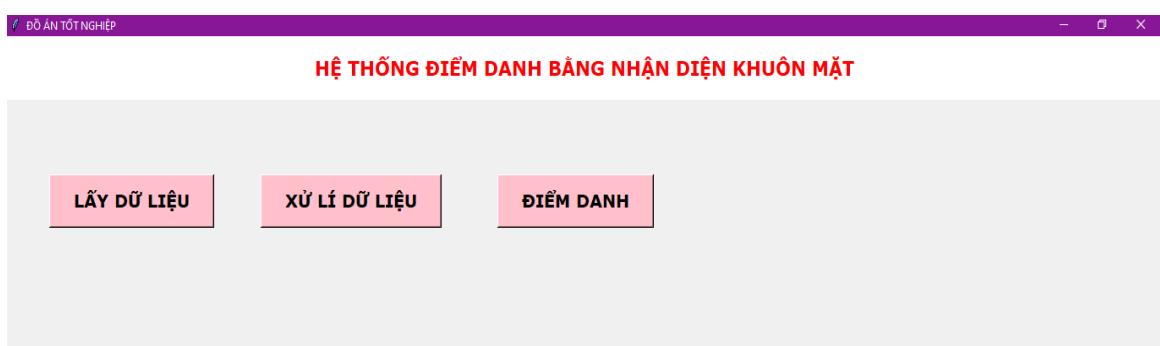
LbMain = Label(win, text = "HỆ THỐNG ĐIỂM DANH BẰNG NHẬN  
DIỆN KHUÔN MẶT", bg = 'White', fg = 'red',  
font = 'verdana 16 bold ', width =100,  
padx = 450, pady = 20)
LbMain.pack()

Photo = Button(win, text = "LẤY DỮ LIỆU", bg ='pink', fg  
= 'black', font = 'verdana 14 bold', padx = 20, pady =  
10)
Photo.place( x = 50, y = 150)

Train = Button(win, text = "XỬ LÍ DỮ LIỆU", bg ='pink',  
fg = 'black', font = 'verdana 14 bold', padx = 20, pady =  
10)
Train.place( x = 300, y = 150)

Test = Button(win, text = "ĐIỂM DANH", bg ='pink', fg =  
'black', font = 'verdana 14 bold', padx = 20, pady = 10)
Test.place( x = 580, y = 150)

win.mainloop()
```



Hình 4. 10: Giao diện được tạo bằng Tkinter Với Button.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

### 4.4.1.3. Cấu trúc giao diện của hệ thống



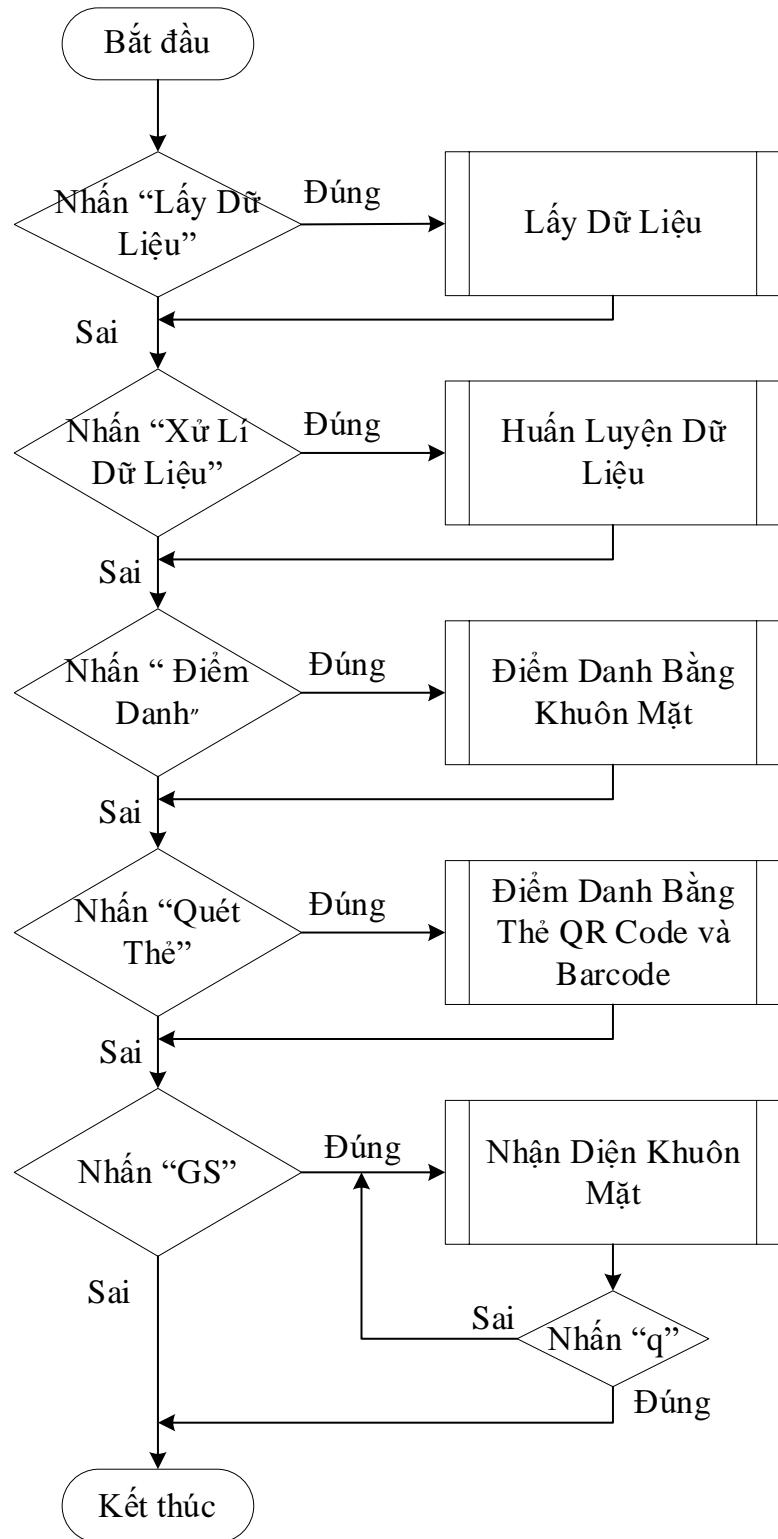
Hình 4. 11: Giao diện hoàn chỉnh

Chức năng:

- **Nút Lấy Dữ Liệu:** Chụp hình Sinh Viên để đưa vào Cơ Sở Dữ Liệu với Thông tin: Tên và Mã Số Sinh Viên. Số lượng Ảnh chụp 200 Tấm.
- **Nút Xử Lý Dữ Liệu:** Tiến hành huấn luyện dữ liệu đã thu thập.
- **Nút Điểm Danh:** Điểm danh Sinh viên bằng Khuôn mặt.
- **Nút Quét Thẻ:** Điểm danh Sinh viên bằng mã QR Code và mã vạch.
- **Nút Xác Nhận:** Để lưu thông tin Sinh viên đã điểm danh vào file Excel.
- **Nút GS (Giám sát):** Dùng để giám sát Sinh viên và quan sát lỗi khi nhận diện sai thông tin của Sinh viên.
- **Sinh Viên Điểm Danh:** Nơi hiển thị thông tin của Sinh viên đã điểm danh và ấn nút Xác Nhận để lưu thông tin đó vào file Exel.

#### 4.4.2. Lưu đồ lập trình hệ thống hệ thống

##### 4.4.2.1. Lưu đồ giải thuật tổng thể



Hình 4. 12: Lưu đồ tổng thể hệ thống

Giải thích lưu đồ giải thuật:

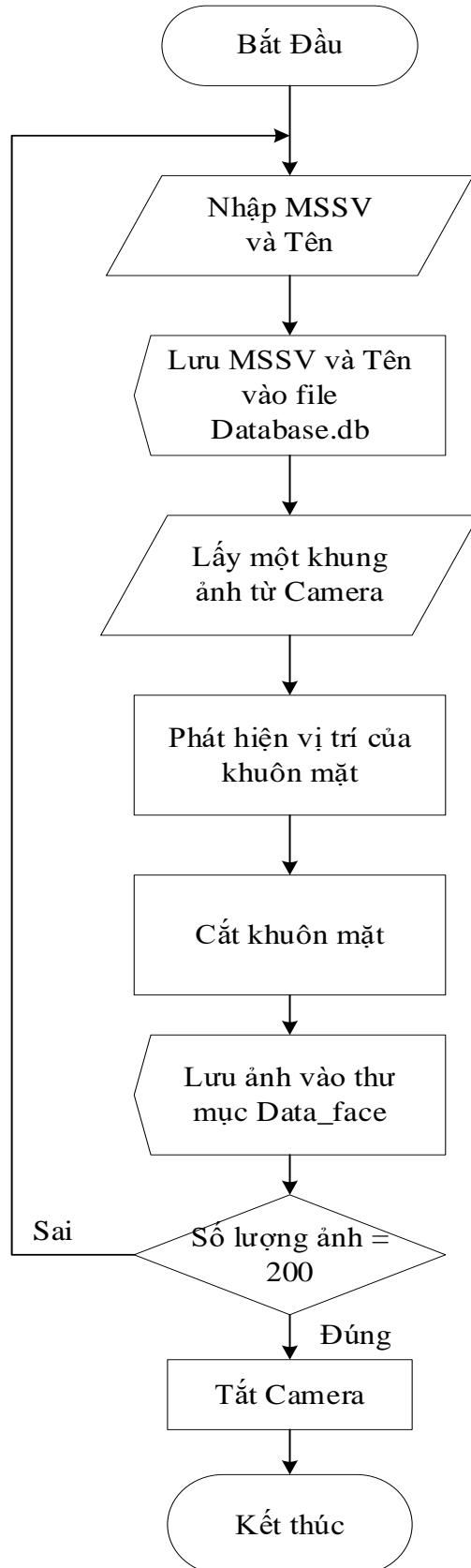
- Đầu tiên, hệ thống sẽ chờ nhấn nút:

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

- Nếu nút “Lấy Dữ Liệu” được nhấn. Hệ thống sẽ tiến hành lấy dữ liệu của Sinh viên bao gồm: Tên và Mã số Sinh viên. Tiếp theo, hệ thống sẽ chụp 200 tấm ảnh và kết thúc.
  - Nếu nút “Xử Lý Dữ Liệu” được nhấn. Hệ thống sẽ tiến hành huấn luyện dữ liệu với tất cả các ảnh đã thu thập được và kết thúc.
  - Nếu nút “Điểm Danh” được nhấn. Hệ thống sẽ tiến hành điểm danh bằng khuôn mặt và kết thúc.
  - Nếu nút “Quét thẻ” được nhấn. Hệ thống sẽ tiến hành điểm danh sinh viên bằng thẻ chứa mã QR code và Barcode và kết thúc.
- Nếu nút “GS” được nhấn. Hệ thống sẽ hiển thị camera theo thời gian thực với thông tin sinh viên và nhấn “q” để kết thúc việc giám xác đó.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

### 4.4.2.2. Lưu đồ giải thuật lấy dữ liệu



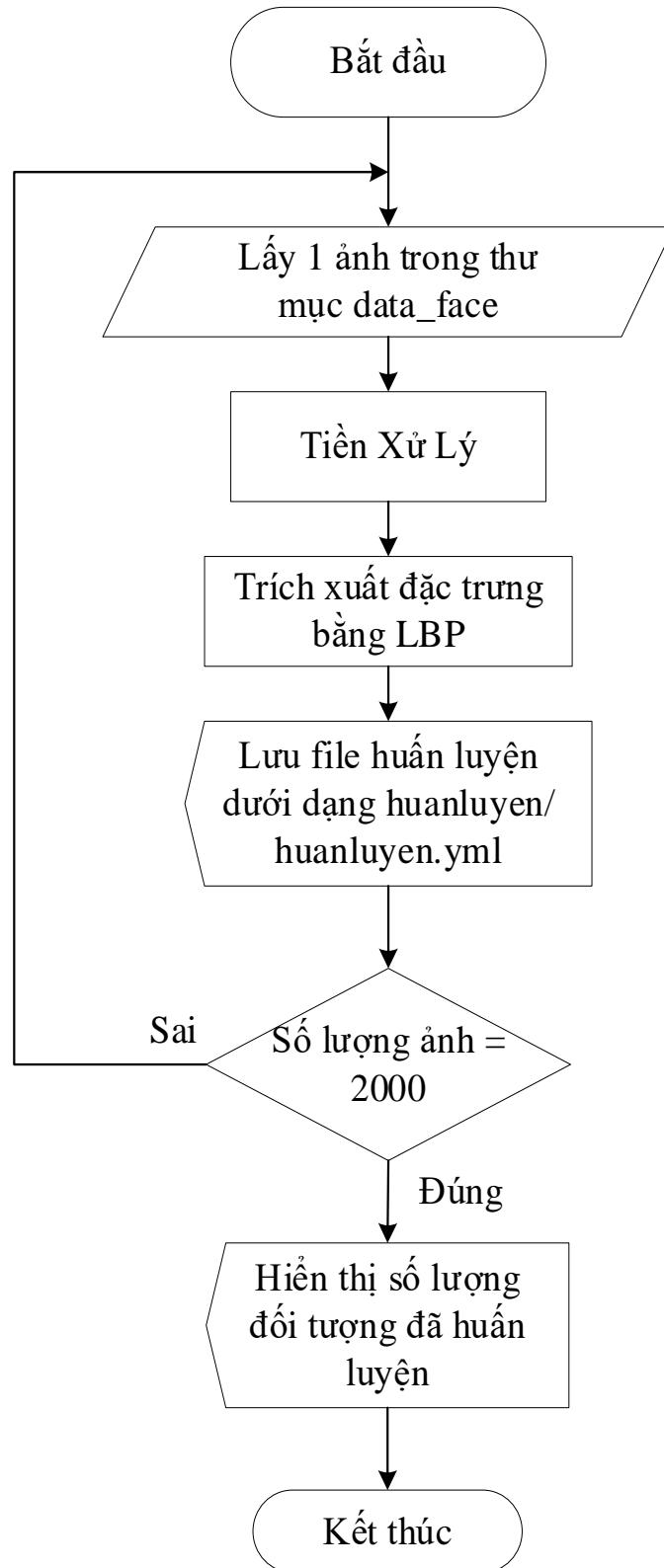
Hình 4. 13: Lưu đồ giải thuật Lấy Dữ Liệu

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

Giải thích lưu đồ giải thuật:

- Đầu tiên, hệ thống sẽ yêu cầu nhập thông tin bao gồm: Tên và Mã Số Sinh viên để lưu vào cơ sở dữ liệu Database.db.
- Tiếp theo, camera sẽ bật lên. Hệ thống sẽ lấy một khung hình từ camera để xác định xem trong khung hình đó có khuôn mặt hay không (bằng haarcascade \_frontalface\_ default.xml). Nếu có hệ thống sẽ định vị trí khuôn mặt và vẽ đường viền xung quanh khuôn mặt đã xác định.
- Hệ thống sẽ cắt từng khuôn mặt trong khung hình đó. Và lưu vào file data\_face với thông tin của khung hình bao gồm: Tên, Mã số sinh viên và số thứ tự.
- Nếu hệ thống xác định trong khung hình đó không có khuôn mặt thì sẽ lấy khung hình tiếp theo từ Camera để xử lí.
- Cứ như thế thực hiện đến khi nào đủ 200 khung hình lưu vào file data\_face. Hệ thống sẽ kết thúc.

#### 4.4.2.3. Lưu đồ giải thuật Huấn luyện Dữ liệu



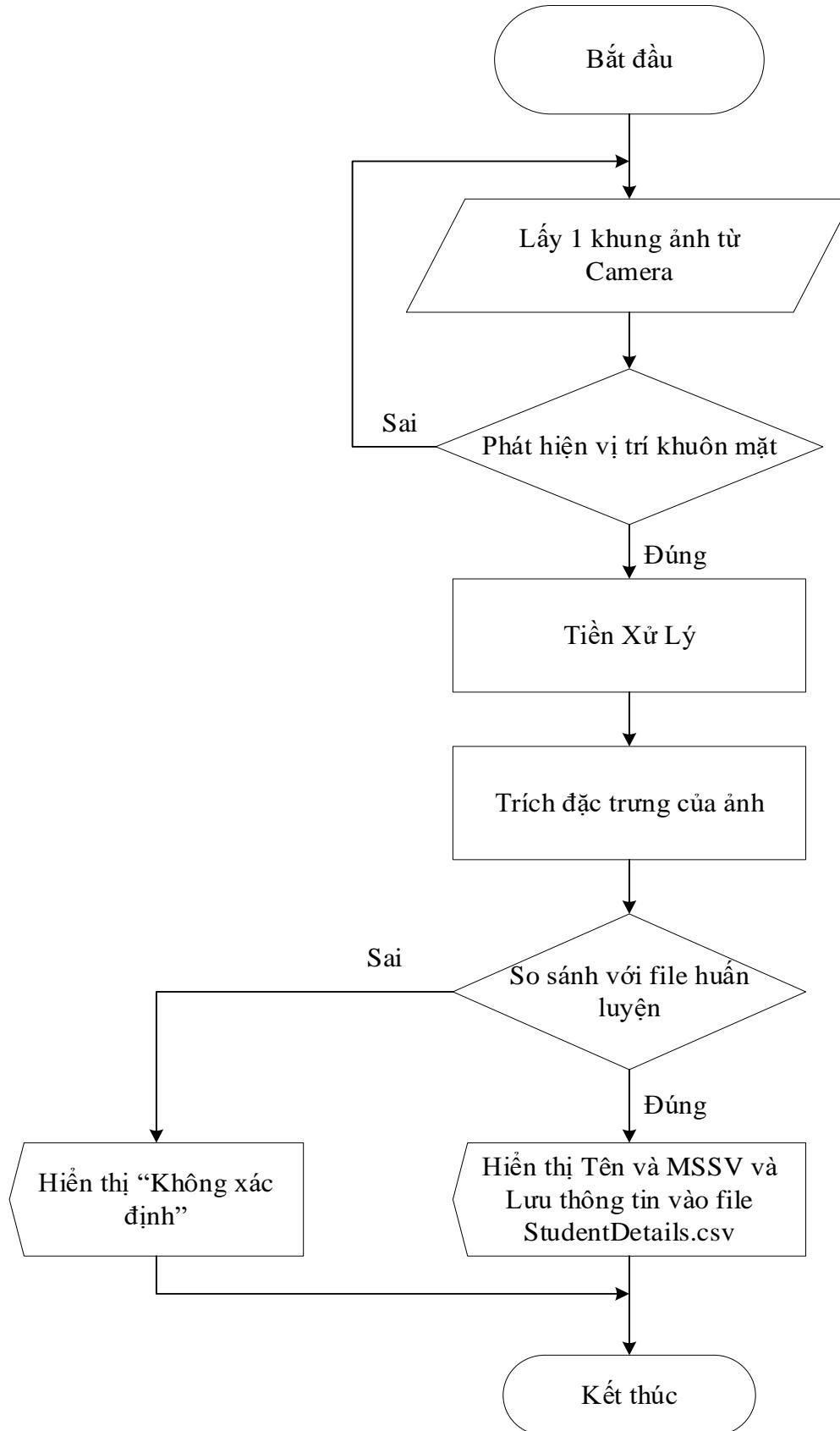
Hình 4. 14: Lưu đồ giải thuật Huấn luyện Dữ liệu

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

Giải thích lưu đồ giải thuật:

- Đầu tiên, hệ thống sẽ trỏ vào thư mục chứa các khung hình đã thu thập được: data\_face.
- Tiền xử lí: ảnh sẽ được chuyển thành ảnh xám để đưa vào hệ thống để trích xuất đặc trưng bằng phương pháp LBP của từng ảnh vào lưu vào file huanluyen/huanluyen.yml.
- Cứ như thế thực hiện đến khi nào hết 2000 ảnh đã thu thập trong thư mục data\_face.
- Sau khi thực hiện huấn luyện xong 2000 ảnh. Hệ thống hiển thị số lượng đối tượng đã được huấn luyện trong mô hình và Kết thúc.

#### 4.4.2.4. Lưu đồ giải thuật Nhận Diện Khuôn mặt



Hình 4. 15: Lưu đồ giải thuật Nhận Diện

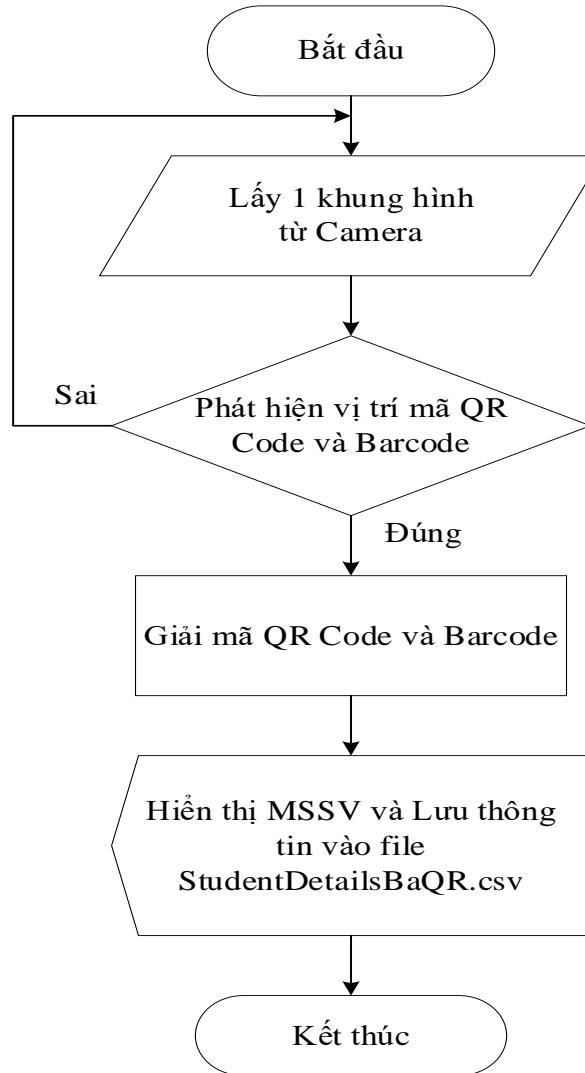
## CHƯƠNG 4: THI CÔNG HỆ THỐNG

Giải thích lưu đồ giải thuật:

- Đầu tiên, hệ thống sẽ đọc file dữ liệu lưu thông tin của Sinh viên: Database.db và file huấn luyện: huanluyen/huanluyen.yml.
- Tiếp theo bật camera lên và hệ thống sẽ lấy một khung hình từ camera theo thời gian thực để tiến hành việc điểm danh.
- Mô hình sẽ xác định xem trong khung hình đó có khuôn mặt hay không. Nếu có khuôn mặt trong khung hình. Hệ thống sẽ tiến hành cắt khuôn mặt đó, chuyển thành ảnh xám và thực hiện trích xuất đặc trưng bằng phương pháp LBP từ khung hình đó và so sánh với dữ liệu đã huấn luyện.
- Nếu đối tượng có đặc trưng giống với đặc trưng của dữ liệu đã huấn luyện. Hệ thống sẽ đưa thông tin của sinh viên đó hiển thị lên màn hình theo thời gian thực.
- Nếu trong khung hình không có khuôn mặt thì hệ thống sẽ lấy khung hình tiếp theo từ Camera.
- Khi xác định được thông tin của đối tượng điểm danh: Nếu đối tượng đó có trong cơ sở dữ liệu thì hệ thống sẽ hiển thị “Tên và Mã số Sinh viên” lên màn hình. Nếu không có trong cơ sở dữ liệu hệ thống sẽ hiển thị “Không xác định”.
- Thông tin sinh viên điểm danh sẽ được lưu vào file Exel: StudentDetails.csv bao gồm: Tên, Mã số sinh viên, Ngày, Giờ.
- Hệ thống Kết thúc việc điểm danh.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

### 4.4.2.5. Lưu đồ giải thuật điểm danh bằng QR Code



Hình 4. 16: Lưu đồ giải thuật Điểm danh bằng QR Code và Barcode

Giải thích lưu đồ giải thuật:

- Đầu tiên, hệ thống bắt camera và lấy một khung hình từ camera theo thời gian thực để thực hiện điểm danh bằng quét thẻ.
- Hệ thống sẽ xác định xem trong khung hình đó có mã QR code và Barcode hay không.
- Nếu xác định được vị trí của mã QR code và Barcode, hệ thống sẽ tiến hành giải mã mã QR code và Barcode thành ký tự hiện thị thông tin mã lên màn hình.
- Nếu không xác định được vị trí mã QR code và Barcode, hệ thống sẽ lấy khung hình tiếp theo từ camera.

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

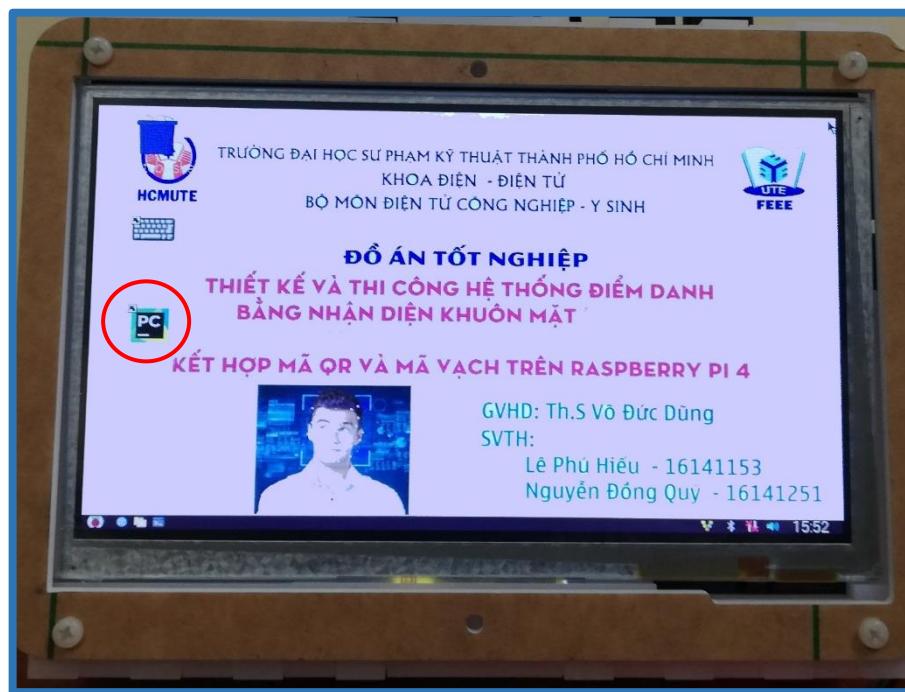
- Thông tin sẽ được lưu vào file StudentDetailsBaQR.csv gồm: Mã số Sinh viên, ngày, giờ. Và hệ thống kết thúc.

Lưu thông tin text vào file Exel và Kết thúc.

### 4.4.3. Hướng dẫn sử dụng

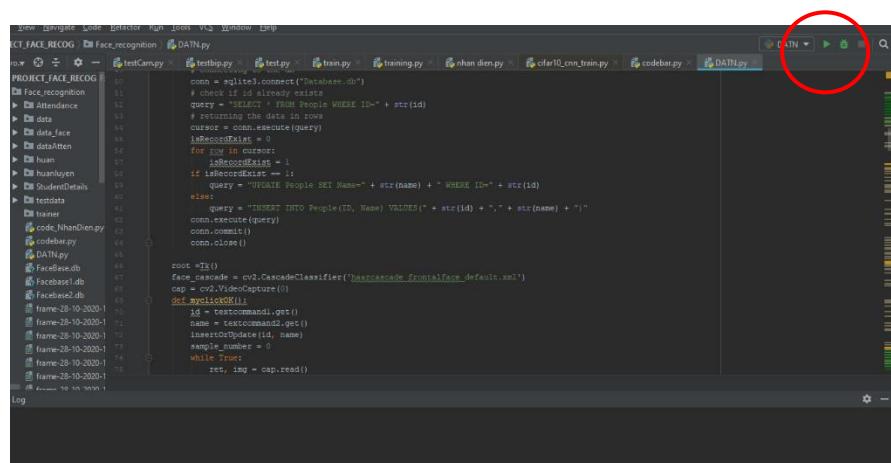
**Bước 1:** Cấp nguồn và khởi động hệ thống.

**Bước 2:** Khởi động phần mềm Pycharm, nháy đúp và biểu tượng PC trên màn hình.



Hình 4. 17: Khởi động chương trình Pycharm

**Bước 3:** Chạy file code của hệ thống, bằng cách ấn vào mũi tên màu xanh trên cùng bên trái.



Hình 4. 18: Chạy file chương trình chính

## CHƯƠNG 4: THI CÔNG HỆ THỐNG

**Bước 4:** Giao tiếp với giao diện chính của hệ thống.



Hình 4. 19: Giao diện chính của hệ thống

- **Nút Lấy Dữ Liệu:** Chụp hình Sinh Viên để đưa vào Cơ Sở Dữ Liệu với Thông tin: Tên và Mã Số Sinh Viên. Số lượng Ảnh chụp 200 Tấm.
- **Nút Xử Lý Dữ Liệu:** Tiến hành huấn luyện dữ liệu đã thu thập.
- **Nút Điểm Danh:** Điểm danh Sinh viên bằng Khuôn mặt.
- **Nút Quét Thẻ:** Điểm danh Sinh viên bằng mã QR Code và mã vạch.
- **Nút Xác Nhận:** Để lưu thông tin Sinh viên đã điểm danh vào file Excel khi người thực hiện điểm danh kiểm tra đúng thông tin của mình.
- **Nút GS ( Giám sát):** Dùng để giám sát Sinh viên và quan sát lỗi khi nhận diện sai thông tin của Sinh viên.
- **Sinh Viên Điểm Danh:** Nơi hiển thị thông tin của Sinh viên đã điểm danh và ấn nút Xác Nhận để lưu thông tin đó vào file Exel.

**Bước 5:** Xem thông tin điểm danh ( chỉ dành cho giáo viên hoặc người quản lý)

Giáo viên hoặc người quản lý và đúng địa chỉ [/home/pi/DATN/StudentDetails]

## CHƯƠNG 4: THI CÔNG HỆ THỐNG



The screenshot shows a terminal window titled "StudentDetails\SStudentDetailBarQR.csv - Mousepad". The window contains a list of student attendance records. The first few lines of the data are:

```
Id,Date,Time  
16141153,2020-12-23,14:11:12  
Id,Date,Time  
16141251- Quy,2020-12-23,14:11:35  
Id,Date,Time  
16141153,2020-12-23,14:12:43  
Id,Date,Time  
16141251- Quy,2020-12-23,14:15:57  
Id,Date,Time  
16141308,2020-12-23,14:17:14  
Id,Date,Time  
16141251 Nguyen Dong Quy,2021-01-06,15:57:11  
Id,Date,Time  
16141153,2021-01-06,16:05:31|  
Id,Date,Time  
16141251,2021-01-06,16:05:42  
Id,Date,Time  
16141251,2021-01-06,16:05:54  
Id,Date,Time  
16141308
```

The terminal window is running on a Raspberry Pi, as indicated by the prompt "pi@raspberrypi: ~". The system tray shows the date and time as 11:31.

Hình 4. 20: Địa chỉ kết quả điểm danh

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

### CHƯƠNG 5. KẾT QUẢ - NHẬN XÉT VÀ ĐÁNH GIÁ

Mục tiêu ban đầu của đề tài “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện khuôn mặt kết hợp mã QR và mã vạch trên Raspberry Pi 4**” mục tiêu ban đầu giúp việc quản lý học sinh, sinh viên một cách chính xác, công bằng và minh bạch. Dễ dàng sử dụng và theo dõi đối tượng. Dựa trên ngôn ngữ Python với thư viện chính là OpenCV và được thực hiện trên Kit Raspberry. Sau quá trình nghiên cứu và thực hiện đề tài, nhóm thực hiện đã rút ra được nhiều vấn đề khác nhau, từ việc sử dụng phần mềm, các phương pháp giải thuật, cho tới sử dụng phần cứng. Thời gian thực hiện trong khoảng thời gian 17 tuần. Trong đó, gồm các vấn đề sau:

- Đối với phần cứng: Biết sử dụng Kit Raspberry, Kết nối và sử dụng màn hình cảm ứng điện dung, đưa ra ý tưởng thiết kế hộp chứa thiết bị và giá đỡ cho hệ thống, cài đặt hệ điều hành cho Raspberry, biết sử dụng Camera.
- Đối với phần mềm: Biết cách lập trình cơ bản trên Python cùng với thư viện OpenCV, lập trình thiết kế giao diện giao tiếp, và các vấn đề liên quan đến đề tài như: machine learning, deep learning, cơ sở dữ liệu SQL, thuật toán về nhận diện khuôn mặt, mã Barcode, mã QR,....

#### 5.1. Kết quả

##### 5.1.1. Kết quả phần cứng

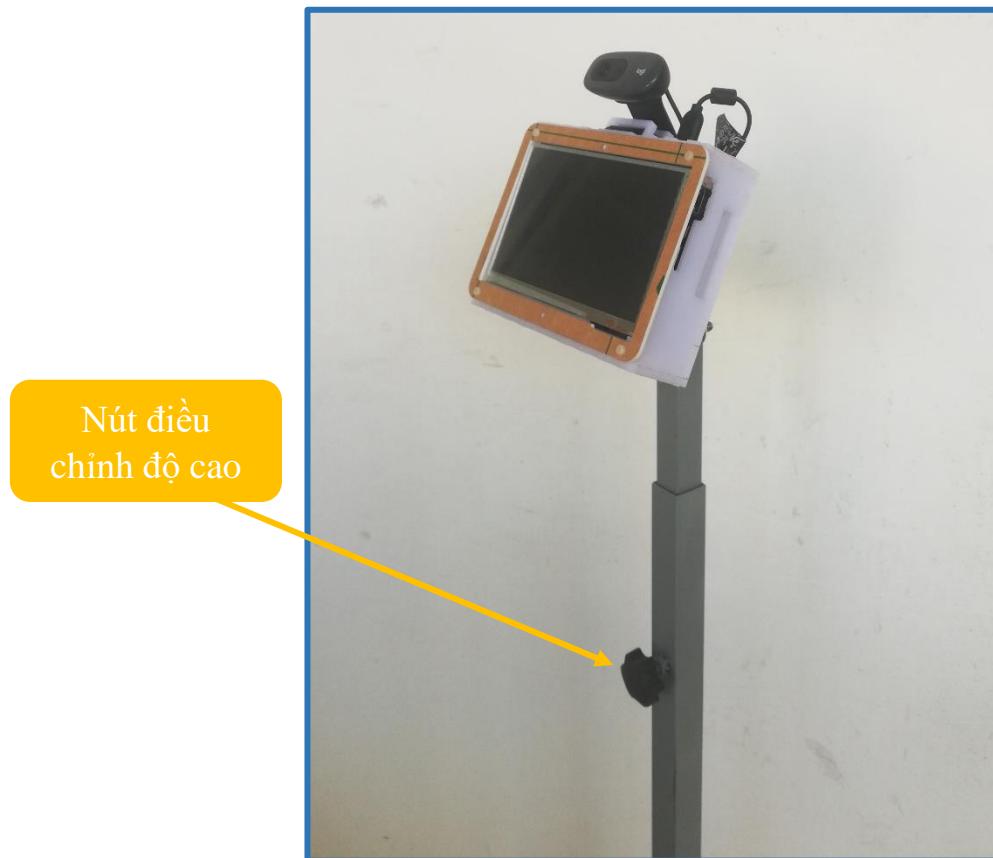
Mô hình hệ thống điểm danh hoàn thiện:



Hình 5. 1: Kết nối hoàn chỉnh phần cứng

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

Mô hình hệ thống điểm danh hoàn thiện bao gồm camera, màn hình giao tiếp chính, bộ xử lý Raspberry Pi 4, giá đỡ và dây kết nối.



Hình 5. 2: Lắp màn hình vào giá đỡ

Hệ thống hoạt động ổn định thời gian dài. Hệ thống có thể điều chỉnh độ cao thấp nhất 1 mét, cao nhất 1,9 mét. Hệ thống có thể điều chỉnh được góc nghiêng phù hợp lên đến 90 độ.

### 5.1.2. Kết quả phần mềm

Phần mềm nhóm đã viết và sử dụng các chương trình: lấy dữ liệu, huấn luyện dữ liệu, nhận diện, đọc mã QR và mã Barcode.

Để thực hiện các chương trình trên, người dùng giao tiếp với hệ thống thông qua giao diện trên màn hình cảm ứng điện dung.

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5. 3: Màn hình giao tiếp người dùng hoàn chỉnh.

### 5.1.3. Kết quả thực hiện thử nghiệm hệ thống

Để có một kết quả khách quan nhất và đánh giá hiệu quả hoạt động của hệ thống, nhóm thực hiện nghiên cứu tiến hành thử nghiệm cho việc điểm danh thực tế về nhận diện khuôn mặt và bằng thẻ sinh viên do chính nhóm thực hiện tự thiết kế.

Mô hình điểm danh của nhóm phải được thực hiện ở điều kiện ánh sáng vừa đủ không bị chói sáng hoặc bị thiếu sáng. Phòng nền phía sau khu vực điểm danh cùng một màu không có ngoại cảnh khác. Nhóm tiến hành thử nghiệm trên phòng nền màu trắng để điểm danh.

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5. 4: Phòng nền phía sau khu vực điểm danh

Như ta thấy ở hình 5.4, phòng nền thực tế phía sau người điểm danh (khung màu xanh lá) là bức tường màu trắng. Phòng nền hiển thị trong màn hình (khung màu vàng) không có ngoại cảnh thì hệ thống của nhóm thực hiện sẽ hoạt động tốt.

### a. Kết quả điểm danh bằng nhận diện khuôn mặt.

Thử nghiệm bao gồm 06 sinh viên là Lê Phú Hiếu, Nguyễn Đồng Quý, Nguyễn Thị Minh Trang, Nguyễn Thu Vy, Lê Thị Thu Ba và Trần Thanh Nho. Mỗi bạn thực hiện việc điểm danh 10 lần, riêng bạn Trần Thanh Nho không thực hiện lấy dữ liệu ban đầu, để kiểm tra trường hợp “không xác định” của hệ thống.

**Bước 1:** Thực hiện lấy dữ liệu.

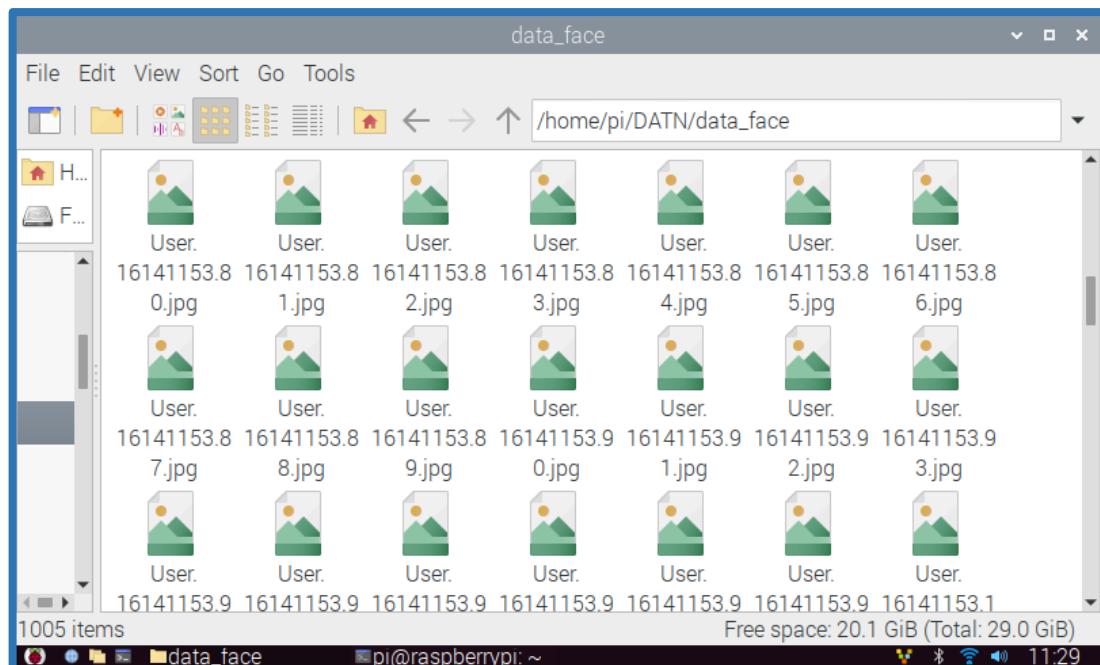
## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5. 5: Thực hiện lấy dữ liệu sinh viên Nguyễn Thu Vy

Trong bước này sẽ lấy dữ liệu 05 bạn sinh viên Phú Hiếu, Thu Vy, Đồng Quý, Minh Trang, Thu Ba. Mỗi sinh viên hệ thống sẽ tự động xác định vị trí khuôn mặt và chụp 200 tấm hình. Như vậy tổng thể ta có 1000 tấm hình là cơ sở dữ liệu. Mỗi ảnh sinh viên sẽ được lưu là mã số sinh viên trong trong data\_face.

Để thực hiện bước này sinh viên nhấn và phím “ LẤY DỮ LIỆU” trên màn hình cảm ứng.



Hình 5. 6: Ảnh sau khi lấy dữ liệu lưu trong data\_face.

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

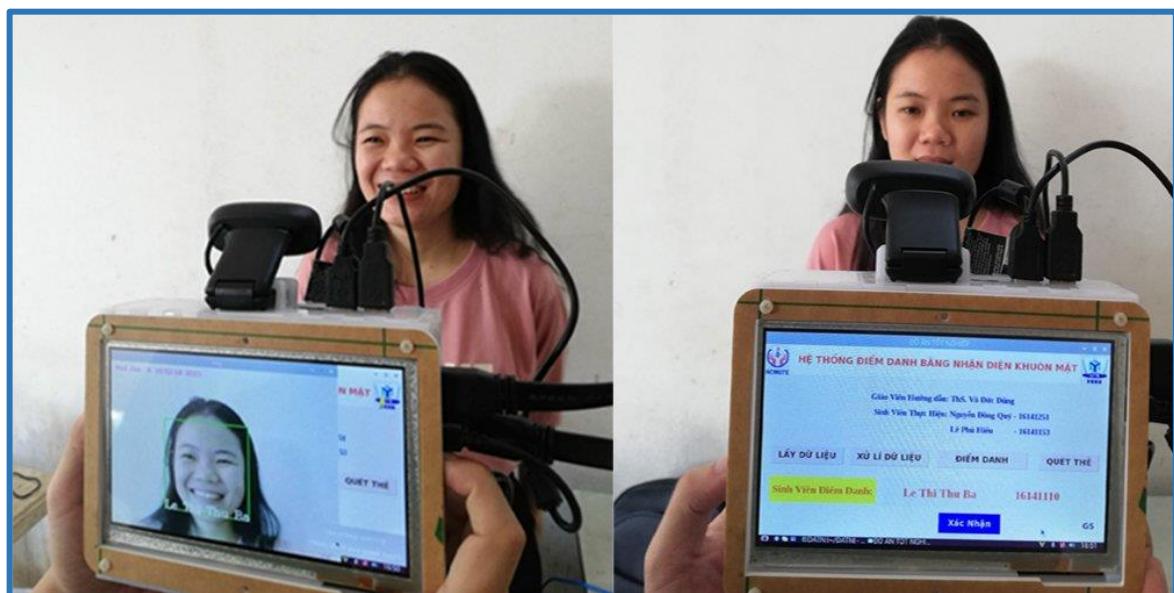
### Bước 2: Huấn luyện dữ liệu

Khi đã lấy hết dữ liệu của 05 bạn sinh viên, tiến hành cho hệ thống huấn luyện học tập dữ liệu.

Để thực hiện bước này sinh viên nhấn và phím “XỬ LÝ DỮ LIỆU” trên màn hình cảm ứng.

### Bước 3: Tiến hành điểm danh nhận diện khuôn mặt

Sinh viên nhấn vào phím “ĐIỂM DANH” trên màn hình cảm ứng. Khi có kết quả hiện ra, sinh viên kiểm tra đúng thông tin và bấm phím “XÁC NHẬN” để lưu kết quả.

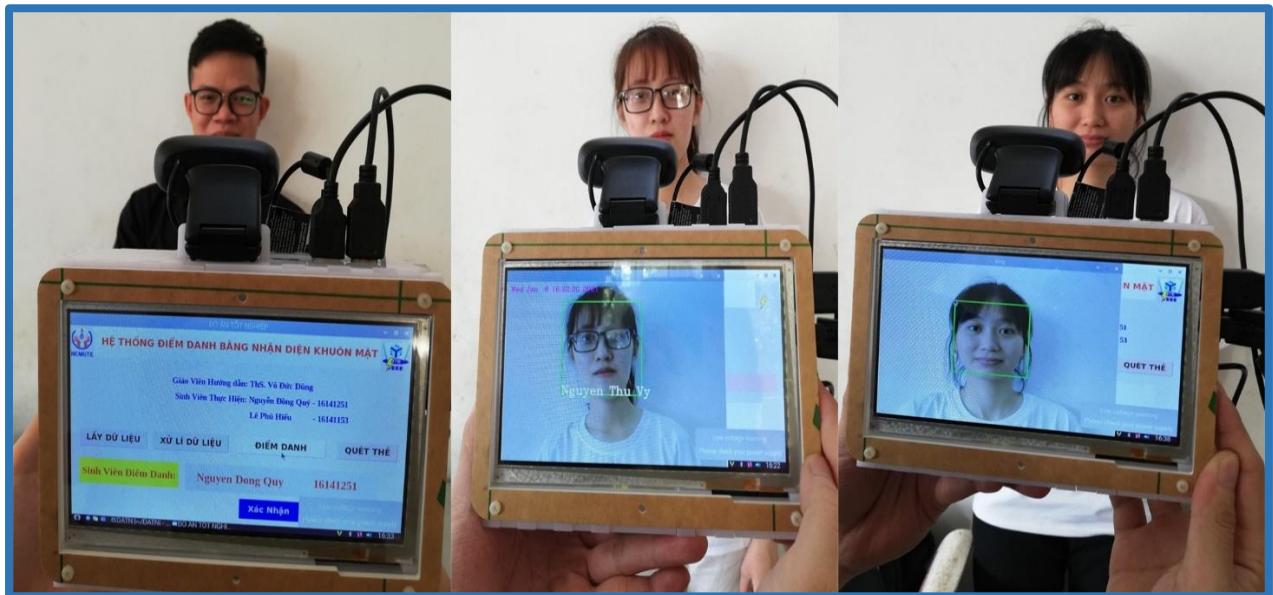


Hình 5. 7: Điểm danh sinh viên Lê Thị Thu Ba

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5. 8: Điểm danh sinh viên Lê Phú Hiếu



Hình 5. 9: Điểm danh các sinh viên khác

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ



Hình 5. 10: Trường hợp “ Không xác định” (Trần Thanh Nho).

Kết quả điểm danh được lưu vào file excel “StudentDetails”. Thông tin sinh viên gồm có mã số sinh viên, họ tên không dấu và thời gian thực khi điểm danh.

StudentDetails\StudentDetails.csv - Mousepad	
File	Edit
16141251, Nguyen Dong Quy, 2020-12-22, 09:28:37	
16141251, Nguyen Dong Quy, 2020-12-22, 14:08:51	
16141153, Le Phu Hieu, 2020-12-22, 14:09:06	
16141153, Le Phu Hieu, 2020-12-22, 14:09:14	
16141153, Le Phu Hieu, 2020-12-22, 14:09:25	
16141153, Le Phu Hieu, 2020-12-22, 14:09:43	
16141153, Le Phu Hieu, 2020-12-22, 14:09:51	
16141153, Le Phu Hieu, 2020-12-22, 14:10:01	
16141153, Le Phu Hieu, 2020-12-22, 14:10:35	
16141251, Nguyen Dong Quy, 2020-12-22, 14:10:55	
16141251, Nguyen Dong Quy, 2020-12-22, 14:11:03	
16141153, Le Phu Hieu, 2020-12-22, 14:11:15	
16141251, Nguyen Dong Quy, 2020-12-22, 14:11:22	
16141153, Le Phu Hieu, 2020-12-22, 14:11:58	
16141153, Le Phu Hieu, 2020-12-22, 14:12:11	
16141251, Nguyen Dong Quy, 2020-12-22, 14:12:21	
16141251, Nguyen Dong Quy, 2020-12-22, 14:12:34	
16141251, Nguyen Dong Quy, 2020-12-22, 14:27:12	
16141251, Nguyen Dong Quy, 2020-12-22, 15:14:28	
16141251, Nguyen Dong Quy, 2020-12-22, 15:15:25	
16141348, Nguyen Thu Vy, 2020-12-22, 15:41:46	

Hình 5. 11: Kết quả điểm danh bằng nhận diện khuôn mặt.

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

Sau khi tiến hành thực nghiệm việc điểm danh mỗi sinh viên 10 lần trong điều kiện thuận lợi. Nhóm thực hiện đã ghi chép và tổng hợp được bảng sau:

Bảng 5. 1: Kết quả tiến hành thực nghiệm điểm danh bằng nhận diện khuôn mặt.

STT	Họ và tên	Số lần chính xác/ 10 lần thử	Tỷ lệ chính xác	Tỉ lệ sai số
1	Lê Phú Hiếu	9	90%	10%
2	Nguyễn Đồng Quý	9	90%	10%
3	Nguyễn Thu Vy	8	80%	20%
4	Nguyễn Thị Minh Trang	10	100%	0%
5	Lê Thị Thu Ba	9	90%	10%
6	Trần Thanh Nho	10 (Không xác định)	100%	0%

Bảng 5. 2: Bảng thống kê chi tiết số lần đúng và số lần sai.

Người thực hiện điểm danh	Thông tin hiển thị trên màn hình								
	Phú Hiếu	Đồng Quý	Thu Vy	Minh Trang	Thu Ba	Thanh Nho	Không xác định	Độ chính xác	
Phú Hiếu	9	1	0	0	0	0	0	90%	
Đồng Quý	0	9	1	0	0	0	0	90%	
Thu Vy	0	1	8	0	0	0	1	80%	
Minh Trang	0	0	0	10	0	0	0	100%	
Thu Ba	0	0	0	0	9	0	1	90%	
Thanh Nho	0	0	0	0	0	0	10	100%	

b. Kết quả điểm danh bằng mã QR và mã vạch.

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

Để thí nghiệm điểm danh bằng thẻ, nhóm thực hiện đã tiến hành thiết kế thẻ sinh viên bao gồm 2 mặt: mặt trước là thông tin và mã QR code, mặt sau là mã vạch (Barcode).

Nhấn vào phím “QUÉT THẺ” trên giao diện điều khiển và đưa thẻ sinh viên đối diện với camera để điểm danh bằng thẻ.



Hình 5. 12: Thiết kế thẻ sinh viên

Bảng 5. 3: Kết quả tiến hành thực nghiệm điểm danh bằng mã QR.

STT	Họ và tên	Số lần chính xác/ 10 lần thử bằng mã QR	Tỷ lệ chính xác	Tỉ lệ sai số
1	Lê Phú Hiếu	10	100%	0%
2	Nguyễn Đồng Quý	10	100%	0%
3	Nguyễn Thu Vy	10	100%	0%
4	Nguyễn Thị Minh Trang	10	100%	0%

Bảng 5. 4: Kết quả tiến hành thực nghiệm điểm danh bằng mã vạch (Barcode).

STT	Họ và tên	Số lần chính xác/ 10 lần thử bằng mã vạch	Tỷ lệ chính xác	Tỉ lệ sai số

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

1	Lê Phú Hiếu	10	100%	0%
2	Nguyễn Đồng Quý	10	100%	0%
3	Nguyễn Thu Vy	10	100%	0%
4	Nguyễn Thị Minh Trang	10	100%	0%



Hình 5. 13: Kết quả điểm danh bằng thẻ QR code và barcode

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

StudentDetails\SStudentDetailBarQR.csv - Mousepad	
File	Edit
Id,Date,Time	
16141348,2021-01-12,14:39:24	
Id,Date,Time	
16141348,2021-01-12,14:39:31	
Id,Date,Time	
16141251,2021-01-12,14:44:05	
Id,Date,Time	
16141251,2021-01-12,14:44:21	
Id,Date,Time	
16141348,2021-01-12,14:45:06	
Id,Date,Time	
16141348,2021-01-12,14:45:14	
Id,Date,Time	
16141308,2021-01-12,14:46:01	
Id,Date,Time	
16141308,2021-01-12,14:46:13	
Id,Date,Time	
16141153,2021-01-12,14:47:01	
Id,Date,Time	
16141153,2021-01-12,14:47:16	

Hình 5. 14: Kết quả được lưu và file StudentDailBarQR.csv

### ❖ Tổng hợp kết quả thực nghiệm:

Bảng 5. 5: Bảng tổng hợp kết quả thực nghiệm.

STT	Cách thức điểm danh	Trung bình tỷ lệ chính xác	Trung bình tỷ lệ sai số	Đánh giá
1	Nhận diện khuôn mặt	92%	8%	Đạt
2	Mã QR code	100%	0%	Đạt
3	Mã vạch (Barcode)	100%	0%	Đạt

- Điểm danh bằng nhận diện khuôn mặt kết quả đạt được độ chính xác trung bình trên 92%. Thời gian điểm danh từ 10s – 15s trên một sinh viên.

- Điểm danh bằng mã QR code kết quả đạt được độ chính xác tuyệt đối 100%. Thời gian điểm danh từ 1s – 2s trên một sinh viên.

- Điểm danh bằng mã vạch (Barcode) kết quả đạt được độ chính xác tuyệt đối 100%. Thời gian điểm danh từ 2s – 3s trên một sinh viên.

- Kết quả được lưu và file excel gồm có họ tên, mã số sinh viên, thời gian thực để theo dõi và quản lý.

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

**Thực nghiệm ở điều kiện không thuận lợi: Có ánh sáng chiếu trực tiếp, phía sau khu vực điểm danh có ngoại cảnh tác động.**



Hình 5. 15: Thực nghiệm ở điều kiện không thuận lợi

Sau khi thực nghiệm ở môi trường ánh sáng chiếu trực tiếp, có ngoại cảnh tác động, nhóm đạ ghi chép và tổng hợp lại như sau:

**\* Điểm danh bằng nhận diện khuôn mặt:**

Bảng 5. 6: Kết quả tiến hành thực nghiệm điểm danh bằng nhận diện khuôn mặt điều kiện không thuận lợi.

STT	Họ và tên	Số lần chính xác/ 10 lần thử	Tỷ lệ chính xác	Tỉ lệ sai số
1	Lê Phú Hiếu	8	80%	20%
2	Nguyễn Đồng Quý	7	70%	30%
3	Nguyễn Thu Vy	7	70%	30%
4	Nguyễn Thị Minh Trang	9	90%	10%

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

Bảng 5. 7: Bảng ma trận nhầm lẫn điểm danh bằng nhận diện khuôn mặt điều kiện không thuận lợi.

Người thực hiện điểm danh	Thông tin hiển thị trên màn hình						
		Phú Hiếu	Đồng Quý	Thu Vy	Minh Trang	Không xác định	Độ chính xác
Phú Hiếu	8	1	0	0	2		80%
Đồng Quý	0	7	2	0	1		70%
Thu Vy	0	1	7	0	2		70%
Minh Trang	1	0	0	9	0		90%

Khi thực hiện ở môi trường không thuận lợi trên tỷ lệ nhận diện chính xác còn lại 77,5%. Trường hợp không xác định tăng lên so với lần thực nghiệm trước.

Một số nguyên nhân là cho việc nhận diện sai chính là do có nhiều ngoại cảnh tác động hệ thống xác định sai vị trí khuôn mặt. Hoặc do ánh sáng chiếu trực tiếp vào camera làm thay đổi cường độ sáng tối của từng pixel.

### \* Điểm danh bằng mã QR và mã vạch:

Bảng 5. 8: Kết quả tiến hành thực nghiệm điểm danh bằng mã QR và mã vạch trong môi trường không thuận lợi.

STT	Họ và tên	Tỷ lệ chính xác bằng mã QR	Tỷ lệ chính xác bằng mã vạch
1	Lê Phú Hiếu	100%	100%
2	Nguyễn Đồng Quý	100%	90%
3	Nguyễn Thu Vy	100%	100%
4	Nguyễn Thị Minh Trang	100%	100%

Kết quả gần như được giữ nguyên so với trường hợp thuận lợi. Chỉ có trường hợp sinh viên Nguyễn Đồng Quý đọc sai mã vạch 01/10 lần thử.

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

Tuy nhiên tốc độ đọc được mã vạch tăng lên từ 4 – 5 giây.

### **Trường hợp thực nghiệm khác:**

\* **Điểm danh nhận diện khuôn mặt mang khẩu trang:**



Hình 5. 16: Sinh viên thử nghiệm mang khẩu trang khi điểm danh

Sau thực hiện 10 lần việc mang khẩu trang khi điểm danh nhận diện khuôn mặt. Tỷ lệ nhận diện chính xác khuôn mặt là 30%, nhận diện sai là 40% , không xác định là 30%. Hệ thống xác định vị trí khuôn mặt tốn nhiều thời gian hơn.

\* **Điểm danh bằng mã QR bị che khuất một số khu vực**



## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

Hình 5. 17: Các phần mã vạch bị che khuất

Nếu mã QR bị che khuất các khu vực nằm trong khu màu đỏ thì hệ thống sẽ không đọc được. Phần mã trong khung màu xanh, hệ thống vẫn đọc được mã QR.

### \* *Điểm danh bằng mã vạch bị đứt hoặc mất*



Hình 5. 18: Mã vạch bị đứt hoặc mất

Khi khảo sát trường hợp mã vạch bị đứt hoặc mất một phần thì hệ thống vẫn có thể đọc chính xác.

### \* *Điểm danh bằng mã thẻ sử dụng lâu bị phai màu và mờ*



Hình 5. 19: Mã thẻ bị phai màu và mờ

Khi khảo sát trường hợp mã thẻ bị phai màu và mờ do sử dụng lâu hệ thống vẫn có thể đọc chính xác.

## 5.2. Nhận xét và đánh giá

### 5.2.1. Nhận xét

Dựa vào kết quả quá trình thực nghiệm nhóm thực hiện có một số nhận xét sau:

+ Trong môi trường thuận lợi ánh sáng vừa phải, phông nền phía sau là tường trắng: Khi điểm danh bằng nhận dạng khuôn mặt tỷ lệ nhận dạng đúng trung bình trên 90%. Tốc độ điểm danh một sinh viên từ 10 – 15 giây. Khi điểm danh bằng thẻ với mã

## CHƯƠNG 5: KẾT QUẢ – NHẬN XÉT VÀ ĐÁNH GIÁ

QR và mã vạch (Barcode) tỉ lệ chính xác là 100%. Tốc độ điểm danh từ 1 - 2 giây/ sinh viên.

- + Những sinh viên đeo kính có thể bị nhận diện sai, nhưng tỉ lệ sai có thể chấp nhận được.
- + Khi khởi động hệ thống và lấy thông tin sinh viên còn khá lâu từ 1 phút đến 2 phút.
- + Mô hình nhận diện khuôn mặt chính xác khi camera không bị ánh sáng chiếu trực tiếp vào, phía sau khu vực người điểm danh không có nhiều ngoại cảnh, phông nền một màu, như trong thực nghiệm nhóm đã sử dụng phông nền trắng.
- + Khi càng nhiều dữ liệu được thu thập thì tốc độ xử lý của hệ thống càng chậm.
- + Chế độ giám sát chỉ được từ 2 – 3 sinh viên tốc độ xử lý chậm không như mong muốn.

### 5.2.2. Đánh giá

Đánh giá một cách khách quan tốc độ xử lý ảnh, tỷ lệ chính xác khi điểm danh theo hệ thống mục tiêu ban đầu đạt kết quả khá ổn. Tuy nhiên, với mô hình thực tế cần điểm danh nhanh với độ chính xác cao hơn nữa thì Raspberry chưa đáp ứng được nhu cầu. Vì thời gian thực hiện đề tài giới hạn, cũng như giới hạn về mặt chi phí nên nhóm thực hiện chưa giải quyết được hết các vấn đề phát sinh hay cải tiến thêm để hệ thống có thể đạt hiệu quả cao hơn, đáp ứng nhu cầu thực tế hơn mà chỉ dừng lại ở việc xây dựng mô hình nhỏ gọn, đơn giản, mô phỏng, giải quyết được mục tiêu căn bản đề ra là điểm danh bằng nhận diện khuôn mặt kết hợp thêm điểm danh QR code và Barcode, bằng kiến thức xử lý ảnh, máy học và điều khiển thông minh đã được học tập trên trường.

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1. Kết luận

Sau thời gian nghiên cứu, tìm hiểu với sự giúp đỡ của thầy ThS. Võ Đức Dũng, đề tài “**Thiết kế và thi công hệ thống điểm danh bằng nhận diện khuôn mặt kết hợp mã QR và mã vạch trên Raspberry Pi 4**” mà nhóm đã hoàn thành đáp ứng tốt các yêu cầu đề ra. Các khái niệm hoạt động đúng, đảm bảo cập nhật dữ liệu chính xác, hệ thống chạy ổn định trong thời gian dài không phát sinh lỗi, còn riêng vấn đề sử dụng thì khá dễ dàng, an toàn, thân thiện phụ hợp trong môi trường giáo dục.

- **Đồ án do nhóm thực hiện đã đạt được khoảng 95% mục tiêu đặt ra.**
- Raspberry Pi 4 còn khá hạn chế khi quá nhiều dữ liệu và nhiều tác vụ cùng một lúc thì không thể đáp ứng tốc độ xử lý nhanh được.
- Camera Logitech C270, tuy có độ phân giải không quá cao nhưng ngược lại nó có giá thành phù hợp, khả năng kết nối linh hoạt cùng với chất lượng ảnh phù hợp với việc xử lý ảnh.
- Sau nhiều lần chạy thực nghiệm, mô hình hệ thống đã thiết kế và xây dựng hoạt động tương đối ổn định: thời gian từ chụp ảnh cho đến khi đưa ra kết quả nhận dạng cho mỗi sinh viên là 10s – 15s. Qua nhiều lần khảo sát, hệ thống nhận dạng sai với sai số trung bình nhỏ hơn 10% đối với điểm danh bằng nhận diện khuôn mặt. Chính xác tuyệt đối 100% đối với điểm danh bằng thẻ sinh viên.

#### 6.1.1. Ưu điểm

- Là chương trình được lập trình do đó tiết kiệm được chi phí thiết kế và thi công phần cứng.
- Do kích thước mô hình nhỏ nên khả năng linh hoạt và tiện lợi sẽ dễ dàng hơn. Từ đó có thể phát triển hệ thống một cách hiệu quả hơn.
- Giao diện hiển thị đẹp, rõ ràng có tính thẩm mỹ cao, độ chính xác khi điểm danh bằng nhận diện khuôn mặt trên 90% và điểm danh bằng thẻ đạt lên đến 100% tỉ lệ chính xác.
- Kết quả thực tế trên Kit Raspberry Pi 4 cho độ chính xác khá cao.
- Đáp ứng được nhu cầu thực tế, dễ dàng nghiên cứu và mở rộng thêm.

## CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1.2. Hạn chế

- Hệ thống của nhóm hoạt động tốt nhất khi các khuôn mặt điểm danh phải chính diện với vị trí camera, ánh sáng vừa phải không chói sáng hoặc thiếu sáng. Phòng nền phía sau khu vực điểm danh cùng một màu không có ngoại cảnh.
- Để đáp ứng nhu cầu thực tế là tốc độ xử lý phải nhanh, bộ xử lý trung tâm kit Raspberry Pi chưa đủ mạnh để giải quyết vấn đề tốc độ điểm danh.

### 6.2. Hướng phát triển

#### 6.2.1. Hướng khắc phục

- Tiền xử lý ảnh qua nhiều lớp sẽ giúp hệ thống cho ra kết quả chuẩn xác hơn.
- Điều kiện ngoại cảnh cần phù hợp về khoảng cách, ánh sáng, hạn chế các đối tượng làm nhiễu khung ảnh, hướng và góc nghiêng của khuôn mặt,... để đảm bảo độ tin cậy và giảm sai số tối mức thấp nhất cho hệ thống.
- Cải thiện Camera với chất lượng tốt hơn có thể sử dụng Camera hồng ngoại để không phụ thuộc vào điều kiện thiếu sáng.
- Nâng cấp Kit Raspberry Pi lên phiên bản cao hơn để việc xử lý dữ liệu một cách nhanh hơn.

#### 6.2.2. Hướng phát triển

Kết hợp với ứng dụng điện thoại thông minh, webserver trong việc điểm danh và quản lý thông tin sinh viên, học sinh của toàn trường.

## TÀI LIỆU THAM KHẢO

- [1] Trần Thị Linh Đa, Nguyễn Châu Ngân, Thiết kế và thi công hệ thống điểm danh nhân viên sử dụng vi điều khiển ARM, Đồ án tốt nghiệp, trường ĐHSPKT, Tp. HCM, 2019.
- [2] Lê Hoàng Thành, Hồ Đình Vương, Thiết kế và thi công hệ thống bảo mật ứng dụng xử lý ảnh, Đồ án tốt nghiệp, trường ĐHSPKT, Tp. HCM, 2018.
- [3] PGS. TS Nguyễn Thanh Hải, “Giáo trình Xử Lý Ảnh”, Nhà xuất bản ĐH Quốc Gia, Tp. HCM, 2014.
- [4] Trần Văn Tuấn, Phạm Văn Long, Thiết kế và thi công hệ thống phân loại sản phẩm bít chì theo màu sắc, Đồ án tốt nghiệp, Trường ĐH SPKT, Tp. HCM, 2020.
- [5] Hà Duy Khánh, Đồ án Phân loại sản phẩm dùng Kit Raspberry, Đồ Án Tốt Nghiệp Đại Học, Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh, 2018.
- [6] Prof. Bojana Dalbelo Basic Assoc. Prof. Jan Snajder, “Artificial Intelligence”, University of Zagreb, Faculty of Electrical Engineering and Computing, 2018-2019.
- [7] Trần Như Ý, Nguyễn Văn Tùng, Ngô Dương Hà, Phương pháp phát hiện người dựa trên chuyển động, Trường ĐH Công Nghệ Thực Phẩm, Tp. HCM, 2017.
- [8] Nông Văn Tân, Trần Thị Mỹ Khiêm, Ứng dụng xử lý ảnh vào việc phát hiện ngủ gật dùng kit Raspberry, Đồ án tốt nghiệp, Trường ĐH SPKT, tp.HCM, 2018.
- [9] Vo Anh Quoc, Tran Van Son, Building attendance system prototype based on face recognition using machine learning, Đồ án tốt nghiệp, ĐH. SPKT, Tp. HCM, 2019.
- [10] TS. Phạm Ngọc Hưng, TS. Phạm Ngọc Thuận, Hướng dẫn lập trình đọc mã Barcode – QR Code, ĐH Bách Khoa Hà Nội.  
([https://sites.google.com/site/embedded247/npcourse/impcourse/barcode\\_qrcode](https://sites.google.com/site/embedded247/npcourse/impcourse/barcode_qrcode))

## PHỤ LỤC

Code chương trình:

```
from tkinter import *
import os
import sqlite3
import cv2
import time
import numpy as np
from PIL import ImageTk, Image
import pandas as pd
import datetime
import csv
from pyzbar.pyzbar import decode

win = Tk()
win.title('ĐỒ ÁN TỐT NGHIỆP')

LbMain = Label(win, text = "HỆ THỐNG ĐIỂM DANH BẰNG NHẬN DIỆN KHUÔN MẶT",
bg = 'White', fg = 'red',
font = 'verdana 16 bold ', width =100, padx = 450, pady =
20)
LbMain.pack()

infogv = Label(win, text="Giáo Viên Hướng dẫn: ThS. Võ Đức Dũng" ,fg="blue"
, width=40 ,height=2 ,font=('times', 14, ' bold '))
infogv.place(x=230, y=90)

infosvl = Label(win, text="Sinh Viên Thực Hiện: Nguyễn Đồng Quý - 16141251"
,fg="blue" ,width=50 ,height=2 ,font=('times', 14, ' bold '))
infosvl.place(x=230, y=125)

infosv2 = Label(win, text="Lê Phú Hiếu - 16141153" ,fg="blue"
, width=40 ,height=2 ,font=('times', 14, ' bold '))
infosv2.place(x=365, y=160)

lbl3 = Label(win, text="Sinh Viên Điểm Danh:",width=20 ,fg="red"
, bg="yellow" ,height=2 ,font=('times', 18, ' bold '))
lbl3.place(x=20, y=300)

messageName = Label(win, text="" ,fg="red" ,bg="white",activeforeground =
"green",width=20 ,height=2 ,font=('times', 20, ' bold '))
messageName.place(x=280, y=300)

messageMssv = Label(win, text="" ,fg="red" ,bg="white",activeforeground =
"green",width=15 ,height=2 ,font=('times', 20, ' bold '))
messageMssv.place(x=540, y=300)

photo = PhotoImage(file = r"logoSPKT.png")
photoLogo = photo.subsample(23,23)
labelSPKT = Label(win, image = photoLogo).place(x= 1, y=1)

photo1 = PhotoImage(file = r"logoDDT.png")
photoLogo1 = photo1.subsample(10,10)
labelSPKT1 = Label(win, image = photoLogo1).place(x= 730, y=1)

def TakePhoto():
    def insertOrUpdate(id, name):
        # connecting to the db
```

```

conn = sqlite3.connect("Facebase2.db")
# check if id already exists
query = "SELECT * FROM People WHERE ID=" + str(id)
# returning the data in rows
cursor = conn.execute(query)
isRecordExist = 0
for row in cursor:
    isRecordExist = 1
if isRecordExist == 1:
    query = "UPDATE People SET Name=" + str(name) + " WHERE ID=" +
str(id)
else:
    query = "INSERT INTO People(ID, Name) VALUES(" + str(id) + ", "
+ str(name) + ")"
conn.execute(query)
conn.commit()
conn.close()

root =Tk()
face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)
def myclickOK():
    id = textcommand1.get()
    name = textcommand2.get()
    insertOrUpdate(id, name)
    sample_number = 0
    while True:
        ret, img = cap.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        for (x, y, w, h) in faces:
            sample_number += 1

            if not os.path.exists('data_face'):
                os.makedirs('data_face')

            cv2.imwrite('data_face/User.' + str(id) + "." +
str(sample_number) + ".jpg", img[y:y + h, x:x + w])
            cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

            cv2.imshow('img', img)
            cv2.waitKey(1);
            if (sample_number > 200):
                cap.release()
                cv2.destroyAllWindows()
                break;
#REGISTER INFORMATION THEMES
mssv = Label(root, text = "MSSV:")
mssv.pack()
textcommand1 = Entry(root, width = 50)
textcommand1.pack()
HovaTen = Label(root, text="Ho va Ten:")
HovaTen.pack()
textcommand2 = Entry(root, width=50)
textcommand2.pack()
myButtonOK = Button(root, text = "OK", command = myclickOK)
myButtonOK.pack()
root.mainloop()

def TrainData():
    recognizer = cv2.face.LBPHFaceRecognizer_create()

```

```

path = 'data_face'

def getImagesWithID(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    faces = []
    IDs = []

    for imagePath in imagePaths:
        faceImg = Image.open(imagePath).convert('L')
        faceNp = np.array(faceImg, 'uint8')
        ID = int(os.path.split(imagePath)[-1].split('.')[1])
        faces.append(faceNp)
        IDs.append(ID)
        cv2.imshow('training', faceNp)
        cv2.waitKey(10)
    return np.array(IDs), faces

print("\n [INFO] Training faces. It will take a few seconds. Wait ...")

Ids, faces = getImagesWithID(path)
recognizer.train(faces, Ids)

if not os.path.exists('trainer'):
    os.makedirs('trainer')

recognizer.save('huanluyen/huanluyen.yml')
print("\n [INFO] {} faces trained. Exiting Program".format(len(np.unique(Ids))))
cv2.destroyAllWindows()

def AttendentData():
    face_cascade =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read("huanluyen/huanluyen.yml")
    columnName = ['Id', 'Name', 'Date', 'Time']
    attendant = pd.DataFrame(columns=columnName)
    # TIME
    localtime = time.localtime(time.time())
    localtime_fulltime = time.asctime(localtime)

    def getProfile(Id):
        conn = sqlite3.connect("Facebase2.db")
        query = "SELECT * FROM People WHERE ID=" + str(Id)
        cursor = conn.execute(query)
        profile = None
        for row in cursor:
            profile = row
        conn.close()
        return profile

    cap = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_COMPLEX
    sample_number = 0
    while True:
        ret, img = cap.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.3, 5)
        img = cv2.putText(img, localtime_fulltime, (10, 25), font, 0.6,
(255, 0, 255), 1, cv2.LINE_AA)
        for (x, y, w, h) in faces:
            sample_number +=1

```

```

        cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
        roi_gray = gray[y:y + h, x:x + w]
        roi_color = img[y:y + h, x:x + w]

        Id, conf = recognizer.predict(roi_gray)
        if conf < 60:
            profile = getProfile(Id)
            if profile != None:

                cv2.putText(img, str(profile[1]), (x, y + h), font, 1, (255, 255, 255), 2)
                mssvOut = profile[0]
                NameOut = profile[1]

            else:
                cv2.putText(img, "Unknown", (x, y + h), font, 1, (255, 255, 255), 2)
                mssvOut = " "
                NameOut = "Không xác định!"
        attendant = attendant.drop_duplicates(subset=['Id'], keep='first')
        cv2.imshow('img', img)
        if (cv2.waitKey(1) == ord('q')):
            break
        if (sample_number > 0):
            cap.release()
            cv2.destroyAllWindows()
            break;
    messageName.configure(text = str(NameOut))
    messageMssv.configure(text = str(mssvOut))

def confirm():
    NameOut = messageName.cget('text')
    MssvOut = messageMssv.cget('text')
    ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    with open("StudentDetails\StudentDetails.csv", 'a+') as file:
        writer = csv.writer(file)
        writer.writerow([MssvOut, NameOut, date, timeStamp])

def manager():
    face_cascade =
    cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read("huanluyen/huanluyen.yml")
    # TIME
    localtime = time.localtime(time.time())
    localtime_fulltime = time.asctime(localtime)

    def getProfile(Id):
        conn = sqlite3.connect("Facebase2.db")
        query = "SELECT * FROM People WHERE ID=" + str(Id)
        cursor = conn.execute(query)
        profile = None
        for row in cursor:
            profile = row
        conn.close()
        return profile

    cap = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_COMPLEX
    sample_number = 0
    while True:
        ret, img = cap.read()

```

```

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
img = cv2.putText(img, localtime_fulltime, (10, 25), font, 0.6,
(255, 0, 255), 1, cv2.LINE_AA)
for (x, y, w, h) in faces:
    sample_number += 1
    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
    roi_gray = gray[y:y + h, x:x + w]
    roi_color = img[y:y + h, x:x + w]

    Id, conf = recognizer.predict(roi_gray)
    if conf < 50:
        profile = getProfile(Id)
        if profile != None:
            cv2.putText(img, str(profile[1]), (x, y + h), font, 1,
(255, 255, 255), 2)
    else:
        cv2.putText(img, "Unknown", (x, y + h), font, 1, (255, 255,
255), 2)
cv2.imshow('img', img)
if (cv2.waitKey(1) == ord('q')):

    cap.release()
    cv2.destroyAllWindows()
    break
def barcode():
    cap = cv2.VideoCapture(0)
    cap.set(3, 640)
    cap.set(4, 480)
    count = 0
    while True:
        success, img = cap.read()
        for barcode in decode(img):
            count += 1
            print(barcode.data)
            myData = barcode.data.decode('utf-8')
            print(myData)
            ts = time.time()
            date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
            timeStamp =
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            with open("StudentDetails\SStudentDetailBarQR.csv", 'a+') as
datafile:
                writer = csv.writer(datafile)

                writer.writerow(["Id", "Date", "Time"])

                writer.writerow([myData, date, timeStamp])
            pts = np.array([barcode.polygon], np.int32)
            pts = pts.reshape((-1, 1, 2))
            cv2.polylines(img, [pts], True, (255, 0, 255), 5)
            pts2 = barcode.rect
            cv2.putText(img, myData, (pts2[0], pts2[1]),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0, 255), 2)

            cv2.imshow('Result', img)
            if (cv2.waitKey(1) == ord('q')):
                break
            if (count > 0):
                cap.release()
                cv2.destroyAllWindows()

```

```

        break;
messageName.configure(text = myData)
messageMssv.configure(text = ' ')

Photo = Button(win, text = "LẤY DỮ LIỆU", command = TakePhoto, bg ='pink',
fg = 'black', font = 'verdana 14 bold', padx = 15, pady = 7)
Photo.place( x = 20, y = 230)

Train = Button(win, text = "XỬ LÍ DỮ LIỆU", command = TrainData, bg
='pink', fg = 'black', font = 'verdana 14 bold', padx = 15, pady = 7)
Train.place( x = 200, y = 230)

Test = Button(win, text = "ĐIỂM DANH FACE", command = AttendentData, bg
='pink', fg = 'black', font = 'verdana 14 bold', padx = 15, pady = 7)
Test.place( x = 400, y = 230)

BarCode = Button(win, text = "QUÉT THẺ", command = barcode, bg ='pink', fg
= 'black', font = 'verdana 14 bold', padx = 15, pady = 7)
BarCode.place( x = 640, y = 230)

ConfirmAttendent = Button(win, text = "Xác Nhận", command = confirm, bg
='blue', fg = 'white', font = 'verdana 14 bold', padx = 20, pady = 10)
ConfirmAttendent.place( x = 410, y = 375)

Manager = Button(win, text = "GS", command = manager, bg ='white', fg =
'black', font = 'verdana 14 bold', padx = 5, pady = 5)
Manager.place( x = 740, y = 380)

win.mainloop()

```

