

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN ĐIỆN TỬ**

-----oo-----



**LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC**

**HỆ THỐNG NHẬN DẠNG NGƯỜI BẰNG GIỌNG NÓI  
VÀ KHUÔN MẶT**

**GVHD: ThS HỒ TRUNG MỸ**

**SVTH: LÊ TIẾN ĐẠT**

**MSSV: 1710948**

**SVTH: VÕ MAI TRÍ LUẬN**

**MSSV: 1712083**

**TP. HỒ CHÍ MINH, THÁNG 8 NĂM 2021**

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  
TRƯỜNG ĐẠI HỌC BÁCH KHOA

Độc lập – Tự do – Hạnh phúc.

-----☆-----

-----☆-----

Số: \_\_\_\_\_ /BKĐT

Khoa: **Điện – Điện tử**

Bộ Môn: **Điện Tử**

## NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP

1. HỌ VÀ TÊN: **LÊ TIẾN ĐẠT** MSSV: 1710948  
**VÕ MAI TRÍ LUẬN** MSSV: 1712083
2. NGÀNH: **ĐIỆN TỬ - VIỄN THÔNG** LỚP: DD17DV02
3. Đề tài: Hệ thống nhận dạng người bằng giọng nói và khuôn mặt
4. Nhiệm vụ (Yêu cầu về nội dung và số liệu ban đầu):
  - Thiết kế hệ thống nhận dạng người chính chủ bằng giọng nói và khuôn mặt. Nhận dạng giọng nói bao gồm người nói và mật khẩu sáu số ngẫu nhiên, các số từ “Không” đến “Chín”
  - Lựa chọn thuật toán nhận dạng và mô phỏng lại các thuật toán trên phần mềm MATLAB.
  - Xây dựng lại các thuật toán và thực hiện trên phần cứng và tự lập trình cho phần cứng
  - Hệ thống nhận dạng tiêu thụ điện năng thấp và bảo mật thông tin cho người dùng.
  - Tỷ lệ nhận dạng tốt
5. Ngày giao nhiệm vụ luận văn: 17/03/2021
6. Ngày hoàn thành nhiệm vụ: 23/07/2021
7. Họ và tên người hướng dẫn:  
**ThS. HỒ TRUNG MỸ** Phản hướng dẫn  
.....

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

Tp.HCM, ngày..... tháng..... năm 20  
**CHỦ NHIỆM BỘ MÔN**

**NGƯỜI HƯỚNG DẪN CHÍNH**

### PHẦN DÀNH CHO KHOA, BỘ MÔN:

Người duyệt (chấm sơ bộ):.....

Đơn vị:.....

Ngày bảo vệ : .....

Điểm tổng kết: .....

Nơi lưu trữ luận văn: .....

## LỜI CẢM ƠN

*Lời đầu tiên, cho chúng em xin gửi lời cảm ơn chân thành nhất đến người thầy đã giúp đỡ và hỗ trợ chúng em hoàn thành đề cương lần này, đó là thầy Hồ Trung Mỹ, thầy đã cho chúng em nhiều lựa chọn trong việc chọn đề tài luận văn cũng như hướng dẫn và gợi ý cho chúng em rất nhiều điều trong quá trình làm luận văn. Nếu không có thầy định hướng, giúp đỡ chúng em thì chúng em khó lòng hoàn thành được đề tài một cách tốt nhất.*

*Thứ hai, chúng em xin gửi lời cảm ơn chân thành nhất đến các thầy cô giảng viên bộ môn Điện tử nói riêng cũng như khoa Điện-Điện Tử ở trường đại học Bách Khoa TP.HCM nói chung đã hỗ trợ và giúp đỡ cho chúng em trong quá trình làm luận văn.*

*Ngoài ra, chúng em cũng xin gửi lời cảm ơn sâu sắc nhất đến bạn bè, gia đình, tất cả mọi người đã giúp đỡ, động viên và hỗ trợ chúng em trong thời gian thực hiện cho đến lúc hoàn thành luận văn.*

*Trong quá trình nghiên cứu và thực hiện luận văn, chúng em mặc dù đã cố gắng hết sức để hoàn thành luận văn, song do khả năng cũng như kinh nghiệm của bản thân chúng em có hạn nên luận văn vẫn không thể tránh khỏi những hạn chế nhất định và còn một số thiếu sót. Vì vậy, chúng em rất mong nhận được các sự góp ý chân thành từ các thầy, các cô và các bạn để bổ sung và hoàn thiện hơn trong lần nghiên cứu tiếp theo của chúng em.*

*Xin chân thành cảm ơn tất cả mọi người!*

Tp. Hồ Chí Minh, ngày 23 tháng 07 năm 2021 .

Sinh viên

Lê Tiến Đạt Võ Mai Trí Luận

## TÓM TẮT LUẬN VĂN

Luận văn này trình bày về công nghệ nhận dạng con người dựa trên sinh trắc học. Ở đề tài lần này, nhóm em sẽ thực hiện một hệ thống nhận dạng đặc trưng giọng nói, mật khẩu giọng nói và khuôn mặt của một người chính chủ được chỉ định từ trước. Đối với phần nhận dạng đặc trưng giọng nói và mật khẩu của giọng nói, nhóm em sử dụng giải thuật trích xuất đặc trưng tiếng nói MFCC và Vector Quantization để huấn luyện và nhận dạng. Ở phần nhận dạng giọng nói này, nhóm em sử dụng vi điều khiển STM32F407VG Discovery cho việc xử lý và nhận dạng. Đối với nhận dạng khuôn mặt, vì lý do không đủ khả năng để thực hiện trên vi điều khiển nên nhóm em đã chọn máy tính nhúng Raspberry Pi 4 để thực hiện giải thuật Principal Component Analysis (PCA) cho việc trích xuất đặc trưng và huấn luyện để nhận dạng khuôn mặt. Khi hệ thống nhận dạng được mật khẩu và đặc trưng giọng nói của người dùng là chính chủ, người dùng sẽ tiếp tục nhận dạng khuôn mặt. Thông qua hệ thống, khi ba lớp bảo mật đều nhận dạng đúng chính chủ, hệ thống sẽ bật khóa chốt cửa và hiển thị thông báo cho người dùng qua LCD. Kết quả giải thuật nhận dạng trên MATLAB đạt hiệu suất khá cao, nhưng khi thực hiện trên phần cứng, do những điều kiện về dữ liệu, giải thuật, phần cứng chưa được tối ưu hóa nên kết quả nhận dạng thấp hơn so với nhận dạng trên máy tính.

## MỤC LỤC

DANH SÁCH HÌNH .....	vi
1. GIỚI THIỆU .....	1
1.1    Tổng quan .....	1
1.2    Tình hình nghiên cứu trong và ngoài nước .....	2
1.3    Nhiệm vụ luận văn .....	4
2. LÝ THUYẾT .....	6
2.1 Nhữn g vấn đề cơ bản của tiếng nói con người .....	6
2.1.1    Nguyên lý hình thành tiếng nói con người.....	6
2.1.2    Đặc tính vật lý của tiếng nói .....	6
2.1.3 Cơ chế hoạt động của tai.....	7
2.2    Tìm hiểu về lý thuyết cơ bản của nhận dạng khuôn mặt .....	8
2.3 Tổng quan về nhận dạng giọng nói và nhận dạng người nói .....	9
2.3.1 Sơ lược về nhận dạng giọng nói và nhận dạng người nói.....	9
2.3.2 Phân loại các hệ thống nhận dạng giọng nói.....	10
2.3.3 Các hướng nghiên cứu .....	11
2.3.4 Cơ sở lý thuyết về xử lý tín hiệu.....	12
2.3.5 Trích xuất đặc trưng tiếng nói.....	16
2.3.6 Các phương pháp huấn luyện và nhận dạng .....	26
2.4 Tìm hiểu lý thuyết về công nghệ nhận dạng khuôn mặt .....	31
2.4.1 Phát hiện khuôn mặt (Face Detection).....	31
2.4.2 Nhận dạng khuôn mặt (Face Recognition) .....	36
2.5 Áp dụng thuật toán PCA cho nhận dạng khuôn mặt.....	38
2.5.1    Giới thiệu và lý do chọn PCA.....	38
2.5.2    Cơ sở toán học .....	38
2.5.3    Thuật toán PCA .....	41
2.5.3 Nhận dạng khuôn mặt .....	46

3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG .....	47
3.1    Đặc tả phần cứng .....	47
3.2    Sơ đồ khối .....	49
3.2.1 Sơ đồ khối tổng quát .....	49
3.2.2 Sơ đồ chi tiết .....	50
3.3    Sơ lược về phần cứng .....	52
3.3.1 Vi điều khiển STM32F407VG.....	52
3.3.2 Micro MP45DT02.....	56
3.3.3    Máy tính nhúng Raspberry Pi 4 .....	57
3.3.4    Raspberry Pi Camera Module V2 .....	59
3.3.5    Khóa chốt điện DC12V LY-03 .....	61
3.3.6    Module relay với opto cách ly 12V .....	61
3.3.7    Màn hình LCD Text 1602 Xanh lá .....	62
4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM .....	64
4.1 Quy trình tách đặc trưng và huấn luyện tạo codebook để nhận dạng giọng nói .....	67
4.1.1    Trích đặc trưng.....	68
4.1.2    Huấn luyện và nhận dạng giọng nói .....	74
4.2 Quy trình huấn luyện và nhận dạng khuôn mặt dùng PCA .....	77
4.2.1 Thuật toán PCA để huấn luyện dữ liệu nhận dạng.....	78
4.2.2 Nhận dạng khuôn mặt .....	80
4.3 Thử nghiệm giải thuật trên MATLAB .....	81
4.4 Lập trình thuật toán trên vi điều khiển và máy tính nhúng .....	83
4.4.1    Khái quát về vi điều khiển .....	83
4.4.2    Phần mềm dùng cho vi điều khiển .....	83
4.4.3    Thực hiện giải thuật trên vi điều khiển .....	86
4.4.4    Khái quát về máy tính nhúng và ngôn ngữ dùng cho nó .....	90
4.4.5 Phản hồi .....	91

---

5. KẾT QUẢ THỰC HIỆN .....	101
5.1 Nguồn thực nghiệm để nhận dạng chính chủ .....	101
5.1.1 Nhận dạng trên phần mềm MATLAB .....	101
5.1.2 Nhận dạng trên phần cứng .....	105
5.2 Kết quả lập trình ứng dụng trên nền tảng MATLAB.....	106
5.3 Kết quả phần cứng .....	115
5.4 Đánh giá độ chính xác của mô hình.....	124
5.4.1 Thực hiện thuật toán trên phần mềm MATLAB.....	124
5.4.2 Thực hiện thuật toán trên phần cứng.....	126
5.4.3 Đánh giá hiệu quả nhận dạng của phần mềm và phần cứng .....	126
6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	128
6.1 Kết luận.....	128
6.2 Hướng phát triển .....	129
7. TÀI LIỆU THAM KHẢO .....	130
8. PHỤ LỤC .....	132

## DANH SÁCH HÌNH

Hình 1. 1 Nhận dạng khuôn mặt.....	1
Hình 1. 2 Nhận dạng vân tay.....	2
Hình 1. 3 Nhận dạng giọng nói.....	2
Hình 1. 4 Khoảng cách từ micro đến miệng.....	3
Hình 1. 5 Góc nghiên của camera so với khuôn mặt .....	4
Hình 2. 1 Cấu tạo các bộ phận phát âm con người.....	6
Hình 2. 2 Cấu tạo tai con người .....	8
Hình 2. 3 Qui trình chuyển tín hiệu analog sang digital.....	13
Hình 2. 4 Độ phân giải trong ADC .....	13
Hình 2. 5 Qui trình lấy mẫu.....	14
Hình 2. 6 Qui trình lượng tử hóa .....	15
Hình 2. 7 Qui trình số hóa .....	15
Hình 2. 8 Quá trình trích đặc trưng LPC.....	18
Hình 2. 9 Quá trình trích đặc trưng MFCC .....	19
Hình 2. 10 Quá trình windowing .....	19
Hình 2. 11 Các loại cửa sổ trong quá trình Windowing .....	20
Hình 2. 12 Biểu đồ phổ âm thanh theo chu kỳ .....	21
Hình 2. 13 Dãy bộ lọc Mel-filterbank .....	21
Hình 2. 14 Quá trình chuyển từ spectrum sang Mel-scale power spectrum.....	22
Hình 2. 15 Phương pháp ngưỡng kép.....	24
Hình 2. 16 Phương pháp Vector Quantization .....	27
Hình 2. 17 Giải thuật Vector Quantization .....	28
Hình 2. 18 Mô hình Markov .....	30
Hình 2. 19 Mô hình Hidden Markov .....	31
Hình 2. 20 Các đặc trưng Haar-like .....	32
Hình 2. 21 Bốn đặc trưng Haar-like cơ bản.....	32

---

Hình 2. 22 Đặc trưng cạnh .....	32
Hình 2. 23 Đặc trưng đường .....	33
Hình 2. 24 Đặc trưng xung quanh tâm.....	33
Hình 2. 25 Integral Image.....	33
Hình 2. 26 Cách tính 4 điểm trên Integral Image.....	34
Hình 2. 27 Bộ phân loại AdaBoost .....	35
Hình 2. 28 Kết hợp các bộ phân loại yếu thành bộ phân loại mạnh.....	36
Hình 2. 29 Sơ đồ nhận diện khuôn mặt .....	36
Hình 2. 30 Trước và sau khi giảm chiều dữ liệu bằng PCA.....	38
Hình 3. 1 Sơ đồ chi tiết giữa vi điều khiển và máy tính nhúng .....	50
Hình 3. 2 Sơ đồ chi tiết các khối còn lại .....	51
Hình 3. 3 Vi điều khiển STM32F4 Discovery .....	52
Hình 3. 4 Website của thư viện CMSIS-DSP .....	55
Hình 3. 5 Micro MP45DT02.....	56
Hình 3. 6 Raspberry Pi 4 .....	57
Hình 3. 7 Các chân ra của Raspberry Pi 4 .....	58
Hình 3. 8 Camera Module V2 .....	60
Hình 3. 9 Sau khi nối Camera Module V2 với Raspberry Pi 4 .....	60
Hình 3. 10 Khóa chốt cửa điện.....	61
Hình 3. 11 Module relay .....	61
Hình 3. 12 LCD Text 1602 .....	62
Hình 4. 1 Giải thuật huấn luyện của nhận dạng giọng nói.....	65
Hình 4. 2 Giải thuật nhận dạng giọng nói .....	66
Hình 4. 3 Thuật toán trích xuất đặc trưng tiếng nói MFCC.....	67
Hình 4. 4 Sơ đồ biên độ và pha.....	68
Hình 4. 5 Tín hiệu âm thanh của từ “Một” trước và sau khi Pre-emphasis.....	69
Hình 4. 6 Tín hiệu âm thanh của 6 âm từ “một” đến “sáu”.....	70

---

Hình 4. 7 Tín hiệu âm thanh sau khi tách bằng phương pháp ZCR và STE .....	71
Hình 4. 8 Các từ sau khi đã được tách ra.....	71
Hình 4. 9 Cửa sổ Hamming .....	72
Hình 4. 10 Bộ lọc Mel-filter Bank .....	73
Hình 4. 11 Acoustic vectors của từ “Một” .....	74
Hình 4. 12 Thuật toán huấn luyện và nhận dạng tiếng nói bằng VQ.....	75
Hình 4. 13 Codebook sau khi lượng tử hóa vector .....	76
Hình 4. 14 Lưu đồ giải thuật tổng quát của nhận dạng khuôn mặt.....	77
Hình 4. 15 Thuật toán huấn luyện của nhận dạng khuôn mặt bằng PCA.....	78
Hình 4. 16 Thuật toán nhận dạng khuôn mặt bằng PCA.....	80
Hình 4. 17 Phần mềm STM32CubeMX.....	84
Hình 4. 18 Cấu hình xung clock cho vi điều khiển.....	84
Hình 4. 19 Phần mềm Keil uVision 5 .....	85
Hình 4. 20 Các thư viện cần thiết cho chương trình.....	86
Hình 4. 21 Tham số Hamming Window .....	87
Hình 4. 22 Tham số Mel-FilterBank .....	88
Hình 4. 23 Tham số codebook của từ KHÔNG .....	88
Hình 4. 24 Lưu đồ giải thuật tổng quát nhận dạng giọng nói .....	89
Hình 4. 25 Ngôn ngữ lập trình Python .....	91
Hình 4. 26 Ảnh cần tải để flash lên SD card .....	92
Hình 4. 27 Phần mềm Win32 Disk Imager .....	92
Hình 4. 28 Cách bật uart cho Raspberry Pi .....	93
Hình 4. 29 Sơ đồ chân Raspberry Pi .....	93
Hình 4. 30 Cách kết nối với Raspberry Pi qua Serial .....	94
Hình 4. 31 Cách thay đổi địa chỉ wifi cho board Pi.....	94
Hình 4. 32 Phần mềm PuTTY.....	95
Hình 4. 33 Cài đặt phần mềm xrdp cho board Pi .....	96

Hình 4. 34 Phần mềm xrdp .....	96
Hình 4. 35 Cách tìm địa chỉ IP để kết nối bằng VNC Viewer .....	97
Hình 4. 36 Phần mềm Pycharm.....	97
Hình 4. 37 Phần mềm Thonny Python IDE.....	98
Hình 4. 38 Thư viện OpenCV.....	98
Hình 4. 39 Thư viện NumPy .....	99
Hình 4. 40 Thư viện matplotlib .....	100
Hình 5. 1 Dữ liệu khoảng cách của chính chủ so với đầu vào chính chủ .....	103
Hình 5. 2 Giao diện phần mềm mô phỏng nhận dạng khuôn mặt và giọng nói bằng MATLAB .....	106
Hình 5. 3 Trường hợp nhận dạng khuôn mặt đúng chính chủ .....	108
Hình 5. 4 Trường hợp nhận khuôn mặt sai chính chủ .....	109
Hình 5. 5 Dạng sóng của tín hiệu âm thanh theo miền thời gian, văn bản từ “một” tới “sáu” .....	110
Hình 5. 6 Phổ biên độ của tín hiệu vừa thu được ở trên.....	110
Hình 5. 7 Acoustic Vector của tín hiệu âm thanh từ “một” đến “sáu” .....	111
Hình 5. 8 Trường hợp nhận dạng đúng giọng nói chính chủ nhưng sai mật khẩu .....	113
Hình 5. 9 Trường hợp nhận dạng giọng nói sai cả giọng chính chủ và mật khẩu .....	113
Hình 5. 10 Trường hợp nhận dạng đúng giọng chính chủ và mật khẩu .....	114
Hình 5. 11 Trường hợp nhận dạng đúng cả 3 lớp bảo mật .....	115
Hình 5. 12 Mặt trước phần cứng .....	116
Hình 5. 13 Các khối chức năng của phần cứng .....	117
Hình 5. 14 Kết quả LCD khi nhận dạng đúng mật khẩu .....	118
Hình 5. 15 Kết quả LED trái khi nhận dạng đúng giọng nói .....	118
Hình 5. 16 Kết quả LCD khi nhận dạng sai mật khẩu .....	119
Hình 5. 17 Kết quả LED trái khi nhận dạng sai mật khẩu .....	119
Hình 5. 18 Kết quả LCD khi nhận dạng sai chính chủ .....	120
Hình 5. 19 Kết quả hai LED khi nhận dạng sai chính chủ .....	120
Hình 5. 20 Kết quả LCD khi nhận dạng đúng chính chủ .....	121

Hình 5. 21 Kết quả hai LED khi nhận dạng đúng chính chủ .....	121
Hình 5. 22 Kết quả khóa chốt mở khi nhận dạng đúng chính chủ.....	122
Hình 5. 23 Kết quả nhận dạng khuôn mặt chính chủ.....	123
Hình 5. 24 Kết quả nhận dạng khuôn mặt người lạ .....	123

## DANH SÁCH BẢNG

Bảng 1. 1 Lịch phân chia công việc theo tuần.....	xii
Bảng 3. 1 Các chân của LCD .....	63
Bảng 5. 1 Bảng dữ liệu để tìm ngưỡng giọng nói chính chủ từ người 1 đến người 9 .....	102
Bảng 5. 2 Bảng dữ liệu tìm ngưỡng giọng nói chính chủ từ người 11 đến người thứ 20.....	103
Bảng 5. 3 Bảng chức năng hai đèn LED trong hệ thống.....	116
Bảng 5. 4 Độ chính xác thuật toán nhận dạng người nói trên MATLAB .....	124
Bảng 5. 5 Độ chính xác của thuật toán nhận dạng mật khẩu số trên MATLAB .....	125
Bảng 5. 6 Độ chính xác của thuật toán nhận dạng khuôn mặt trên MATLAB.....	125
Bảng 5. 7 Độ chính xác nhận dạng người nói trên phần cứng.....	126
Bảng 5. 8 Độ chính xác nhận dạng mật khẩu số trên phần cứng .....	127
Bảng 5. 9 Độ chính xác nhận dạng khuôn mặt trên phần cứng.....	127

## BẢNG KẾ HOẠCH THỰC HIỆN

Nội dung	Thời gian																
	Tháng 3				Tháng 4 & 5				Tháng 6 & 7								
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	
Nội dung 1	Red	Red	Red														
Nội dung 2	Blue	Blue	Blue														
Nội dung 3				Blue	Blue												
Nội dung 4						Blue	Blue	Blue									
Nội dung 5				Red	Red												
Nội dung 6					Red	Red	Red	Red									
Nội dung 7									Green	Green	Green	Green					
Nội dung 8													Green	Green	Green		
Nội dung 9															Green	Green	
Nội dung 10																	Green
Nội dung 11	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green

Bảng 1.1 Lịch phân chia công việc theo tuần

- Nội dung 1: Tìm tài liệu tham khảo về lý thuyết về nhận dạng khuôn mặt (face recognition)
- Nội dung 2: Tìm tài liệu tham khảo về lý thuyết nhận dạng giọng nói (speaker identification) và nhận dạng văn bản.
- Nội dung 3: Tìm hiểu đánh giá các giải thuật, tổng hợp, chỉnh sửa từ đó lựa chọn và thực hiện giải thuật về trích xuất đặc trưng giọng nói trên MATLAB.
- Nội dung 4: Tìm hiểu đánh giá các giải thuật, tổng hợp, chỉnh sửa từ đó lựa chọn và thực hiện giải thuật về huấn luyện đặc trưng và nhận dạng giọng nói trên MATLAB
- Nội dung 5: Tìm hiểu đánh giá các giải thuật, tổng hợp, chỉnh sửa từ đó lựa chọn và thực hiện giải thuật về huấn luyện và nhận dạng khuôn mặt trên MATLAB
- Nội dung 6: Thu thập dữ liệu về giọng nói và khuôn mặt của chính chủ và những người khác chính chủ

- Nội dung 7: Tìm hiểu phần cứng, lựa chọn phần cứng, nhúng dữ liệu đặc trưng lên phần cứng. Đánh giá giải thuật về thời gian thực hiện trên kit thử nghiệm
- Nội dung 8: Thực hiện giải thuật nhận dạng khuôn mặt và giọng nói trên phần cứng và thử nghiệm kết quả. Kiểm chứng độ chính xác của thiết kế
- Nội dung 9: Thực hiện đóng gói toàn bộ phần cứng cho dự án để nhận dạng khuôn mặt, giọng nói và mật khẩu số.
- Nội dung 10: Soạn slide thuyết trình và luyện tập thuyết trình cho đợt bảo vệ.
- Nội dung 11: Viết báo cáo luận văn.
- Sinh viên thực hiện: Tiến Đạt 

Trí Luận 

Cả hai 

- Quy định riêng của nhóm:
  - Họp nhóm để làm việc, thảo luận phải tới đúng giờ
  - Hoàn thành deadline đã đề ra
  - Các thành viên trong nhóm tôn trọng lẫn nhau, giúp đỡ nhau cùng tiến bộ
  - Tiền để xây dựng và phát triển đồ án phải chia đều cho các thành viên trong nhóm.

## 1. GIỚI THIỆU

### 1.1 Tổng quan

Sinh trắc học hay công nghệ sinh trắc học (Biometric) là công nghệ sử dụng những thuộc tính vật lý, đặc điểm sinh học của mỗi người như vân tay, móng mắt, khuôn mặt, giọng nói,...để nhận dạng. Với sự phát triển của khoa học máy tính, nhận dạng cá nhân bằng sinh trắc học ngày càng đáng tin cậy và chính xác cao trong một số ứng dụng của chính phủ và thương mại, ví dụ như: kiểm soát an ninh biên giới quốc tế; quyền truy cập bảo mật vào các tòa nhà, máy tính xách tay và điện thoại di động,... Bên cạnh việc tăng cường bảo mật, các hệ thống sinh trắc học cũng nâng cao sự tiện lợi của người dùng bằng cách giảm bớt các nhu cầu về thiết kế và ghi nhớ nhiều mật khẩu phức tạp.



Hình 1. 1 Nhận dạng khuôn mặt

Một hệ thống sinh trắc học phải đối mặt với các vấn đề liên quan như tính không phổ biến của sinh trắc học, các lỗi và hạn chế khi nhận dạng. Nhận dạng sinh trắc học cũng có thể sai sót đối với những người dùng không thể cung cấp đầu vào vật lý mà nó yêu cầu cũng như các thay đổi đối với người dùng hiện tại. Ngoài ra, các cuộc tấn công giả mạo cũng là một hạn chế trong lĩnh vực nhận dạng sinh trắc học. Trong lĩnh vực nhận dạng sinh trắc học, thì quen thuộc nhất vẫn là nhận dạng dấu vân tay (fingerprint recognition), nhận dạng gương mặt (face recognition).



Hình 1. 2 Nhận dạng vân tay

Trong những năm gần đây, với sự phát triển của lĩnh vực Deep learning và xử lý tiếng nói (Speech Processing) thì nhận dạng người nói (Speaker Identification) đã trở nên tiềm năng rất lớn và được ứng dụng trong các lĩnh bảo mật kinh tế, tài chính nói riêng cũng như bảo mật, an ninh xã hội, quốc phòng nói chung. Trong đề tài này, chúng em sẽ thực hiện nhận dạng giọng nói với một số lượng nhỏ từ vựng các số từ “không đến chín” do một số người phát âm. Ngoài ra, chúng em sẽ nhận dạng khuôn mặt với một số lượng nhỏ các khuôn mặt thu thập được từ mọi người xung quanh.



Hình 1. 3 Nhận dạng giọng nói

## 1.2 Tình hình nghiên cứu trong và ngoài nước

Việc ứng dụng một hệ thống nhận dạng giọng nói để xác định người nói hoặc nhận dạng văn bản mà người đó nói cho đến thời điểm hiện tại vẫn rất còn nhiều hạn chế. Đây là một đối tượng không ổn định với nhiều khó khăn như sau:

- Tại nhiều thời điểm khác nhau thì giọng nói người đó về cơ bản vẫn có sự khác nhau, lý do là vì người nói chịu ảnh hưởng từ các tác động bên ngoài hoặc bên trong cơ thể.
- Giọng nói của một người cũng sẽ thay đổi theo thời gian, độ tuổi của người đó
- Người nói ở các tình trạng sức khỏe khác nhau sẽ có tiếng nói khác nhau.
- Giọng nói còn không ổn định một phần do tốc độ nói.

Ngoài ra, còn các ảnh hưởng khác của ngoại cảnh như:

- Nhiều và tiếng ồn của môi trường xung quanh người nói lúc thu âm
- Chất lượng của thiết bị ghi âm
- Khoảng cách từ miệng người nói đến thiết bị ghi âm

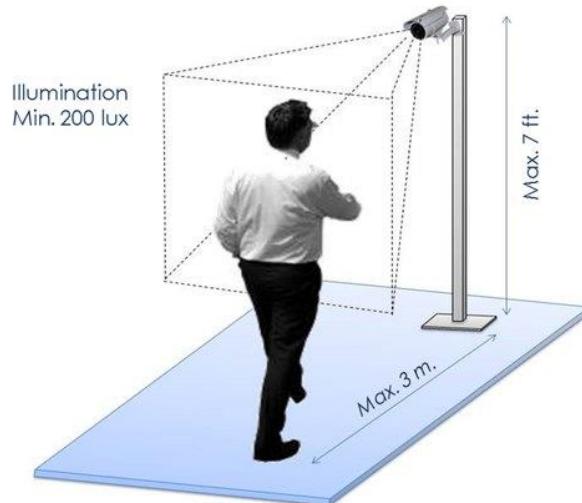


Hình 1.4 Khoảng cách từ micro đến miệng

Nhận dạng khuôn mặt hiện nay được thực hiện với rát nhiều thuật toán khác nhau, được chia làm ba loại: phương pháp tiếp cận toàn cục, phương pháp tiếp cận dựa trên các đặc điểm cục bộ và phương pháp lai. Qua nhiều lần xem xét các bài báo khoa học nghiên cứu độ chính xác giữa các thuật toán và cân nhắc tính phức tạp để khi nhúng thuật toán xuống phần cứng có thể khả thi, nhóm em chọn phương pháp tiếp cận toàn cục, cụ thể là phương pháp Eigenfaces-PCA. Phương pháp Eigenfaces-PCA là phương pháp trích rút đặc trưng nhằm giảm số chiều của ảnh, tuy đơn giản nhưng mang lại hiệu quả tương đối tốt.

Nhận dạng khuôn mặt người hiện nay cũng còn tồn tại các khó khăn trong quá trình nhận dạng như:

- Môi trường xung quanh: độ sáng tối trên khuôn mặt
- Khuôn mặt thay đổi dần theo thời gian
- Các vật dụng xuất hiện như mắt kính
- Góc nghiêng của khuôn mặt so với camera



Hình 1. 5 Góc nghiêng của camera so với khuôn mặt

Do tồn tại nhiều khó khăn như vậy, nên chúng em rút ra kết luận là điều kiện lý tưởng nhất cho việc nhận dạng tiếng nói và khuôn mặt là tiếng nói và khuôn mặt phải đồng bộ và ổn định cả trong lúc huấn luyện và lúc nhận dạng về môi trường và điều kiện thu thập dữ liệu.

Để thực hiện được nhận dạng giọng nói trên phần cứng, nhóm em ưu tiên chọn các phương pháp đơn giản về hướng tiếp cận nhưng vẫn đảm bảo hiệu quả tương đối tốt nhờ vào thực nghiệm nhiều trên bộ dữ liệu huấn luyện.

### 1.3 Nhiệm vụ luận văn

#### Các đối tượng và phạm vi nghiên cứu:

- Các khái niệm cơ bản trong lĩnh vực xử lý tín hiệu nói chung và xử lý tiếng nói nói riêng
- Các khái niệm cơ bản trong lĩnh vực nhận dạng khuôn mặt và nhận dạng giọng nói.
- Các khái niệm cơ bản trong lĩnh vực trích xuất đặc trưng âm thanh và khuôn mặt

- Các kỹ thuật, phương pháp huấn luyện và nhận dạng danh tính người dựa trên tiếng nói và khuôn mặt

 **Nhiệm vụ chính của đề tài:**

- Tìm tài liệu tham khảo về lý thuyết về nhận dạng khuôn mặt (face recognition)
- Tìm tài liệu tham khảo về lý thuyết nhận dạng giọng nói (speaker identification) và nhận dạng văn bản.
- Tìm hiểu đánh giá các giải thuật, tổng hợp, chỉnh sửa từ đó lựa chọn và thực hiện giải thuật về trích xuất đặc trưng giọng nói trên MATLAB.
- Tìm hiểu đánh giá các giải thuật, tổng hợp, chỉnh sửa từ đó lựa chọn và thực hiện giải thuật về huấn luyện đặc trưng và nhận dạng giọng nói trên MATLAB
- Tìm hiểu đánh giá các giải thuật, tổng hợp, chỉnh sửa từ đó lựa chọn và thực hiện giải thuật về huấn luyện và nhận dạng khuôn mặt trên MATLAB
- Thu thập dữ liệu về giọng nói và khuôn mặt của chính chủ và những người khác chính chủ
- Tìm hiểu phần cứng, lựa chọn phần cứng, nhúng dữ liệu đặc trưng lên phần cứng. Đánh giá giải thuật về thời gian thực hiện trên kit thử nghiệm
- Thực hiện giải thuật nhận dạng khuôn mặt và giọng nói trên phần cứng và thử nghiệm kết quả. Kiểm chứng độ chính xác của thiết kế
- Thực hiện đóng gói toàn bộ phần cứng cho dự án để nhận dạng khuôn mặt, giọng nói và mật khẩu số.
- Soạn slide thuyết trình và luyện tập thuyết trình cho đợt bảo vệ.
- Viết báo cáo luận văn.
- Đánh giá khả năng thực hiện được của đề tài, rút ra các bài học.

## 2. LÝ THUYẾT

### 2.1 Những vấn đề cơ bản của tiếng nói con người

#### 2.1.1 Nguyên lý hình thành tiếng nói con người

Khi người nói phát âm, một luồng hơi từ phổi được đẩy lên tạo một áp lực thanh quản. Khi chịu áp lực đó, thanh quản mở ra giúp luồng hơi không khí thoát qua, lúc đó áp lực đó sẽ giảm xuống khiến cho thanh quản tự động đóng lại. Sau đó, quá trình lại tái diễn khi người nói phát âm tiếp, các chu kỳ đóng mở thanh quản này lặp đi lặp lại liên tục sẽ tạo ra các tần số sóng âm.

Sau khi thanh quản tạo ra các tần số sóng âm cơ bản, đến các cơ quan khác như: vòm họng, khoang miệng, lưỡi, răng, môi,...Sẽ đóng vai trò như một bộ cộng hưởng có khả năng thay đổi hình dạng linh hoạt. Bộ cộng hưởng này sẽ có tác dụng khuếch đại một vài tần số cũng như triệt tiêu đi một vài tần số khác để tạo ra âm thanh. Chính vì các cơ quan trên có khả năng thay đổi hình dạng linh hoạt nên âm thanh của người nói cũng sẽ thay đổi theo để tạo thành tiếng nói.



Hình 2. 1 Cấu tạo các bộ phận phát âm con người

#### 2.1.2 Đặc tính vật lý của tiếng nói

**Độ cao:** là mức độ cao thấp của âm thanh, nó thuộc vào tần số dao động của không khí chứa âm thanh đó trong một khoảng thời gian nhất định. Tần số dao động càng lớn thì độ cao âm thanh càng cao.

Độ lớn: là cường độ hay biên độ của âm thanh dao động. Trong ngôn ngữ, độ lớn khi phát âm phụ âm thường cao hơn nguyên âm, tạo nên sự khác biệt lớn giữa phụ âm và nguyên âm trong tiếng nói.

Độ dài: là trường độ của âm thanh, phụ thuộc vào sự chấn động lâu hoặc dài của các phần tử không khí.

Âm sắc: là sắc thái riêng do mỗi cá thể khác nhau tạo ra. Âm sắc là nguyên nhân tạo nên sự khác biệt giữa giọng nói giữa người này với người khác. Âm sắc có được là do hiện tượng cộng hưởng tạo nên.

Tiếng ồn: còn được gọi là nhiễu, nhiễu sẽ làm giảm độ rõ ràng của tín hiệu âm thanh

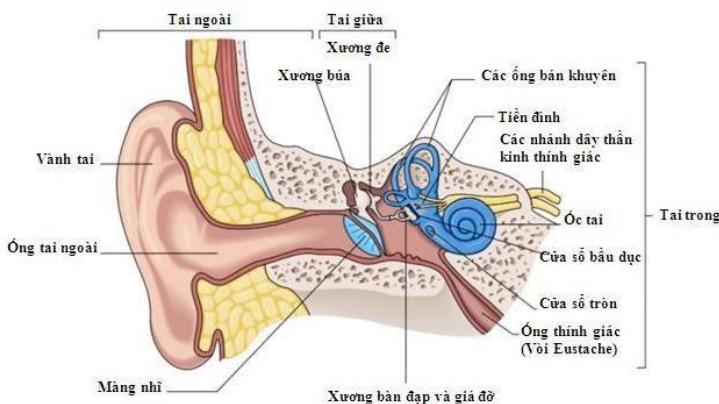
### 2.1.3 Cơ chế hoạt động của tai

Trong lĩnh vực nhận dạng giọng nói, việc hiểu cơ chế nghe của con người quan trọng hơn cách con người tạo ra tiếng nói

Tai người do cơ chế hoạt động phi tuyến tính, tức là độ cảm nhận âm thanh ở tần số 20000Hz không phải gấp 1000 lần tần số 20Hz. Tai người thường sẽ nhạy cảm ở tần số thấp, kém nhạy cảm ở tần số cao.

Khi âm thanh truyền tới tai, nó sẽ va đập vào màng nhĩ, màng nhĩ sẽ rung lên, truyền rung động lên ba xương nhỏ: malleus, incus, stapes tới ống tai. Ống tai là một bộ phận dạng xoắn, rỗng như con ốc. Ống tai chứa các dịch nhầy bên trong giúp truyền âm thanh, dọc theo ống tai là các tế bào lông giúp cảm nhận âm thanh. Các tế bào lông này rung lên khi có sóng truyền qua, uốn cong và nó sẽ gửi tín hiệu thần kinh lên não bộ qua dây thần kinh thính giác. Các tế bào lông ở đoạn đầu cứng hơn, sẽ rung động với các tần số cao. Càng sâu vào trong, các tế bào lông sẽ càng bớt cứng, đáp ứng với các tần số thấp. Chính vì lý do cấu tạo ống tai với số lượng lớn các tế bào lông đáp ứng được với tần số thấp nhiều hơn nên việc cảm nhận tai người sẽ phi tuyến tính, nhạy cảm ở tần số thấp, và kém nhạy cảm ở tần số cao.

Thần kinh thính giác sẽ gửi các tín hiệu đến não bộ, ở đó các tín hiệu này sẽ được hiểu là âm thanh.



Hình 2.2 Cấu tạo tai con người

## 2.2 Tìm hiểu về lý thuyết cơ bản của nhận dạng khuôn mặt

Nhận dạng khuôn mặt (Face Recognition) là một phương pháp sinh trắc học để xác định hoặc xác minh một cá nhân nào đó bằng cách so sánh dữ liệu hình ảnh chụp trực tiếp hoặc hình ảnh kỹ thuật số với bản ghi được lưu trữ cho người đó. Xét về nguyên tắc chung, nhận dạng khuôn mặt có sự tương đồng rất lớn với nhận dạng vân tay và móng mắt. Tuy nhiên, ở mỗi lĩnh vực, việc trích chọn đặc trưng sẽ khác nhau. Trong khi nhận dạng vân tay và móng mắt đã có thể ứng dụng rộng rãi trên nhiều lĩnh vực thì nhận dạng khuôn mặt vẫn còn nhiều thách thức. So với hai lĩnh vực nhận dạng kia, nhận dạng khuôn mặt có nguồn dữ liệu phong phú hơn, vì chúng ta có thể nhìn thấy khuôn mặt con người ở bất cứ đâu từ ảnh, video, đường phố,... Ngoài ra nhận dạng khuôn mặt cũng có lợi thế khi không cần đến sự hợp tác của người cần được nhận dạng, vì camera có thể quay và chụp lại được khuôn mặt của con người một cách dễ dàng hơn so với lấy vân tay và móng mắt.

Các hệ thống nhận dạng khuôn mặt thường được sử dụng cho các mục đích an ninh như kiểm soát an ninh tại tòa nhà, sân bay, máy ATM, tra cứu thông tin của tội phạm, phát hiện tội phạm ở nơi công cộng,... Và còn nhiều ứng dụng khác được ứng dụng rộng rãi trong cuộc sống.

Bên cạnh những thành công đã được ghi nhận thì nhận dạng khuôn mặt cũng còn gặp nhiều khó khăn như về độ sáng, góc nghiên, kích thước hình ảnh, diện mạo, biểu hiện cảm xúc của khuôn mặt hay ảnh hưởng của tham số môi trường.

Các bước trong nhận dạng khuôn mặt trong đề tài này bao gồm: phát hiện khuôn mặt (Face Detection), trích chọn đặc trưng (feature extraction), nhận dạng khuôn mặt (Face Recognition)

## 2.3 Tổng quan về nhận dạng giọng nói và nhận dạng người nói

### 2.3.1 Sơ lược về nhận dạng giọng nói và nhận dạng người nói

Nhận dạng giọng nói (Speech Recognition) là chuyển đổi tín hiệu ngôn ngữ từ dạng âm thanh thành dạng văn bản. Hiểu một cách chính xác, nhận dạng giọng nói là phân chia và đính nhãn ngôn ngữ cho tín hiệu tiếng nói.

Nhận dạng người nói (Speaker Identification) là dùng các phương pháp xử lý tín hiệu số, trích xuất và huấn luyện đặc trưng để xác định cụ thể người nói đang là ai khi ghi âm được tiếng nói của người đó.

Nhận dạng danh tính người nói có nhiều ứng dụng:

- Ứng dụng trong tài chính, ngân hàng: việc bảo mật thông tin tài chính hay các tài khoản ngân hàng với mã pin hay mật khẩu thường đã lỗi thời và khó kiểm soát, quản lý cũng như dễ bị đánh cắp thông tin
- Ứng dụng trong pháp y và các hoạt động hành pháp: việc ứng dụng nhận dạng danh tính người nói có thể được ứng dụng trong việc nhận dạng danh tính người đã mất, nếu có được dữ liệu giọng nói của người đó. Với ứng dụng này, việc điều tra và giải quyết các vấn đề liên quan đến người đã mất sẽ dễ dàng hơn.
- Ứng dụng quản lý dùng cho chính phủ: việc thu thập một lượng lớn dữ liệu giọng nói từ công dân, chính phủ có thể ứng dụng nhận dạng danh tính để dễ kiểm soát và quản lý công dân hơn, ngoài thẻ căn cước và chứng minh nhân dân.
- Ngoài các ứng dụng trên, nhận dạng danh tính người nói còn rất nhiều ứng dụng tiềm năng khác có thể được triển khai trong tương lai.

### 2.3.2 Phân loại các hệ thống nhận dạng giọng nói

#### a. Nhận dạng từ liên tục và nhận dạng từ tách biệt

Một hệ thống nhận dạng tiếng nói (Speech Recognition) chia làm 2 dạng: nhận dạng từ mà người nói phát âm một cách liên tục, và nhận dạng từng từ mà người nói phát âm.

Nhận dạng liên tục các từ:

- Nhận dạng liên tục các từ là nhận dạng tiếng nói được phát liên tục trong một chuỗi tín hiệu, ví dụ như một câu nói, mệnh lệnh hoặc một đoạn văn bản được đọc bởi người dùng.
- Hệ thống nhận dạng liên tục này rất phức tạp vì các từ được phát âm ra một cách liên tục rất khó xử lý theo kịp, nhất là đối với hệ thống nhận dạng theo thời gian thực. Nếu người nói phát âm các từ liên tục mà không có khoảng nghỉ giữa các khoảng thì công việc nhận dạng càng trở nên khó khăn, điều này thường xảy ra trong thực tế.

Nhận dạng từng từ:

- Nhận dạng từng từ mà người nói phát âm tức là đối với mỗi từ cần nhận dạng sẽ được người nói phát âm một cách rời rạc, có khoảng nghỉ trước và sau khi phát âm một từ.
- Hệ thống loại này đơn giản hơn hệ thống nhận dạng liên tục và có nhiều ứng dụng thực tiễn.
  - Với đề tài lần này, nhóm em chọn nhận dạng từng từ và sẽ đọc mật khẩu có sự ngắt quãng giữa các từ để dễ dàng hơn trong quá trình nhận dạng

#### b. Nhận dạng phụ thuộc văn bản (Text-dependent) và độc lập văn bản (Text-independent)

Nhận dạng phụ thuộc văn bản:

- Hệ thống nhận dạng cần được biết trước văn bản mà người nói phát âm.
- Các văn bản được phát âm sẽ được cố định trước khi nói
- Dùng cho các ứng dụng kiểm soát chặt chẽ đầu vào là giọng nói của người nói
- Việc biết trước văn bản được nói có thể cải thiện hiệu suất của hệ thống nhận dạng này
- Các văn bản được nói thường sẽ rất ngắn

Nhận dạng độc lập văn bản:

- Nhận dạng độc lập văn bản sẽ không cần biết trước văn bản được phát âm bởi người nói
- Người nói sẽ được chọn văn bản được hệ thống nhận dạng khi phát âm hoặc các đoạn tiếng nói hội thoại.
- Dùng cho các ứng dụng kiểm soát ít chặt chẽ đầu vào bởi tiếng nói người nói hơn kiểu nhận dạng phụ thuộc
- Là hệ thống nhận dạng linh hoạt hơn nhưng đồng thời cũng có nhiều vấn đề khó khăn
- Người nói sẽ phát âm bất kỳ văn bản chưa được hệ thống biết trước, sau khi nhận dạng thì hệ thống sẽ biết được văn bản được nói là gì
- Đoạn văn bản được nói thường sẽ rất dài

### 2.3.3 Các hướng nghiên cứu

#### a. Phương pháp ngữ âm - âm vị học:

Phương pháp ngữ âm - âm vị học dựa trên cơ sở lý thuyết khẳng định sự tồn tại hữu hạn và duy nhất những đơn vị ngữ âm cơ bản trong ngôn ngữ nói con người được gọi là âm vị, bao gồm: nguyên âm - phụ âm, âm vô thanh-âm hữu thanh, âm vang-âm bẹt,... Các âm vị này có thể xác định được bằng tập các đặc trưng trích xuất trong phổ của tín hiệu tiếng nói theo thời gian. Đặc trưng quan trọng nhất của âm vị là formant. Đó là các vùng tần số có cộng hưởng cao nhất của tín hiệu. Ngoài ra còn một số đặc trưng khác như âm vực (cao độ - pitch), âm lượng,...

Về lý thuyết, phương pháp này có vẻ rất đơn giản. Tuy nhiên, khi người ta thử nghiệm trong thực tế cho thấy rằng phương pháp này cho kết quả nhận dạng không chính xác. Một số nguyên nhân dẫn đến vấn đề này do:

- Phương pháp cần rất nhiều kiến thức về ngữ âm học. Mà những kiến thức này nhìn chung vẫn còn chưa được nghiên cứu đầy đủ.
- Formant chỉ ổn định đối với các nguyên âm, còn đối với phụ âm formant rất khó xác định và không ổn định. Ngoài ra, việc xác định các formant cho độ chính xác không cao. Đặc biệt là khi có tác động của nhiễu (là vấn đề rất hay xảy trong thực tế).
- Trên thực tế, rất khó phân biệt các âm vị dựa trên phổ, nhất là các phụ âm vô thanh, có một số phụ âm rất giống nhiễu.

**b. Phương pháp nhận dạng mẫu:**

Phương pháp nhận dạng mẫu dựa vào lý thuyết xác suất và thống kê để nhận dạng dựa trên ý tưởng là so sánh đối tượng cần nhận dạng với các mẫu được thu thập trước đó để tìm mẫu nào giống với đối tượng nhất. Hệ thống nhận dạng sẽ trải qua 2 giai đoạn.

- Giai đoạn huấn luyện: thu thập mẫu, phân lớp và huấn luyện hệ thống ghi nhớ các mẫu đã được thu thập đó.
- Giai đoạn nhận dạng: dựa vào đối tượng cần nhận dạng. Sau đó, so sánh với các mẫu và đưa ra kết quả là mẫu nào giống với đối tượng nhất.

Đa phần các hệ nhận dạng thành công trên thế giới đều sử dụng phương pháp này.

Phương pháp nhận dạng mẫu có các ưu điểm sau:

- Sử dụng đơn giản, dễ hiểu, mang tính toán học cao (lý thuyết nhiều về xác suất thống kê, lý thuyết máy học,...)
- Giúp cho hệ thống nhận dạng ít bị ảnh hưởng bởi những biến thể về bộ từ vựng, tập đặc trưng, môi trường xung quanh...
- Phương pháp này cho kết quả tốt. Trên thực tế, các thử nghiệm đã chứng minh được điều này.

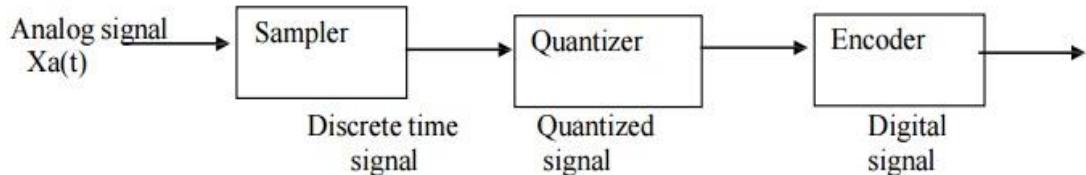
**c. Phương pháp trí tuệ nhân tạo:**

Phương pháp trí tuệ nhân tạo sẽ nghiên cứu cách học nói và học nghe của con người, để thực hiện phương pháp này cần phải tìm hiểu các quy luật của ngữ âm, ngữ pháp, ngữ nghĩa, ngữ cảnh,... và tích hợp chúng vào để bổ sung cho các phương pháp khác nhằm nâng cao kết quả nhận dạng.

Đối với các hệ thống nhận dạng mẫu, người ta sẽ cải tiến bằng cách với mỗi đối tượng cần nhận dạng, hệ thống sẽ chọn ra một số mẫu giống với đối tượng nhất. Sau đó, người ta sẽ kiểm chứng tiếp các kết quả đó bằng các luật ngữ pháp, ngữ nghĩa, ngữ cảnh,... để tìm ra mẫu nào là mẫu phù hợp nhất cho hệ thống nhận dạng.

**2.3.4 Cơ sở lý thuyết về xử lý tín hiệu****a. Bộ ADC (Analog to Digital Converter):**

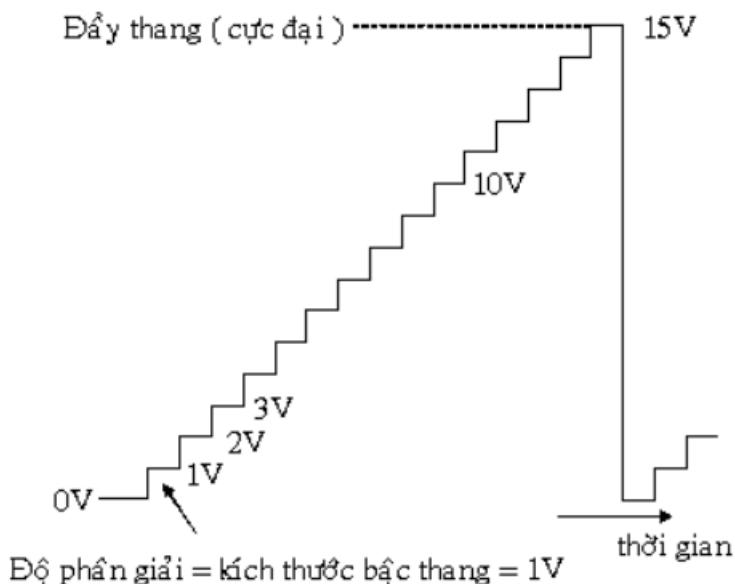
Vì âm thanh là dạng tín hiệu tương tự, trong khi đó để máy tính hiểu được thì chúng ta cần làm việc với các tín hiệu số rời rạc. Do đó, chúng ta cần bộ ADC để chuyển tín hiệu tương tự sang tín hiệu số. Quá trình này gọi là biến đổi A/D.



Hình 2. 3 Qui trình chuyển tín hiệu analog sang digital

Các chỉ số kỹ thuật chủ yếu của ADC:

- Độ phân giải: độ phân giải của một bộ ADC biểu thị bằng số bit của tín hiệu số đầu ra. Khi số lượng bit nhiều thì sai số lượng tử càng nhỏ và độ chính xác càng cao.



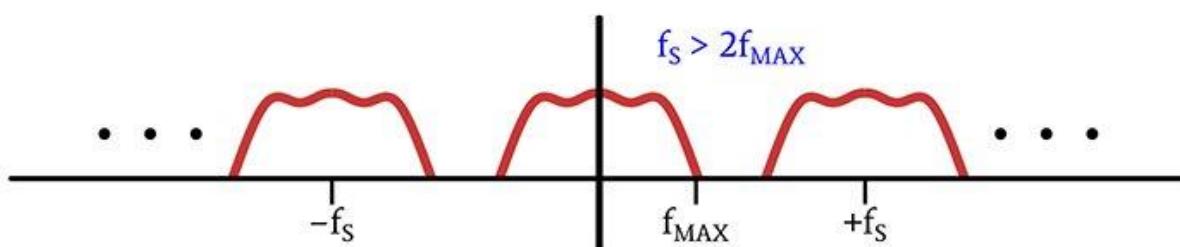
Hình 2. 4 Độ phân giải trong ADC

- Tần số lấy mẫu:

- Trong xử lý tín hiệu, lấy mẫu là chuyển đổi một tín hiệu liên tục sang một tín hiệu rời rạc. Một mẫu chứa một giá trị hoặc tập hợp các giá trị tại một điểm trên trục thời gian (hoặc không gian).
- Tần số lấy mẫu hay tỷ lệ lấy mẫu  $f_s$  được định nghĩa là số lượng các mẫu thu được trong một giây, với công thức là  $f_s = 1/T$ . Tần số lấy mẫu có đơn vị là Hz hoặc số mẫu/ 1 giây.
- Trong một số trường hợp, chúng ta có thể phục hồi lại được hoàn toàn và chính xác tín hiệu ban đầu, bằng cách áp dụng định lý lấy mẫu Nyquist-Shannon.
- Định lý lấy mẫu Nyquist-Shannon đảm bảo rằng các tín hiệu có tần số giới hạn có thể được tái tạo hoàn toàn từ phiên bản mẫu của nó, nếu như tần số lấy mẫu lớn hơn hoặc bằng 2 lần tần số tối đa của tín hiệu gốc.

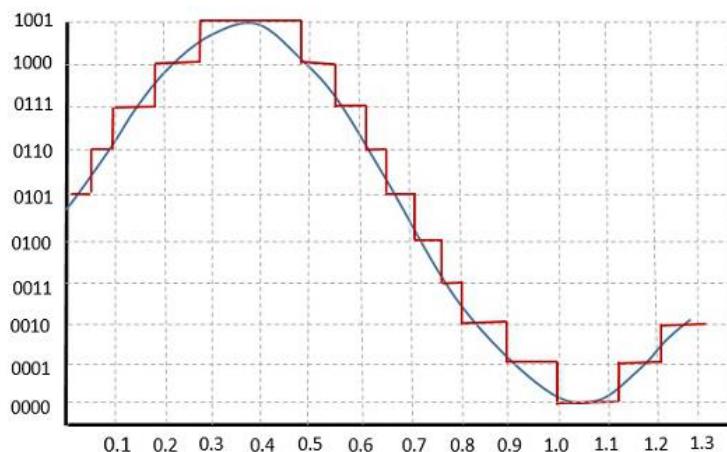
Để thực hiện chuyển đổi từ tín hiệu tương tự sang tín hiệu số, phải trải qua 3 bước sau:

- Lấy mẫu (Sampling): chia nhỏ tín hiệu tương tự thành các frame bằng nhau và chúng sẽ có những đoạn overlap lên nhau 1 khoảng gần bằng 1/3 đoạn frame. Khi lấy mẫu ta áp dụng định lý lấy mẫu Nyquist:  $f_s \geq 2 * F_{MAX}$ , trong đó:
  - $f_s$ : tần số lấy mẫu
  - $F_{MAX}$ : tần số lớn nhất của tín hiệu lấy mẫu
  - Tần số con người có thể nghe được nằm trong khoảng: 20Hz – 20 000 Hz
  - Tần số giọng nói con người nằm trong khoảng: 200-9000Hz
  - Tần số mà con người thường hay nói nhất: 300 Hz – 4000Hz



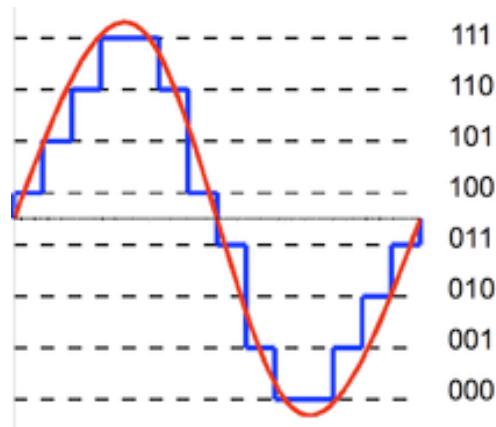
Hình 2.5 Qui trình lấy mẫu

- Lượng tử hóa (quantization): là một quá trình xấp xỉ một tập đại lượng có giá trị tương đối lớn hoặc thay đổi liên tục bằng một lượng có giá trị nhỏ hơn. Mục đích của quá trình này trong ADC là để biểu diễn lại một cách chính xác nhất các tín hiệu tương tự bằng tín hiệu số. Quá trình này còn thể hiện theo cách là quá trình chuyển đổi tín hiệu rời rạc có biên độ liên tục sang tín hiệu rời rạc có biên độ rời rạc (tín hiệu số) với mỗi mẫu tín hiệu được biểu diễn bằng một giá trị được chọn từ tập hữu hạn các giá trị có thể có.



Hình 2.6 Qui trình lượng tử hóa

- Số hóa (digitization): là quá trình biểu diễn mỗi giá trị rời rạc của tín hiệu số bằng một dãy số nhị phân b bit.



Hình 2.7 Qui trình số hóa

## b. Biến đổi Fourier rời rạc:

Về phép biến đổi Fourier:

- Về mặt toán học: Biến đổi Fourier là phép toán chuyển một hàm với giá trị phức của các biến thành hàm khác
- Trong xử lý tín hiệu số để hiểu được định nghĩa của Fourier cần biết được 2 khái niệm:
  - Miền thời gian: là miền xác định của hàm dựa theo thời gian
  - Miền tần số: Là miền mới của tín hiệu dựa trên tần số sau việc chuyển đổi tín hiệu từ miền thời gian. Biến đổi Fourier là việc biến đổi tín hiệu từ miền thời gian sang miền tần số. Biến đổi Fourier đóng vai trò quan trọng trong xử lý ảnh, có khả năng linh hoạt cao trong thiết kế và tiến hành các phương pháp lọc trong việc nâng cao chất lượng ảnh, phục hồi ảnh, nén ảnh...

Công thức biến đổi Fourier rời rạc (DFT):

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad k=0, \dots, N-1$$

Với  $e$  là cơ số của logarit tự nhiên,  $i$  là đơn vị ảo ( $i^2 = -1$ ), và  $\pi$  là pi

Công thức biến đổi Fourier rời rạc ngược (IDFT):

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} kn} \quad n=0, \dots, N-1$$

### 2.3.5 Trích xuất đặc trưng tiếng nói

Trích xuất đặc trưng của mẫu là một trong những phần quan trọng của một hệ thống nhận dạng. Những đối tượng khác nhau sẽ có một hay nhiều đặc trưng. Các đặc trưng giữa các đối tượng càng khác nhau thì việc nhận dạng có độ chính xác càng cao.

Hiện tại, các phương pháp chủ yếu dùng để trích xuất đặc trưng phổ biến bao gồm: Linear Prediction Coding (LPC), Mel – Frequency Cepstrum Coefficients (MFCC), Principle Components Analysis (PCA) và một số phương pháp khác.

### a. Mô hình LPC (Linear Prediction Coding):

Mô hình LPC được sử dụng rộng rãi trong các hệ thống nhận dạng giọng nói vì:

- LPC là một mô hình tốt đối với tín hiệu tiếng nói, đặc biệt đối với trạng thái nói tiếng nói gần ổn định. Mô hình LPC cho ta một xấp xỉ tương đối tốt của phổ âm thanh. Tuy nhiên, trong các vùng ngắn và không âm, mô hình LPC hoạt động kém hiệu quả hơn vùng có âm thanh, nhưng nó vẫn cung cấp một mô hình có thể sử dụng tốt cho mục đích nhận dạng tiếng nói.
- Cách mà LPC được ứng dụng trong việc phân tích tín hiệu tiếng nói dẫn đến một sự phân tách hợp lý các âm nguồn âm thanh. Do đó, việc biểu diễn chi tiết các đặc điểm của các dải âm thanh là hoàn toàn có thể.
- Phương pháp tính toán của LPC chính xác về mặt toán học và đơn giản, trực tiếp trong việc cài đặt lên cả phần cứng hoặc phần mềm. Số lượng tính toán trong xử lý LPC cũng ít hơn trong phương pháp filter bank.
- Mô hình LPC hoạt động tốt trong các ứng dụng nhận dạng. Kinh nghiệm cho thấy, các hệ thống nhận dạng sử dụng mô hình LPC cho kết quả tốt hơn so với các hệ thống sử dụng filter bank. Ý tưởng cơ bản của mô hình LPC là một mẫu tiếng nói cho trước tại thời điểm  $n$ ,  $s(n)$  có thể được xấp xỉ bởi một tổ hợp tuyến tính của  $p$  mẫu tín hiệu quá khứ, theo biểu thức sau:

$$(n) \approx a_1(n-1) + a_2s(n-2) + \dots + a_ps(n-p)$$

Trong đó các hệ số  $a_1, a_2, \dots, a_p$  được coi như không đổi trong khung thời gian phân tích. Biến đổi công thức (1), thêm vào đại lượng  $G.u(n)$  ta có:

$$s(n) = \sum_{i=1}^p a_i s(n-i) + Gu(n)$$

Trong đó  $u(n)$  là kích thích chuẩn hóa và  $G$  là hệ số của kích thích. Bằng biến đổi sang miền  $Z$  ta có quan hệ:

$$S(z) = \sum_{i=1}^p a_i z^{-i} S(z) + GU(z)$$

Dựa trên mô hình liên hệ chính xác giữa  $s(n)$  và  $u(n)$  ta coi tổ hợp tuyến tính của các tín hiệu quá khứ là một ước lượng của  $\tilde{s}(n)$

$$\tilde{s}(n) = \sum_{k=1}^p a_k s(n-k)$$

Sai số ước lượng  $e(n)$  được định nghĩa:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^p a_k s(n-k)$$

Với hàm truyề̂n sai số:

Vấn đề cơ bản của phân tích dự đoán tuyến tính là xác định tập các hệ số  $\{a_k\}$  tiên đoán trực tiếp từ tín hiệu tiếng nói để các đặc tính phổ của bộ lọc trùng với tín hiệu sóng tiếng nói trong cửa sổ phân tích.

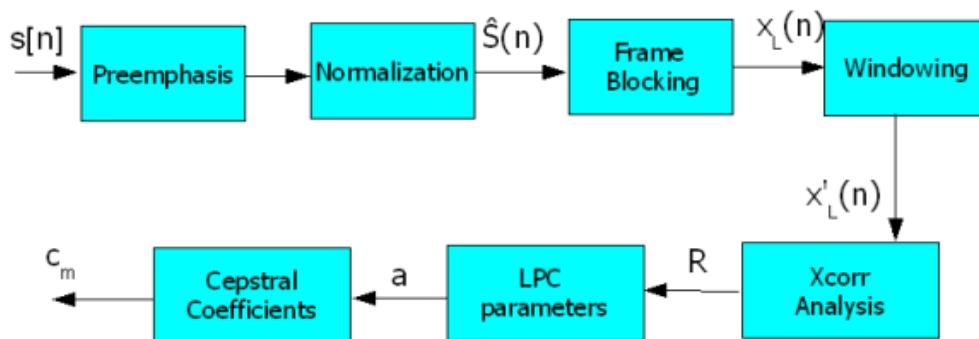
Do các đặc điểm phổ tàn của tiếng nói thay đổi theo thời gian, do vậy các hệ số tiên đoán tại một thời điểm  $n$  phải được ước lượng từ một phân đoạn ngắn của tín hiệu tiếng nói xảy ra gần  $n$ . Và như thế, hướng tiếp cận cơ bản là tìm một tập các hệ số tiên đoán có sai số dự đoán bình phương đạt cực tiểu trên một phân đoạn ngắn của tín hiệu sóng tiếng nói. Thông thường, tín hiệu tiếng nói được phân tích trên các khung liên tiếp với độ dài khoảng 10ms.

Bài toán này được giải dựa trên phương pháp tự tương quan, khi đó các hệ số  $a_k$  ước lượng sẽ là nghiệm của phương trình:

$$\sum_{k=1}^p r_n |i-k| \hat{a}_k = r_n(i), k \leq i \leq p$$

Với  $r(k)$  là hệ số tự tương quan của tín hiệu dời đi  $k$  mẫu:

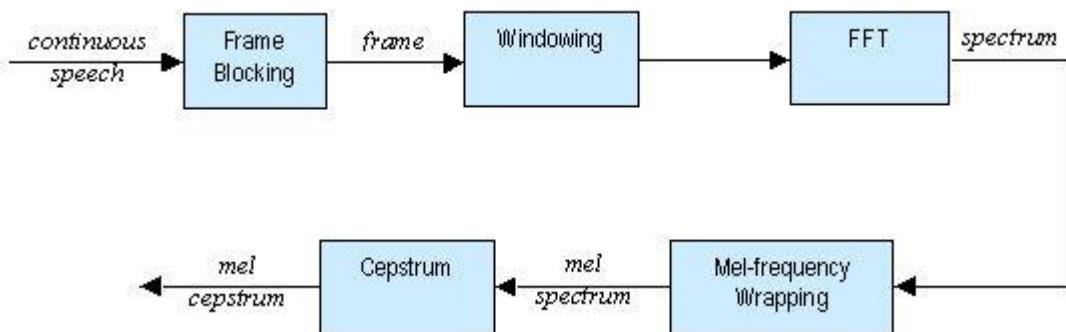
$$r(k) = \sum_{n=-\infty}^{\infty} x(n)x(n+k)$$



Hình 2.8 Quá trình trích đặc trưng LPC

## b. Phương pháp MFCC (Mel-Frequency Cepstrum Coefficients):

Phương pháp phân tích hệ số phô theo thang độ Mel (MFCC):



Hình 2. 9 Quá trình trích đặc trưng MFCC

- **Frame Blocking:**

- Trong bước này, tín hiệu tiếng nói liên tục được phân thành các khung gồm N mẫu, với các khung liền kề được chồng lấp bởi M mẫu ( $M < N$ ).
- Khung đầu tiên bao gồm N mẫu đầu tiên. Khung thứ hai bắt đầu M mẫu sau khung đầu tiên và chồng lên nó bởi N trừ đi M mẫu,...Quá trình này tiếp tục lặp lại cho đến khi tất cả tín hiệu tiếng nói được tính trong một hoặc nhiều khung.

- **Windowing:**

- Thay vì biến đổi Fourier trên cả đoạn âm thanh dài, ta trượt một cửa sổ dọc theo tín hiệu để lấy ra các frame rồi mới áp dụng FFT (Fast Fourier Transform) trên từng frame này. Tốc độ nói của con người trung bình khoảng 3 đến 4 từ mỗi giây, mỗi từ khoảng 3-4 âm, mỗi âm chia thành 3-4 phần, như vậy 1 giây âm thanh được chia thành 36 - 40 phần, ta chọn độ rộng mỗi frame khoảng 20-25ms là vừa đủ rộng để bao 1 phần âm thanh. Các frame phải được overlap lên nhau khoảng 10ms (khoảng 1/3 độ dài frame).

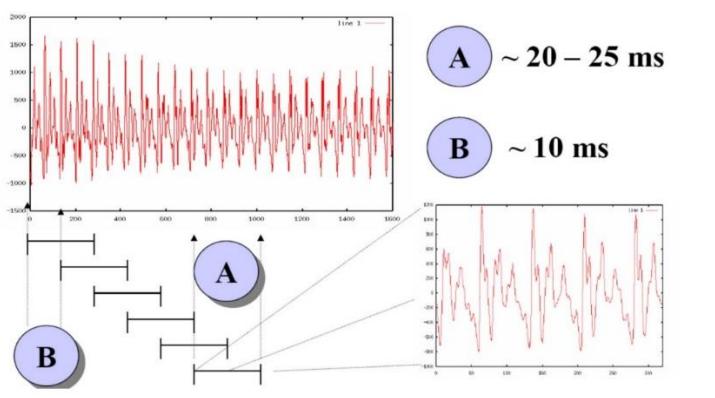
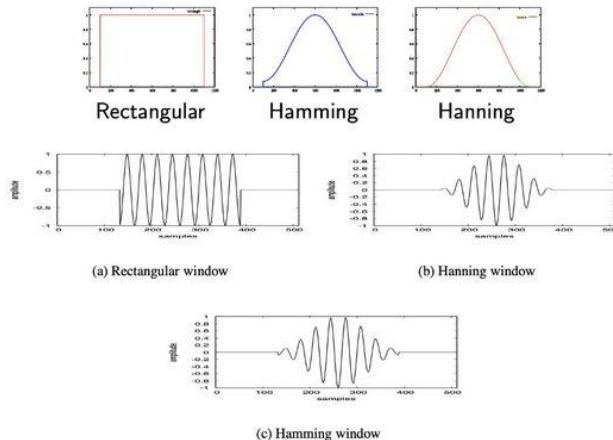


Image from Bryan Pellom

Hình 2. 10 Quá trình windowing

- Tuy nhiên, nếu cắt frame sẽ làm cho các giá trị ở 2 biên của frame bị giảm đột ngột về giá trị 0, sẽ dẫn tới hiện tượng là khi FFT sang miền tần số sẽ có rất nhiều nhiễu ở tần số cao. Để khắc phục được điều này, ta cần làm mượt bằng cách nhân chập frame với 1 vài loại window. Có nhiều loại window phổ biến là Hamming Window, Hanning Window,... có tác dụng làm giá trị biên frame giảm xuống từ từ.



(Taylor, fig 12.1)

Hình 2. 11 Các loại cửa sổ trong quá trình Windowing

### ○ Khối FFT (Fast Fourier transform):

- Bước xử lý tiếp theo là Fast Fourier Transform, chuyển đổi từng khung hình gồm N mẫu từ miền thời gian sang miền tần số. FFT là một thuật toán nhanh để triển khai Biến đổi Fourier rời rạc (DFT), được định nghĩa trên tập N mẫu  $\{x_n\}$ , như sau:

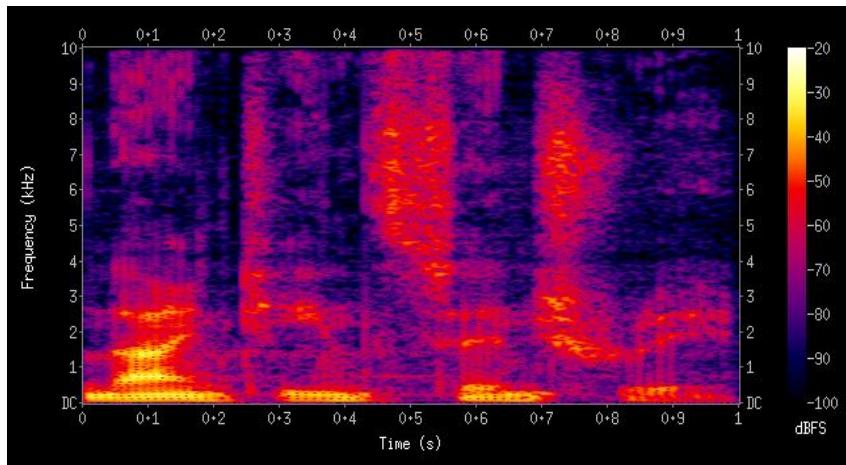
$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn}$$

Với  $k = 0, 1, 2, \dots, N-1$

- Nói chung  $X[k]$  là số phức và chúng ta chỉ xem xét giá trị tuyệt đối của chúng (độ lớn tần số). Chuỗi kết quả  $\{X_k\}$  được hiểu như sau: tần số dương tương ứng  $0 \leq f < \frac{F_s}{2}$  với

các giá trị  $0 \leq n \leq \frac{N}{2} - 1$ , trong khi tần số âm tương ứng với  $-\frac{F_s}{2} < f < 0$ . Ở đây,  $F_s$  biểu thị tần số lấy mẫu.

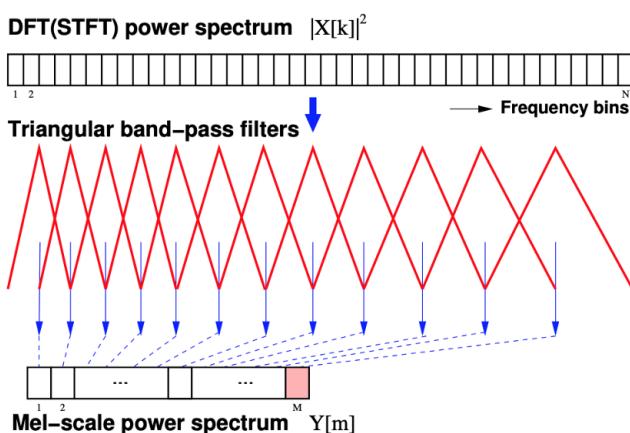
- Kết quả sau bước này thường được gọi là phô hoặc biểu đồ chu kỳ:



Hình 2. 12 Biểu đồ phô âm thanh theo chu kỳ

- **Khối Mel filterbank:**

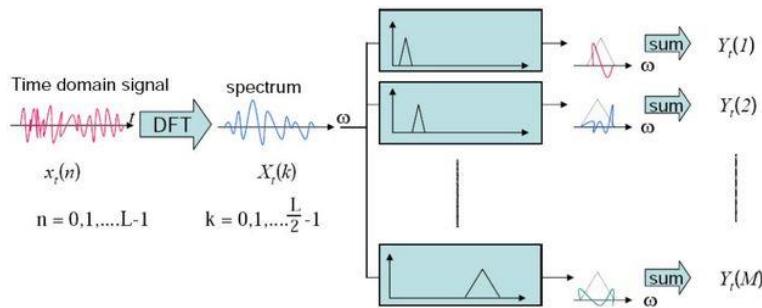
- Do tai người cảm nhận âm thanh theo hướng phi tuyến tính, cảm nhận tốt ở tần số thấp và kém nhạy cảm ở tần số cao, nên ta cần một cơ chế mapping dựa theo cách cảm nhận của tai người.



Hình 2. 13 Dãy bộ lọc Mel-filterbank

- Đầu tiên, ta sẽ bình phương các giá trị trong spectrogram thu được DFT power spectrum (phô công suất). Sau đó, ta sẽ áp dụng một tập các bộ lọc thông dải Mel-scale filter trên từng khoảng tần số (mỗi filter sẽ áp dụng trên một dải tần số nhất

định). Giá trị output của từng filter là năng lượng dải tần số mà filter đó bao phủ được. Ta sẽ thu được Mel-scale power spectrum.



Hình 2. 14 Quá trình chuyển từ spectrum sang Mel-scale power spectrum

### ○ Khối Cepstrum:

- Trong bước cuối cùng này, chúng tôi chuyển đổi phổ log mel thành miền thời gian. Kết quả được gọi là the mel frequency cepstrum coefficients (MFCC). Biểu diễn cepstral của phổ tiếng nói cung cấp một biểu diễn tốt về các đặc tính phổ cục bộ của tín hiệu cho phép phân tích khung đã cho. Bởi vì các hệ số phổ mel là số thực, chúng ta có thể chuyển đổi chúng sang miền thời gian bằng cách sử dụng Biến đổi Cosin rời rạc (DCT). Do đó, nếu chúng ta biểu thị các hệ số phổ công suất mel đó là kết quả của bước cuối cùng là  $\tilde{S}_k$ ,  $k = 0, 1, \dots, K-1$ , chúng ta có thể tính được các hệ số MFCC,  $\tilde{c}_n$  như

$$\tilde{c}_n = \sum_{k=1}^K (\log \tilde{S}_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K}\right], \quad n = 0, 1, \dots, K-1$$

Lưu ý rằng ta đã loại trừ thành phần đầu tiên, khỏi DCT vì nó đại diện cho giá trị trung bình của tín hiệu đầu vào, mang ít thông tin cụ thể về người nói.

### ○ Voice endpoint detection:

- Sử dụng phương pháp Endpoint detection để cắt ra từng từ trong một câu nói của người nói và đồng thời loại phần im lặng của tiếng nói.
- Sử dụng kết hợp 2 phương pháp là Zero Crossing Rate và Short-term Energy.

### ○ Short-term energy:

- Giả sử tín hiệu thời gian của tiếng nói là  $x(n)$ , hàm cửa sổ  $w(n)$  và tín hiệu giọng nói khung thứ  $i$  thu được sau quá trình xử lý khung là  $y_i(n)$ , khi đó  $y_i(n)$  thỏa mãn:

$$y_i(n) = w(n) * x((i-1)*inc + n),$$

$$1 \leq n \leq L, 1 \leq i \leq f_n$$

- Trong công thức trên,  $w(n)$  là hàm cửa sổ, nói chung là cửa sổ hình chữ nhật hoặc cửa sổ Hamming. Còn  $y_i(n)$  là giá trị biên độ của một khung,  $n = 1, 2, \dots, L$ ;  $i = 1, 2, \dots, f_n$ ;  $L$  là chiều dài khung;  $inc$  là chiều dài dịch chuyển khung;  $f_n$  là tổng số khung hình sau khi kết thúc khung.
- Công thức tính short-term energy của tín hiệu tiếng nói  $y_i(n)$  của khung thứ  $i$  là:

$$E(i) = \sum_{n=0}^{L-1} |y_i(n)|^2$$

#### ○ Biên độ trung bình (average amplitude):

- Biên độ trung bình của tín hiệu tiếng nói được định nghĩa là:

$$M(i) = \frac{1}{L} \sum_{n=0}^{L-1} |y_i(n)| \quad 1 \leq i \leq f_n$$

- $M(i)$  cũng là một đặc trưng của năng lượng một khung tín hiệu. Sự khác biệt giữa  $M(i)$  và  $E(i)$  là bất kể kích thước của giá trị được lấy mẫu trong quá trình tính toán, nó sẽ không được so sánh với bình phương của tín hiệu. Sự khác biệt lớn sẽ mang lại một số lợi ích trong một số lĩnh vực. Công dụng chính của hàm năng lượng ngắn hạn (short-term energy) và biên độ trung bình ngắn hạn (average amplitude) là: phân biệt các đoạn có tiếng và các đoạn ít tiếng, vì giá trị  $E(i)$  trong âm hữu thanh lớn hơn nhiều so với âm nhẹ ít tiếng.

#### ○ Short-term average zero-crossing rate:

- Tỷ lệ vượt mức 0 trung bình ngắn hạn cho biết số lần dạng sóng tín hiệu tiếng nói vượt qua trục hoành (điểm 0) trong khung tín hiệu tiếng nói. Đối với tín hiệu lời nói liên tục, điểm giao nhau bằng không có nghĩa là dạng sóng miền thời gian đi qua trục thời gian; trong khi đối với các tín hiệu rời rạc, nếu các giá trị mẫu liên kề thay đổi miền tín hiệu, nó được gọi là giao nhau bằng không. Tỷ lệ vượt 0 trung bình ngắn hạn là số lần giá trị mẫu thay đổi dấu hiệu.

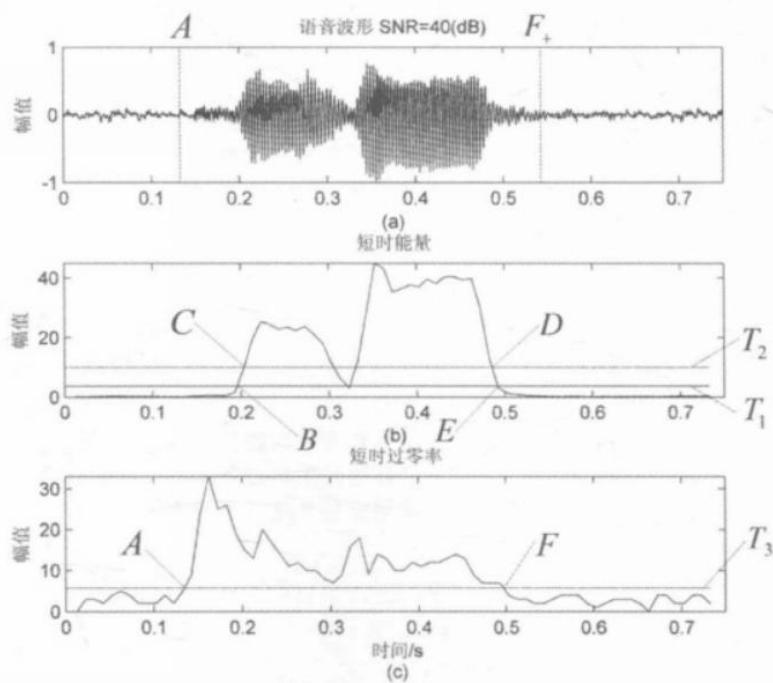
$$Z(i) = \frac{1}{2} \sum_{n=0}^{L-1} |sgn[y_i(n)] - sgn[y_i(n-1)]|$$

Trong công thức trên,  $sgn[]$  là một hàm tương trưng, có thể hiểu là:

$$\text{sgn}[x] = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

- **Quy tắc của Voice Endpoint Detection:**

- Phương pháp ngưỡng kép là phương pháp chính của Voice Endpoint Detection
- Phương pháp ngưỡng kép ban đầu được đề xuất dựa trên năng lượng trung bình ngắn hạn và tỷ lệ vượt ngưỡng trung bình ngắn hạn. Các nguyên âm được tìm thấy trong năng lượng trung bình thời gian, và các chữ cái đầu là phụ âm. Tần suất của chúng cao hơn và tỷ lệ vượt qua trung bình ngắn hạn tương ứng cao hơn. Phương pháp ngưỡng kép được thực hiện bằng cách sử dụng phán đoán hai cấp độ.



Hình 2. 15 Phương pháp ngưỡng kép

- **Mức độ phán đoán đầu tiên:**

- Thực hiện phán đoán sơ bộ dựa trên ngưỡng cao hơn ngưỡng T2 được chọn trên đường bao năng lượng ngắn hạn của giọng nói (được biểu thị bằng một đường ngang đứt nét), nghĩa là giọng nói cao hơn ngưỡng T2 phải là giọng nói (nghĩa là giữa đoạn CD). Nó phải là lời nói và điểm bắt đầu và điểm kết thúc của lời nói phải nằm ngoài điểm thời gian tương ứng với giao điểm của ngưỡng và vùng năng lượng ngắn hạn (nghĩa là nằm ngoài đoạn CD).

- Xác định ngưỡng thấp hơn ngưỡng trên năng lượng trung bình (được biểu thị bằng đường liền nét ngang trong hình) và tìm kiếm từ điểm C sang trái và từ điểm D sang phải để tìm hai giao điểm của đường bao năng lượng ngắn hạn và ngưỡng T1. Có hai điểm B và E nên đoạn BE là điểm đầu và điểm cuối của đoạn tiếng được xác định theo phương pháp ngưỡng kép theo năng lượng ngắn hạn.
- **Mức độ phán đoán thứ hai:**
  - Dựa trên tỷ lệ vượt 0 trung bình ngắn hạn, tìm kiếm từ điểm B sang trái và từ điểm E sang phải để tìm hai điểm A và F có tỷ lệ vượt 0 trung bình ngắn hạn thấp hơn một ngưỡng nhất định là ngưỡng T3, đây là điểm bắt đầu và điểm kết thúc của đoạn nói.
  - Theo hai mức độ phán đoán này, vị trí điểm bắt đầu A và vị trí điểm kết thúc F của tiếng nói. Tuy nhiên, xem xét rằng vùng im lặng giữa các từ trong khi phát âm tiếng nói sẽ có độ dài tối thiểu để biểu thị khoảng dừng giữa các lần phát âm, người ta đánh giá rằng đoạn giọng nói kết thúc khi nó nhỏ hơn ngưỡng T3 và đáp ứng độ dài tối thiểu như vậy, thực tế là tương đương với việc kéo dài phần cuối của tiếng nói như trong hình, điểm bắt đầu và điểm kết thúc của giọng nói được đánh dấu là A và F+ trên biểu đồ tín hiệu tiếng nói (xem hình vẽ rằng điểm kết thúc là F, nhưng quá trình xử lý thực tế được mở rộng đến F+).
  - Trong hoạt động cụ thể của phát hiện điểm cuối, bước đầu tiên là chia giọng nói thành các khung và năng lượng trung bình ngắn hạn có thể thu được trên cơ sở phân khung. Tỷ lệ bằng không, sau đó so sánh và phán đoán theo từng khung giá trị ngưỡng.

### c. So sánh hai phương pháp và đưa ra kết luận:

chúng em lựa chọn phương pháp phân tích hệ số phổ theo thang độ Mel (MFCC) vì đây là phương pháp trích chọn tham số tiếng nói được sử dụng rộng rãi hơn bởi tính hiệu quả của nó thông qua phân tích cepstral theo thang đo Mel.

Phương pháp này khác biệt so với LPC vì nó xây dựng dựa trên sự cảm nhận của tai người đối với các tần số khác nhau. Với các tần số thấp (dưới 1000Hz), độ cảm nhận của tai người là tuyến tính. Đối với các tần số cao, độ biến thiên tuân theo hàm logarit. Các băng lọc tuyến tính ở tần số thấp và biến thiên theo hàm logarit ở tần số cao được sử dụng để trích chọn các đặc trưng âm học quan trọng của tiếng nói.

### 2.3.6 Các phương pháp huấn luyện và nhận dạng

Các đặc trưng sau khi được tạo thành, dù bằng bất cứ phương pháp nào cũng sẽ được dùng để huấn luyện tạo cơ sở dữ liệu và nhận dạng về sau

Các kỹ thuật chính được sử dụng cho việc nhận dạng tiếng nói có thể kể đến:

#### a. Vector quantization (VQ):

VQ là phương pháp ánh xạ những vector trong một không gian lớn thành một số lượng hữu hạn các vector cũng nằm trong không gian đó. Mỗi vùng các không gian rộng lớn gọi là một bó (cluster) có thể đặc trưng bởi tâm của nó gọi là “codeword”. Tập hợp các codeword này gọi là “codebook”.

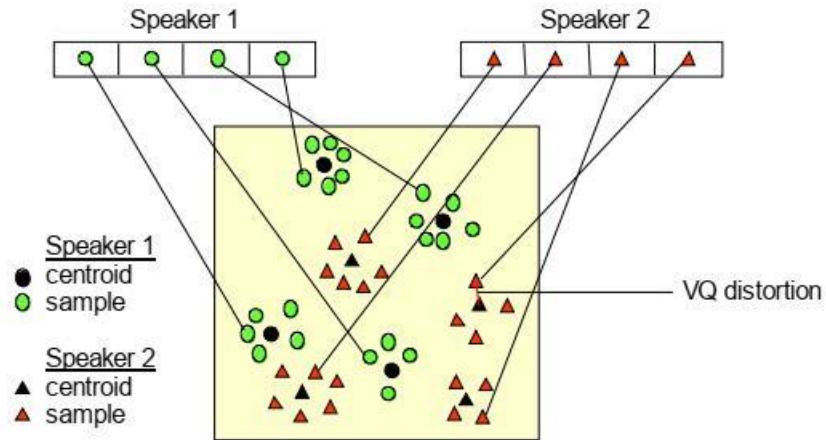
Các ưu điểm của VQ:

- Giảm thiểu lượng dữ liệu lưu trữ
- Giảm thời gian tính toán độ giống nhau giữa các vector phô. Trong nhận dạng tiếng nói, một lượng lớn phép tính dùng để tính sự giống nhau giữa hai cặp phô. Dựa vào VQ, việc tính toán đó được giảm xuống qua việc tìm sự giống nhau của 2 cặp vector codebook trong bảng tìm kiếm.
- Biểu diễn rời rạc về mặt âm học của tiếng nói. Nhờ gán nhãn cho từng frame của từng từ, quá trình chọn codebook tốt nhất cho từ đó trong các hệ nhận dạng tiếng nói chỉ đơn thuần dựa trên các nhãn này

Các khuyết điểm của VQ :

- Việc lượng tử vector chắc chắn dẫn đến sai số lượng tử hóa. Điều này dẫn đến thông tin phô bị sai lệch.
- Việc chọn kích thước codebook cho VQ không đơn giản. Tăng kích thước sẽ giảm sai số lượng tử nhưng lại dẫn đến vấn đề không gian lưu trữ các vector trong codebook. Vì vậy khi cài đặt VQ, chúng ta phải cân nhắc 3 yếu tố: sai số lượng tử, thời gian tìm kiếm vector trong codebook và không gian lưu trữ các vector trong codebook.
- Hình bên dưới chỉ nguyên lý để minh họa cho giải thuật. Trong hình vẽ này, chỉ có 2 giọng nói và 2 chiều của không gian acoustic vector được trình bày. Vòng tròn chỉ acoustic vector của người thứ nhất, hình tam giác chỉ acoustic vector của người thứ 2.

Trong quá trình huấn luyện, thuật toán tạo chùm (sẽ được trình bày sau) được dùng để tạo ra một VQ codebook của từ đó.



Hình 2. 16 Phương pháp Vector Quantization

- Để nhận dạng, khoảng cách Euclid được dùng để tính khoảng cách từ mỗi acoustic vector đến codeworld gần nhất của mỗi codebook đã được huấn luyện (nằm trong database). Tiếng nói nào tương ứng với tổng các khoảng cách Euclid đến một VQ codebook nào đó nhỏ nhất thì tương ứng với tiếng nói đã tạo nên VQ codebook đó.
- Khoảng cách Euclid:
- Khoảng cách Euclid giữa 2 vector n chiều  $a(a_1, a_2, \dots, a_n)$  và  $b(b_1, b_2, \dots, b_n)$  được tính như sau:

$$l = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + \dots + (b_n - a_n)^2}$$

Thuật toán tạo chùm – clustering the training vector:

- Mục đích của thuật toán tạo chùm là tạo mỗi VQ codebook cho mỗi từ thu được từ acoustic vector đã tạo ra từ phần trước. Thuật toán LBG (Linde, Buzo & Gray, 1980) là thuật toán nổi tiếng để nhóm L vector huấn luyện thành M codebook vector.
- Thuật toán LBG được tóm tắt như sau:
  - Khởi tạo 1 vector codebook, là trung tâm của tất cả các acoustic vector thu nhận được từ phần trước.
  - Nhân đôi kích thước của codebook bằng cách chia đôi codebook hiện hành  $y_n$  theo quy tắc sau:

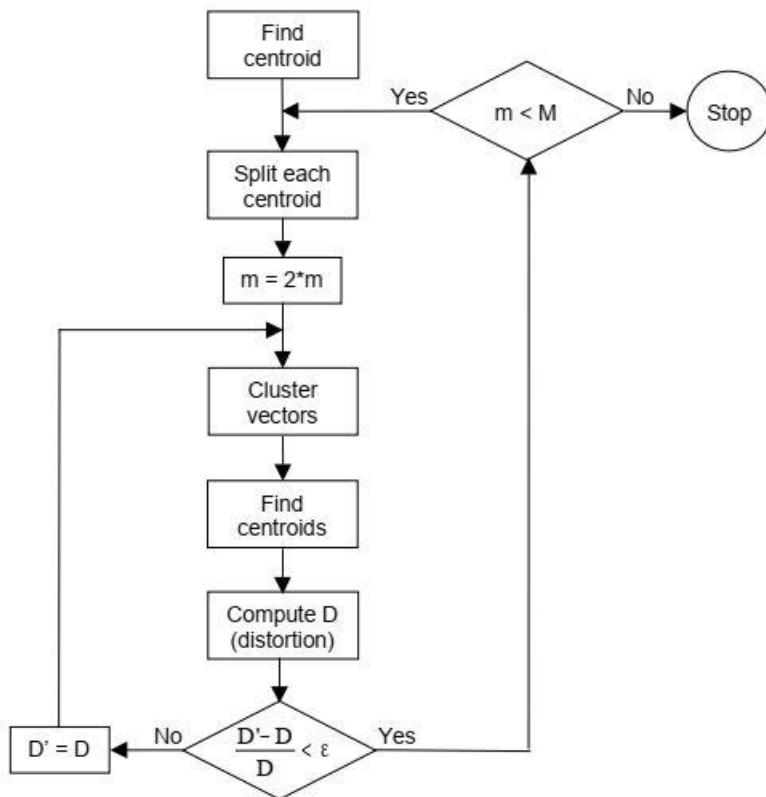
$$y_n^+ = y_n(1 + \varepsilon)$$

$$y_n^- = y_n(1 - \varepsilon)$$

Với các  $n = 1 \dots$  (kích thước của codebook)

$\varepsilon$ : hệ số chia. Ví dụ  $\varepsilon = 0.01$

- Nearest Neighbor Search: với các vector huấn luyện, ở đây là acoustic vector, tìm các codeword trong codebook hiện thời gần với nó nhất (theo khoảng cách Euclid), xếp xép các vector này vào trong các cell tương ứng.
- Centroid Update: cập nhật codeword trong mỗi cell ở bước 3, dùng centroid của mỗi acoustic vector tương ứng với cell đó, thông thường nhất là dùng trung bình cộng các acoustic vector của cell đó.
- Lặp lại bước 3 & 4 cho đến khi khoảng cách trung bình nhỏ hơn ngưỡng cho trước
- Lặp lại bước 2, 3 & 4 cho đến khi kích thước của codebook đến giá trị mong muốn



Hình 2. 17 Giải thuật Vector Quantization

Với: M: kích thước của codebook cần tìm

Cluster vectors: là bước (3), nearest neighbour search.

Find centroids: là bước (4).

Nhận dạng bằng VQ:

- Quá trình nhận dạng có thể miêu tả thông qua các bước như sau:
  - Với mỗi âm thanh phát ra trích đặc trưng của nó, là các Acoustic Vector như đã nói ở trên.
  - Tính khoảng cách của Acoustic vector này với từng Codebook của các từ đã huấn luyện.
  - Tìm khoảng cách Euclidean nhỏ nhất của các Acoustic Vector này đến từng Codeword của các Codebook này. Tính trung bình khoảng cách này. Ứng với mỗi Codebook của từ đã được huấn luyện, ta có tương ứng mỗi khoảng cách như vậy.
  - VD: Giả sử từ cần nhận dạng sau khi rút trích đặc trưng có T vector:  $x_1, x_2, x_3, \dots, x_T$ . Và codebook cho từ thứ i là  $\{y_{1i}, y_{2i}, y_{3i}, \dots, y_{Mi}\}$ , với M là kích thước của codebook. Độ méo trung bình của từ cần nhận dạng sau khi lượng tử ứng với codebook thứ i là:

$$D_i = \frac{1}{T} \sum_{t=1}^T \min[d(x_t, y_{mi})] \quad \text{với } 1 \leq m \leq M$$

- Tìm khoảng cách nhỏ nhất của Acoustic Vector thu được với Codebook của các từ đã được huấn luyện, ta sẽ gán từ thu được với từ (đã huấn luyện) tương ứng có khoảng cách nhỏ nhất.

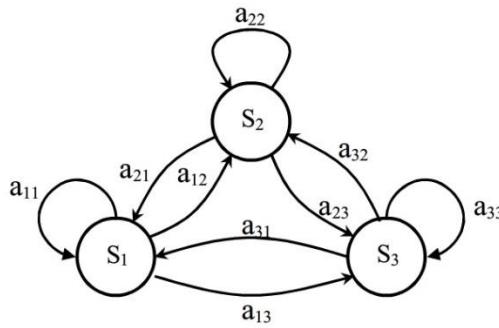
$$j = \operatorname{argmin}[D_i] \quad \text{với } 1 \leq i \leq V; V \text{ là số từ vựng}$$

## b. Hidden Markov Model:

Mô hình Markov:

- Xét một hệ thống gồm N trạng thái phân biệt, được đánh số thứ tự 1, 2, ..., N. Tại thời điểm t bất kỳ, hệ thống có thể chuyển từ trạng thái  $S_i$  sang một trong N – 1 trạng thái còn lại hoặc chuyển trở lại chính trạng thái  $S_i$ .
- Như vậy, ở thời điểm t, từ trạng thái  $S_i$  có N nhánh thao tác chuyển trạng thái. Mỗi nhánh này có một độ đo khả năng xảy ra (xác suất xảy ra), được gọi là xác suất chuyển trạng thái.

**Ví dụ:**



Hình 2.18 Mô hình Markov

- Gọi  $q_t$  là trạng thái đạt được ở thời điểm  $t$ ,  $a_{ij}$  là xác suất chuyển trạng thái từ  $S_i$  sang trạng thái  $S_j$  (xác suất  $S_j$  xảy ra với điều kiện  $S_i$  đã xảy ra). Xác suất chuyển trạng thái  $a_{ij}$  này không phụ thuộc vào thời gian  $t$  và độc lập với các trạng thái đã chuyển trước đó. Duy nhất chỉ phụ thuộc vào trạng thái hiện tại. Quá trình mang tính ngẫu nhiên này được gọi là “có thuộc tính Markov”. Ta có:

$$a_{ij} = p(q_{t+1} = S_j \mid q_t = S_i) \quad 1 \leq i, j \leq N$$

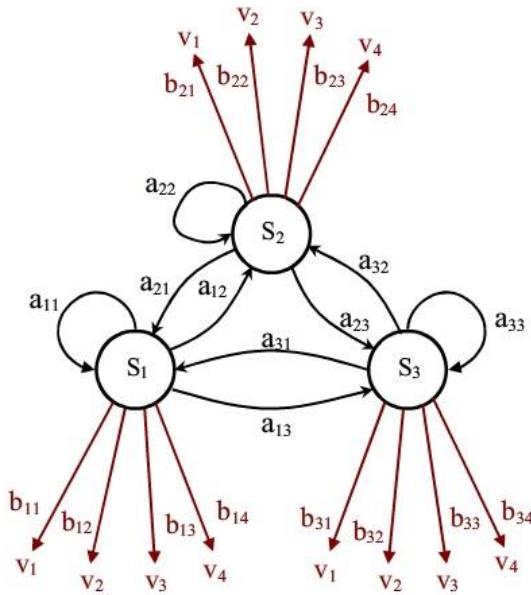
Và  $a_{ij}$  phải thỏa mãn các ràng buộc xác suất:  $a_{ij} \geq 0$ ,  $\sum_{j=1}^N a_{ij} = 1$

- Kết xuất của hệ thống là một chuỗi các trạng thái tại các thời điểm  $t$  tương ứng. Ta biết được trạng thái nào ở thời điểm  $t$  nào, vì vậy mô hình này gọi là **mô hình Markov Hiện (Observable Markov Model)**.
- Xác suất chuyển trạng thái  $a_{ij}$  cho tất cả các trạng thái trong hệ thống có thể được mô tả bằng ma trận chuyển trạng thái:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}$$

Mô hình Markov ẩn:

- Mô hình Markov ẩn là dạng mở rộng của mô hình Markov. Trong mô hình Markov, các sự kiện quan sát được nằm trong mỗi trạng thái và phụ thuộc vào và phụ thuộc vào hàm mật độ xác suất trong các trạng thái đó.



Hình 2. 19 Mô hình Hidden Markov

- Hình trên minh họa mô hình Markov ẩn 3 trạng thái với các sự kiện có thể quan sát được trong mỗi trạng thái là  $V = \{v_1, v_2, v_3, v_4\}$ . Khả năng (xác suất) quan sát được sự kiện  $v_k$  trong trạng thái  $S_j$  phụ thuộc vào hàm xác suất  $b_j(k)$ . Hàm  $b$  được gọi là hàm mật độ xác suất của các sự kiện được quan sát.

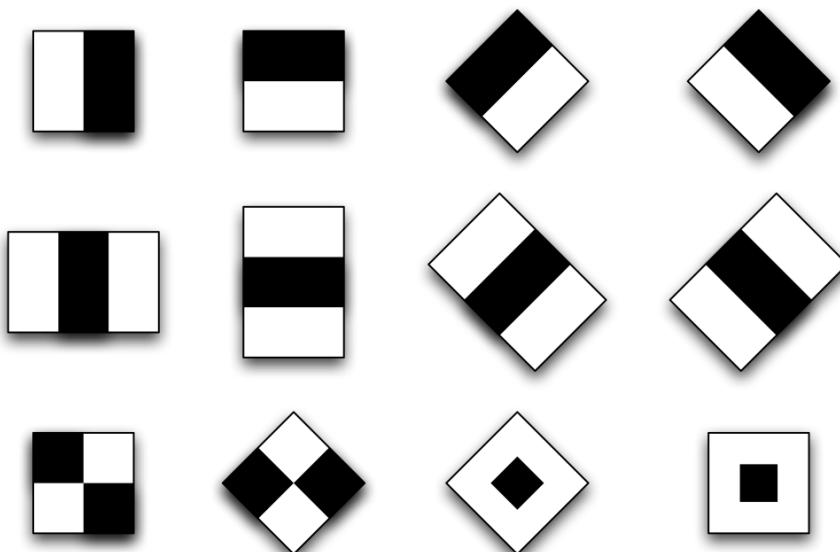
## 2.4 Tìm hiểu lý thuyết về công nghệ nhận dạng khuôn mặt

### 2.4.1 Phát hiện khuôn mặt (Face Detection)

#### a. Các đặc trưng Haar-like:

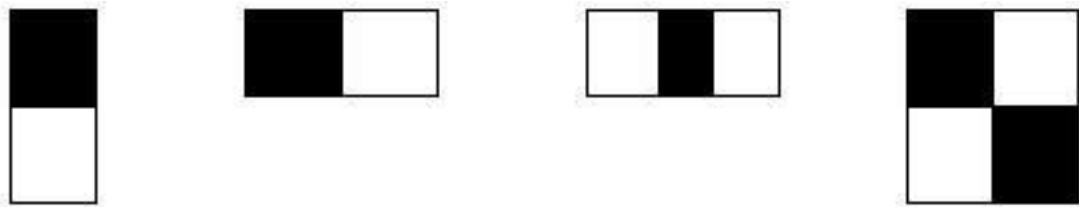
Trong bài toán lần này, nhóm em chọn thuật toán Viola Jones và tính toán các đặc trưng Haar-like để tiến hành nhận diện khuôn mặt

Các đặc trưng Haar-like là những hình chữ nhật được phân thành các vùng khác nhau như hình:



Hình 2. 20 Các đặc trưng Haar-like

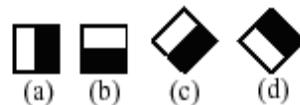
Đặc trưng do Viola và Jones công bố gồm 4 đặc trưng cơ bản để xác định khuôn mặt người. Mỗi đặc trưng Haar-like là sự kết hợp của hai hay ba hình chữ nhật trắng đen như trong hình sau:



Hình 2. 21 Bốn đặc trưng Haar-like cơ bản

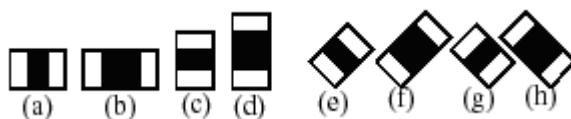
Để sử dụng các đặc trưng này vào việc xác định khuôn mặt người, 4 đặc trưng Haar-Like cơ bản được mở rộng ra và được chia làm 3 đặc trưng như sau:

- Đặc trưng cạnh (edge feature):



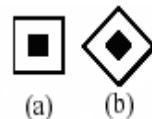
Hình 2. 22 Đặc trưng cạnh

- Đặc trưng đường (line feature):



Hình 2.23 Đặc trưng đường

- Đặc trưng xung quanh tâm (center-surround features)

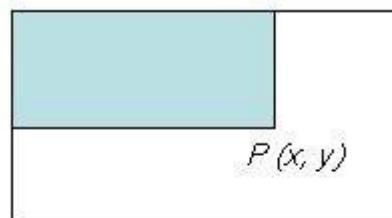


Hình 2.24 Đặc trưng xung quanh tâm

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

$$f(x) = \text{Tổng}_\text{vùng\_đen}(\text{các mức xám của pixel}) - \text{Tổng}_\text{vùng\_trắng}(\text{các mức xám của pixel})$$

Viola và Jones đưa ra một khái niệm gọi là Integral Image, là một mảng 2 chiều với kích thước bằng với kích thước của ảnh cần tính đặc trưng Haar-Like, với mỗi phần tử của mảng này được tính bằng cách tính tổng của điểm ảnh phía trên (dòng-1) và bên trái (cột-1) của nó.



Hình 2.25 Integral Image

Công thức tính Intergral image:

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

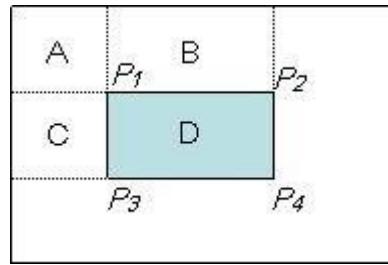
Sau khi tính được Integral Image, việc tính tổng các giá trị mức xám của một vùng bất kỳ nào đó trên ảnh thực hiện rất đơn giản theo cách sau:

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

$$D = A + B + C + D - (A+B) - (A+C) + A$$

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

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

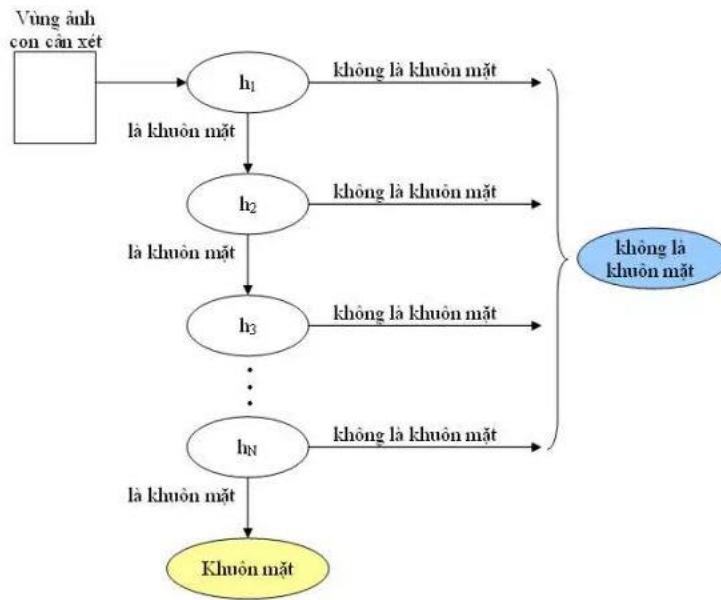


Hình 2. 26 Cách tính 4 điểm trên Integral Image

### b. Bộ phân loại AdaBoost:

AdaBoost là một bộ phân loại mạnh phi tuyến phức dựa trên hướng tiếp cận boosting. AdaBoost cũng hoạt động trên nguyên tắc kết hợp tuyến tính các weak classifiers để hình thành một trong các classifiers

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:



Hình 2. 27 Bộ phân loại AdaBoost

Trong đó,  $h(k)$  là các bộ phân loại yếu, được biểu diễn như sau:

$$h_k = \begin{cases} 1 & \text{nếu } p_k f_k(x) < p_k \theta_k \\ 0 & \text{ngược lại} \end{cases}$$

với  $x$ : cửa sổ con cần xét

$\theta_k$ : ngưỡng

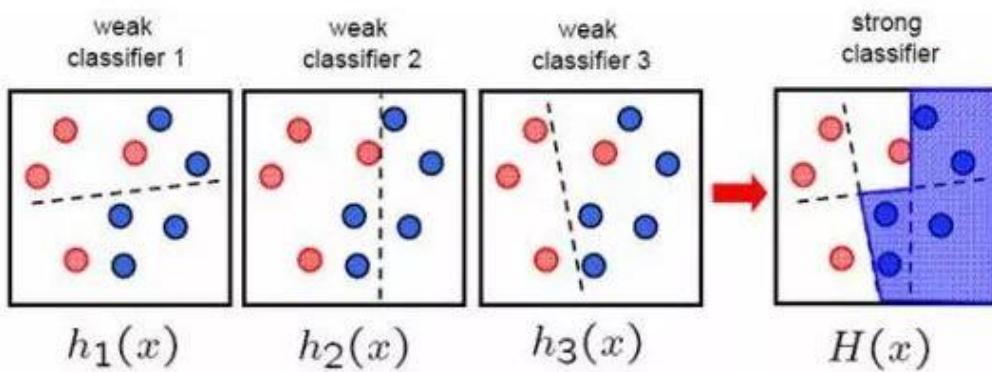
$f_k$ : giá trị của đặc trưng Haar-like

$p_k$ : hệ số quyết định chiều của phương trình

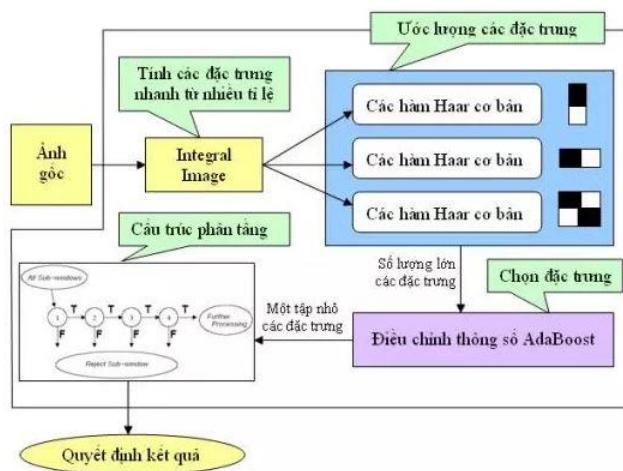
AdaBoost sẽ kết hợp các bộ phân loại yếu thành bộ phân loại mạnh như sau:

$$H(x) = \sum (\alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_n h_n(x))$$

Với  $\alpha_t \geq 0$  là hệ số chuẩn hóa cho các bộ phân loại yếu



Hình 2.28 Kết hợp các bộ phân loại yếu thành bộ phân loại mạnh



Hình 2.29 Sơ đồ nhận diện khuôn mặt

#### 2.4.2 Nhận dạng khuôn mặt (Face Recognition)

Bài toán nhận dạng khuôn mặt được ứng dụng nhiều trong các lĩnh vực đời sống đặc biệt là ở những lĩnh vực công nghệ cao, yêu cầu về an ninh, bảo mật. Do đó để hệ thống nhận dạng khuôn mặt hoạt động được mạnh mẽ với tốc độ và độ tin cậy cao, có rất nhiều phương pháp nhận dạng khuôn mặt được đưa ra. Trên thực tế người ta hay chia ra phương pháp nhận dạng khuôn mặt thành 3 loại:

- Nhận dạng dựa trên phương pháp tiếp cận toàn cục
- Nhận dạng dựa trên phương pháp tiếp cận các đặc điểm cục bộ
- Phương pháp nhận dạng lai (hybrid) là sự kết hợp của hai phương pháp tiếp cận toàn cục và phương pháp tiếp cận dựa trên các đặc điểm cục bộ

### a. Nhận dạng dựa trên phương pháp tiếp cận các đặc điểm cục bộ

Đây là phương pháp nhận dạng khuôn mặt dựa trên việc xác định các đặc trưng hình học của các chi tiết trên khuôn mặt như vị trí, diện tích, khoảng cách của mắt mũi miệng,... và mỗi quan hệ giữa chúng ví dụ như khoảng cách giữa 2 mắt.

Ưu điểm của phương pháp này là nó gần với cách mà con người sử dụng để nhận biết khuôn mặt. Hơn nữa với việc xác định đặc tính và các mối quan hệ, phương pháp này cho kết quả tốt trong các điều kiện không có kiểm soát

Nhược điểm của phương pháp này là thực hiện thuật toán khá phức tạp do việc xác định mối quan hệ giữa các đặc tính, phải đòi hỏi các thuật toán phức tạp và phương pháp này sẽ không hoạt động hiệu quả khi kích thước hình ảnh nhỏ vì rất khó phân biệt được các đặc tính.

Có thể kể đến một vài phương pháp tiếp cận các đặc điểm cục bộ như: Local Feature Based, LBP, Gabor Wavelets

### b. Nhận dạng dựa trên xét toàn bộ khuôn mặt

Nội dung chính của hướng tiếp cận này là xem mỗi ảnh có kích thước MxN là một vector trong không gian có MxN chiều. Ta xây dựng một không gian mới có chiều nhỏ hơn sao cho khi biểu diễn không gian đó các đặc điểm chính trên khuôn mặt không bị mất đi. Trong không gian đó các ảnh của cùng một người sẽ được tập trung lại thành một nhóm gần nhau và cách xa so với các nhóm khác. Hai phương pháp thường được sử dụng trong hướng tiếp cận này là:

- PCA (Principle Components Analysis)
- LDA (Linear Discriminant Analysis)

Cả PCA và LDA đều là phương pháp chuyển đổi tuyến tính: trong khi LDA là phương pháp học giám sát thì PCA là phương pháp học không giám sát, PCA bỏ qua việc đánh nhãn các lớp (class).

Cần lưu ý rằng LDA đưa ra các giả định về sự bằng nhau của các lớp phân phối chuẩn và các lớp hiệp phương sai. Trong một bài báo khoa học so sánh thực nghiệm 2 phương pháp có chỉ ra rằng PCA có xu hướng dẫn đến kết quả phân loại tốt hơn trong nhiệm vụ nhận dạng hình ảnh nếu số lượng mẫu cho một lớp nhất định là tương đối nhỏ: A. M. Martinez and A. C. Kak. PCA versus LDA. IEEE

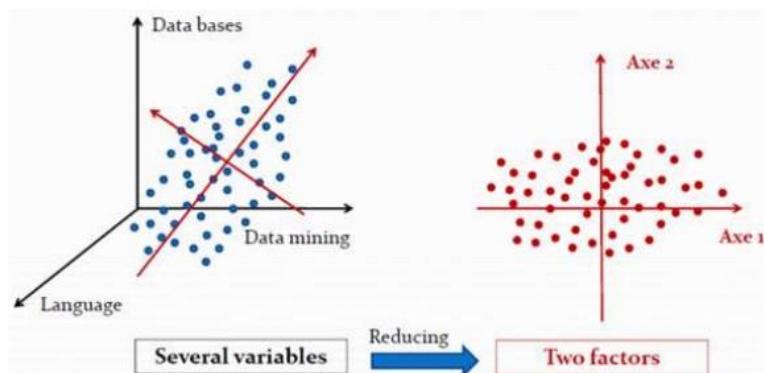
Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No.2, pp. 228-233, 2001.

Ở đề tài lần này, chúng em sẽ chọn phương pháp PCA nhằm mục đích trích rút đặc trưng cho công việc nhận dạng khuôn mặt, lý do là vì giải thuật của phương pháp này đơn giản hơn nhận dạng dựa trên đặc trưng khuôn mặt nhưng vẫn cho ra được hiệu quả nhận dạng tương đối tốt

## 2.5 Áp dụng thuật toán PCA cho nhận dạng khuôn mặt

### 2.5.1 Giới thiệu và lý do chọn PCA

Về bản chất, PCA tìm ra một không gian mới với số chiều nhỏ hơn không gian cũ. Các trục tọa độ không gian mới được xây dựng sao cho trên mỗi trục, độ biến thiên của dữ liệu trên đó là lớn nhất có thể. Trong không gian mới này người ta hy vọng rằng việc phân loại sẽ mang lại kết quả tốt hơn so với không gian ban đầu.



Hình 2. 30 Trước và sau khi giảm chiều dữ liệu bằng PCA

Thuật toán PCA được chọn thay vì các thuật toán vì tính đơn giản của thuật toán. Ngoài ra, thuật toán PCA còn mang lại độ chính xác tương đối tốt, chỉ kém hơn một chút so với các thuật toán phức tạp còn lại.

### 2.5.2 Cơ sở toán học

Kỳ vọng (mean): là giá trị mong muốn, nói đơn giản là trung bình cộng của toàn bộ các giá trị

Cho N giá trị  $x_1, x_2, \dots, x_N$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

Phương sai (variance): Là trung bình cộng của bình phương khoảng cách từ mỗi điểm tới kỳ vọng, phương sai càng nhỏ thì các điểm dữ liệu càng gần với kỳ vọng, tức các điểm dữ liệu càng giống nhau. Phương sai càng lớn thì ta nói dữ liệu càng có tính phân tán

$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$$

Hiệp phương sai (covariance): Là độ đo sự biến thiên cùng nhau của hai biến天然 (phân biệt với phương sai – đo mức độ biến thiên của một biến). Nếu 2 biến có xu hướng thay đổi cùng nhau (nghĩa là, khi một biến có giá trị cao hơn giá trị kỳ vọng thì biến kia có xu hướng cũng cao hơn giá trị kỳ vọng), thì hiệp phương sai giữa hai biến này có giá trị dương. Mặt khác, nếu một biến nằm trên giá trị kỳ vọng còn biến kia có xu hướng nằm dưới giá trị kỳ vọng, thì hiệp phương sai của hai biến này có giá trị âm. Nếu hai biến này độc lập với nhau thì giá trị bằng 0

$$COV(X, Y) = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{N}$$

Ma trận hiệp phương sai:

- Cho N điểm dữ liệu được biểu diễn bởi các vector cột  $x_1, \dots, x_N$ , khi đó, vector kỳ vọng và ma trận hiệp phương sai của toàn bộ dữ liệu được định nghĩa là:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

$$S = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T = \frac{1}{N} \hat{X} \hat{X}^T$$

- Ma trận hiệp phương sai là một ma trận đối xứng. Hơn nữa, nó là một ma trận nửa xác định dương.
- Mọi phần tử trên đường chéo của ma trận hiệp phương sai là các số không âm. Chúng cũng chính là phương sai của từng chiều của dữ liệu.
- Các phần tử ngoài đường chéo  $S_{ij}$ ,  $i \neq j$  thể hiện sự tương quan giữa thành phần thứ i và thứ j của dữ liệu, còn được gọi là hiệp phương sai. Giá trị này có thể dương, âm hoặc bằng 0. Khi nó bằng 0, ta nói rằng hai thành phần i, j trong dữ liệu là không tương quan

- Nếu ma trận hiệp phương sai là ma trận đường chéo, ta có dữ liệu hoàn toàn không tương quan giữa các chiều.

$$S = [\text{var}(x) \quad \text{cov}(x,y)\text{cov}(y,x) \quad \text{var}(y)]$$

Trị riêng (eigenvalue), vector riêng (eigenvector) của covariance matrix:

- Cho một ma trận vuông  $A \in R^{n \times n}$ , nếu số vô hướng  $\lambda$  và vector  $x \neq 0 \in R^n$  thoả mãn:

$$Ax = \lambda x$$

thì  $\lambda$  được gọi là một trị riêng của  $A$  và  $x$  được gọi là vector riêng tương ứng với trị riêng đó.

- Trị riêng là nghiệm của phương trình đặc trưng

$$\det(A - \lambda I) = 0$$

- Một trị riêng có thể có nhiều vector riêng
- Mỗi vector riêng chỉ ứng với một trị riêng duy nhất
- Nếu  $x$  là một vector riêng của  $A$  ứng với  $\lambda$  thì  $kx$ ,  $k \neq 0$  cũng là vector riêng ứng với trị riêng đó
- Mọi ma trận vuông bậc  $n$  đều có  $n$  trị riêng (kể cả lặp) và có thể là các số phức.
- Với ma trận đối xứng, tất cả các trị riêng đều là các số thực
- Với ma trận xác định dương, tất cả các trị riêng của nó đều là các số thực dương. Với ma trận nửa xác định dương, tất cả các trị riêng của nó đều là các số thực không âm.
- Phương pháp giải tìm trị riêng, vector riêng:
  - o Bước 1: giải phương trình đặc trưng để tìm trị riêng

$$\det(A - \lambda I) = 0$$

- o Bước 2: giải hệ phương trình tìm vector riêng  $u_i$  tương ứng với trị riêng  $\lambda_i$ .

$$(A - \lambda I)u = 0$$

$$|x_1 - x_2| + |y_1 - y_2|$$

- Vector Norm:
- Ta có vector  $\vec{a} = [-5, 6, 8, -10]$ . Một tập hợp bất kì của các số vector  $\vec{x} = (x_1, x_2, \dots, x_N)$  sẽ đại diện cho một vector. Các chuẩn của vector được định nghĩa như sau:

- Chuẩn 1 - norm 1 của một vector là tổng các giá trị tuyệt đối của các phần tử của vectơ. Ví dụ đối với vector  $\vec{a}$  kể trên có norm 1 là 29. Nó được biểu diễn theo công thức toán học sau:

$$\|\vec{x}\| = \sum_{i=1}^N |x_i|$$

- Chuẩn 2 - norm 2 của một vector là căn bậc hai của tổng bình phương của mỗi phần tử của vectơ. Ví dụ đối với vector  $\vec{a}$  kể trên có norm 2 là 15. Nó được biểu diễn theo công thức toán học sau:

$$\|\vec{x}\| = \sqrt{\sum_{i=1}^N |x_i|^2}$$

### 2.5.3 Thuật toán PCA

PCA (Principal Component Analysis) về cơ bản là một phương pháp giảm kích thước đơn giản, biến đổi các cột của bộ dữ liệu thành một tập các đặc trưng mới. Nó thực hiện điều này bằng cách tìm ra một tập hợp các hướng mới (như trục x và y) giải thích sự biến đổi tối đa trong dữ liệu, tức là hướng đó ta tìm được tối đa của phương sai.

Áp dụng phương pháp PCA này vào trong trích chọn đặc trưng và nhận dạng khuôn mặt, quá trình nhận dạng bao gồm những bước sau:

**Bước 1:** Chuẩn bị dữ liệu huấn luyện, thu thập các ảnh mặt người bao gồm ảnh chính chủ và ảnh không chính chủ. Các ảnh phải có điều kiện chụp giống nhau

**Bước 2:** Chuẩn hóa tập ảnh huấn luyện về kích thước

Để thuận tiện cho quá trình minh họa, ta xét một tập dữ liệu đầu vào hai chiều. Để đảm bảo yêu cầu trong quá trình xử lý, tất cả dữ liệu của tập huấn luyện phải được chuẩn hóa về cùng một điều kiện ban đầu. Với M khuôn mặt từ  $F_1, F_2, \dots, F_M$

**Bước 3:** Thực hiện phép trừ trung bình

Các ảnh khuôn mặt có kích thước NxN được biểu diễn thành vector  $N^2 \times 1$

Ví dụ với tập bốn ảnh có giá trị sau:

$$\bullet \quad \Gamma_1 = \begin{bmatrix} 1 & 224 & 35 \\ 56 & 98 & 158 \\ 50 & 0 & 78 \end{bmatrix} \quad \Gamma_2 = \begin{bmatrix} 25 & 100 & 59 \\ 60 & 79 & 99 \\ 157 & 115 & 200 \end{bmatrix}$$

$$\Gamma_3 = \begin{bmatrix} 19 & 30 & 159 \\ 199 & 40 & 69 \\ 75 & 32 & 199 \end{bmatrix} \quad \Gamma_4 = \begin{bmatrix} 20 & 30 & 1 \\ 234 & 177 & 61 \\ 161 & 213 & 11 \end{bmatrix}$$

- Thực hiện chuyển đổi từ ma trận có kích thước NxN thành N^2x1

$$\bullet \quad \Gamma_1 = \begin{bmatrix} 1 \\ 56 \\ 50 \\ 224 \\ 98 \\ 0 \\ 35 \\ 158 \\ 78 \end{bmatrix} \quad \Gamma_2 = \begin{bmatrix} 25 \\ 60 \\ 157 \\ 100 \\ 79 \\ 115 \\ 59 \\ 99 \\ 200 \end{bmatrix} \quad \Gamma_3 = \begin{bmatrix} 19 \\ 199 \\ 75 \\ 30 \\ 40 \\ 32 \\ 159 \\ 69 \\ 199 \end{bmatrix} \quad \Gamma_4 = \begin{bmatrix} 20 \\ 234 \\ 161 \\ 30 \\ 30 \\ 177 \\ 213 \\ 1 \\ 61 \\ 11 \end{bmatrix}$$

Với tập gồm n ảnh mặt  $\Gamma_1, \Gamma_2, \dots, \Gamma_M$  ta tính trung bình của tập hợp đó:

$$\bullet \quad \Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

$$\bullet \quad \Psi = \begin{bmatrix} 18.75 \\ 137.25 \\ 110.75 \\ 96 \\ 98.5 \\ 90 \\ 63.5 \\ 96.75 \\ 122 \end{bmatrix}$$

- Sự khác biệt giữa mỗi ảnh so với trị trung bình của nó:

$$\bullet \quad \Phi_i = \Gamma_i - \Psi$$

$$\bullet \quad \Phi_1 = \begin{bmatrix} -45.25 \\ -81.25 \\ -60.75 \\ 128 \\ -0.5 \\ -90 \\ -28.5 \\ 61.25 \\ -44 \end{bmatrix} \quad \Phi_2 = \begin{bmatrix} 6.25 \\ -77.25 \\ 46.25 \\ 4 \\ -19.5 \\ 25 \\ -4.5 \\ 2.25 \\ 78 \end{bmatrix} \quad \Phi_3 = \begin{bmatrix} 0.25 \\ 61.75 \\ -35.75 \\ -66 \\ -58.5 \\ -58 \\ 95.5 \\ -27.75 \\ 77 \end{bmatrix} \quad \Phi_4 = \begin{bmatrix} 11.25 \\ 96.75 \\ 50.25 \\ -66 \\ 78.5 \\ 123 \\ -62.5 \\ -35.75 \\ -111 \end{bmatrix}$$

**Bước 4:** Tạo ma trận hiệp phương sai

Khi dữ liệu đưa vào là hai chiều thì ma trận hiệp phương sai được tạo ra là ma trận hai chiều. Nếu khi dữ liệu đưa vào là N chiều thì ma trận hiệp phương sai được tạo ra sẽ là N chiều. Tất cả các phần tử không nằm trên đường chéo là các đại lượng dương, và với kì vọng là cả hai biến x và y cùng tăng lên. Tập hợp một số lượng lớn các vector là đối tượng chính của PCA

$$C = \sum_{i=1}^M \Phi_i \cdot \Phi_i^T = A \cdot A^T$$

Với ma trận A là tập hợp các ảnh  $A = [\Phi_1, \Phi_2, \dots, \Phi_M]$ .

Ma trận hiệp phương sai khi tìm được sẽ có kích thước là  $N^2 \times N^2$  vì kích thước này quá lớn nên ta phải tính bằng cách  $A^T \cdot A$  do ma trận này có kích thước là  $M \times M$

$$C = \begin{bmatrix} 4332.4 & -171 & -1386.5 & -2650.4 \\ -171 & 1527.6 & -148.2 & -1225.6 \\ -1386.5 & -148.2 & 3205.3 & -1671.3 \\ -2650.4 & -1225.6 & -1671.3 & 5156.5 \end{bmatrix}$$

**Bước 5:** Tìm các giá trị riêng và vector riêng tương ứng từ ma trận hiệp phương sai

Ma trận hiệp phương sai là ma trận vuông, có thể tìm được các vector riêng và trị riêng mà trị riêng của ma trận này dựa vào các bước được trình bày ở phần 2.5.2. Không gian được tạo nên từ các vector riêng được gọi là không gian con của tập ảnh huấn luyện

- Bốn trị riêng:

$$\lambda_1 = 181 \quad \lambda_2 = 18656 \quad \lambda_3 = 48837 \quad \lambda_4 = 78114$$

- Các vector riêng:

$$v_1 = \begin{bmatrix} -0.458 \\ -0.511 \\ -0.49 \\ -0.501 \end{bmatrix} \quad v_2 = \begin{bmatrix} -0.320 \\ -0.842 \\ -0.410 \\ -0.137 \end{bmatrix} \quad v_3 = \begin{bmatrix} -0.587 \\ 0.098 \\ 0.753 \\ -0.278 \end{bmatrix} \quad v_4 = \begin{bmatrix} -0.56 \\ -0.139 \\ -0.119 \\ 0.807 \end{bmatrix}$$

Khi tính được các trị riêng, vector riêng. Nhưng do ma trận hiệp phương sai là ma trận  $M \times M$ , ta cần tìm ma trận  $N^2 \times 1$ , ta áp dụng công thức sau:

$$u_1 = A \cdot v_1$$

$$u_1 = \begin{bmatrix} 11.77 \\ -2.01 \\ -3.46 \\ 4.73 \\ -0.47 \\ -4.75 \\ -0.13 \\ 2.3 \\ -1.82 \end{bmatrix} \quad u_2 = \begin{bmatrix} 7.57 \\ 52.47 \\ -11.72 \\ -8.22 \\ 29.8 \\ 14.67 \\ -17.68 \\ -5.21 \\ -67.9 \end{bmatrix} \quad u_3 = \begin{bmatrix} 24.23 \\ 59.72 \\ -0.69 \\ -106 \\ -67.49 \\ -22.58 \\ 105.5 \\ -46.69 \\ 122.31 \end{bmatrix} \quad u_4 = \begin{bmatrix} 33.52 \\ 126.9 \\ 72.39 \\ -117.6 \\ 73.33 \\ 153.06 \\ -45.21 \\ -60.16 \\ -84.94 \end{bmatrix}$$

Trị riêng thì không cần tính lại vì với M trị riêng của ma trận kích thước MxM thì ma trận  $N^2 \times 1$ . Các vector riêng là độc lập tuyến tính và chính là các trục của không gian mới

#### Bước 6: Xác định các thành phần và vector đặc trưng

Công việc tiếp theo là nén dữ liệu và giảm số chiều vector. Ta tìm được các vector riêng từ ma trận hiệp phương sai và sắp xếp các trị riêng có giá trị từ lớn đến nhỏ.

Các vector riêng ứng với các trị riêng lớn nhất được chọn làm các thành phần chính vì chúng chứa nhiều thông tin có giá trị, có thể lược bỏ các thành phần ít quan trọng của dữ liệu. Khi làm thế thì dữ liệu sẽ bị mất thông tin nhưng với trị riêng càng nhỏ thì thông tin mất sẽ càng nhỏ

Chỉ giữ lại K chiều vector riêng ứng với những trị riêng lớn nhất trong số M vector riêng và  $K << N^2$ . Đây là mục đích của thuật toán PCA giải quyết

#### Bước 7: Biểu diễn trên không gian mới

Với mỗi  $\Phi_i$  khuôn mặt thì sẽ được biểu diễn ứng với K vector riêng tìm được trong không gian mới  $\Omega_i$ :

$$\Omega_i = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_K \end{bmatrix} \text{ trong đó } w_K = u_K^T \cdot \Phi_i$$

$$\text{Cụ thể hơn là: } \Omega_i = \begin{bmatrix} u_1^T \cdot \Phi_i \\ u_2^T \cdot \Phi_i \\ \dots \\ u_K^T \cdot \Phi_i \end{bmatrix}$$

Không gian  $\Omega_i$  mới tìm này chỉ còn kích thước Kx1, các bước trong việc thiết lập cơ sở dữ liệu mới đã xong với số chiều nhỏ hơn ban đầu, thuận lợi trong việc khi cơ sở dữ liệu có dữ liệu là rất lớn.

### Nhận xét thuật toán PCA:

PCA là phương pháp dùng để tạo nên các đặc trưng cho khuôn mặt, nó là nền tảng để phát triển nên những thuật toán khác. Thực tế có rất nhiều người sử dụng PCA kết hợp với những phương pháp khác như mạng Neural, xác suất thống kê để xây dựng nên những thuật toán có độ chính xác cao.

#### Ưu điểm của thuật toán PCA:

- Dễ cài đặt, nếu sử dụng với bài toán cho nhận dạng các khuôn mặt giống nhau thì áp dụng theo lý thuyết sẽ cho ra được kết quả với độ chính xác khá cao. Nếu áp dụng cho bài toán tìm vị trí khuôn mặt thì cần phải có một thuật toán nữa để áp dụng cho thực tế
- Dễ dàng tìm được các đặc trưng tiêu biểu của đối tượng cần nhận dạng mà không cần phải xác định thành phần và mối quan hệ giữa các thành phần đó
- Thuật toán có thể thực hiện tốt với độ phân giải cao, do PCA sẽ thu gọn ảnh thành một ảnh có kích thước nhỏ hơn
- PCA có thể kết hợp với các phương pháp khác như mạng Neural, support vector machine,... để mang lại hiệu quả nhận dạng cao hơn
- Thuật toán PCA đơn giản và dễ áp dụng hơn các thuật toán khác.

#### Nhược điểm của PCA:

- Các mẫu khuôn mặt hoàn toàn phụ thuộc vào tập huấn luyện, có nghĩa là các khuôn mặt trong ảnh kiểm tra (đầu vào) phải gần giống với ảnh huấn luyện về kích thước, tư thế, góc chụp, độ sáng. Nên đôi khi áp dụng trong các môi trường và background khác nhau sẽ cho ra kết quả nhận dạng sai.
- PCA phân loại theo chiều phân bố lớn nhất của tập vector. Tuy nhiên, chiều phân bố lớn nhất không phải lúc nào cũng mang lại hiệu quả tốt nhất cho bài toán nhận dạng. Đây là nhược điểm cơ bản của PCA
- PCA rất nhạy cảm với nhiễu

### 2.5.3 Nhận dạng khuôn mặt

Đưa một khuôn mặt bất kỳ  $\Gamma$  vào kiểm tra. Các bước thực hiện cũng giống như việc tạo cơ sở dữ liệu ở trên

**Bước 1:** Đầu tiên là tính chênh lệch trung bình

$$\Phi = \Gamma - \Psi$$

**Bước 2:** Tìm không gian mới

$$\Omega = \begin{bmatrix} u_1^T \cdot \Phi \\ u_2^T \cdot \Phi \\ \vdots \\ u_K^T \cdot \Phi \end{bmatrix}$$

**Bước 3:** Tính khoảng cách Euclid, tìm khoảng cách càng gần với ảnh thứ i trong M ảnh được dùng làm cơ sở dữ liệu.

$$D_r = \min_M \|\Omega - \Omega_i\|$$

Thiết lập một ngưỡng  $T_r$  để so sánh với khoảng cách vừa tìm được  $D_r$ , nếu  $D_r < T_r$  thì có thể đây là ảnh của khuôn mặt thứ i trong cơ sở dữ liệu.

### 3. THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

Trong phần thiết kế và thực hiện phần cứng, nhóm em sẽ viết đặc tả thiết kế cho phần cứng. Nhóm em sẽ nêu ra các yêu cầu thiết kế, hướng giải quyết và giới hạn của phần cứng.

Cũng trong phần này, nhóm em sẽ trình bày sơ đồ khái tổng quát và sơ đồ khái chi tiết của phần cứng và chức năng của từng khái của sơ đồ khái tổng quát.

Ngoài ra, nhóm em sẽ liệt kê ra các linh kiện, phần cứng cần sử dụng cho đê tài và thông số kỹ thuật của chúng.

#### 3.1 Đặc tả phần cứng

##### Yêu cầu thiết kế:

- Bộ từ vựng: “Không”, “Một”, “Hai”, “Ba”, “Bốn”, “Năm”, “Sáu”, “Bảy”, “Tám”, “Chín”
- Đáp ứng thời gian thực
- Thời gian xử lý nhanh
- Hoạt động ổn định
- Có độ bảo mật cao
- Có thể hoạt động trong thời gian dài
- Có tính di động
- Điện năng tiêu thụ thấp

##### Hướng giải quyết:

Thiết kế bộ từ vựng: nhóm em đã thu thập dữ liệu giọng nói chính chủ và 19 người khác. Riêng dữ liệu giọng nói chính chủ, em đã lặp lại 60 lần mỗi từ để có được 60 mẫu, 10 mẫu đối 10 người thu thập được ở ngoài đời thực và 1 mẫu của 10 người thu được từ internet.

Để có thể mang đi được nhiều nơi để lắp đặt và có tính di động, nhóm em đã chọn máy tính nhúng Raspberry Pi cho phần nhận dạng khuôn mặt. Ngoài ra, một phần vì dùng vi điều khiển để nhận dạng khuôn mặt vẫn được nhưng nhóm em vẫn chưa đủ khả năng để nghiên cứu và làm thành công. Riêng đối với phần nhận dạng giọng nói, nhóm em đã có nhiều nguồn tham khảo nên vẫn chọn vi điều khiển cụ thể là STM32F4 Discovery để thực hiện,

vì vi điều khiển này có giá thành không quá cao, xử lý tốt và phù hợp với nhận dạng giọng nói vì có sẵn mô đun micro.

Đáp ứng thời gian thực và xử lý nhanh: phần này để thực hiện được tốt, nhóm em chọn Raspberry Pi 4 thay vì Raspberry Pi 3. Vì mặc dù dòng Pi 4 cao hơn Pi 3 một chút nhưng dòng Pi 4 nhóm em đã có sẵn do trước đó đã làm qua một project với nó nên có thể tận dụng được. Ngoài ra, dòng Pi 4 có khả năng xử lý nhanh và tốt hơn Pi 3. Nếu sử dụng Pi 3 cho nhận dạng khuôn mặt sẽ có delay trong vấn đề thời gian

Hoạt động ổn định: Vì vấn đề chọn 2 phần cứng khác nhau cho 2 ứng dụng khác nhau nên việc hoạt động sẽ ổn định hơn, mặc dù nhược điểm lớn nhất sẽ là không tối ưu hóa phần cứng. Đây cũng là một điểm trừ của sản phẩm nhóm em

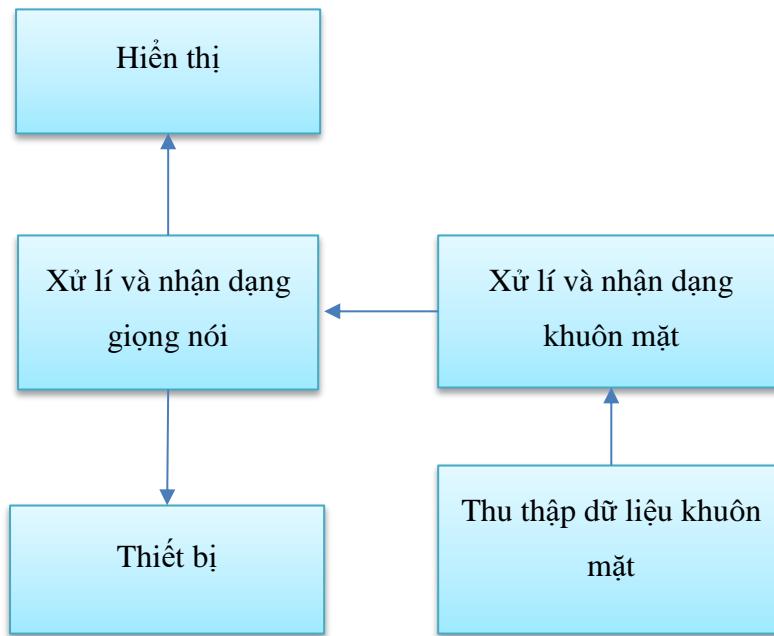
Có thể hoạt động thời gian dài và có độ bền cao: Vì sử dụng riêng 2 phần cứng khác nhau nên hệ thống có thể duy trì hoạt động với thời gian dài

#### **Giới hạn của phần cứng:**

- Tính thẩm mỹ kém
- Chưa tối ưu hóa phần cứng
- Không thực hiện được mạch PCB

## 3.2 Sơ đồ khối

### 3.2.1 Sơ đồ khối tổng quát



### Hệ thống bao gồm các khối:

Xử lý và nhận dạng giọng nói: đây là khối sử dụng board STM32F407VG Discovery để thu thập dữ liệu giọng nói chính chủ, xử lý và tiến hành nhận dạng mật khẩu và đặc trưng của giọng nói. Ở khối này, giải thuật MFCC sẽ được dùng để trích xuất đặc trưng giọng nói và lưu lại trên vi điều khiển. Sau đó, trên vi điều khiển sẽ được lập trình giải thuật Vector Quantization để huấn luyện và so sánh các mẫu giọng nói trong tập huấn luyện với giọng nói đầu vào thu được từ micro.

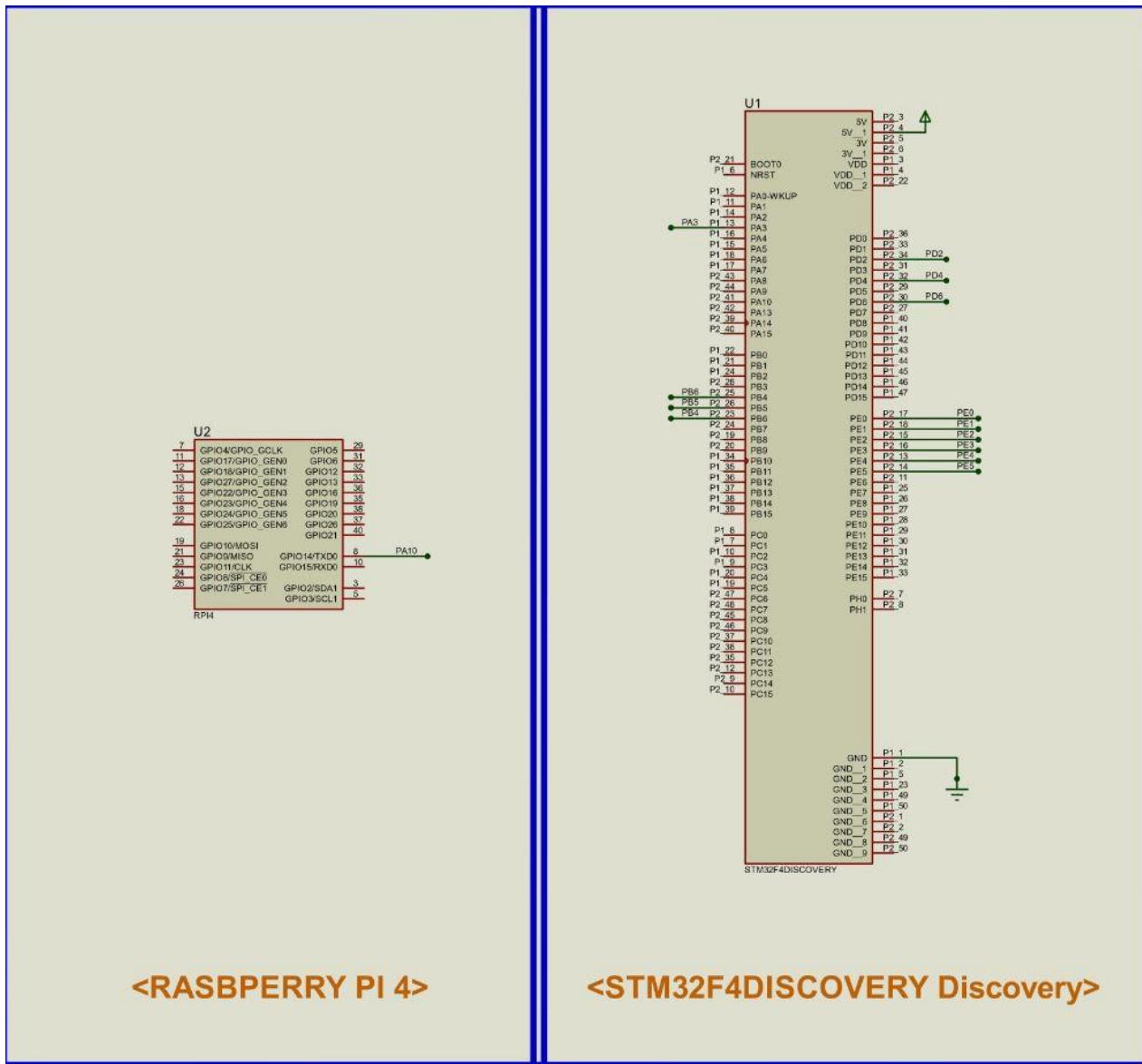
Xử lý và nhận dạng khuôn mặt: đây là khối sử dụng Raspberry Pi 4 để thực hiện xử lý và nhận dạng khuôn mặt. Ở khối này, giải thuật PCA sẽ được xây dựng và thực hiện việc trích xuất vector riêng để tính trọng số của các ảnh trong tập huấn luyện và tiến hành so sánh với trọng số của ảnh đầu vào và tiến hành nhận dạng. Nếu nhận dạng được khuôn mặt chính chủ, khối này sẽ gửi tín hiệu FACEOK thông qua UART sang STM32 còn nếu không có chính chủ khối sẽ gửi FERROR sang STM32.

Thu thập dữ liệu khuôn mặt: đây là khối sử dụng camera Pi V2 để thu thập dữ liệu khuôn mặt phục vụ cho việc trích xuất đặc trưng và huấn luyện cũng như chụp ảnh lại người cần nhận dạng để tiến hành nhận dạng khuôn mặt chính chủ

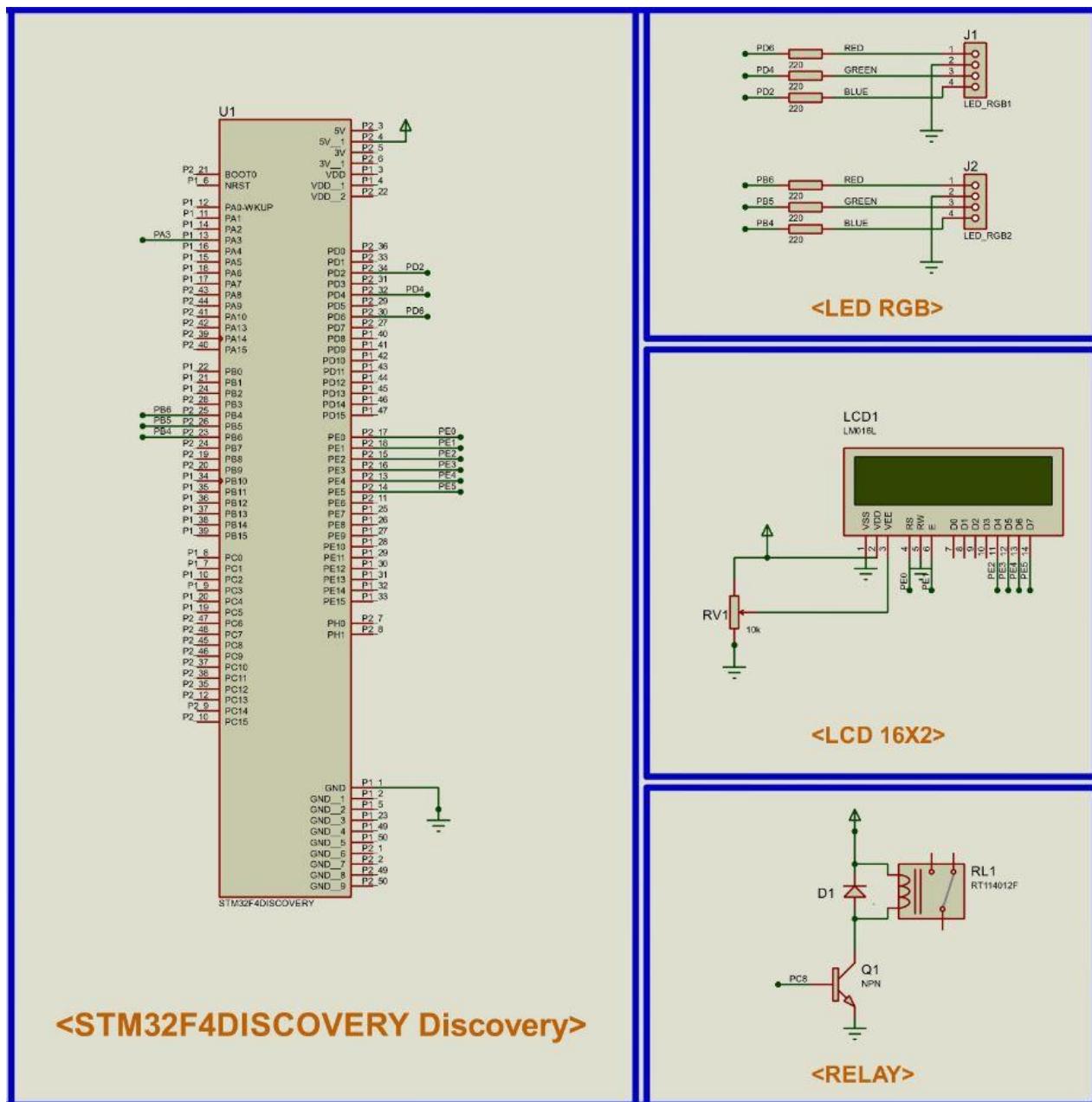
Hiển thị: khối này dùng màn hình LCD 16x2 để thông báo các thông tin nhận dạng đạt hay không. Ngoài ra, ở khối này sẽ có thêm 2 LED RGB đại diện cho nhận dạng khuôn mặt và giọng nói.

Thiết bị: khối này dùng một relay 12V và một khóa chốt cửa điện để mô phỏng lại việc mở cửa

### 3.2.2 Sơ đồ chi tiết



Hình 3. 1 Sơ đồ chi tiết giữa vi điều khiển và máy tính nhúng



Hình 3. 2 Sơ đồ chi tiết các khối còn lại

### 3.3 Sơ lược về phần cứng

#### 3.3.1 Vi điều khiển STM32F407VG



Hình 3. 3 Vi điều khiển STM32F4 Discovery

Kit STM32F407 Discovery hiện là loại kit được sử dụng ở rất nhiều trường đại học hiện nay trong giảng dạy vi điều khiển ARM, nếu so sánh về ngoại vi và sức mạnh của STM32 so với các dòng ARM của các hãng khác thì ở cùng 1 tầm giá, ARM của ST vượt trội về cấu hình và ngoại vi hơn rất nhiều.

##### a. Thông số kỹ thuật:

- Vi điều khiển STM32F407VGT6 - FLASH 1024KB, RAM 192 KB
- Bảng phát triển tích hợp trình gỡ lỗi trình giả lập ST-Link / V2 (nhưng chỉ cung cấp giao diện SWD bên ngoài)
- Đèn báo và nút nhấn reset
- Cung cấp điện:
  - o Lấy 5V qua USB
  - o Có thể cung cấp bên ngoài 5V và 3V
  - o USB OTG FS mini-AB
  - o Cảm biến:
    - o LIS302DL (Cảm biến chuyển động ST MEMS, gia tốc kế đầu ra kỹ thuật số 3 trục)
    - o MP45DT02 (Cảm biến âm thanh ST MEMS, micrô kỹ thuật số đa hướng)
    - o Trình điều khiển âm thanh
    - o CS43L22 (bộ giải mã âm thanh với trình điều khiển loa lớp D tích hợp)
    - o Các quy tắc dẫn đến tất cả các cổng IO, giúp dễ dàng thực hiện các thử nghiệm liên quan.

## b. Thư viện DSP của CMSIS:

Arm Cortex Microcontroller Software Interface Standard (CMSIS) là một nhà phân phối độc lập các lớp thư viện cho mọi vi xử lý dòng Cortex

Thư viện CMSIS-DSP bao gồm:

- Các hàm bộ lọc FIR/IIR
- Hàm xử lí ma trận
- Hàm tính biên độ và pha cho các phép toán phức
- FFT, IFFT
- Các hàm tính toán như cosin, sin,...
- Các hàm tính toán min, max, trung bình, năng lượng,...
- Hỗ trợ chuyển kiểu nhanh

Một số function của thư viện DSP được dùng trong luận văn này:

- o Nhân ma trận :

```

o    /**
o     * @brief Floating-point matrix multiplication
o     * @param[in] pSrcA points to the first input matrix structure
o     * @param[in] pSrcB points to the second input matrix structure
o     * @param[out] pDst points to output matrix structure
o     * @return The function returns either
o         * <code>ARM_MATH_SIZE_MISMATCH</code> or <code>ARM_MATH_SUCCESS</
o         code> based on the outcome of size checking.
o     */
o     arm_status arm_mat_mult_f32(
o         const arm_matrix_instance_f32 * pSrcA,
o         const arm_matrix_instance_f32 * pSrcB,
o         arm_matrix_instance_f32 * pDst);

```

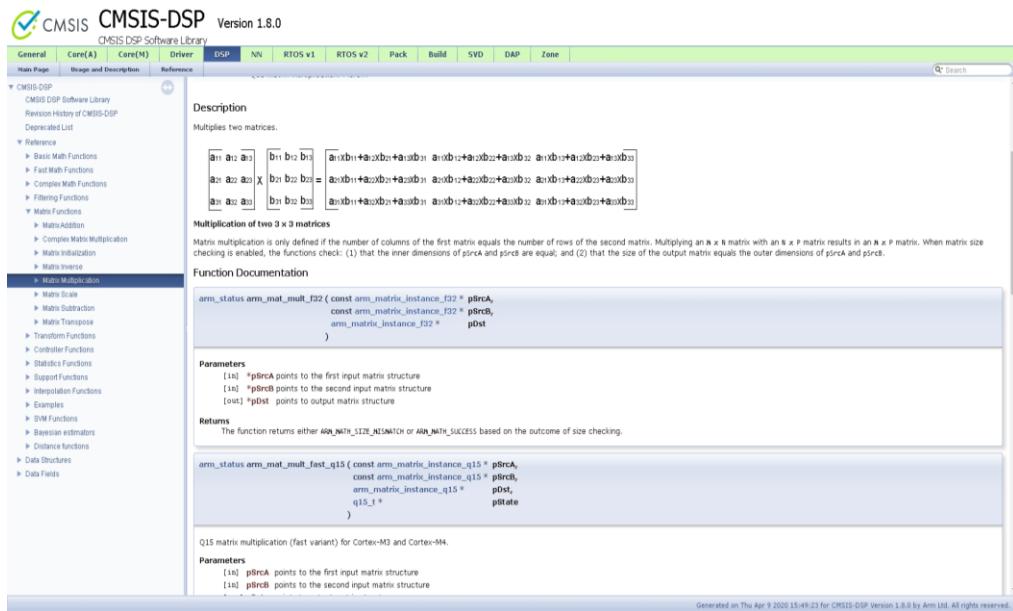
Tính toán FFT/IFFT :

```
○ void arm_rfft_fast_f32(  
○     arm_rfft_fast_instance_f32 * S,  
○     float32_t * p, float32_t * pOut,  
○     uint8_t ifftFlag);
```

Nhân vector:

```
/**  
* @brief Floating-point vector multiplication.  
* @param[in]    *pSrcA points to the first input vector  
* @param[in]    *pSrcB points to the second input vector  
* @param[out]   *pDst points to the output vector  
* @param[in]    blockSize number of samples in each vector  
* @return none.  
*/  
  
void arm_mult_f32(  
    float32_t * pSrcA,  
    float32_t * pSrcB,  
    float32_t * pDst,  
    uint32_t blockSize)
```

Chi tiết về thuật toán hoặc cách sử dụng của các function có thể xem tại website của thư viện:



Hình 3. 4 Website của thư viện CMSIS-DSP

### c. Chức năng DMA:

DMA (Direct Memory Access) là một cơ chế truyền dữ liệu tốc độ cao từ ngoại vi tới bộ nhớ cũng như từ bộ nhớ tới bộ nhớ. Dữ liệu có thể được di chuyển một cách nhanh chóng mà không cần tới tác vụ từ CPU, tiết kiệm tài nguyên CPU cho các hoạt động khác.

DMA của STM32:

- STM32 có 2 bộ DMA với 12 kênh (7 kênh DMA1 và 5 kênh DMA2), mỗi bộ quản lý việc truy cập bộ nhớ từ một hoặc nhiều ngoại vi. DMA cũng có chức năng phân xử độ ưu tiên giữa các DMA request.
- 12 kênh DMA độc lập, có thể thiết lập được. 7 kênh DMA1 và 5 kênh DMA2
- Software trigger được hỗ trợ cho mỗi kênh, và được lập trình bởi phần mềm.
- Độ ưu tiên giữa các kênh DMA có thể lập trình bởi phần mềm (có 4 cấp ưu tiên là very high, high, medium, low) hoặc phần cứng.
- Phụ thuộc vào kích thước giữa nguồn và đích (byte, half word, word). Địa chỉ nguồn/đích phải phù hợp với kích thước dữ liệu.

- Hỗ trợ truyền tải giữa:
  - o Memory to memory
  - o Peripheral to memory
  - o Memory to peripheral
  - o Peripheral to peripheral
  - o Có thể truy cập vào Flash, Sram, APB1, APB2 và AHB như nguồn và đích.
  - o Dữ liệu truyền nhận hỗ trợ tới 65536 bit

### 3.3.2 Micro MP45DT02



Hình 3. 5 Micro MP45DT02

MP45DT02-M là một micro MEMS kỹ thuật số nhỏ gọn, công suất thấp, đa hướng được hãng ST tích hợp trên 1 số board mạch STM32.

- o Thông số:
  - Tiêu thụ điện năng thấp (1.2-3.3V)
  - Thu âm đa hướng
  - Hoạt động tốt trong khoảng -30°C đến +85°C
  - Xuất tín hiệu dưới dạng PDM với tùy chọn mono hay stereo
- o Nguyên lý hoạt động:
  - Thành phần cảm biến âm thanh gồm 2 miếng kim loại dẫn điện. Một miếng cố định và một miếng được gắn cố định để silicon tạo thành một tụ điện biến dung

$$C = \frac{\varepsilon \cdot S}{9 \cdot 10^9 \cdot 4\pi \cdot d}$$

với d là khoảng cách giữa 2 miếng kim loại

- Sóng âm khi lan truyền đến cảm biến thì tấm kim loại không cố định sẽ thay đổi vị trí làm cho giá trị d thay đổi dẫn đến điện dung của tụ điện

biến đổi, cuối cùng làm cho điện áp giữa 2 đầu tụ điện thay đổi. Tiếp đến, IC sẽ xuất ra dữ liệu dưới dạng PDM dựa vào sự thay đổi đó.

### 3.3.3 Máy tính nhúng Raspberry Pi 4

Raspberry Pi là cái máy tính giá 35USD kích cỡ như iPhone và chạy HĐH Linux. Với mục tiêu chính của chương trình là giảng dạy máy tính cho trẻ em. 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



Hình 3. 6 Raspberry Pi 4

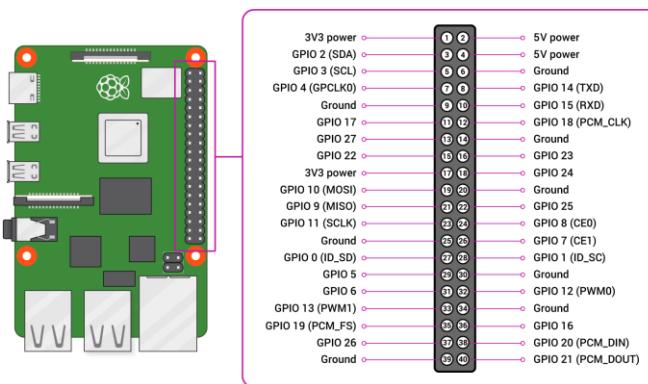
Raspberry Pi sản xuất bởi 3 OEM: Sony, Qsida, Egoman. Và được phân phối chính bởi Element14, RS Components và Egoman.

Nhiệm vụ ban đầu của dự án Raspberry Pi là tạo ra máy tính rẻ tiền có khả năng lập trình cho những sinh viên, nhưng Pi đã được sự quan tâm từ nhiều đối tượng khác nhau. Đặc tính của Raspberry Pi xây dựng xoay quanh bộ xử lý SoC Broadcom BCM2835 (là chip xử lý mobile mạnh mẽ có kích thước nhỏ hay được dùng trong điện thoại di động) bao gồm CPU, GPU, bộ xử lý âm thanh /video, và các tính năng khác ... tất cả được tích hợp bên trong chip có điện năng thấp này .

#### a. Thông số kỹ thuật:

- CPU ARM Cortex-A72 lõi tứ 64-bit 1.5GHz (ARM v8, BCM2837)
- RAM 1GB, 2GB hoặc 4GB (LPDDR4)
- Tích hợp Wireless LAN (băng tần kép 802.11 b/g/n/ac)
- Header GPIO 40 pin
- Card đồ họa OpenGL ES, 3.0
- Cổng CSI cho camera

- Giắc video composite và âm thanh analog 3,5mm
- Khe cắm thẻ micro-SD
- Nguồn USB Type-C
- Công suất tiêu thụ: mức tiêu thụ điện năng tối đa của Pi 4 là khoảng 7,6W khi tải và 3,4W khi không tải.
- Nguồn điện cung cấp: bộ cấp nguồn USB Type-C, ở mức 5.1V / 3A.
- Hệ điều hành: Pi có thể chạy trên các hệ điều hành khác nhau như Raspberry OS (tiền thân là Raspian), Ubuntu,... hoặc các hệ điều hành đa phương tiện khác như Kodi OSMC và LibreElec.
- GPIO:



Hình 3. 7 Các chân ra của Raspberry Pi 4

Trong 40 chân GPIO bao gồm:

- 26 chân GPIO. Khi thiết lập là input, GPIO có thể được sử dụng như chân interrupt, GPIO 14 & 15 được thiết lập sẵn là chân input.
- 1UART, 1 I2C, 2 SPI, 1 PWM (GPIO 4)
- 2 chân nguồn 5V, 2 chân nguồn 3.3V, 8 chân GND
- 2 chân ID EEPROM

Điện áp hoạt động:

- 2 chân 5V
- 2 chân 3V3
- Các chân đất 0V
- Các chân inputs và outputs – 3V3 (high) và 0V (low)

Các chức năng khác của GPIO pins:

- PWM (pulse-width modulation)
  - o Phần mềm có sẵn trên tất cả các chân
  - o Phần cứng khả dụng trên GPIO12, GPIO13, GPIO18, GPIO19
- SPI
  - o SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)
  - o SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)
- I2C
  - o Dữ liệu: (GPIO2); Clock (GPIO3)
  - o Dữ liệu EEPROM: (GPIO0); EEPROM Clock (GPIO1)
- Serial
  - o TX (GPIO14)
  - o RX (GPIO15)
- HDMI:
  - o Raspberry Pi 4 có hai cổng micro HDMI, cho phép kết nối hai màn hình riêng biệt. Cần cáp micro HDMI-to-HDMI hoặc cáp HDMI-to-HDMI cùng với bộ chuyển đổi micro HDMI-to-HDMI để kết nối Raspberry Pi 4 với màn hình.

### b. Ứng dụng:

Sử dụng Raspberry Pi như máy tính để bàn nhưng với giá thành rẻ hơn và kích cỡ nhỏ hơn, dễ tích hợp vào các hệ thống nhận dạng như nhận dạng khuôn mặt, giọng nói, xe cộ,...

**Ưu Điểm:** giá rẻ, nhỏ gọn, siêu tiết kiệm điện, GPU mạnh, phục vụ cho nhiều mục đích, khả năng hoạt động liên tục 24/7

**Nhược điểm:** CPU cấu hình thấp, Lan 100, không có tích hợp WiFi (có thể mua USB WiFi gắn vào), yêu cầu phải có kiến thức cơ bản về Linux, điện tử

#### 3.3.4 Raspberry Pi Camera Module V2



Hình 3. 8 Camera Module V2

Raspberry Pi Camera Module V2 có một cảm biến 8-megapixel của Sony IMX219

Camera Module có thể được sử dụng để quay video độ nét cao, cũng như chụp hình ảnh tĩnh. Nó khá dễ dàng để sử dụng cho người mới bắt đầu, nhưng cũng có rất nhiều giải pháp mở rộng để cung cấp cho người dùng yêu cầu cao. Có rất nhiều demo của người dùng về công dụng của Camera Module như chụp Time-Lapse, Slow-Motion và rất nhiều ứng dụng khác

#### **Thông số kỹ thuật:**

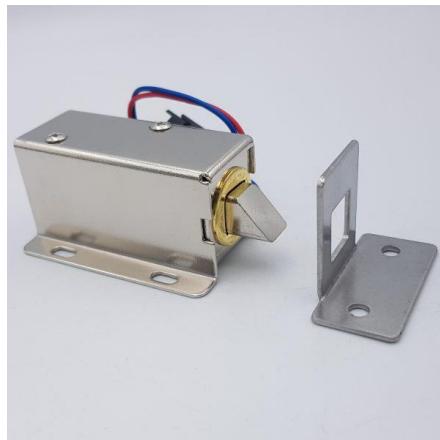
- Ống kính tiêu cự cố định
- Cảm biến độ phân giải 8 megapixel cho khả năng chụp ảnh kích thước 3280 x 2464
- Hỗ trợ video 1080p30, 720p60 và 640x480p90
- Kích thước 25mm x 23mm x 9mm
- Trọng lượng chỉ hơn 3g
- Kết nối với Raspberry Pi thông qua cáp ribbon đi kèm dài 15 cm
- Camera Module được hỗ trợ với phiên bản mới nhất của Raspbian

Kết nối camera pi với board Pi: Cắm 15 chân của cable vào cổng CSI (Camera Serial Interface).



Hình 3. 9 Sau khi nối Camera Module V2 với Raspberry Pi 4

### 3.3.5 Khóa chốt điện DC12V LY-03



Hình 3. 10 Khóa chốt cửa điện

Khóa chốt điện Solenoid Lock LY-03, có chức năng hoạt động như một ổ khóa cửa sử dụng Solenoid để kích đóng mở bằng điện, được sử dụng nhiều trong nhà thông minh hoặc các loại tủ, cửa điện,... Khóa sử dụng điện áp 12/24VDC, là loại thường đóng với chất lượng tốt, độ bền cao.

Thông số kỹ thuật:

- Vật liệu: Thép không gỉ
- Nguồn điện: 12V DC
- Dòng điện làm việc: 0.8A
- Công suất: 9.6W
- Yêu cầu nguồn cấp: 12VDC/1A
- Kích thước: L54xD38xH28

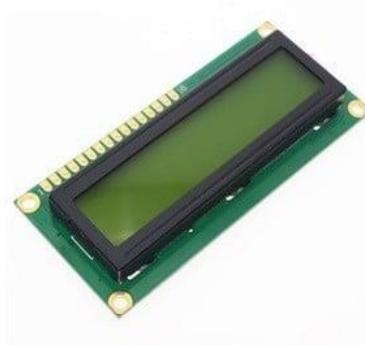
### 3.3.6 Module relay với opto cách ly 12V



Hình 3. 11 Module relay

Relay 1 kênh 12V được thiết kế chắc chắn, khả năng cách điện tốt, có sẵn 4 lỗ gắn ốc 3mm. Trên module đã có sẵn mạch kích relay sử dụng transistor và IC cách ly quang giúp cách ly hoàn toàn mạch điều khiển (vi điều khiển) với rơ le bảo đảm vi điều khiển hoạt động ổn định. Tín hiệu kích hoạt rơ le đóng: LOW, tín hiệu HIGH sẽ làm rơ le mở trở lại.

### 3.3.7 Màn hình LCD Text 1602 Xanh lá



Hình 3. 12 LCD Text 1602

Màn hình text LCD1602 xanh lá sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ sử dụng thích hợp cho những người mới học và làm dự án. Trong luận văn này em sử dụng giao tiếp 4bit để điều khiển màn hình

#### a. Thông số kỹ thuật:

- Điện áp hoạt động là 5 V.
- Kích thước: 80 x 36 x 12.5 mm
- Chữ đen, nền xanh lá
- Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
- Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.
- Có đèn led nền, có thể dùng biến trờ hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu
- Có bộ ký tự được xây dựng hỗ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet để biết thêm chi tiết.

**b. Các chân của LCD:**

Chân	Ký hiệu	Mô tả	Giá trị
1	VSS	GND	0V
2	VCC		5V
3	V0	Độ tương phản	
4	RS	Lựa chọn thanh ghi	RS=0 (mức thấp) chọn thanh ghi lệnh RS=1 (mức cao) chọn thanh ghi dữ liệu
5	R/W	Chọn thanh ghi đọc/viết dữ liệu	R/W=0 thanh ghi viết R/W=1 thanh ghi đọc
6	E	Enable	
7	DB0	Chân truyền dữ liệu	8 bit: DB0DB7
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	A	Cực dương led nền	0V đến 5V
16	K	Cực âm led nền	0V

Bảng 3.1 Các chân của LCD

## 4. THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

Trong phần này, nhóm em sẽ trình bày các lưu đồ giải thuật của nhận dạng khuôn mặt và nhận dạng giọng nói bao gồm giải thuật trích xuất đặc trưng MFCC, giải thuật lấy mẫu, huấn luyện và nhận dạng giọng nói VQ và giải thuật giảm chiều không gian PCA để nhận dạng khuôn mặt.

Ngoài ra, chúng em sẽ giải thích chi tiết các bước tiến hành của các giải thuật trên và trình bày một vài lưu đồ giải thuật của chương trình nhận dạng giọng nói trên vi điều khiển.

Cuối cùng, nhóm em sẽ trình bày thêm các phần mềm cần sử dụng cho luận văn cũng như cách tạo một project cho STM32F4 Discovery và cách cấu hình cho Raspberry Pi 4.

Bên cạnh đó, nhóm em sẽ trình bày các hàm chương trình mà nhóm em sẽ sử dụng trong phần mềm nhận dạng bằng MATLAB.

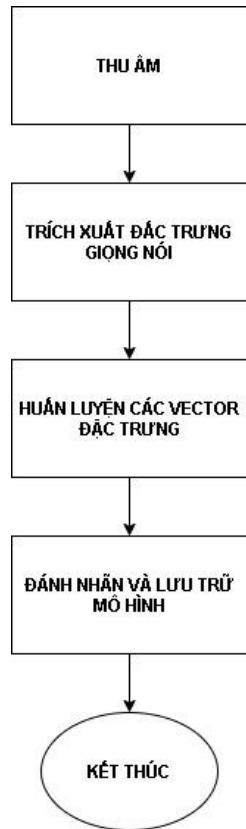
Yêu cầu đặt ra cho phần mềm mô phỏng thuật toán trên MATLAB:

- Hệ thống nhận dạng người trên phần mềm có độ chính xác trên 80% trở lên
- Thuật toán sử dụng cho phần mềm đơn giản
- Phần mềm thiết kế với giao diện dễ nhìn, trực quan và dễ sử dụng, thao tác

✚ Lưu đồ tổng quát nhận diện giọng nói:

Lưu đồ tổng quát bao gồm 2 phần: huấn luyện và nhận dạng

○ Huấn luyện mô hình:



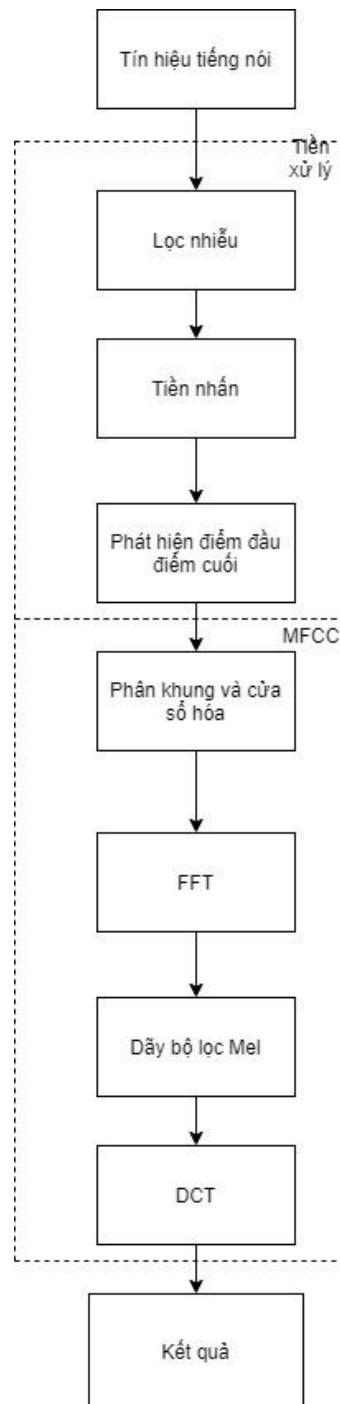
Hình 4. 1 Giải thuật huấn luyện của nhận dạng giọng nói

- Nhận dạng :



Hình 4. 2 Giải thuật nhận dạng giọng nói

#### 4.1 Quy trình tách đặc trưng và huấn luyện tạo codebook để nhận dạng giọng nói



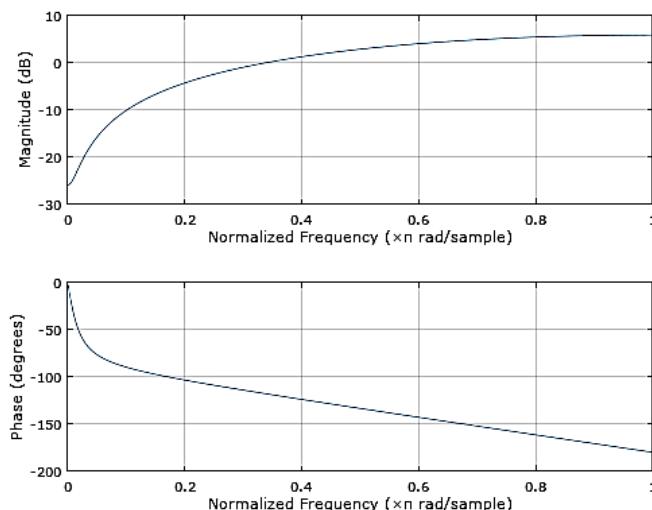
Hình 4. 3 Thuật toán trích xuất đặc trưng tiếng nói MFCC

#### 4.1.1 Trích đặc trưng

Nhóm em chọn tần số lấy mẫu là  $fs = 16\text{kHz}$ , số bit trên 1 mẫu là 8 bit, số kênh thu âm là 1 (Mono).

#### Bước 1: Pre-Emphasis

Tín hiệu tiếng nói thường thu ở môi trường thực tế, do vậy sẽ có thêm nhiều tín hiệu nhiễu (tần số thấp) được thêm vào tín hiệu của chúng ta. Và khi nghe, chúng ta có thể dễ dàng nhận ra nhiễu do nhiễu có cường độ lớn bằng một phần đáng kể của tiếng nói khi thu âm. Bằng việc thực hiện tăng cường cường độ ở vùng tần số cao lên nhằm tăng năng lượng của vùng có tần số cao (vùng tần số có tiếng nói), có thể hiểu là làm tiếng nói lớn lên hơn để ảnh hưởng của các âm thanh môi trường và nhiễu trở nên không đáng kể, làm cho thông tin được rõ ràng hơn.



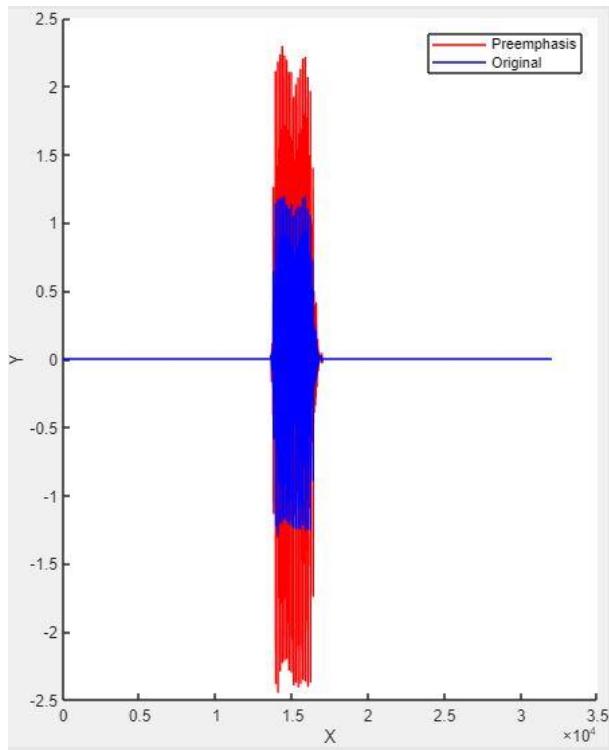
Hình 4.4 Sơ đồ biên độ và pha

Tín hiệu  $s(n)$  được cho qua một bộ lọc FIR thông cao:

$$H(z)=1-az^{-1}$$

$$\tilde{x}(n) = x(n) - ax(n-1) \text{ với } 0.9 \leq a \leq 1$$

Thường chọn  $a = 0.95$



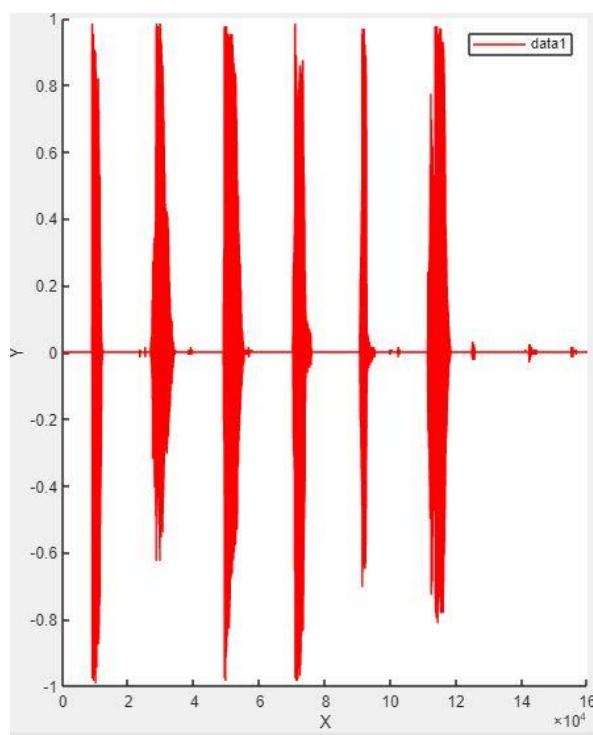
Hình 4. 5 Tín hiệu âm thanh của từ “Một” trước và sau khi Pre-emphasis

## Bước 2: Dùng Voice Endpoint Detection để cắt khoảng lặng có biên độ nhỏ

Nhằm mục đích giảm khối lượng tính toán và tăng độ chính xác, chỉ khi nào có tín hiệu tiếng nói thu được thì mới xử lý. Công việc này được gọi là tách từ (endpoint detection)

Phương pháp phổ biến nhất dùng để tách từ là dùng năng lượng (Short-term energy) kết hợp với tỉ lệ điểm qua điểm không (Zero Crossing Rate).

Chúng em sử dụng phương pháp Short-term energy kết hợp với Zero Crossing Rate vì tính hiệu quả và độ chính xác của nó.

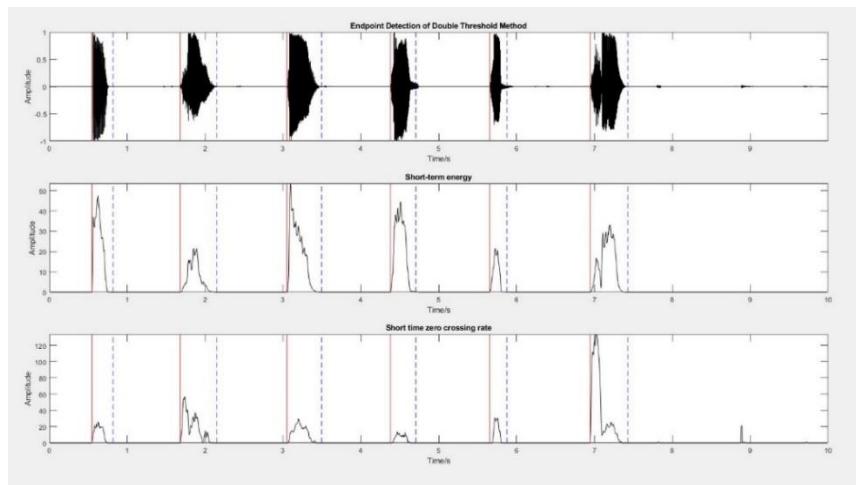


Hình 4. 6 Tín hiệu âm thành của 6 âm từ “một” đến “sáu”

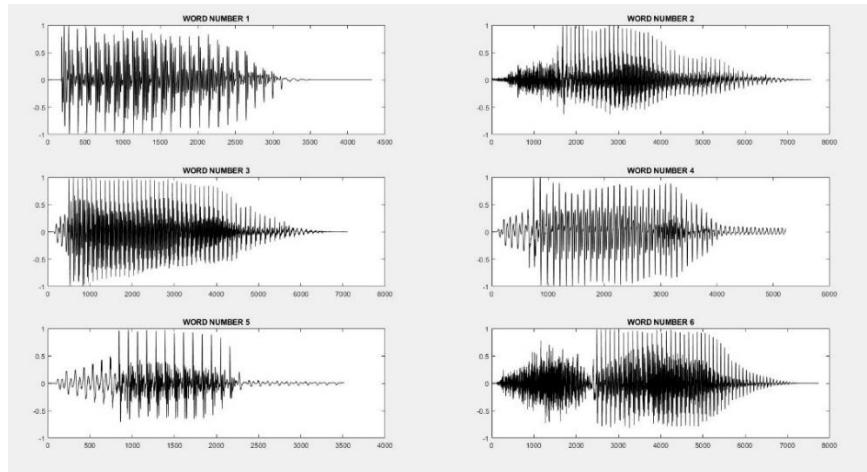
Để tách được các từ trong một câu nói, chúng em sử dụng phương pháp ngưỡng kép với việc đặt các giá trị ngưỡng của phương pháp ngưỡng năng lượng (STE-Short-term Energy) và trung bình tỉ lệ các điểm vượt qua điểm không (Short term Average Zero Crossing Rate) dựa trên thực nghiệm:

- Ngưỡng năng lượng (STE):
  - $T_1=0.5$
  - $T_2=1.5$

- Trung bình tỉ lệ các điểm vượt qua điểm không(STA ZCR): T3=0.7



Hình 4. 7 Tín hiệu âm thanh sau khi tách bằng phương pháp ZCR và STE



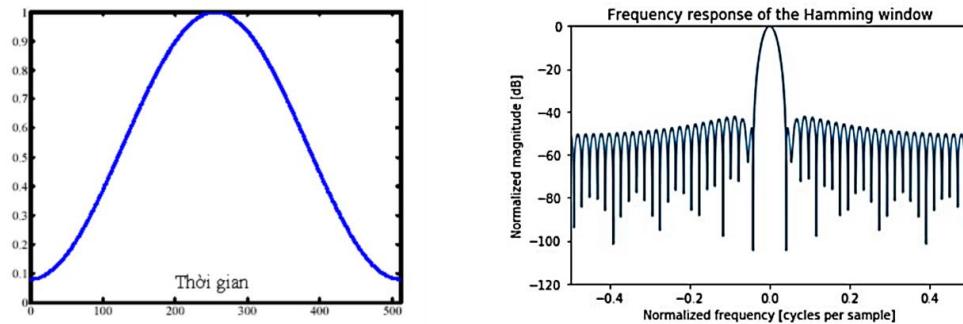
Hình 4. 8 Các từ sau khi đã được tách ra

### Bước 3: Frame Blocking

Tín hiệu được phân thành những khung, mỗi khung N mẫu, độ chồng lấp overlap có M mẫu: M thường xấp xỉ  $1/3N$ , với  $N = 256$  để dễ cho việc tính toán FFT.

#### Bước 4: Cửa sổ hóa

- Sử dụng cửa sổ Hamming dạng như sau:



Hình 4.9 Cửa sổ Hamming

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \leq n \leq N-1$$

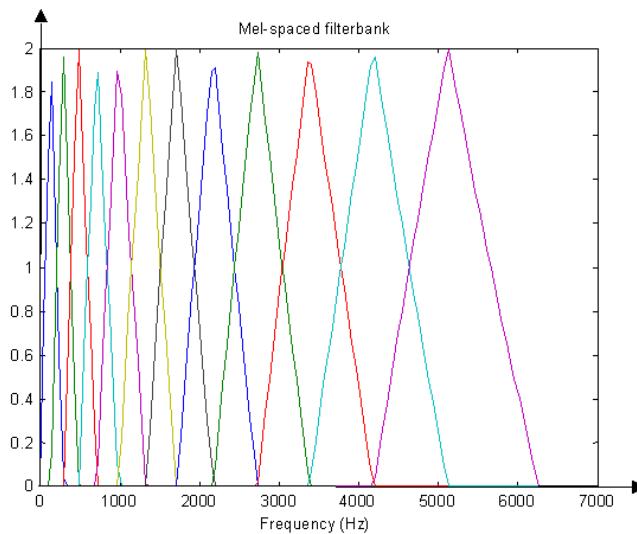
#### Bước 5: FFT

Mục đích của quá trình này là chuyên đổi tín hiệu từ miền thời gian sang tín hiệu miền tần số. FFT là giải thuật nhanh của phép biến đổi Fourier rời rạc (DFT).

#### Bước 6: Mel Frequency Wrapping

- Tạo các bộ lọc Mel theo công thức:

$$\begin{aligned}
 H_m(k) &= 0 && \text{nếu } k < f(m-1) \\
 &= \frac{k-f(m-1)}{f(m)-f(m-1)} && \text{nếu } f(m-1) \leq k < f(m) \\
 &= 1 && \text{nếu } k = f(m) \\
 &= \frac{f(m+1)-k}{f(m+1)-f(m)} && \text{nếu } f(m) < k < f(m+1) \\
 &= 0 && \text{nếu } k > f(m+1)
 \end{aligned}$$



Hình 4. 10 Bộ lọc Mel-filter Bank

Để xác định mel – spectrum, cho biên độ phổ tần số sau bước FFT ở trên qua bộ lọc mel, với công thức tính như sau:

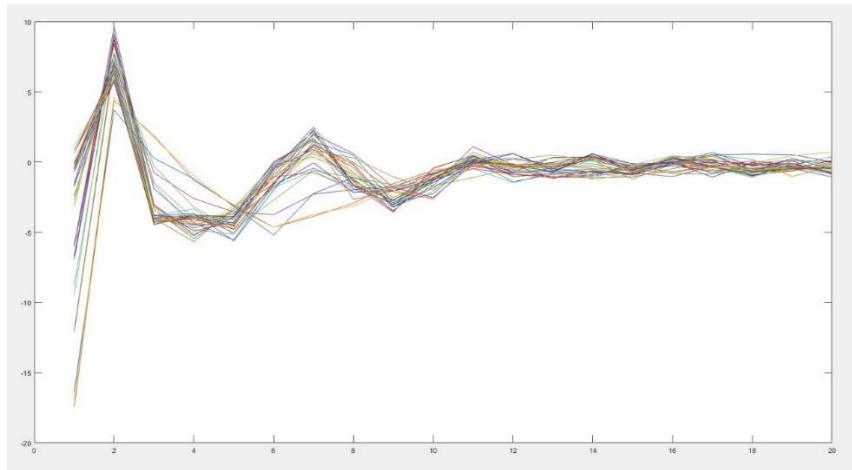
$$\tilde{S}(l) = \sum_{k=0}^{N/2} X(k)^* M_i(k) \quad l=0,1,\dots,L-1$$

Kết quả thu được đến bước này là một ma trận (số điểm hệ số mel – spectrum). Các hệ số này được đưa vào bước cuối cùng để tìm Acoustic vector, là đặc trưng của giọng nói.

### Bước 7: Cepstrum

Ở bước này dùng DCT (discrete cosin transform – biến đổi cosin rời rạc) thay vì IFFT, vì các lý do sau:

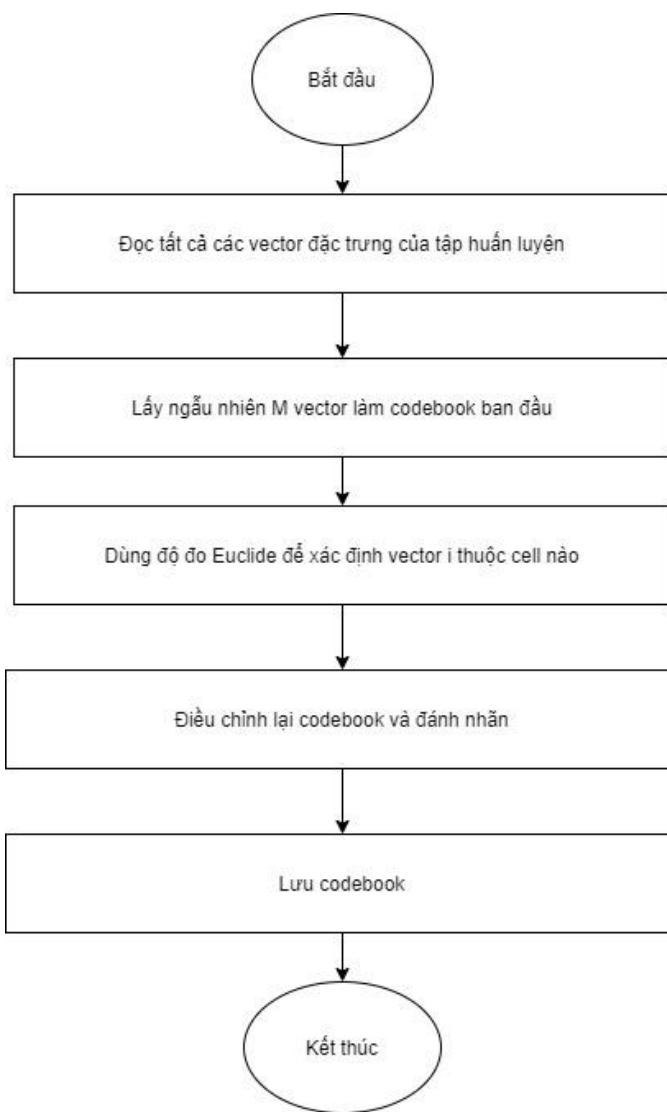
- Tín hiệu là tín hiệu thực
- IFFT áp dụng cho tín hiệu là số phức, trong khi DCT là số thực
- Sau khi qua biến đổi DCT ta đã thu được các đặc trưng của tiếng nói, là một chuỗi các acoustic vector của các frame liên tiếp nhau. Các acoustic vector này được dùng để huấn luyện và nhận dạng tiếng nói



Hình 4. 11 Acoustic vectors của từ “Một”

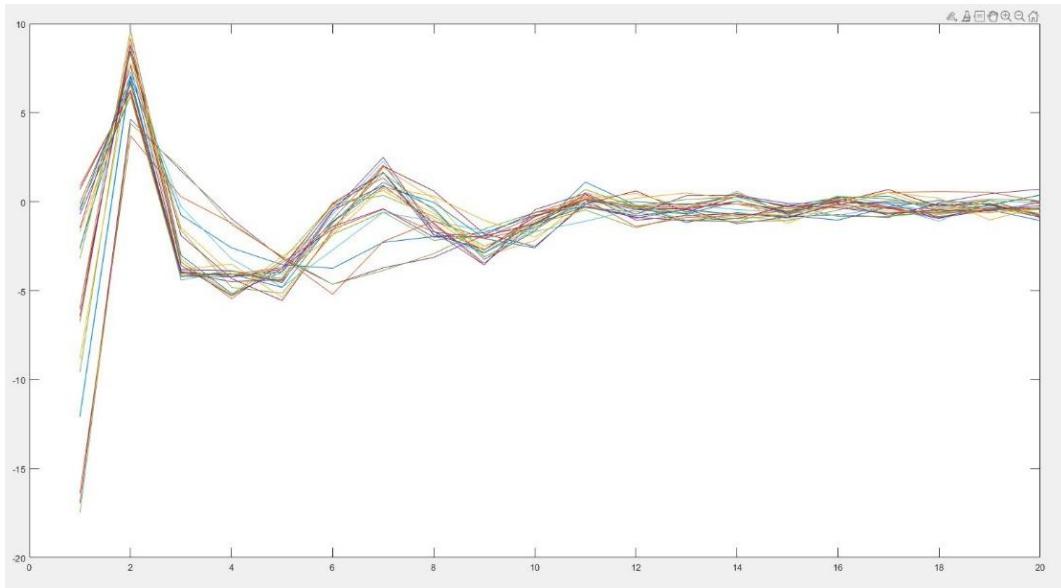
#### 4.1.2 Huấn luyện và nhận dạng giọng nói

Sau khi trích đặc trưng thì ta đã có được 33 vector đặc trưng mfcc của các frame trong từ “một”. Tiếp theo là bước dùng Vector Quantization tạo codebook cho các từ trong tập huấn luyện, các từ này phải trải qua bước tính Acousitc vector trước khi tới bước tính codebook:



Hình 4. 12 Thuật toán huấn luyện và nhận dạng tiếng nói bằng VQ

- Sau khi tạo codebook với số centroids là 32, kết quả có dạng:

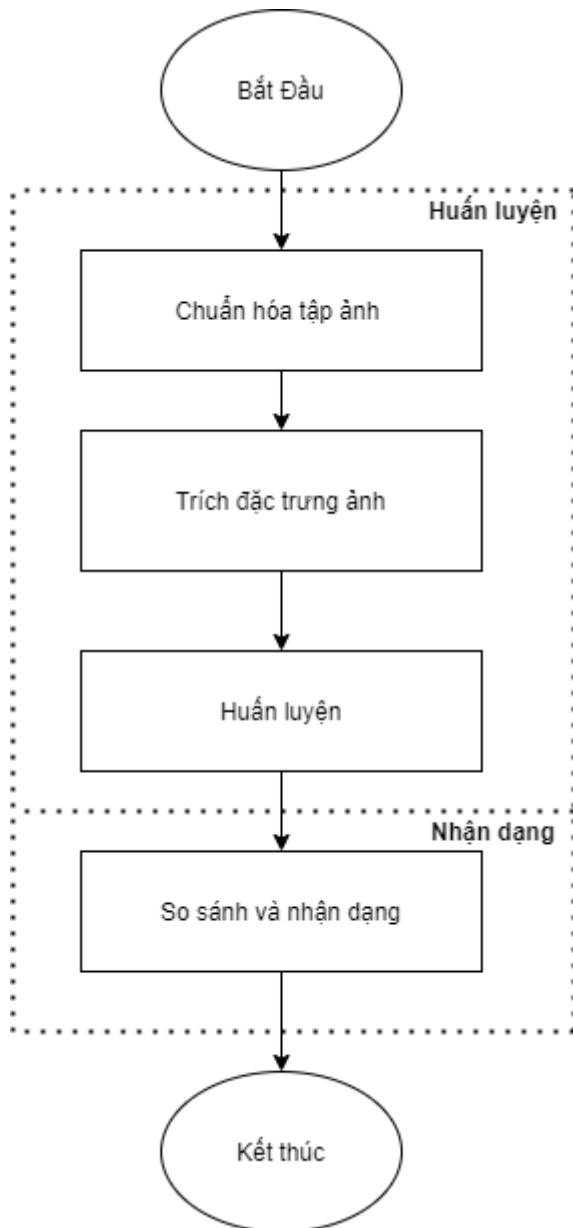


Hình 4. 13 Codebook sau khi lượng tử hóa vector

Sau khi tính được codebook cho từng file tiếng nói trong tập huấn luyện và tính được cepstrum mfcc của tiếng nói đầu vào. Lúc này, ta sẽ tính khoảng cách Euclidean giữa cepstrum của tiếng nói đầu vào và codebook của từng file huấn luyện.

Nếu khoảng cách Euclidean nào nhỏ dưới ngưỡng cho trước (ngưỡng này tính được dựa trên thực nghiệm) thì sẽ xác định đúng là người nói chính chủ. Ngược lại thì đây là người lạ

## 4.2 Quy trình huấn luyện và nhận dạng khuôn mặt dùng PCA



Hình 4. 14 Lưu đồ giải thuật tổng quát của nhận dạng khuôn mặt

Phần nhận dạng chia làm hai phần chính. Đó là phần trích đặc trưng, huấn luyện và phần so sánh và nhận dạng khuôn mặt. Để huấn luyện và nhận dạng ảnh, ta cần phải chuẩn hóa lại ảnh đầu vào và ảnh huấn luyện.

#### 4.2.1 Thuật toán PCA để huấn luyện dữ liệu nhận dạng



Hình 4. 15 Thuật toán huấn luyện của nhận dạng khuôn mặt bằng PCA

Bắt đầu tiến trình huấn luyện, chúng ta thu thập lần lượt 30 ảnh và 27 ảnh có khuôn mặt của chính chủ chụp bằng camera của laptop và camera của board Pi

Dùng thuật toán Viola Jones và các đặc trưng Haar like để tách khuôn mặt trong ảnh chụp.

Tiếp theo, ta sẽ chuẩn hóa các giá trị pixel của mỗi ảnh sau bước trên trong tập huấn luyện về giá trị từ 0 đến 255, có nghĩa là chuyển đổi ảnh từ ảnh màu sang ảnh xám

Sau đó, ta sẽ chuẩn hóa kích thước của tất cả các ảnh trong tập huấn luyện về cùng 1 kích thước là 128x128 pixel đối với ảnh chụp bằng camera Pi và 100x90 pixel đối với ảnh chụp bằng camera laptop. Tiếp tục, ta làm phẳng các ảnh bằng cách biến đổi kích thước ảnh về dạng một chiều có dạng 1xM với M là 128x128 hoặc 100x90.

Tiếp tục quá trình, ta tạo data X có số chiều MxN với N là số ảnh cần huấn luyện là 27 đối với board Pi và 30 đối với PC, M là số pixel hay kích thước mỗi ảnh là 128x128 hay 100x90.

Để chuẩn hóa từng ảnh trong tập huấn luyện, ta tính trung bình của data X và từ đó lấy data X ban đầu trừ trung bình X tính được ở trên để ra được ảnh chuẩn hóa (ma trận phương sai)

$$\mathbf{X} = \mathbf{X} - \mathbf{X}_{\text{tb}}$$

Sau đó, ta tính ma trận hiệp phương sai Q từ ma trận phương sai tính được ở trên, theo công thức sau:

$$\mathbf{Q} = \frac{1}{N} \sum_{i=0}^n \mathbf{X}_i * \mathbf{X}_i'$$

với N là số ảnh ta phải huấn luyện

Với n là số chiều ban đầu và giảm về K chiều

Tính ra các giá trị riêng và các vector riêng từ ma trận Q theo công thức ở mục 2.5.2

Úng với mỗi trị riêng ta có nhiều vector riêng, nhưng ứng với mỗi vector riêng ta chỉ có một trị riêng

Sau đó, ta sắp xếp các vector riêng theo thứ tự giảm dần của trị riêng

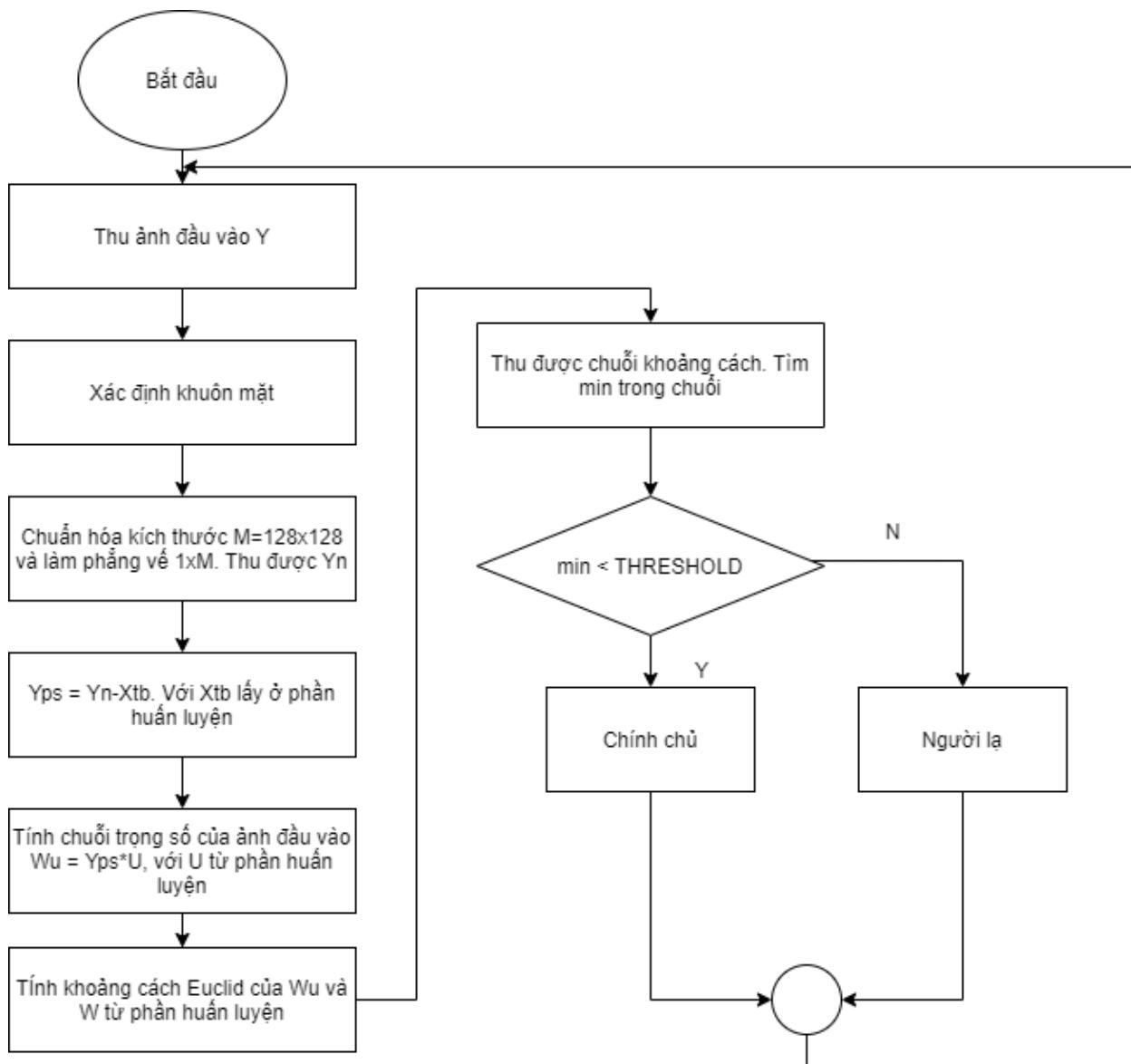
Ta tính ra được một ma trận mới bao gồm nhiều vector riêng đã được sắp xếp theo thứ tự giảm dần của trị riêng. Sau đó ta chọn ra K vector riêng đầu tiên ứng với K trị riêng lớn nhất. Với K << M. Ta thu được một ma trận có số trực thông tin được giảm so với số trực không gian cũ của ảnh. Tạm gọi là ma trận R.

Sau đó, ta nhân ma trận R với ma trận data X chuyển vị từ ma trận X có kích thước MxN ở bước trên. Ta thu được ma trận có số chiều không gian nhỏ hơn với trực thông tin có độ biến thiên dữ liệu cao nhất và các dữ liệu lúc này đã được chiếu lên trực thông tin mới. Tạm

gọi là ma trận U. Ta sẽ dùng ma trận U chiếu lên ảnh được chuẩn hóa (ma trận phương sai) của data X tập huấn luyện và ảnh đầu vào.

Ta thu được chuỗi các trọng số của ảnh được huấn luyện W từ việc nhân dữ liệu X đại diện cho tập huấn luyện đã chuẩn hóa với ma trận U đã được chuẩn hóa và hiệu chỉnh lại kích thước 128x128

#### 4.2.2 Nhận dạng khuôn mặt



Hình 4. 16 Thuật toán nhận dạng khuôn mặt bằng PCA

Đầu tiên, ta dùng camera Pi hoặc camera PC/laptop để ghi hình lại ảnh khuôn mặt chính chủ hoặc bất kì một người nào khác chính chủ. Ta thu được ma trận  $Y$

Tiếp theo, ta dùng thuật toán Viola Jones để xác định vùng có khuôn mặt trên frame ảnh chụp từ camera

Ta sẽ chuẩn hóa ảnh khuôn mặt đầu vào về một kích thước nhất định là 128x128 pixel hoặc 100x90 pixel, sau khi tách vùng có khuôn mặt bằng Viola Jones. Tiếp tục, ta làm phẳng ảnh đầu vào bằng cách biến đổi kích thước ảnh về dạng một chiều có dạng 1xM với M là 128x128 hoặc 100x90. Ta thu được  $Y_n$

Ta có được trung bình của ảnh trong tập huấn luyện. Ta sẽ tính ảnh chuẩn hóa (ma trận phương sai) của ảnh đầu vào cần nhận dạng bằng cách trừ ảnh 1 chiều 1xM cho ảnh trung bình từ tập huấn luyện. Ta thu được ma trận  $Y_{ps}$

$$Y_{ps} = Y_n - X_{tb}$$

Sau đó, để tính trọng số của ảnh đầu vào cần huấn luyện  $W_u$ , ta chiều ảnh đầu vào đã chuẩn hóa lên ma trận U từ giai đoạn huấn luyện bằng cách nhân ảnh đầu vào đã chuẩn hóa đó với ma trận U.

$$W_u = Y_{ps} * U$$

Sau đó, ta tính khoảng cách Euclid (dùng cho board Pi) giữa hai trọng số là trọng số ảnh đầu vào  $W_u$  và trọng số của ảnh trong tập huấn luyện W.

Khoảng cách ta tính được nếu dưới 1 mức ngưỡng nhất định (mức ngưỡng này nhóm em tính thông qua thực nghiệm) thì sẽ xác định đó là chính chủ, còn nếu vượt quá ngưỡng thì không phải là chính chủ. Trong lần thực hiện mô phỏng trên MATLAB và trên board Pi có 2 ngưỡng thực nghiệm khác nhau.

### 4.3 Thủ nghiệm giải thuật trên MATLAB

Nhóm em viết ra các hàm (function) để thực hiện giải thuật nhận dạng giọng nói, mặt khẩu và khuôn mặt

Hàm tách tín hiệu tiếng nói:

```
word = endpointdetection(x, fs); %Voice Endpoint Detection
```

- Với x là tín hiệu tiếng nói đầu vào
- fs là tần số lấy mẫu của tiếng nói, fs=16kHz
- Hàm tiền nhấn của tín hiệu

```
data_plot = pre_emph(data_rec); %Pre-emphasis
```

- Với a=1, b=0.95 thì ta có được bộ lọc ở khâu pre-emphasis
  - data\_rec chính là tín hiệu được thu âm
- Hàm tạo bộ lọc Mel-filter bank:

```
s = melfb_v2(p, n, fs); %mel filter bank
```

- p là số bộ lọc tam giác, p = 20
  - n là số mẫu của 1 frame, n = 256
  - m là số mẫu của overlap frame, m = 100
- Hàm tính các hệ số đặc trưng Acoustic Vector mfcc của tiếng nói:

```
r = mymfcc(s, fs); %trich dac trung mfcc
```

- s là tín hiệu âm thanh đầu vào
  - fs là tần số lấy mẫu, fs = 16kHz
- Hàm tạo codebook:

```
r = vqlbg(d, k); %tao codebook
```

- d là các vector dữ liệu sau khi trích đặc trưng mfcc
  - k là số centroids, là số nhán codebook, ở đây k = 32
- Hàm tính khoảng cách Euclid:

```
d = disteu(x, y); % tính khoảng cách Euclid
```

- x, y là các điểm cần tính khoảng cách

## 4.4 Lập trình thuật toán trên vi điều khiển và máy tính nhúng

### 4.4.1 Khái quát về vi điều khiển

Vi điều khiển là một hệ thống nhúng khép kín với các thiết bị ngoại vi, bộ xử lý và bộ nhớ. Muốn sử dụng vi điều khiển trước tiên ta phải lập trình cho nó. Trước khi lập trình cho vi điều khiển, người viết phải hiểu được cấu tạo phần cứng và các yêu cầu mà mạch điện cần thực hiện.

Chương trình mà lập trình viên viết là một tập hợp các lệnh được tổ chức theo một trình tự hợp lý để giải quyết các yêu cầu lập trình viên. Tập hợp tất cả các lệnh được gọi là tập lệnh. Họ vi điều khiển ARM đều có chung một tập lệnh, các vi điều khiển được cải tiến về sau thường ít thay đổi tập lệnh mà thay vào đó chú trọng phát triển phần cứng.

Với sự hỗ trợ của máy tính, người viết chương trình có thể viết chương trình cho vi điều khiển bằng các ngôn ngữ lập trình cấp cao. Sau khi quá trình viết chương trình được hoàn tất, các trình biên dịch sẽ chuyển các câu lệnh cấp cao thành mã máy một cách tự động. Sau đó, các mã máy này được nạp vào bộ nhớ ROM của vi điều khiển, vi điều khiển sẽ tìm đọc các lệnh từ ROM để thực thi chương trình

Vi điều khiển STM32 sử dụng kiến trúc ARM Cortex M đang dần trở nên phổ biến và được sử dụng rộng rãi trong nhiều ứng dụng vì tính năng mạnh mẽ, chi phí thấp và hiệu suất cao của nó. Do đó, nhóm em chọn dòng vi điều khiển này cho việc lập trình vì nó đáp ứng phần lớn yêu cầu của đề tài.

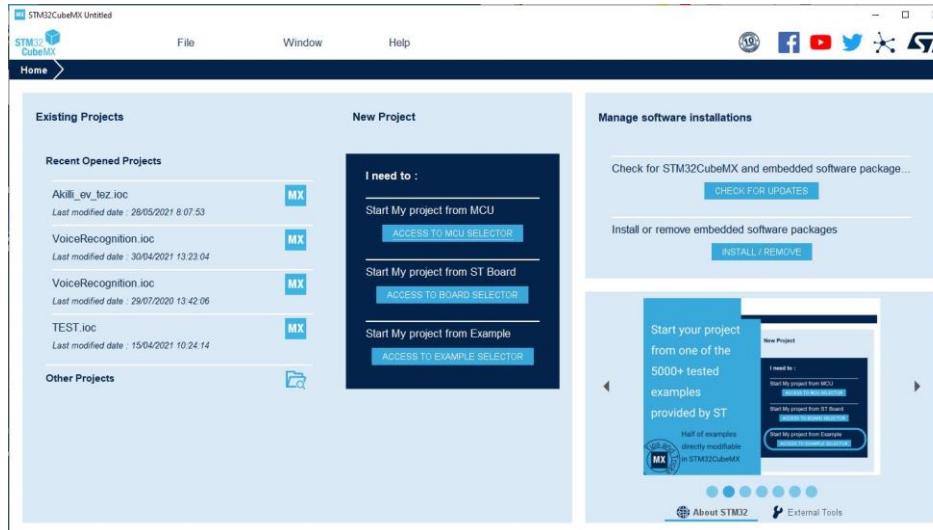
### 4.4.2 Phần mềm dùng cho vi điều khiển

STMicroelectronics đã giới thiệu một công cụ có tên là STM32Cube MX, công cụ này dùng để sinh code cơ bản cho các thiết bị ngoại vi và board STM32. Vì vậy, chúng ta không cần lo lắng về việc code hóa cho các trình điều khiển và các thiết bị ngoại vi cơ bản. Đồng thời, code được tạo này có thể được sử dụng trong Keil uVision và được chỉnh sửa theo yêu cầu. Cuối cùng, code này được nạp vào board STM32 bằng mạch nạp ST-Link từ STMicroelectronics

Các bước thiết lập nền một project cho STM32:

**Bước 1:** Cài đặt tất cả các trình điều khiển thiết bị cho ST-Link V2, công cụ phần mềm STM32Cube MX và Keil uVision và cài đặt thêm các gói thư viện cần thiết

## Bước 2: Mở STM32Cube MX



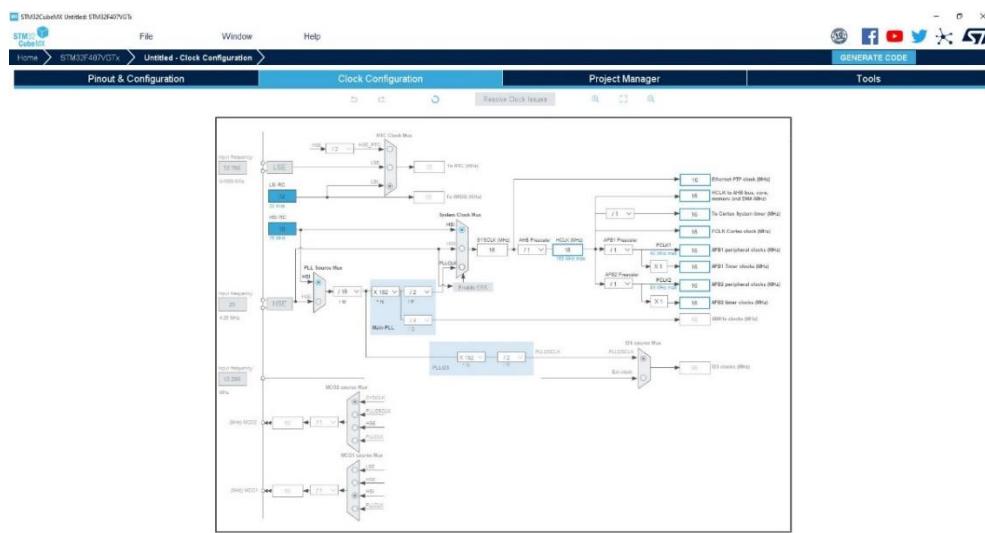
Hình 4. 17 Phần mềm STM32CubeMX

### Bước 3: Click vào Acess to mcu selector

**Bước 4:** Chọn dòng vi điều khiển cần lập trình, ở đây là STM32F407VG

**Bước 5:** Cấu hình ngoại vi bằng cách nhấp vào các chân vi điều khiển ở trong tab **Pinout & Configuration**

**Bước 6:** Cấu hình xung clock trong tab **Clock configuration**



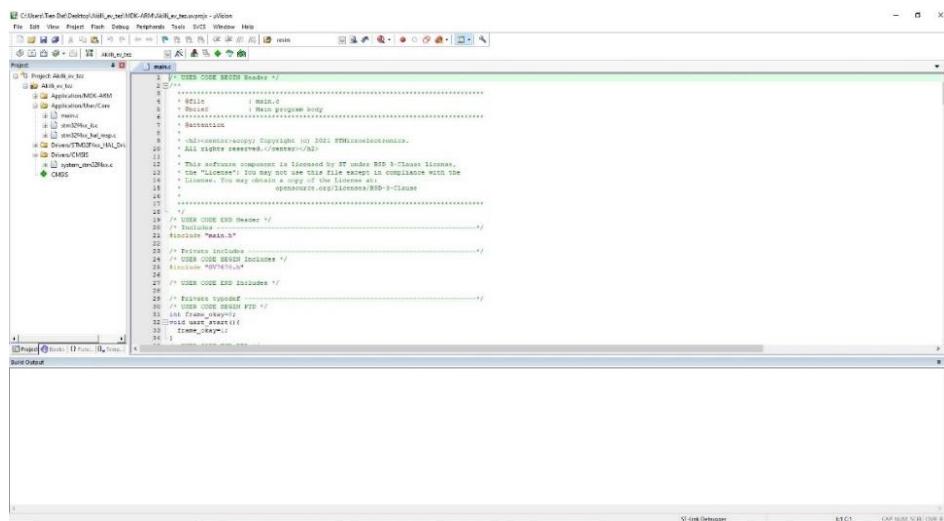
Hình 4. 18 Cấu hình xung clock cho vi điều khiển

**Bước 7:** Vào tab **Project Manager** đặt tên Project, chọn IDE MDK-ARM V5 để có thể sinh code lên Keil C. Chọn nút **Generate** để sinh code

Lập trình trên Keil uVision và Debug:

- Mặc dù nhóm em đã hoàn thành giải thuật trên Matlab nhưng vẫn cần phải chuyển đổi sang ngôn ngữ C vì giữa chúng có nhiều điểm khác biệt như sau:
- Ngôn ngữ C cần phải khai báo biến và kiểu dữ liệu của biến còn Matlab thì không cần
- Ngôn ngữ C không hỗ trợ các hàm tính FFT, DCT, max, min, sum,...
- Ngôn ngữ C không hỗ trợ các thao tác để tối ưu hóa tính toán trên ma trận, trong khi Matlab lại rất mạnh về phần xử lý ma trận
- Ngôn ngữ C giúp tối ưu hóa phần cứng tốt hơn Matlab vì ngôn ngữ C có sử dụng đến con trỏ, khai báo stack, cấp phát bộ nhớ động,...

Keil C v5 là một trong những IDE tốt nhất hiện nay và được các lập trình viên sử dụng trong việc lập trình và phát triển cho các dòng vi điều khiển kiến trúc ARM cho các ứng dụng nhúng. Sau đây là các bước để lập trình và debug trên phần mềm này:



Hình 4. 19 Phần mềm Keil uVision 5

**Bước 1:** Mở Keil C v5 từ project đã được sinh code từ STMCube MX ở phần trên

**Bước 2:** thêm các thư viện cần thiết vào code

```
17 [*****]
18 */
19 /* USER CODE END Header */
20 /* Includes ----- */
21 #include "main.h"
22
23 /* Private includes ----- */
24 /* USER CODE BEGIN Includes */
25
26 /* USER CODE END Includes */
27
```

#### Hình 4. 20 Các thư viện cần thiết cho chương trình

**Bước 3:** Lập trình dựa trên các giải thuật đã trình bày ở trên. Ngoài ra cần khai báo biến và kiểu dữ liệu cũng như kích thước mảng thích hợp

**Bước 4:** Click vào biểu tượng Build hoặc nhấn F7

**Bước 5:** Nếu có báo lỗi thì nhấp đúp vào dòng báo lỗi, trình biên dịch sẽ đưa đến dòng code bị báo lỗi, sau đó ta tìm cách sửa lỗi, build lại lần nữa đến khi không còn lỗi

**Bước 6:** Nhấn tổ hợp phím Ctrl+F7 hoặc bấm vào biểu tượng để debug

**Bước 7:** Tô đen các biến cần theo dõi, nhấp vào chuột phải và add các biến này vào các ô Watch dog 1,2,3. Ngoài ra, có thể theo dõi các thanh ghi ngoại vi hay 32 thanh ghi hệ thống trong mục **Registers window**. Thêm điểm dừng (break point) vào các dòng lệnh cần sửa lỗi.

**Bước 8:** Bấm Run. Nếu có các điểm break point thì chương trình sẽ dừng ngay tại các điểm đó hoặc nếu có lỗi thì chương trình sẽ dừng tại các handler nhằm thông báo cho người dùng

#### 4.4.3 Thực hiện giải thuật trên vi điều khiển

Do tài nguyên của vi điều khiển là rất có hạn, không thể thực hiện các phép tính về ma trận và số thập phân mạnh mẽ như máy tính, vì vậy các tham số tính toán phức tạp sẽ được thực hiện trên MATLAB sau đó ghi ra các header file(.c .h) để đổ xuống bộ nhớ ROM của vi điều khiển, sẵn sàng cho việc thực hiện tính toán và nhận dạng ở vi điều khiển mà không cần phải tính toán lại. Các tham số đều ở dạng số thập phân được khai báo ở định dạng float. Điểm đặc biệt của thư viện arm cmsis chỉ hỗ trợ tính toán mảng 1 chiều, vì vậy cần phải thực hiện xử lý tính toán theo từng frame tín hiệu, đồng thời biểu diễn mảng 2 chiều thành 1 chiều để có thể sử dụng được thư viện.

Các function cần viết lại ở vi điều khiển:

- block\_frames: chia thành từng frame, mỗi frame 256 mẫu, nhân cửa sổ hamming, FFT.
- mfcc : nhân tính hiệu với bộ lọc mel.
- voice\_compare: tính toán khoảng cách với codebook đã được huấn luyện sẵn.

Các tham số cần phải đổ xuống:

- Tham số HammingWindow [256] :

```
const float HamWindow[256] =
{
    0.080000f, 0.080140f, 0.080558f, 0.081256f, 0.082232f,
    0.115287f, 0.119769f, 0.124506f, 0.129496f, 0.134734f,
    0.215734f, 0.223871f, 0.232200f, 0.240716f, 0.249413f,
    0.365931f, 0.376474f, 0.387117f, 0.397852f, 0.408674f,
    0.542834f, 0.554166f, 0.565489f, 0.576797f, 0.588083f,
    0.719302f, 0.729684f, 0.739951f, 0.750097f, 0.760115f,
    0.868261f, 0.876100f, 0.883736f, 0.891163f, 0.898377f,
    0.966858f, 0.970952f, 0.974785f, 0.978353f, 0.981656f,
    0.999965f, 0.999686f, 0.999128f, 0.998290f, 0.997175f,
    0.962504f, 0.957894f, 0.953030f, 0.947916f, 0.942554f,
    0.860222f, 0.851988f, 0.843565f, 0.834958f, 0.826172f,
    0.708810f, 0.698217f, 0.687527f, 0.676747f, 0.665885f,
    0.531500f, 0.520171f, 0.508854f, 0.497557f, 0.486285f,
    0.355493f, 0.345168f, 0.334960f, 0.324878f, 0.314925f,
    0.207794f, 0.200056f, 0.192524f, 0.185203f, 0.178097f,
    0.111063f, 0.107099f, 0.103398f, 0.099962f, 0.096793f,
};
```

Hình 4. 21 Tham số Hamming Window

- Tham số Mel-filter bank (hệ số = 20, số mẫu = 256 , overlap = 100):

```
const float MelFb[MELFB_NUM*129] =
{
    0.00000f, 1.00000f, 0.50000f, 0.00000f,
    0.00000f, 0.00000f, 0.50000f, 1.00000f,
    0.00000f, 0.00000f, 0.00000f, 0.00000f,
    0.00000f, 0.00000f, 0.00000f, 0.00000f
};
```

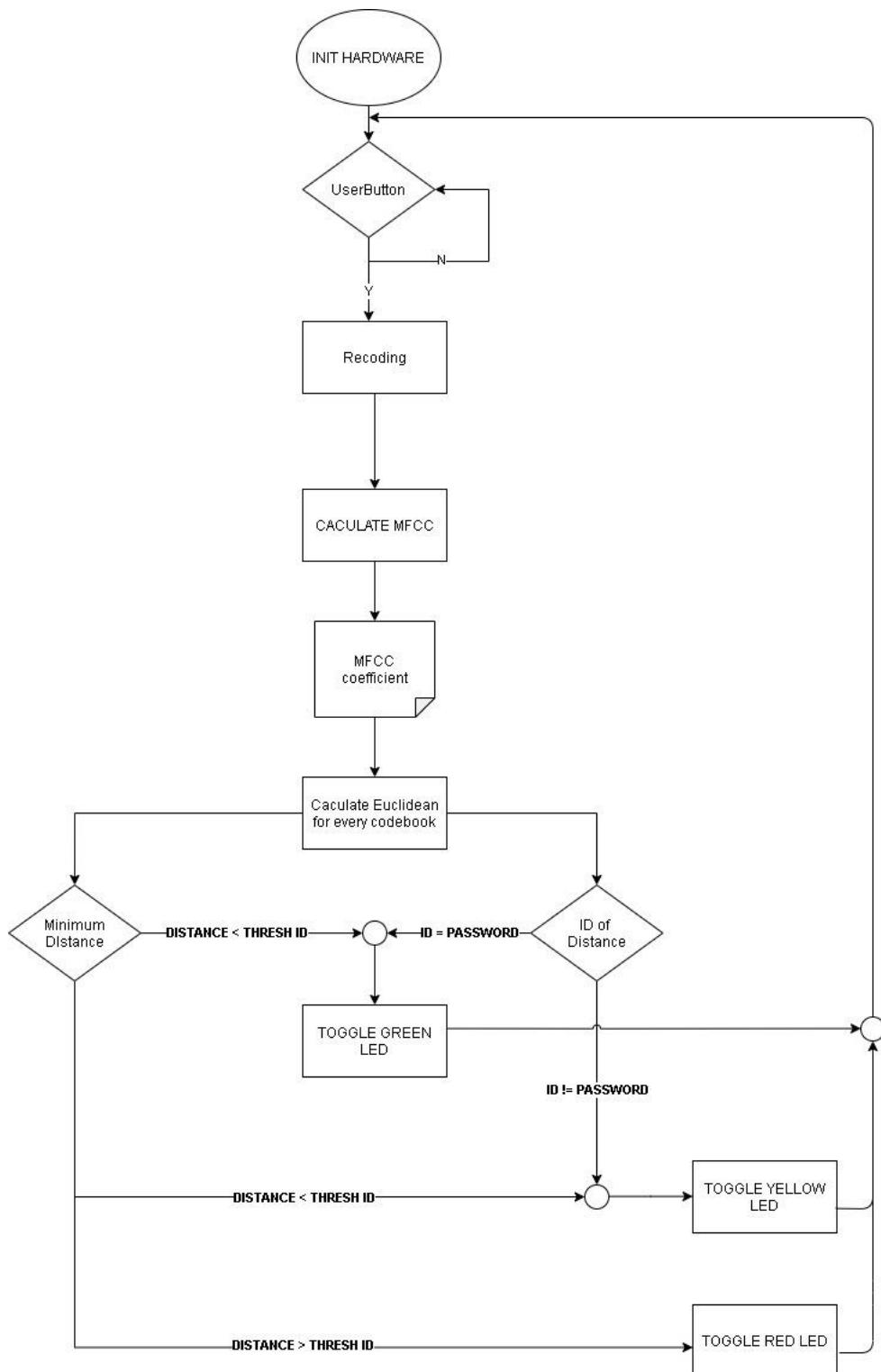
Hình 4. 22 Tham số Mel-FilterBank

- Tham số của Codebook của từ KHONG:

```
const float Word[WORD_NUM][MELFB_NUM*CENTROID] =
{ /* KHONG */
    21.382245f, 19.909606f, 10.749663f, -2.180295f,
    -9.487318f, -5.142074f, -9.885378f, -7.012711f,
    -3.713344f, -0.463904f, -2.437135f, 0.470079f,
    -2.716303f, -1.834091f, -0.515708f, 0.169449f,
    -2.125627f, -1.589971f, 0.769926f, 0.558820f,
    0.459025f, 0.582535f, -0.029237f, 0.197410f,
    -0.939586f, -0.157638f, -0.459787f, -0.373534f,
    -2.158005f, -0.443026f, 0.739773f, 0.884973f,
    -1.544531f, -0.388398f, 1.289659f, 0.868966f,
    0.786064f, -0.018504f, -0.814954f, -1.160644f,
    0.892501f, -1.117151f, 0.850568f, 0.462255f,
    0.622028f, -0.178022f, 0.123894f, -1.665662f,
    0.120029f, -0.285574f, -0.862163f, -1.853380f,
    -1.682685f, -0.390657f, 0.154045f, -0.331414f,
    -0.322369f, -0.141502f, -1.265557f, 0.222306f,
    0.489974f, 0.497535f, 0.068354f, 0.891791f,
    -0.277581f, 0.642911f, -0.834263f, 0.491118f,
    -1.002922f, 0.984803f, -0.235266f, 0.632110f,
    -0.184648f, 0.772822f, -0.144002f, 0.431868f,
    -1.152093f, 0.485742f, -0.347226f, 0.671364f,
},
```

Hình 4. 23 Tham số codebook của từ KHÔNG

- Lưu đồ giải thuật chi tiết:
- + Tổng quát về quá trình thu âm, tính toán, cách thức hoạt động của hệ thống:



Hình 4. 24 Lưu đồ giải thuật tổng quát nhận dạng giọng nói

Bước 1 : Khởi tạo các phần cứng sử dụng trong đề tài: audio, lcd, gpio,...

Bước 2: Chờ nút Userbutton được nhấn, nếu được nhấn hệ thống sẽ bắt đầu ghi âm

Bước 3: Thực hiện các bước tiền xử lí giọng nói sẵn sàng cho việc xử lí tiếp theo, tính toán đặc trưng MFCC.

Bước 4: Các bước tính toán để ra được khoảng cách đến từng code book đã được huấn luyện và đổ vào kit giống như đã trình bày phần trên.

Bước 5: Sau khi có được khoảng cách đến từng code book bắt đầu thực hiện so sánh với ngưỡng đã được khởi tạo từ trước, đồng thời kiểm tra xem khoảng cách nhỏ nhất thuộc codebook nào. Nếu giá trị khoảng cách nhỏ hơn ngưỡng đồng thời đúng ID của password, hệ thống sẽ bật đèn xanh, báo hiệu đúng người và giọng nói. Nếu chỉ có khoảng cách nhỏ hơn ngưỡng và password sai thì bật đèn vàng báo hiệu sai mật khẩu. Khoảng cách lớn hơn ngưỡng thì bật đèn đỏ báo hiệu người lạ.

#### 4.4.4 Khái quát về máy tính nhúng và ngôn ngữ dùng cho nó

##### a. Khái quát về máy tính nhúng:

Máy tính nhúng là một thiết bị, một hệ thống được thiết kế để phục vụ cho một yêu cầu, một bài toán, ứng dụng, một chức năng nhất định nào đó và được ứng dụng nhiều trong lĩnh vực công nghiệp, tự động hóa điều khiển, quan trắc và truyền tin...

Trong luận văn lần này, nhóm em sử dụng máy tính nhúng Raspberry Pi 4 cho ứng dụng nhận dạng khuôn mặt vì độ bền cao, kích thước nhỏ và sử dụng kiến trúc ARM Cortex-A72 có thể xử lý đáp ứng được thời gian thực.

##### b. Ngôn ngữ dùng cho máy tính nhúng:

Trong phần nhận dạng khuôn mặt, nhóm em sử dụng máy tính nhúng Raspberry Pi và ngôn ngữ Python để lập trình cho Raspberry Pi.

##### Giới thiệu về Python:

Python là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn

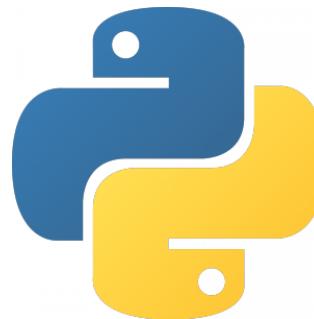
toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có cấu trúc dữ liệu cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở 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 Python đơn giản, dễ học
- Miễn phí, mã nguồn mở
- Khả năng di chuyển
- Khả năng mở rộng và có thể nhúng
- Ngôn ngữ thông dịch cấp cao
- Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến
- Hướng đối tượng

▫ Các ứng dụng của Python:

- Khoa học và tính toán
- Tạo nguyên mẫu phần mềm
- Ngôn ngữ tốt để dạy lập trình



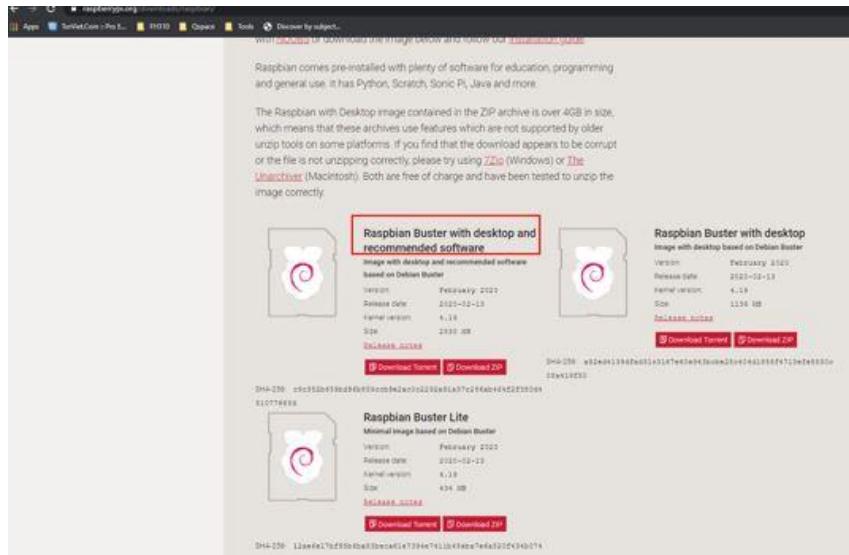
Hình 4. 25 Ngôn ngữ lập trình Python

#### 4.4.5 Phần mềm dùng cho máy tính nhúng

##### a. Cấu hình cho Raspberry Pi:

Bước 1: tải image trên trang chủ Raspberry

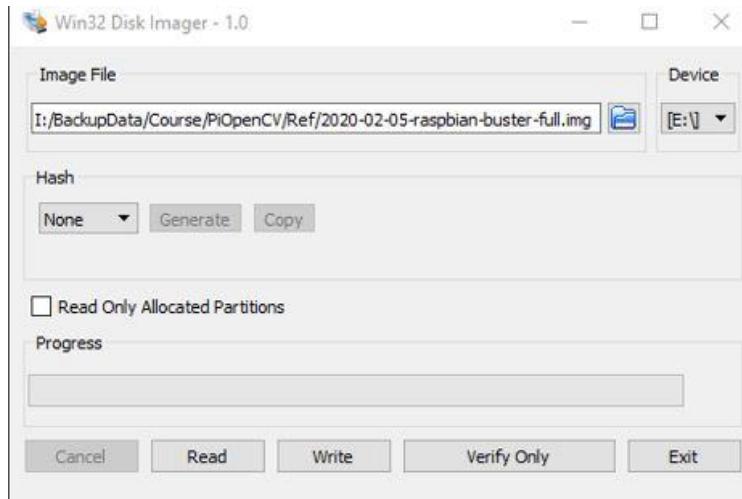
(<https://www.raspberrypi.org/downloads/raspbian/>)



Hình 4. 26 Ảnh càn tải đê flash lên SD card

Bước 2: sử dụng phần mềm Win32 Disk Imager để flash image xuống SD card

- Mở Win32 Disk Imager lên
- Vào ổ của SD card
- Mở file có đuôi .img lên, nó sẽ hiện ra 1 đường dẫn tới file đó
- Chọn **Write** để flash Image có đuôi .img lên thẻ SD



Hình 4. 27 Phần mềm Win32 Disk Imager

Bước 3: Bật UART trên board Raspberry Pi

- Sau khi flash image vào SD card, truy cập vào thư mục /boot trên thẻ nhớ, thêm dòng sau vào cuối file config.txt

### **enable\_uart=1**

- Cắm thẻ nhớ vào board và cấp nguồn



```

config.txt - Notepad
File Edit Format View Help
#framebuffer_width1280
#framebuffer_height720

# uncomment if hdmi display is not detected and composite is being output
#hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1
#hdmi_mode=2

# uncomment to force a HDMI mode rather than DVI. This can make audio work in
# some (but not all) monitor modes
#hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interference, blanking, or
# no display
#config_hdmi_boost=4

# uncomment for composite PAL
#dtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# Uncomment some or all of these to enable the optional hardware interfaces
#dtoverlay=pi2_grenan
#dtoverlay=pi2_ron
#dtoverlay=pi2_m

# Uncomment this to enable infrared communication.
#dtoverlay=pi2-infrared,pin=37
#dtoverlay=pi2-infrared,tx_gpio=pin18

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio=on

[pi4]
# Enable BCM V4 UVD driver on top of the dispmanx display stack
dwcovlvc4-firmware=v3d
max_framebuffers=2

[all]
#dtoverlay=vc4-fkms-v3d
enable_uart=1

```

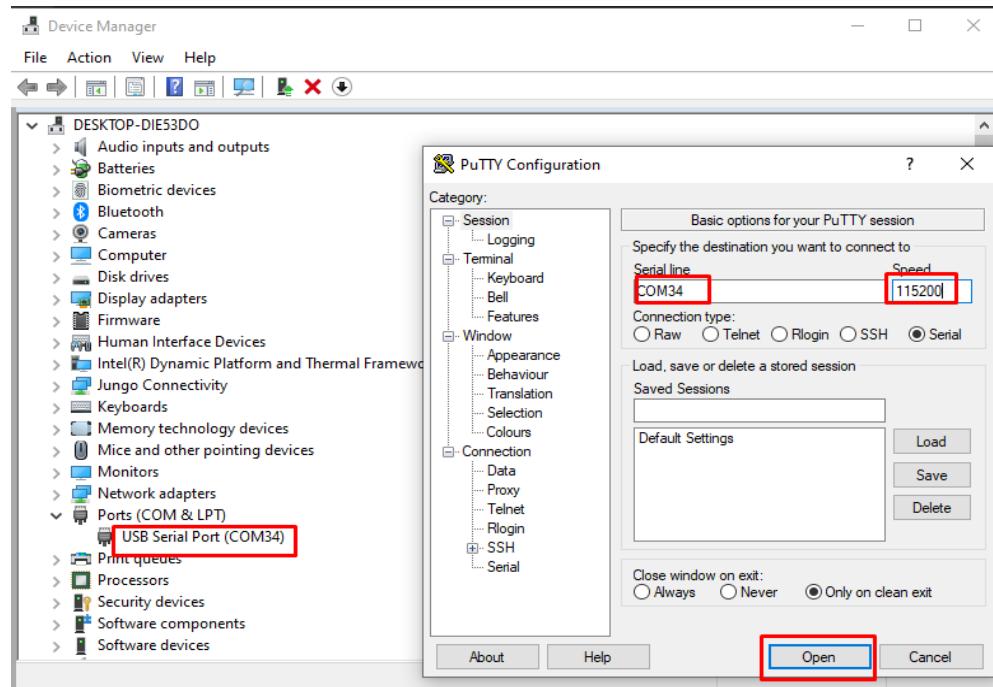
Hình 4. 28 Cách bật uart cho Raspberry Pi

Bước 4: Kết nối board Pi thông qua cổng Serial để tìm địa chỉ IP của board Pi sau khi board Pi đã kết nối với mạng Wifi

- Bước 4.1: Cài đặt phần mềm PuTTY
- Bước 4.2: Kết nối board Pi với PC qua mạch chuyển UART-USB, bật PuTTY lên, tích vào ô serial và nhập cổng COM với giá trị cổng COM xong trong Device Manager



Hình 4. 29 Sơ đồ chân Raspberry Pi



Hình 4. 30 Cách kết nối với Raspberry Pi qua Serial

- Bước 4.3: đăng nhập vào board Pi, nhập
  - user: pi
  - pass: raspberry

Bước 5: Tìm địa chỉ IP của board Pi thông qua kết nối Serial và PuTTY

- Bước 5.1: kiểm tra board Pi sử dụng mạng Wifi nào, gõ:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

The screenshot shows a terminal window with the command 'sudo nano /etc/wpa\_supplicant/wpa\_supplicant.conf' running. The file content is displayed, showing configuration for a network interface named 'wlan0'. The configuration includes the SSID 'HiepLam', the password 'HiepLam123456', and the key management method 'key\_mgmt=WPA-PSK'. The terminal window has a dark background and a light-colored text area.

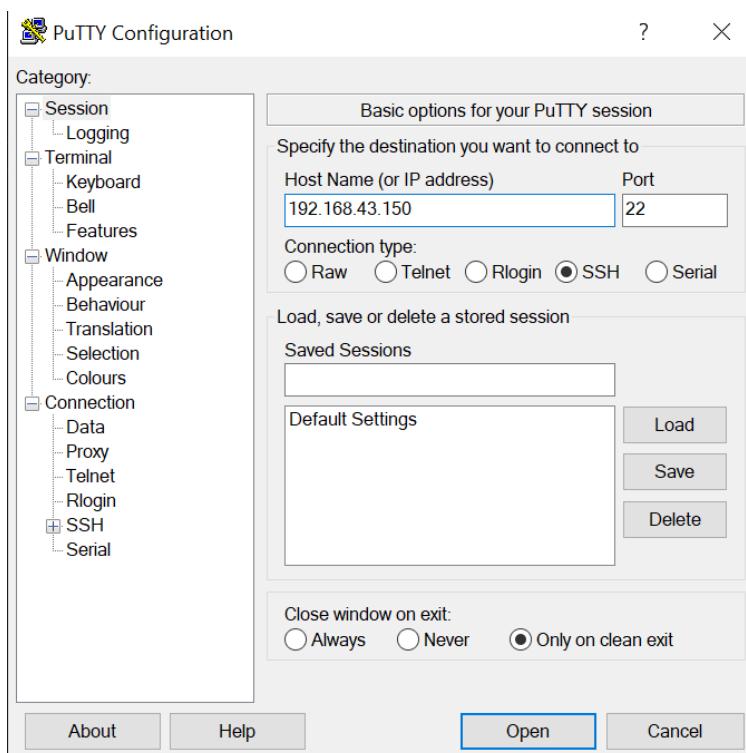
```
wifi
    essid="HiepLam"
    psk="HiepLam123456"
    key_mgmt=WPA-PSK
```

Hình 4. 31 Cách thay đổi địa chỉ wifi cho board Pi

- Thay đổi ssid và psk của board Pi để board Pi kết nối với mạng Wifi giống với mạng Wifi mà máy tính đang sử dụng
- Gõ Ctrl+S và Ctrl+X để lưu lại và thoát ra
- Bước 5.2: tiến hành reboot, gõ lệnh: *sudo reboot*
- Bước 5.3: Gõ *ifconfig* để tìm địa chỉ IP của Raspberry Pi.

Bước 6: Kết nối với Raspberry Pi thông qua SSH để bật camera và VNC Viewer cho board Pi

- Nếu không cần vào giao diện hệ điều hành của board Pi thì ta thông qua kết nối SSH để điều khiển board Pi thông qua terminal một cách gián tiếp
- Trên máy tính win 10, mở phần mềm PuTTY, nhập địa chỉ IP của board Pi và kết nối



Hình 4. 32 Phần mềm PuTTY

Bước 7: Kết nối và điều khiển board Pi qua Remote Desktop Connection trên Win 10 hoặc phần mềm VNC Viewer

- Cách kết nối thông qua Remote Desktop Connection:
  - Trên Pi, cài đặt xrdp - ứng dụng Remote Desktop Server, nếu chưa cài thì gõ lệnh *sudo apt-get install xrdp*

```

1 Change User Password Change password for the 'pi' user
2 Network Options Configure network settings
3 Boot Options Configure options for start-up
4 Localisation Options Set up language and regional settings to match your
5 Interfacing Options Configure connections to peripherals
6 Overclock Configure overclocking for your Pi
7 Advanced Options Configure advanced settings
8 Update Update this tool to the latest version
9 About raspi-config Information about this configuration tool

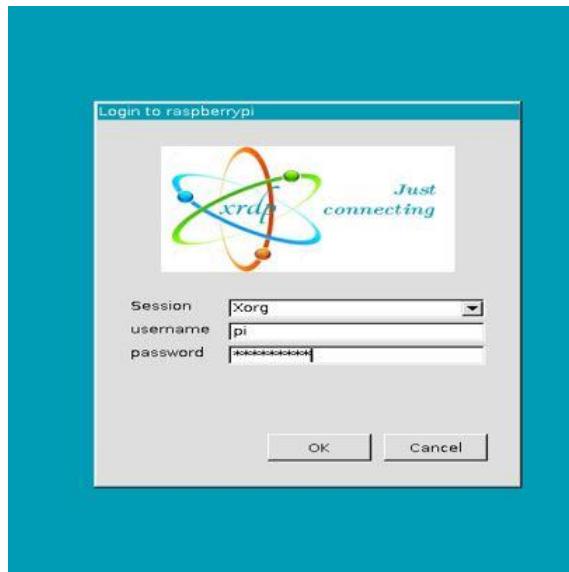
<Select> <Finish>

pi@raspberrypi:~$ sudo apt-get install xrdp
Reading package lists... done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ssl-cert x11-apps x11-session-utils xbitmaps xfonts-75dpi xfonts-base
  xfonts-scalable xorg xorg-docs-core xorgxrdp
Suggested packages:
  openssl-blacklist mesa-utils xorg-docs x11-xfs-utils guacamole
  xrdp-pulseaudio-installer
The following NEW packages will be installed:
  ssl-cert x11-apps x11-session-utils xbitmaps xfonts-75dpi xfonts-base
  xfonts-scalable xorg xorg-docs-core xorgxrdp xrdp
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 10.9 MB of additional disk space will be used.
After this operation, 17.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://mirror.freerdp.org/Raspbian/raspbian buster/main armhf ssl-cert all 1.0.39 [20.8 kB]
Get:2 https://mirror.freerdp.org/Raspbian/raspbian buster/main armhf x11-apps armhf 7.7+7 [541 kB]

```

Hình 4. 33 Cài đặt phần mềm xrdp cho board Pi

- Trên host PC win 10, bật chương trình Remote Desktop Connection, nhập địa chỉ IP của board Pi, thông tin đăng nhập và kết nối



Hình 4. 34 Phần mềm xrdp

- Cách kết nối thông qua VNC Viewer:
  - Trên phần mềm PuTTY sau khi kết nối Serial giữa board Pi và máy tính, gõ `vncserver`
  - Trên màn hình PuTTY sẽ hiện ra địa chỉ để chúng ta nhập vào VNC Viewer

```

pi@raspberrypi:~ login: pi
Mật khẩu: 
Lần đăng nhập: T5 Thg 7 6 20:17:11 +07 2021 trên ttys0
Linux raspberrypi 5.10.17-v7+ #1421 SMP Thu May 27 14:00:13 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
pi@raspberrypi:~$ vncserver
unable to open display ""

pi@raspberrypi:~$ vncserver
VNC(R) Server 6.7.3 ((4262a) ARMv6 (May 13 2020 19:34:20))
Copyright (C) 2001-2019 RealVNC Ltd.
RealVNC and VNC are trademarks of RealVNC Ltd and are protected by trademark
registrations and/or pending trademark applications in the European Union,
United States of America and other jurisdictions.
Protocol version 3.3. RealVNC's VNC server patent 6786746; EU patent 2652951.
See https://www.realvnc.com for information on VNC.
For third party acknowledgments see:
https://www.realvnc.com/docs/etosse.html
OS: Raspbian GNU/Linux 10, Linux 5.10.17, armv7l

On some distributions (in particular Red Hat), you may get a better experience
by running vncserver-vnc4 in conjunction with the system Xorg server, rather
than the Xorg server provided with RealVNC. Most desktop environments and
applications will likely be compatible. For more information on this alternative
implementation, please see: https://www.realvnc.com/doclink/kb-546

Running applications in /etc/vnc/xstartup
VNC Server catchphrase: "Stand human Electra. John Justin common."
signature: 19-0e-65-8a-1d-df-f4-a8

Log file is /home/pi/.vnc/raspberryvnc1.log
New desktop is raspberryvnc1 (192.168.1.6:5500)
pi@raspberrypi:~$ 

```

Hình 4. 35 Cách tìm địa chỉ IP để kết nối bằng VNC Viewer

### b. Phần mềm để soạn thảo code Python trên máy tính nhúng:

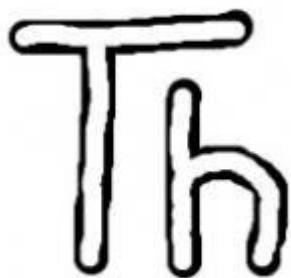
Để soạn thảo code trên máy tính, em sử dụng phần mềm Pycharm

PyCharm là môi trường phát triển tích hợp đa nền tảng (IDE) được phát triển bởi JetBrains và được thiết kế đặc biệt cho Python. PyCharm có mặt trên cả 3 nền tảng Windows, Linux và Mac OS.



Hình 4. 36 Phần mềm Pycharm

Trên máy tính nhúng, em sử dụng Thonny Python IDE để soạn thảo code Python. Thonny là công cụ hướng dẫn hữu ích, hỗ trợ những lập trình viên chưa có kinh nghiệm làm quen với ngôn ngữ lập trình Python thông qua môi trường màn hình nền (IDE) thân thiện. Chương trình cung cấp tính năng đánh dấu cú pháp và hoàn thành mã, cho phép chuyển hướng dễ dàng.



Hình 4. 37 Phần mềm Thonny Python IDE

Phần Face Detection và Face Recognition được mô phỏng trên MATLAB bằng giải thuật Viola Jones và PCA. Vì Python cũng khá mạnh về xử lý ảnh, ma trận, thích hợp dùng cho các thuật toán Machine Learning và ngoài ra nó rất đơn giản, trực quan, linh hoạt nên em chọn ngôn ngữ Python để sử dụng thực hiện trên máy tính nhúng.

Em đã viết lại giải thuật Viola Jones sử dụng cho Face Detection và thực hiện bằng module detector.py. Module detector.py này dùng để tách vùng có khuôn mặt từ các ảnh chụp người chính chủ có trong tập train và đưa sang tập Training. Thuật toán PCA để huấn luyện cho tập dữ liệu được em viết lại trong module train.py và run.py. Module train.py dùng để thực hiện giải thuật PCA ví dụ như tính ảnh trung bình, ảnh chuẩn hóa, ảnh covariance,... trên mỗi ảnh trong tập Training. Còn module run1.py dùng để thực hiện thuật toán PCA trên ảnh thu được từ camera Pi và so sánh giữa trọng số tính được từ ảnh chụp đầu vào đó với mỗi ảnh trong tập Training. Nếu hiệu giữa 2 trọng số dưới một ngưỡng thực nghiệm cho trước thì đó chính là chính chủ.

#### c. Các bộ thư viện và mô đun dùng cho máy tính nhúng:

Thư viện OpenCV:



Hình 4. 38 Thư viện OpenCV

Project OpenCV được bắt đầu từ Intel năm 1999 bởi Gary Bradsky. OpenCV viết tắt cho Open Source Computer Vision Library. OpenCV là thư viện nguồn mở hàng đầu cho Computer Vision và Machine Learning, và hiện có thêm tính năng tăng tốc GPU cho các hoạt động theo real-time.

OpenCV có trên các giao diện C++, C, Python và Java và hỗ trợ Windows, Linux, Mac OS, iOS và Android. OpenCV được thiết kế để hỗ trợ hiệu quả về tính toán và chuyên dùng cho các ứng dụng real-time (thời gian thực). Nếu được viết trên C/C++ tối ưu, thư viện này có thể tận dụng được bộ xử lý đa lõi (multi-core processing).

❖ **Ứng dụng OpenCV:**

- Hình ảnh street view
- Kiểm tra và giám sát tự động
- Robot và xe hơi tự lái
- Phân tích hình ảnh y học
- Tìm kiếm và phục hồi hình ảnh/video
- Phim – cấu trúc 3D từ chuyển động
- Nghệ thuật sắp đặt tương tác

❖ **Thư viện Numpy:**



Hình 4. 39 Thư viện NumPy

Numpy là một thư viện lõi phục vụ cho khoa học máy tính của Python, hỗ trợ cho việc tính toán các mảng nhiều chiều, có kích thước lớn với các hàm đã được tối ưu áp dụng lên các mảng nhiều chiều đó. Numpy đặc biệt hữu ích khi thực hiện các hàm liên quan tới Đại Số Tuyến Tính.

Sử dụng NumPy, lập trình viên có thể thực hiện các thao tác sau:

- Các phép toán toán học và logic trên mảng.
- Các biến đổi Fourier và các quy trình để thao tác shape.
- Các phép toán liên quan đến đại số tuyến tính. NumPy tích hợp sẵn các hàm cho đại số tuyến tính và tạo số ngẫu nhiên.

NumPy là sự thay thế hoàn hảo cho Matlab

✚ Thư viện Matplotlib:



Hình 4. 40 Thư viện matplotlib

Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu nên Matplotlib là một trong những giải pháp như vậy cho người dùng Python. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotlib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở.

Trong luận văn lần này, nhóm em dùng thư viện này để hiển thị ra các ảnh khuôn mặt chính chủ ban đầu, trung bình, phương sai (chuẩn hóa), hiệp phương sai và ảnh chính chủ sau khi chiếu lên ma trận PCA

## 5. KẾT QUẢ THỰC HIỆN

### 5.1 Nguồn thực nghiệm để nhận dạng chính chủ

#### 5.1.1 Nhận dạng trên phần mềm MATLAB

##### a. Nhận dạng giọng nói:

Trong quá trình thực nghiệm để tìm nguồn nhận dạng giọng nói chính chủ, nhóm em đã lấy 60 mẫu giọng nói chính chủ bằng cách lặp lại từ 60 lần trong 55 giây (các số từ 0 đến 9) và lấy 10 mẫu giọng nói người lạ (với 10 người nói tình nguyện) bằng cách lặp lại 10 lần trong 10 giây (các số từ 0 đến 9).

Qua nhiều quá trình so sánh thực nghiệm, do khoảng cách giữa giọng nói chính chủ với tập 60 mẫu giọng chính chủ lớn hơn khoảng cách giữa giọng nói chính chủ với tập 10 mẫu giọng nói người lạ. Do đó, nhóm em quyết định thay vì lấy nguồn toàn bộ hệ thống 10 số thì lấy nguồn của từng số từ 0 đến 9.

Dữ liệu giọng nói được nhóm em thu thập từ bộ dữ liệu 9 người có sẵn trên internet, 10 người thu âm thật ở ngoài và thêm 1 người nói chính chủ thu âm thật.

Riêng người nói chính chủ được thu âm giọng nói 60 lần mỗi số, còn lại 9 mẫu dữ liệu giọng nói trên mạng lặp lại 1 lần và 10 mẫu dữ liệu giọng nói thu trực tiếp lặp lại 10 lần

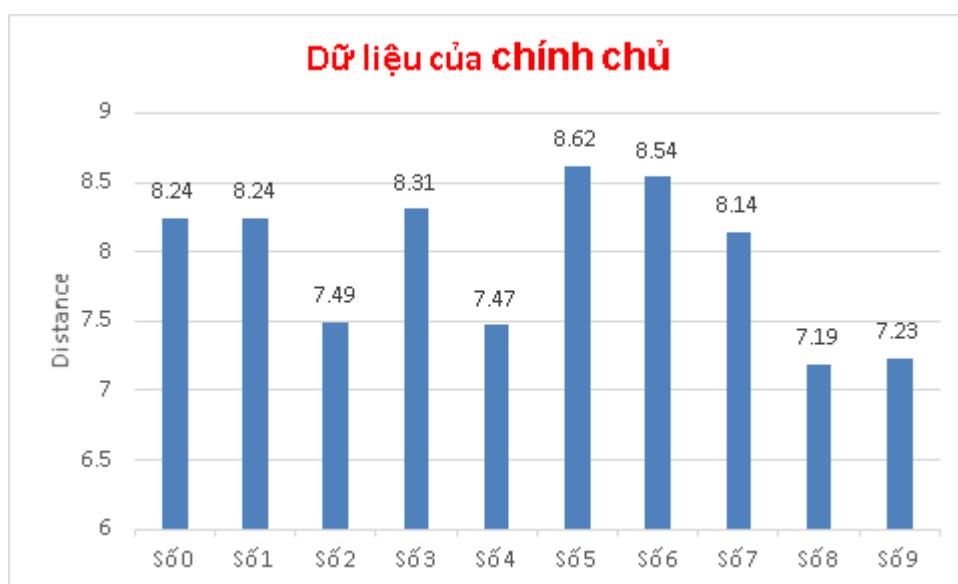
- Bảng dữ liệu khi thực nghiệm tính toán để tìm ngưỡng:

	Người 1	Người 2	Người 3	Người 4	Người 5	Người 6	Người 7	Người 8	Người 9
Số	15.4147 4266	18.9229 5762	15.8128 6256	15.4156 5584	15.6370 9608	19.7145 5184	19.3576 5198	16.0500 6931	16.7429 2398
Số	15.5605 7074	15.5228 6313	13.0628 5997	15.2301 0278	16.2204 102	14.0026 4675	12.8321 8126	13.7268 1987	11.4517 5156
Số	14.4388 6039	23.7796 97	15.8376 6771	15.5285 1754	18.4963 7392	15.9899 2353	16.8372 7752	18.0326 0528	16.0888 7528
Số	9.66744 9517	16.0929 8267	12.0866 1326	13.4684 1954	17.3324 422	12.1943 0592	16.0245 1582	13.5451 5841	15.1026 7093
Số	9.76098 4447	11.5950 6482	14.4779 3232	14.8771 702	11.6202 9577	12.0511 4167	13.0307 8879	15.9760 6923	13.9294 8427
Số	15.0185 5099	18.5316 2763	16.6080 4455	12.7980 4334	14.9944 4397	18.2034 6573	18.4876 6671	16.4459 2178	15.2925 5209
Số	12.4740 4805	16.9183 0886	13.9271 9869	13.3679 4655	15.0162 2485	14.2346 3601	14.6738 8807	13.1669 1818	14.9628 1554
Số	15.0136 2958	15.5681 4945	13.9472 5694	13.9434 1132	19.7734 7563	12.0151 1598	12.1484 879	13.8789 5161	13.4468 1764
Số	11.1714 9841	13.7390 8183	13.5267 6961	12.6129 1265	10.8478 784	16.0597 4478	12.4895 3103	14.7148 9099	13.3708 9957
Số	16.1901 7603	14.1475 3185	14.7700 6076	11.1798 5802	14.5319 9862	15.3404 8951	11.9661 6509	13.6537 0812	11.2168 2549

Bảng 5. 1 Bảng dữ liệu để tìm ngưỡng giọng nói chính chủ từ người 1 đến người 9

Người 10(chính chủ)	Người 11	Người 12	Người 13	Người 14	Người 15	Người 16	Người 17	Người 18	Người 19	Người 20
8.2410796 47	12.013 69928	12.112 7066	10.866 34537	11.209 32506	10.056 04623	11.201 41602	11.343 32472	10.646 59749	9.6090 93362	13.131 32071
8.2445561 99	8.6005 82972	8.7374 55526	9.8001 07544	11.684 89786	10.508 60291	10.322 37134	9.3655 24374	11.815 37648	12.442 87071	9.2149 59728
7.4881997 42	11.592 68409	11.330 8309	9.3293 81851	9.8658 72959	9.8651 2916	11.585 36235	11.081 08461	11.758 80688	10.039 87224	12.157 49359
8.3086287 72	11.402 07637	9.3893 65364	10.757 80865	9.8914 82842	9.6979 96923	9.8828 85776	8.6717 83587	10.473 43352	8.3788 35429	12.190 51654
7.4736496 26	9.5439 65003	8.0102 37153	12.696 5971	8.9982 6301	7.5548 02562	8.0367 04988	8.4973 1339	9.7740 84302	8.4961 82224	9.1560 51422
8.6197216 83	12.025 96163	10.508 95676	10.821 36028	8.7396 34	10.964 32466	10.579 9341	9.1178 81206	10.149 46689	8.9154 3092	12.365 08833
8.5372540 45	13.322 52186	12.779 37849	12.445 68348	11.598 07541	9.8186 06102	10.020 60224	12.243 24851	12.844 68344	11.986 08441	12.331 59053
8.1398757 5	10.464 10112	10.150 13374	12.154 43704	10.054 86513	11.249 29545	12.306 11	9.2692 9868	10.237 47511	11.074 73463	11.832 48932
7.1868237	7.4206 98899	10.092 14174	9.9339 71901	9.0478 89999	7.4473 21164	9.3719 61173	8.9148 04973	7.5850 92332	7.4603 71394	10.421 79596
7.2264676 09	8.2749 52822	9.0793 19249	9.4704 95072	8.8519 80573	10.122 58168	8.8388 885	9.3128 85935	8.3345 21924	8.1559 27249	10.078 53864

Bảng 5. 2 Bảng dữ liệu tìm ngưỡng giọng nói chính chủ từ người 11 đến người thứ 20



Hình 5. 1 Dữ liệu khoảng cách của chính chủ so với đầu vào chính chủ

- Ngưỡng của từng số :

SỐ	NGƯỠNG
KHÔNG	8.6
MỘT	8.1
HAI	7.9
BA	6.5
BỐN	8.7
NĂM	9.0
SÁU	8.3
Bảy	8.6
TÁM	8.0
CHÍN	9.0

### b. Nhận dạng khuôn mặt:

Về vấn đề tìm ngưỡng nhận dạng khuôn mặt chính chủ, nhóm em đã thu thập 30 khuôn mặt của 5 người lạ khác nhau, đại diện cho tập người lạ và 30 khuôn mặt của chính chủ để so sánh tất cả với khuôn mặt đầu vào là chính chủ. Dựa trên quá trình so sánh thực nghiệm, nhóm em đặt ra ngưỡng cho nhận dạng khuôn mặt chính chủ là 14000.

Khoảng cách nhỏ nhất và lớn nhất trong 30 ảnh chính chủ lần lượt là 7763.69 và 29833.5 và khoảng cách nhỏ nhất trong 30 ảnh người lạ là 13226.08.

Do giải thuật trích xuất đặc trưng vector riêng từ PCA là giải thuật đơn giản, không quá phức tạp nên với những yếu tố ánh sáng, nền (background có kèm một ít trong ảnh sau khi cắt khuôn mặt), khuôn mặt chính chủ thay đổi theo thời gian, nên không thể tránh khỏi sai sót và ngưỡng có thể xác định dễ dàng giống như lý thuyết (theo lý thuyết là khoảng cách lớn nhất giữa ảnh chính chủ đầu vào và tập chính chủ nhỏ hơn khoảng cách nhỏ nhất giữa ảnh chính chủ đầu vào và tập người lạ).

### 5.1.2 Nhận dạng trên phần cứng

#### a. Nhận dạng giọng nói:

Đối với nhận dạng giọng nói trên phần cứng, vì lý do bộ nhớ flash của vi điều khiển STM32F407VG không quá lớn, chỉ có 192 KB.

Vì vậy, nhóm em bị hạn chế trong quá trình thu thập và lấy mẫu dữ liệu giọng nói chính chủ các số từ “không” đến “chín”. Nhóm em chỉ lấy 3 mẫu, tức là 3 lần lặp lại cho mỗi số khi thu âm.

Do thời gian và kiến thức có hạn nên nhóm em chỉ mới dừng lại ở bước nhận dạng từng số, chưa hoàn thành kịp thuật toán tách các từ trên vi điều khiển.

Bên cạnh đó, qua nhiều quá trình thực nghiệm gồm thu âm và kiểm tra trên vi điều khiển, chúng em đã thu được các ngưỡng để nhận dạng chính chủ trên vi điều khiển sau:

- Nguồng số 0: 8.0
- Nguồng số 1: 8.0
- Nguồng số 2: 9.0
- Nguồng số 3: 7.0
- Nguồng số 4: 9.3
- Nguồng số 5: 8.0
- Nguồng số 6: 8.0
- Nguồng số 7: 8.5
- Nguồng số 8: 9.3

- Nguồn số 9: 9.0

### b. Nhận dạng khuôn mặt:

- Đối với phần nhận dạng khuôn mặt, qua quá trình thực nghiệm nhóm em chọn nguồn là  $2.5 \times 10^7$  khi thực hiện nhận dạng khuôn mặt chính chủ trên Raspberry Pi 4.

## 5.2 Kết quả lập trình ứng dụng trên nền tảng MATLAB



Hình 5. 2 Giao diện phần mềm mô phỏng nhận dạng khuôn mặt và giọng nói bằng MATLAB

Giao diện MATLAB trên được nhóm em lập trình và thiết kế bằng App Designer.

Giao diện chương trình mô phỏng thuật toán nhận dạng người bằng giọng nói và khuôn mặt của nhóm em dựa trên 3 lớp bảo mật.

Lớp đầu tiên sẽ là lớp nhận dạng khuôn mặt chính chủ, đúng chính chủ thì đèn thứ ba từ trái sang phải sẽ bật xanh, còn không đúng sẽ bật đỏ, ban đầu đèn này có màu xám.

Khi đưa đúng mặt người chính chủ trước webcam của Laptop và bật chương trình lên, bấm chọn nút **Webcam** để bật Webcam trên Laptop. Sau đó, chọn chụp ảnh để **Chụp Ảnh** gương mặt của người dùng.

Vì hệ thống chỉ cần huấn luyện (nút **Train**) một lần đầu tiên, nên những lần sau người dùng không cần phải bấm nút **Train** nữa (nếu cập nhật lại ảnh huấn luyện mới thì phải train lại).

Để train hay huấn luyện cho tập ảnh huấn luyện thì bạn cần nhập số ảnh cần huấn luyện vào và số trị riêng sau khi giảm xuống và sau đó chọn nút **Train**.

Để nhận dạng khuôn mặt là chính chủ hay không, ta chọn nút **Nhận Dạng**. Ở trường hợp đã huấn luyện (train) rồi, ta chỉ cần nhập số ảnh trong tập huấn luyện (ở đây ví dụ là 60 ảnh) và chọn nút **Nhận Dạng** để nhận dạng khuôn mặt là chính chủ hay không. Nếu đúng là chính chủ ảnh sẽ hiển thị trên màn hình **Ảnh sau khi nhận dạng** là ảnh chính chủ trong tập huấn luyện.

Nút **Excel** để xuất thông số khoảng cách giữa ảnh chụp đầu vào với ảnh trong tập huấn luyện (nếu khoảng cách nhỏ dưới ngưỡng thực nghiệm thì xác định là chính chủ) ra một file Excel cho trước, nhằm mục đích dễ dàng thu nhận dữ liệu để tìm ngưỡng cho ảnh chính chủ.

Màn hình đen ở phía tay phải giao diện có tác dụng hiển thị các dòng thông báo sau khi người dùng train ảnh, xuất file Excel hay nhận dạng ảnh.

Nút **Clear** để xóa các dòng chữ hiển thị trên màn hình đen thông báo. Đồng thời, nút này cũng reset lại màu xám ban đầu cho đèn số 3 từ trái sang phải trên giao diện

Nút **Dừng Webcam** để tắt Webcam sau khi Webcam Laptop được bật lên

Nút **Load Ảnh** để tải một ảnh nào đó nằm trong máy tính lên giao diện để nhận dạng (có thể thay cho ảnh chụp từ Webcam máy tính).

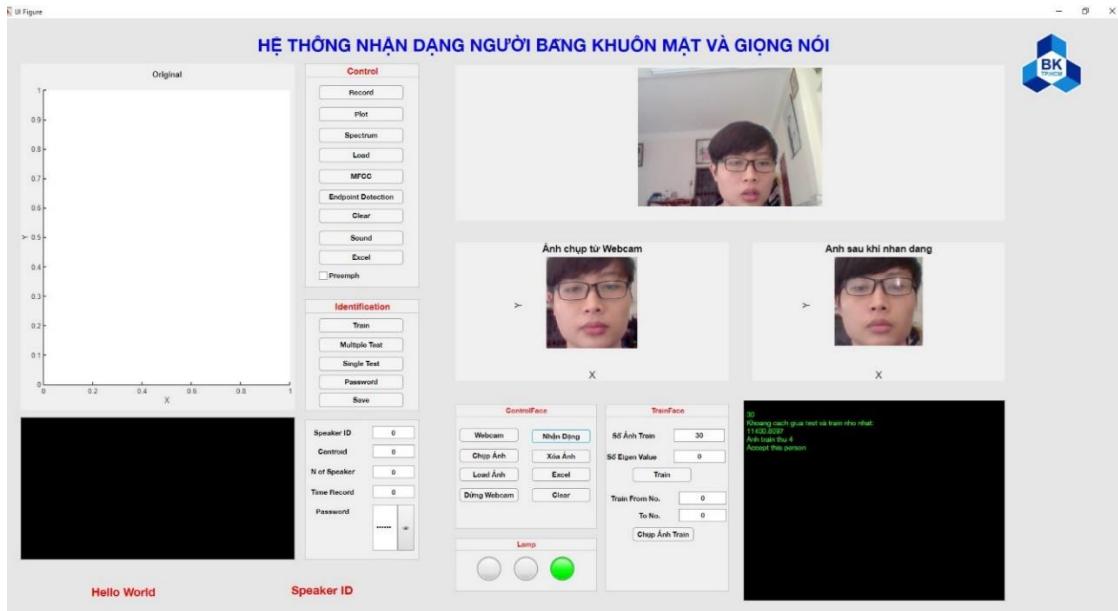
Nút **Xóa Ảnh** để xóa ảnh trên hai màn hình Axes của hệ thống nhận dạng khuôn mặt, bao gồm màn hình **ảnh chụp từ Webcam** và màn hình **Ảnh sau khi nhận dạng**.

Màn hình trên cùng có dòng chữ **Webcam** nhằm mục đích hiển thị ảnh chuyển động thời gian thực từ webcam máy tính.

Ở trong mục **TrainFace** có phần để điền thứ tự ảnh cần huấn luyện từ mục **Train From No.** đến **To No.**. Và ngoài ra, có nút **Chụp Ảnh Train** để chụp ảnh đưa vào tập huấn luyện, nhằm phục vụ cho mục đích tìm ngưỡng để xác định chính chủ hay người lạ.

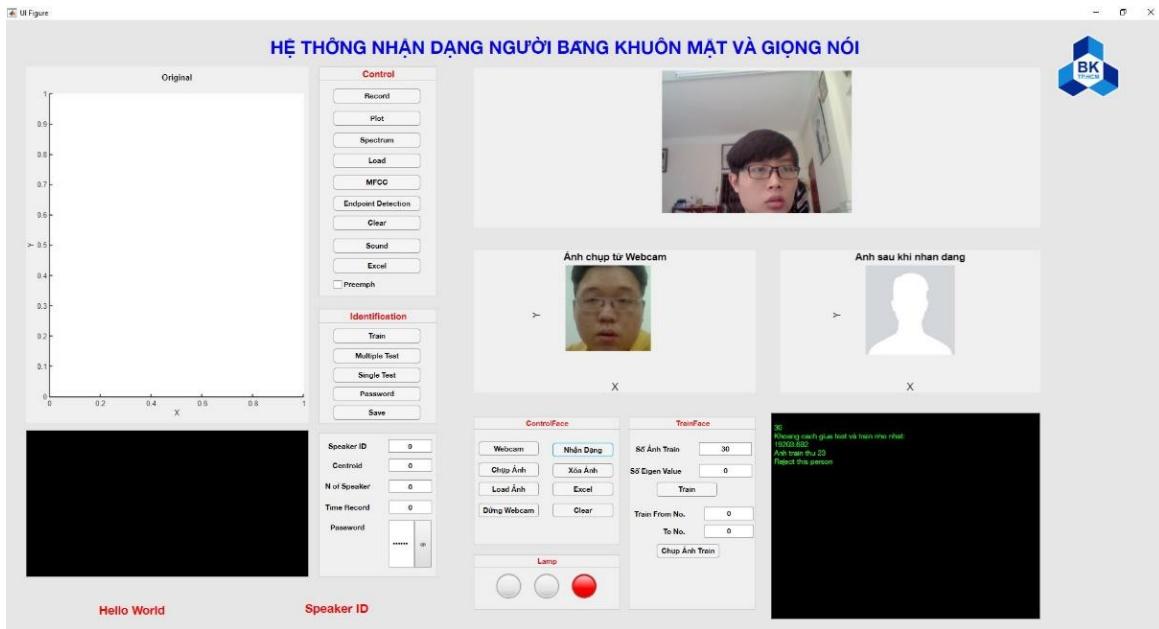
Ở mục **ControlFace** có mục để điền số ảnh chính chủ trong tập huấn luyện. Ví dụ ở đây là 30 ảnh chính chủ chụp ở các background khác nhau.

Khi hệ thống nhận dạng đúng khuôn mặt chính chủ, đèn thứ ba từ trái qua sẽ hiển thị màu xanh lá cây. Các thông báo sẽ hiển thị trên màn hình đen của phần nhận dạng khuôn mặt



Hình 5. 3 Trường hợp nhận dạng khuôn mặt đúng chính chủ

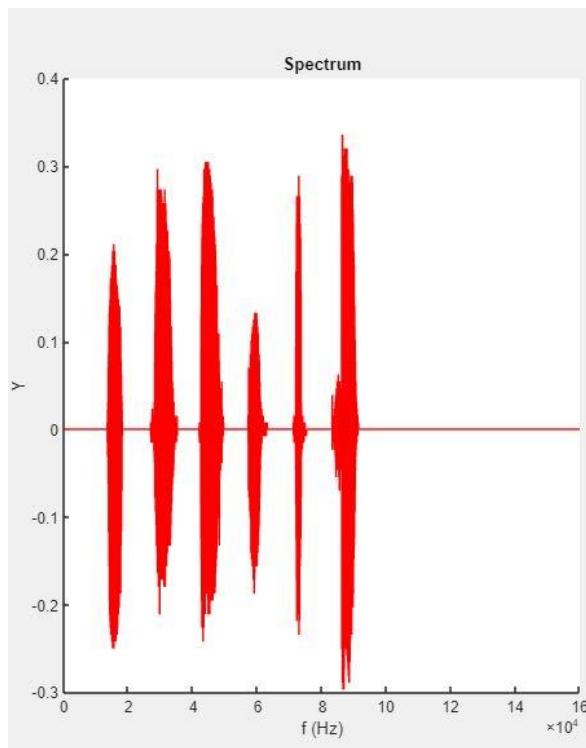
Khi hệ thống nhận dạng mặt người không phải chính chủ, bên màn hình **Ảnh sau khi nhận dạng** sẽ hiện ra hình ảnh đại diện Unknown (không phải chính chủ). Lúc này đèn thứ 3 từ trái sang phải sẽ bật màu đỏ.



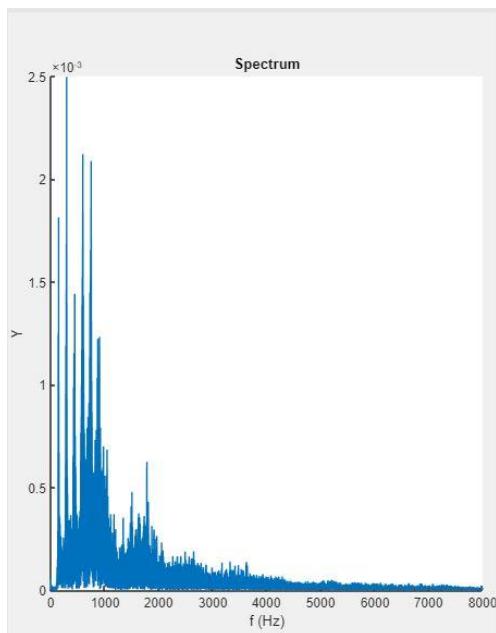
Hình 5. 4 Trường hợp nhận khuôn mặt sai chính chủ

Lớp bảo mật thứ 2 và thứ 3 của hệ thống sẽ là nhận dạng đặc trưng giọng nói của chính chủ và mật khẩu mà người đó nói.

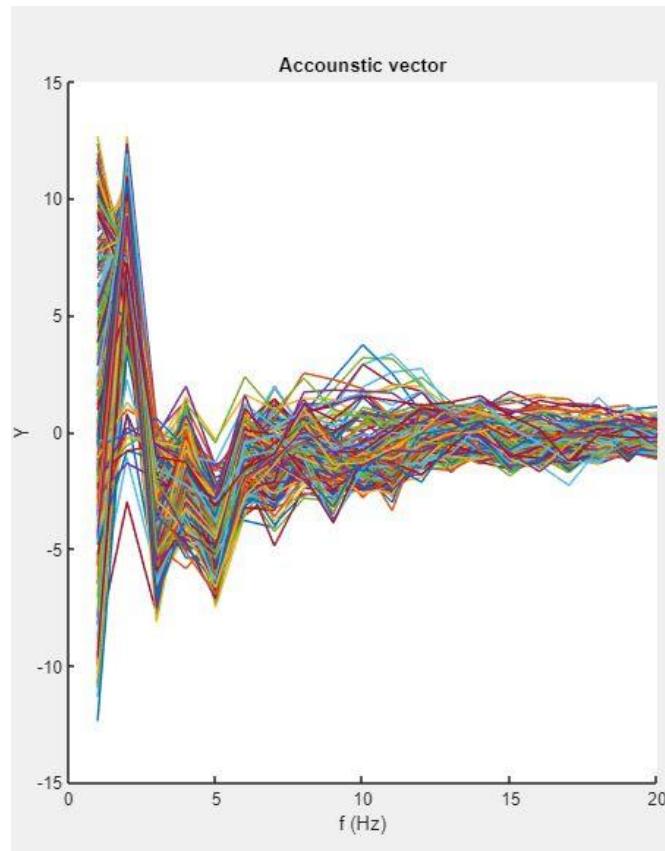
Ở phía ngoài cùng bên trái của giao diện là màn hình hiển thị tín hiệu sóng âm thanh, thông số đặc trưng Acoustic Vector sau khi trích xuất MFCC, phổ biên độ (spectrum) của tín hiệu sau khi thu âm. Bên dưới màn hình này là màn hình màu đen để hiển thị thông báo sau và thông số khoảng cách sau khi nhận dạng, huấn luyện, xuất file Excel hay nhận dạng mật khẩu dạng chữ số.



Hình 5. 5 Dạng sóng của tín hiệu âm thanh theo miền thời gian, văn bản từ “một” tới “sáu”



Hình 5. 6 Phô biên độ của tín hiệu vừa thu được ở trên



Hình 5. 7 Acoustic Vector của tín hiệu âm thanh từ “một” đến “sáu”

Năm bên tay phải của màn hình hiển thị đồ thị là bảng **Control** bao gồm các nút có các chức năng sau:

- **Record:** để thu âm giọng nói
- **Plot:** để vẽ đồ thị tín hiệu âm thanh ban đầu trên màn hình
- **Spectrum:** để vẽ đồ thị tín hiệu phổ biên độ của tín hiệu âm thanh
- **Load:** để tải file âm thanh .wav lên hệ thống
- **MFCC:** để vẽ đồ thị Acoustic Vector sau khi trích đặc trưng sử dụng phương pháp MFCC.
- **Endpoint Detection:** để cắt các từ trong một đoạn âm thanh ra thành từng âm riêng lẻ
- **Clear:** để xóa các dòng thông báo màu xanh lá cây trên màn hình màu đen bên dưới
- **Sound:** để phát lại âm thanh vừa thu âm

- **Excel:** để xuất các dữ liệu về khoảng cách giữa các file .wav trong tập huấn luyện và âm thanh đầu vào vừa thu âm, xuất ra màn hình màu đen
- **Ô tích Preemph:** để vẽ tín hiệu âm thanh sau khi tiền nhấn mạnh (Pre-Emphasis) ra màn hình đồ thị

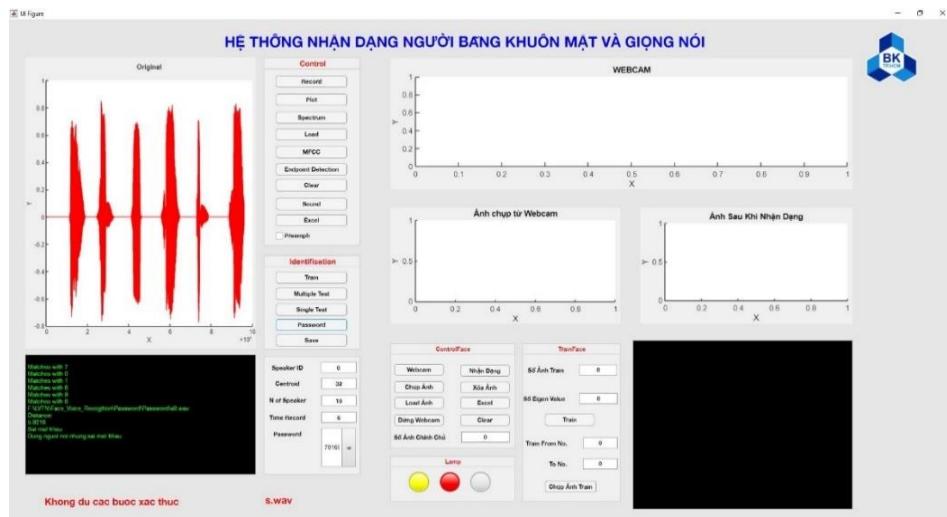
Bên dưới bảng Control là bảng **Identification** bao gồm các nút với các chức năng:

- **Train:** để huấn luyện cho các file âm thanh .wav trong tập huấn luyện
- **Single Test:** để tính khoảng cách nhận dạng từ âm thanh thu vào với các file âm thanh trong tập huấn luyện
- **Password:** để nhận dạng mật khẩu của tiếng nói
- **Save:** để lưu lại file âm thanh vừa thu âm vào một đường dẫn bất kỳ được chỉ định trước trong máy tính (chỉ định bằng phần mềm)

Bên dưới bảng Identification là các ô để điền các thông số:

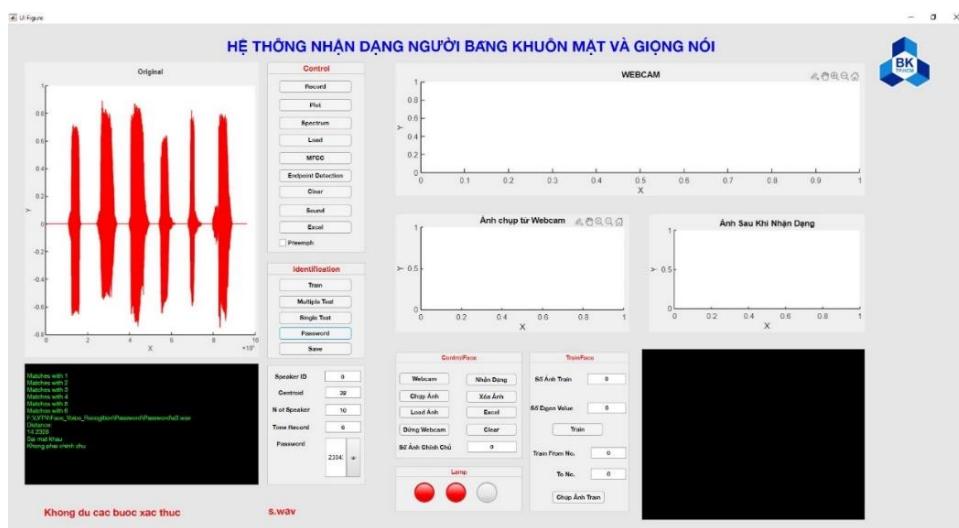
- **Speaker ID:** trước khi muốn lưu lại file âm thanh vừa ghi âm cần phải nhập **Speaker ID** là mã số của file âm thanh vừa thu
- **Centroid:** trước khi huấn luyện hay nhận dạng giọng nói và mật khẩu, ta cần phải nhập số Centroid là số nhân codebook (thường nhập 32).
- **N of Speaker:** trước khi huấn luyện hay nhận dạng giọng nói và mật khẩu, ta cần phải nhập số người file cần huấn luyện. Trong trường hợp huấn luyện và tính khoảng cách giữa file thu âm và các file huấn luyện thì sẽ nhập **N of Speaker** là 20, vì có 20 người trong tập huấn luyện, mỗi người sẽ phát âm từ 0 đến 9. Trong trường hợp nhận dạng mật khẩu, thì ta nhập **N of Speaker** là 10, vì có 10 chữ số có thể hình thành nên mật khẩu là từ 0 đến 9.
- **Time Record:** nhằm nhập số lượng thời gian để thu âm tiếng nói đầu vào dành cho người dùng, tính theo đơn vị giây.
- **Password:** nhằm để nhập mật khẩu vào cho hệ thống (không giới hạn số ký tự nhập vào), hệ thống chỉ dừng lại ở mức nhận dạng mật khẩu là các chuỗi số. Các số khi thu âm phải được người nói phát âm rõ ràng và có khoảng nghỉ giữa các từ. Mật khẩu phải được nhập trước khi chọn nút **Password** ở bảng **Identification** để nhận dạng mật khẩu

Đèn đầu tiên từ trái qua phải là đèn xác định của hệ thống nhận dạng giọng nói và mật khẩu, nếu đèn màu vàng có nghĩa là người nói đúng là chính chủ nhưng sai mật khẩu.



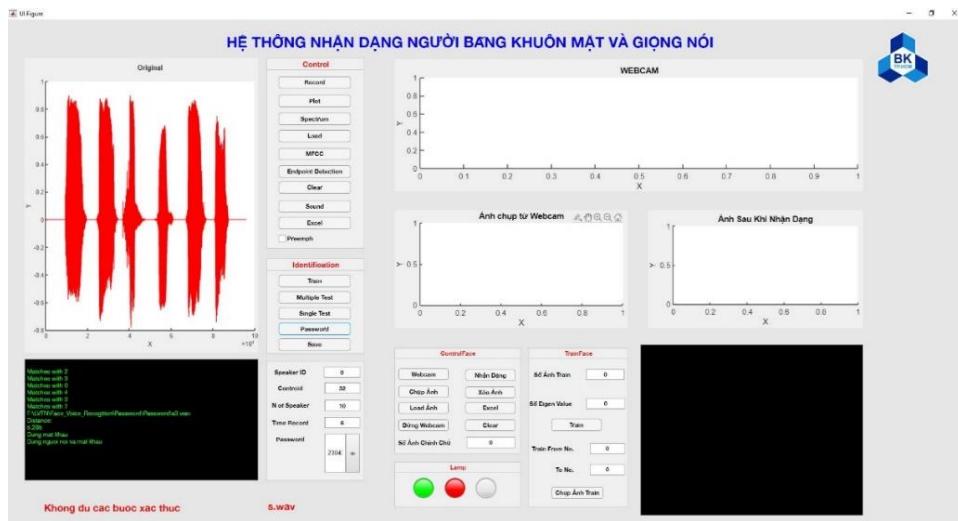
Hình 5. 8 Trường hợp nhận dạng đúng giọng nói chính chủ nhưng sai mật khẩu

Nếu đèn đầu tiên từ trái qua phải có màu đỏ sau khi nhận dạng có nghĩa là hệ thống nhận dạng sai người chính chủ và sai luôn mật khẩu



Hình 5. 9 Trường hợp nhận dạng giọng nói sai cả giọng chính chủ và mật khẩu

Nếu đèn đầu tiên từ trái sang phải hiện màu xanh có nghĩa là hệ thống đã nhận dạng đúng chính chủ và người đó nói đúng mật khẩu được cài đặt trước.

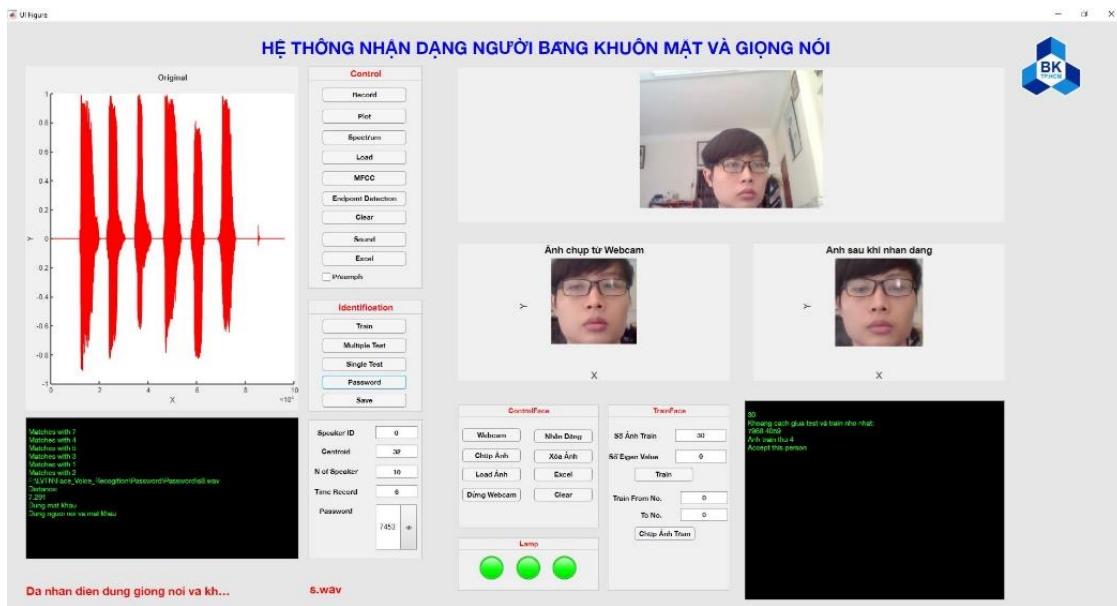


Hình 5. 10 Trường hợp nhận dạng đúng giọng chính chủ và mật khẩu

Sau khi hệ thống nhận dạng đúng cả khuôn mặt, giọng nói và mật khẩu tiếng nói của chính chủ thì hệ thống sẽ hiển thị màu xanh ở đèn số 2 từ trái qua phải. Ngược lại chỉ cần một hoặc hai hoặc ba lớp bảo mật này của hệ thống nhận dạng không phải chính chủ thì đèn số 2 từ trái qua phải sẽ hiển thị màu đỏ.

Để reset lại hai đèn 1,2 từ trái sang trong hệ thống nhận dạng giọng nói và mật khẩu sang màu xám ban đầu, ta nhấn nút **Clear** trong bảng **Control**.

Sau khi đã nhận dạng đúng được cả khuôn mặt chính chủ và giọng nói chính chủ (đặc trưng và mật khẩu) thì cả 3 đèn đều hiện màu xanh lá cây. Đèn giữa sẽ là đèn quyết định cả 3 lớp bảo mật đều đúng. Nếu chỉ cần 1 trong 3 điều kiện bảo mật sai thì đèn ở giữa sẽ hiện màu đỏ.



Hình 5. 11 Trường hợp nhận dạng đúng cả 3 lớp bảo mật

Đánh giá và nhận xét phần mềm: phần mềm mô phỏng giải thuật trên MATLAB có giao diện thân thiện với người dùng, thao tác đơn giản. Hệ thống có nhiều nút chức năng cho người dùng sử dụng. Nhưng hệ thống chụp ảnh và nhận dạng khuôn mặt vẫn hơi bị trễ do xử lý ảnh yêu cầu RAM của PC hay laptop phải mạnh. Ngoài ra, nhận dạng mật khẩu đôi khi bị sai một số từ (như số 4, số 8, số 2).

### 5.3 Kết quả phần cứng

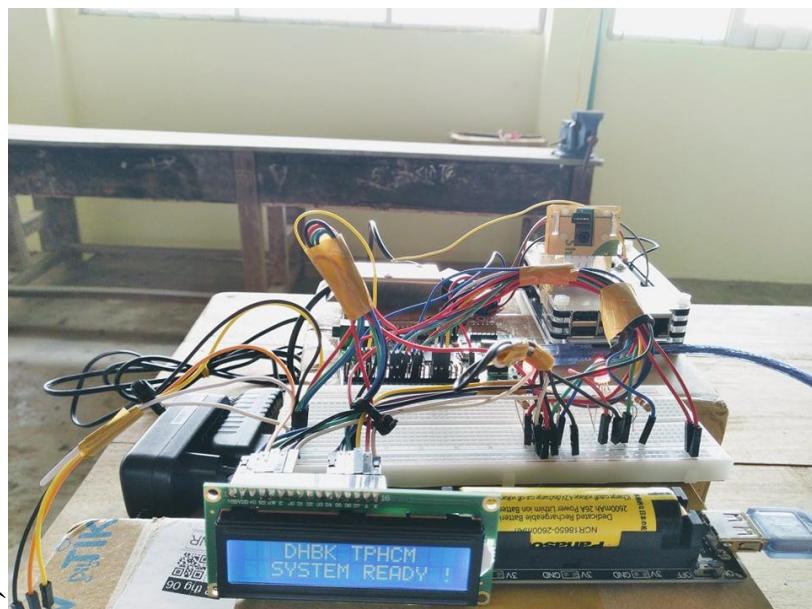
- Ở phần cứng nhóm em chưa thực hiện được password 6 số, nên password trên phần cứng trong luận văn này chỉ có 1 số.
- Hệ thống gồm có hai đèn LED RGB ở bên trái và bên phải.
  - o Đèn bên trái: báo kết quả nhận dạng giọng nói

- Đèn bên phải: báo kết quả nhận dạng khuôn mặt
- Chức năng báo hiệu của đèn hai LED RGB trong hệ thống:

	Màu đỏ	Màu vàng	Màu Xanh
Đèn RGB 1	Nhận dạng ra người nói là người lạ	Nhận dạng sai mật khẩu, đúng người nói chính chủ	Nhận dạng đúng người nói và mật khẩu chính chủ
Đèn RGB 2	Nhận dạng ra khuôn mặt người lạ	Không có	Nhận dạng ra khuôn mặt chính chủ

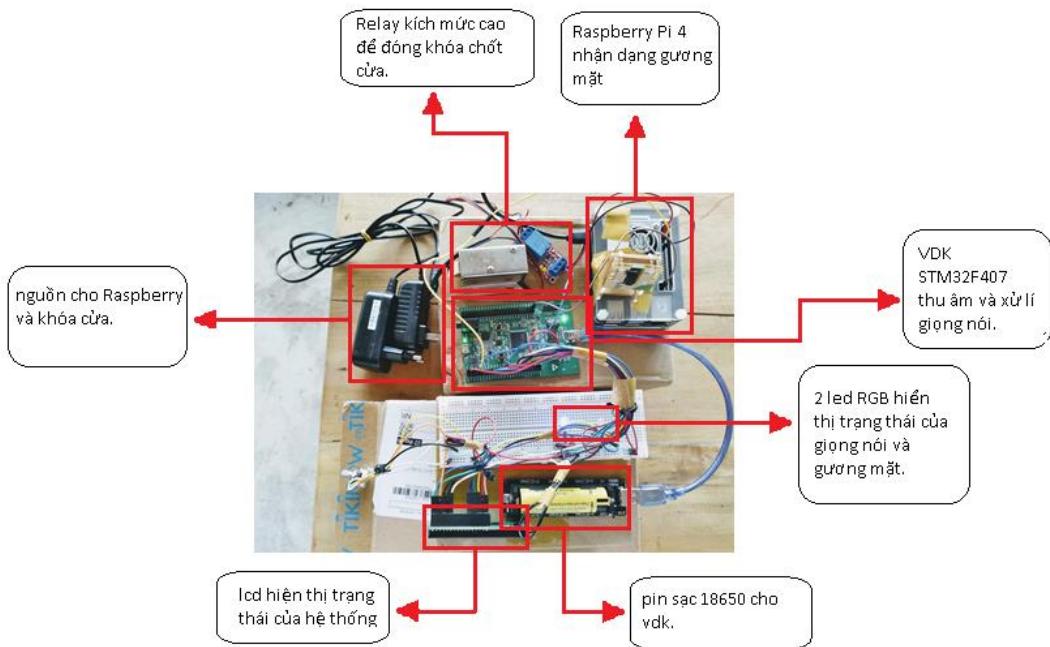
Bảng 5. 3 Bảng chức năng hai đèn LED trong hệ thống

- Mặt trước phần cứng:



Hình 5. 12 Mặt trước phần cứng

- Mặt trên phần cứng:



Hình 5. 13 Các khái niệm của phần cứng

Vì như đã nói ở phần 5.1.2, nhóm em chưa kịp hoàn thành giải thuật tách từ trên vi điều khiển, nên đã không thể nhận dạng mật khẩu 6 chữ số được. Do đó, nhóm em quyết định nhận dạng từng chữ số khác nhau.

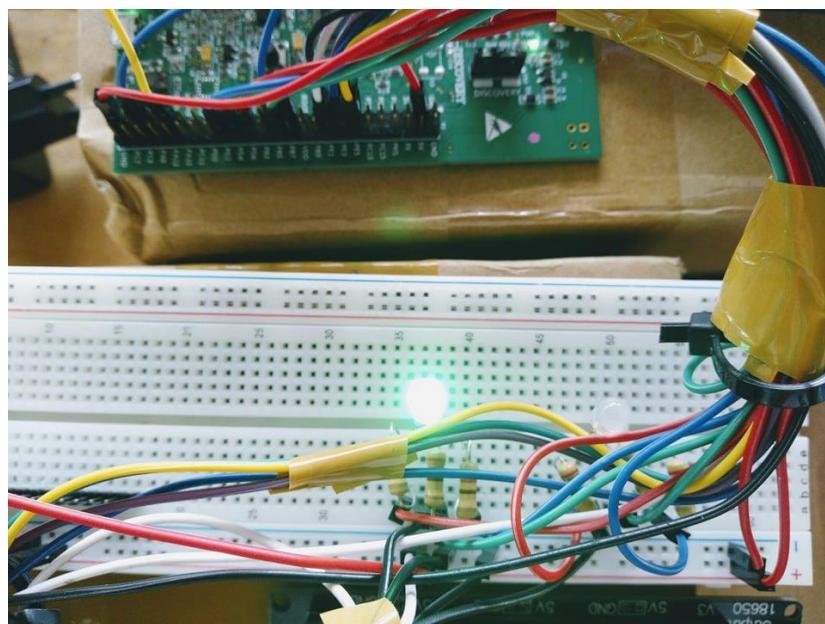
✚ Kết quả hiển thị trên LCD và đèn LED báo:

- Khi nhận dạng đúng mật khẩu, mật khẩu nhóm em thiết lập là số “CHÍN”



Hình 5. 14 Kết quả LCD khi nhận dạng đúng mật khẩu

- Khi nhận dạng đúng mật khẩu và đặc trưng giọng nói, đèn bên trái sẽ báo màu xanh, còn nếu không đúng đặc trưng giọng sẽ báo màu đỏ



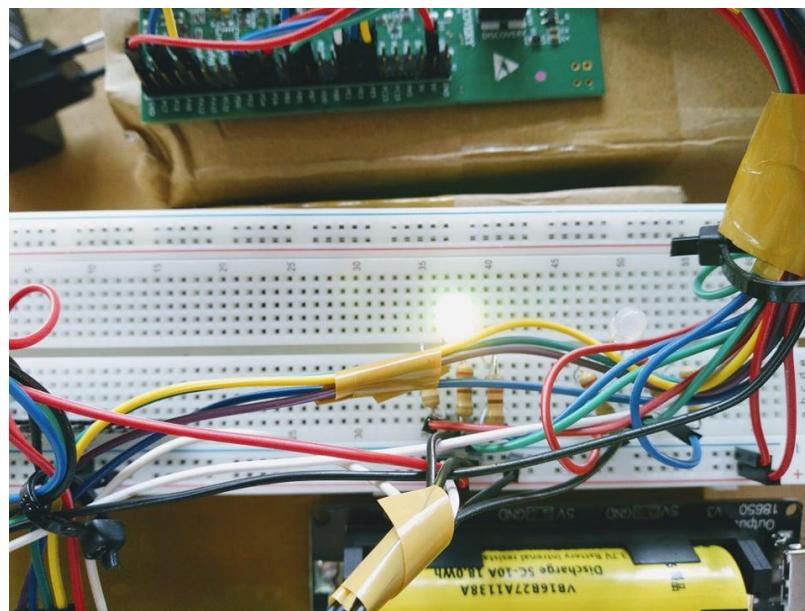
Hình 5. 15 Kết quả LED trái khi nhận dạng đúng giọng nói

- Khi nhận dạng sai mật khẩu nhưng đúng chính chủ, ở đây người đọc đã đọc số “KHÔNG”



Hình 5. 16 Kết quả LCD khi nhận dạng sai mật khẩu

- Khi nhận dạng đúng đặc trưng giọng chính chủ nhưng không đúng mật khẩu, đèn bên trái sẽ báo màu vàng.



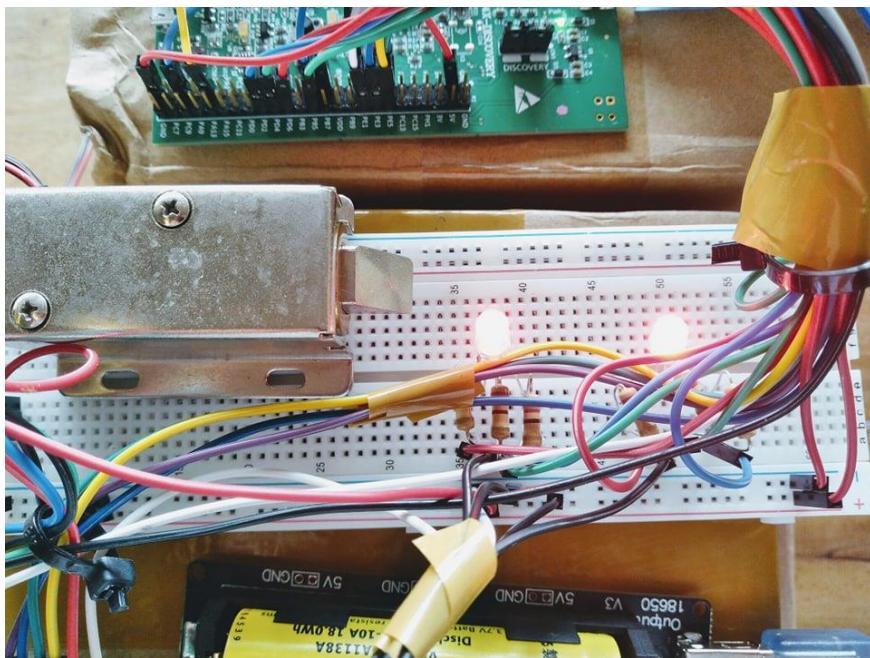
Hình 5. 17 Kết quả LED trái khi nhận dạng sai mật khẩu

- Khi nhận dạng sai chính chủ:



Hình 5. 18 Kết quả LCD khi nhận dạng sai chính chủ

- Khi nhận dạng sai khuôn mặt và giọng nói chính chủ, cả hai đèn sẽ hiển thị màu đỏ



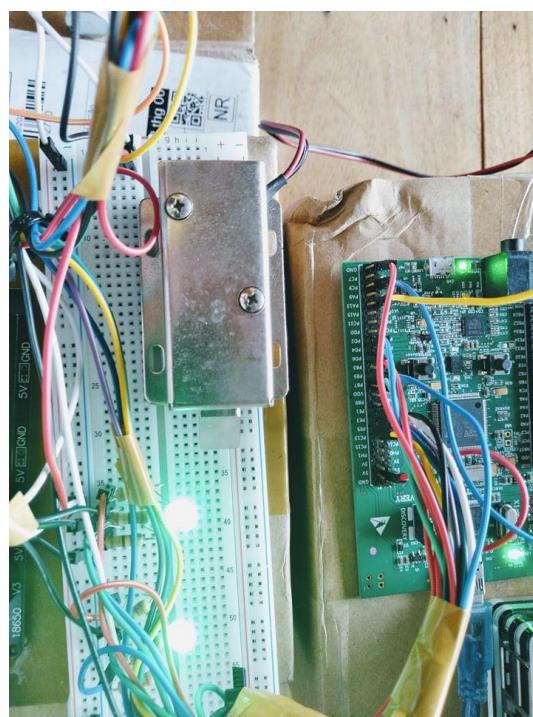
Hình 5. 19 Kết quả hai LED khi nhận dạng sai chính chủ

- Khi nhận dạng đúng giọng nói và gương mặt chính chủ:



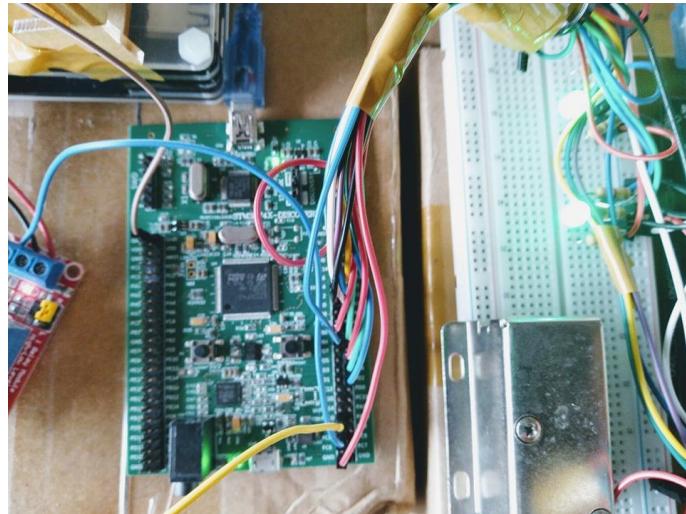
Hình 5. 20 Kết quả LCD khi nhận dạng đúng chính chủ

- Khi nhận dạng đúng khuôn mặt chính chủ, đèn bên phải sẽ báo màu xanh



Hình 5. 21 Kết quả hai LED khi nhận dạng đúng chính chủ

- Khi nhận dạng đúng cả khuôn mặt và giọng nói, mật khẩu của chính chủ, thì khóa cửa điện sẽ tự động thụt vào để mở cửa trong một thời được định sẵn, sau đó sẽ thụt ra lại. Lúc này, cả hai đèn sẽ hiển thị màu xanh.



Hình 5. 22 Kết quả khóa chốt mở khi nhận dạng đúng chính chủ

└ Kết quả nhận diện và nhận dạng khuôn mặt được hiển thị trên Raspberry Pi (qua màn hình laptop):

- Nhận dạng khuôn mặt được ghi lại bằng camera Pi V2 trong 6 frame ảnh đầu tiên. Trong 6 frame ảnh này, nếu đúng từ 4 frame trở lên (đúng chính chủ) thì Raspberry Pi sẽ gửi tín hiệu FACEOK thông qua UART đến STM32, ngược lại thì sẽ gửi FERROR.
- Một số kết quả khi nhận dạng khuôn mặt được hiển thị trên màn hình máy tính



Hình 5. 23 Kết quả nhận dạng khuôn mặt chính chủ



Hình 5. 24 Kết quả nhận dạng khuôn mặt người lạ

Đánh giá và nhận xét kết quả phần cứng: phần cứng tương đối dễ sử dụng, đáp ứng thời gian thực tương đối tốt, tiêu tốn ít điện năng, bền theo thời gian. Tuy nhiên, kèm theo đó là những hạn chế, thứ nhất là hệ thống chưa được tối ưu hóa, chưa thân thiện người dùng. Thứ hai, hệ thống có độ chính xác thấp hơn so với phần mềm. Thứ ba, hệ thống nhận dạng giọng nói vẫn chưa hoàn thành thuật toán nhận dạng sáu số mật khẩu như phần mềm đã làm được.

## 5.4 Đánh giá độ chính xác của mô hình

### 5.4.1 Thực hiện thuật toán trên phần mềm MATLAB

Đối với nhận dạng giọng nói, nhóm em đã thu âm 10 từ vựng từ số 0 đến số 9, mỗi từ phát âm 500 lần do chính chủ phát âm. Trong 500 lần phát âm, nhóm em sẽ đếm số lần vượt ngưỡng của từng số (không phải chính chủ) và đếm số lần nhận dạng số đó sai. Sau đó thu được hai bảng đánh giá sau đây

Từ vựng	Nhận dạng đúng (lần)	Nhận dạng sai (lần)	Hiệu quả (%)
Không	430	70	86
Một	420	80	84
Hai	411	89	82.2
Ba	453	47	90.6
Bốn	401	99	80.2
Năm	439	61	87.8
Sáu	467	33	93.4
Bảy	474	26	94.8
Tám	402	98	80.4
Chín	432	68	86.4
<b>Trung bình</b>	<b>86.58</b>	<b>13.42</b>	<b>86.58</b>

Bảng 5. 4 Độ chính xác thuật toán nhận dạng người nói trên MATLAB

Từ vựng	Nhận dạng đúng (lần)	Nhận dạng sai (lần)	Hiệu quả (%)	Từ hay nhầm lẫn nhất
Không	462	38	92.4	Hai
Một	451	49	90.2	Ba
Hai	443	57	88.6	Năm
Ba	477	23	95.4	Không
Bốn	418	82	83.6	Bảy
Năm	454	46	90.8	Tám
Sáu	476	24	95.2	Bốn
Bảy	482	18	96.4	Ba
Tám	413	87	82.6	Năm
Chín	454	46	90.8	Sáu
<b>Trung bình</b>	<b>90.6</b>	<b>9.4</b>	<b>90.6</b>	

Bảng 5. 5 Độ chính xác của thuật toán nhận dạng mật khẩu số trên MATLAB

- Đối với nhận dạng khuôn mặt, nhóm em chụp ảnh chính chủ 500 lần từ camera trên máy tính để thử nghiệm xác suất chính xác của mô hình

Nhận dạng đúng (lần)	Nhận dạng sai (lần)	Hiệu quả (%)
435	65	87%

Bảng 5. 6 Độ chính xác của thuật toán nhận dạng khuôn mặt trên MATLAB

### 5.4.2 Thực hiện thuật toán trên phần cứng

Đối với nhận dạng giọng nói, nhóm em đã thu âm 10 từ vựng từ số 0 đến số 9, mỗi từ phát âm 500 lần do chính chủ phát âm. Trong 500 lần phát âm, nhóm em sẽ đếm số lần vượt ngưỡng của từng số (không phải chính chủ) và đếm số lần nhận dạng số đó sai. Sau đó thu được hai bảng đánh giá sau đây

Từ vựng	Nhận dạng đúng (lần)	Nhận dạng sai (lần)	Hiệu quả (%)
Không	384	116	76.8
Một	337	163	67.4
Hai	326	174	65.2
Ba	406	94	81.2
Bốn	315	185	63
Năm	368	132	73.6
Sáu	388	112	77.6
Bảy	392	108	78.4
Tám	344	156	68.8
Chín	384	116	76.8
<b>Trung bình</b>	<b>72.88</b>	<b>27.12</b>	<b>72.88</b>

Bảng 5. 7 Độ chính xác nhận dạng người nói trên phần cứng

### 5.4.3 Đánh giá hiệu quả nhận dạng của phần mềm và phần cứng

Độ chính xác của phần mềm sử dụng MATLAB vẫn chưa cao nhưng tốt hơn nhiều so với phần cứng. Vì phần mềm được mô phỏng trên MATLAB với số lần lấy mẫu cao, tập dữ liệu đa dạng. Tuy nhiên, giải thuật đơn giản nên hiệu quả nhận dạng cũng không quá cao.

Độ chính xác của phần cứng tương đối thấp. Vì khi thực hiện giải thuật trên phần cứng khó triển khai giải thuật vì ngôn ngữ C không mạnh ở xử lý ma trận mặc dù giải thuật đơn giản, tài nguyên của phần cứng thấp nên số lần lấy mẫu dữ liệu không cao (3 lần). Ngoài

ra ở vi điều khiển sử dụng micro chưa loại được nhiều nên cũng cho ra kết quả không tốt, hay nhận dạng sai mật khẩu.

Từ vựng	Nhận dạng đúng (lần)	Nhận dạng sai (lần)	Hiệu quả (%)	Từ hay nhầm lẫn nhất
Không	435	65	87	Năm
Một	363	137	72.6	Ba
Hai	338	162	67.6	Năm
Ba	446	54	89.2	Sáu
Bốn	332	168	66.4	Một
Năm	418	82	83.6	Tám
Sáu	369	131	73.8	Không
Bảy	462	38	92.4	Ba
Tám	394	106	78.8	Năm
Chín	436	64	87.2	Sáu
<b>Trung bình</b>	<b>79.86</b>	<b>20.14</b>	<b>79.86</b>	

Bảng 5. 8 Độ chính xác nhận dạng mật khẩu số trên phần cứng

- Ở project lần này, nhóm em sử dụng từ “CHÍN” để làm mật khẩu nhận dạng cho phần nhận dạng giọng nói.
- Nhận dạng khuôn mặt chính chủ trong 500 lần quay, được thực hiện trên phần cứng

Nhận dạng đúng (lần)	Nhận dạng sai (lần)	Hiệu quả (%)
422	78	84.4%

Bảng 5. 9 Độ chính xác nhận dạng khuôn mặt trên phần cứng

## 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

- Những nội dung chính đã được giải quyết:
  - Nghiên cứu tổng quan bài toán nhận dạng giọng nói và khuôn mặt
  - Nghiên cứu các phương pháp trích chọn đặc trưng giọng nói, chi tiết phương pháp trích chọn đặc trưng MFCC
  - Nghiên cứu mô hình VQ ứng dụng trong xác thực người nói và mật khẩu dạng số của người nói
  - Nghiên cứu tổng quan bài toán nhận dạng khuôn mặt
  - Nghiên cứu các phương pháp trích chọn đặc trưng khuôn mặt, chi tiết phương pháp nén dữ liệu PCA
  - Nghiên cứu phương pháp PCA để nhận dạng khuôn mặt một cách tổng quát
- Tìm hiểu và nắm được các nội dung sau:
  - Lý thuyết về tiếng nói, đặc trưng ngữ âm tiếng Việt
  - Kiến thức về xử lý số tín hiệu: DFT, FFT, bộ lọc, ADC,...
  - Các phương pháp trích đặc trưng. Chủ yếu là MFCC
  - Lượng tử hóa vector trong việc nén dữ liệu và nhận dạng tiếng nói
  - Phương pháp nén dữ liệu PCA để huấn luyện và nhận dạng khuôn mặt một cách tổng quan
  - Kiến thức về máy tính nhúng, vi điều khiển
  - Kiến thức lập trình nâng cao

## 6.2 Hướng phát triển

- Nghiên cứu tối ưu hóa phần cứng bằng cách sử dụng vi điều khiển có tài nguyên tốt hơn ví dụ như dòng vi điều khiển STM32F7, không sử dụng máy tính nhúng để tăng độ bảo mật cho hệ thống
- Thiết kế và làm mạch in cho hệ thống để trông đẹp mắt và nhỏ gọn hơn (vì lý do dịch bệnh bất khả kháng nên nhóm không thể ra ngoài đặt mạch hay mua đồ để làm mạch in được)
- Xây dựng mô hình mô phỏng lại cửa ra vào có chốt khóa điện tử
- Về phần thuật toán, tiếp tục nghiên cứu các thuật toán phức tạp hơn và có độ chính xác cao hơn như HOG và SVM cho nhận diện khuôn mặt và LBP cho nhận dạng khuôn mặt; thuật toán Hidden Markov Model, Neural Network, GM,..., ứng dụng cho bài toán nhận dạng người nói và xác thực mật khẩu số của tiếng nói nhằm tăng tính bảo mật cho hệ thống
- Thu thập số lượng lớn hơn dữ liệu người dùng chính chủ và người lạ để tiến hành khiêm thử, điều chỉnh các tham số của hệ thống cho chính xác

## 7. TÀI LIỆU THAM KHẢO

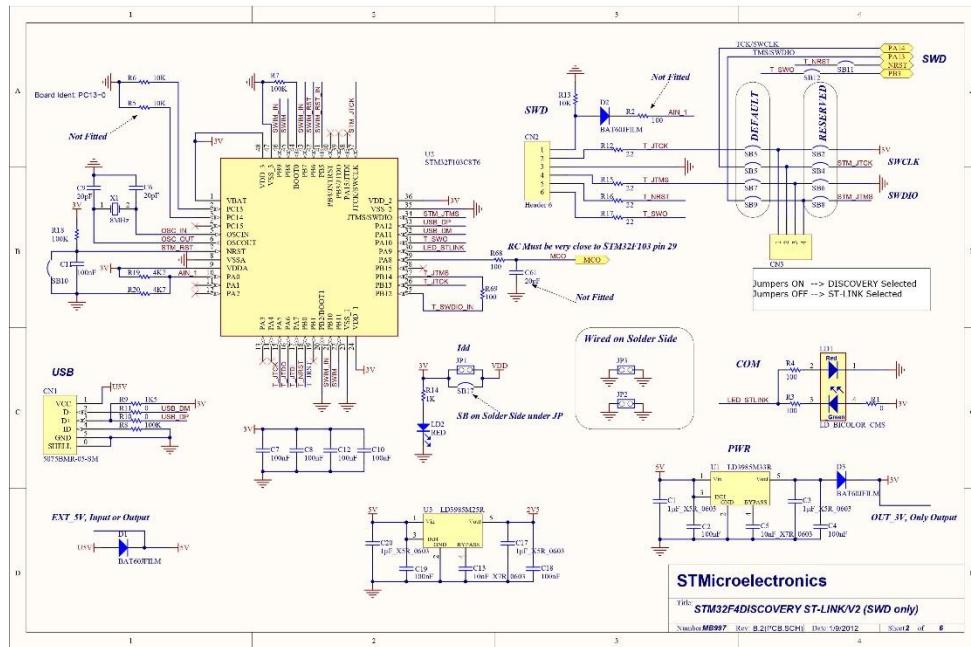
- [1] M. Q. Wang and S. J. Young, "Speech recognition using hidden Markov model decomposition and a general background speech model", *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Francisco, CA, USA, 1992, pp. 253-256 vol.1, doi: 10.1109/ICASSP.1992.225924. Altera Corp., "SDRAM Controller for Altera's DE2/ DE1 boards", www.altera.com
- [2] Bachu R., Kopparthi S., Adapa B., Barkana B. (2010) Voiced/Unvoiced Decision for Speech Signals Based on Zero-Crossing Rate and Energy. In: Elleithy K. (eds) Advanced Techniques in Computing Sciences and Software Engineering. Springer, Dordrecht. [https://doi.org/10.1007/978-90-481-3660-5\\_47](https://doi.org/10.1007/978-90-481-3660-5_47)
- [3] M. Sahidullah and G. Saha, "A Novel Windowing Technique for Efficient Computation of MFCC for Speaker Recognition," in *IEEE Signal Processing Letters*, vol. 20, no. 2, pp. 149-152, Feb. 2013, doi: 10.1109/LSP.2012.2235067
- [4] Mel Frequency Cepstral Coefficient (MFCC) tutorial Practical Cryptography
- [5] Beigi, Homayoon, "Fundamentals of Speaker Recognition"
- [6] H. B. Kekre and V. Kulkarni, "Closed set and open set Speaker Identification using amplitude distribution of different Transforms," *2013 International Conference on Advances in Technology and Engineering (ICATE)*, Mumbai, 2013, pp. 1-8, doi: 10.1109/ICAdTE.2013.6524764.
- [7] Z. Weng, L. Li and D. Guo, "Speaker recognition using weighted dynamic MFCC based on GMM," *2010 International Conference on Anti-Counterfeiting, Security and Identification*, Chengdu, 2010, pp. 285-288, doi: 10.1109/ICASID.2010.5551341.
- [8] C. Bernal-Ruiz, F. E. Garcia-Tapias, B. Martin-del-Brio, A. Bono-Nuez and N. J. Medrano-Marques, "Microcontroller implementation of a voice command recognition system for human-machine interface in embedded systems," 2005 IEEE Conference on Emerging Technologies and Factory Automation, Catania, 2005, pp. 5 pp.-591, doi: 10.1109/ETFA.2005.1612576.
- [9] B. Gerazov, V. Pop-Dimitrijoska, Z. Ivanovski and G. Apostolovska, "Use of Gaussian Mixture Models in Macedonian forensic speaker identification," 2012 20th Telecommunications Forum (TELFOR), Belgrade, 2012, pp. 724-727, doi: 10.1109/TELFOR.2012.6419310.

- [10] A. Zulfiqar, A. Muhammad and M. E. A.M., "A Speaker Identification System Using MFCC Features with VQ Technique," *2009 Third International Symposium on Intelligent Information Technology Application*, Shanghai, 2009, pp. 115-118, doi: 10.1109/IITA.2009.420.
- [11] A. Ashar, M. S. Bhatti and U. Mushtaq, "Speaker Identification Using a Hybrid CNN-MFCC Approach," *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, Karachi, Pakistan, 2020, pp. 1-4, doi: 10.1109/ICETST49965.2020.9080730.
- [12] Trung Thành Nguyễn, “Kiến thức nền tảng xử lý tiếng nói - Speech Processing”, 2020, [www.viblo.asia/](http://www.viblo.asia/)
- [13] Trung Thành Nguyễn, “Feature Extraction - MFCC cho xử lý tiếng nói”, 2020, [www.viblo.asia/](http://www.viblo.asia/)
- [14] Datai@SG, “Principal Component Analysis (PCA)”, 2020, [dtaisg.org/](http://dtaisg.org/)
- [15] Tiep Vu Huu, “Principal Component Analysis (phần ½)”, 2017, [machinelearningcoban.com/](http://machinelearningcoban.com/)
- [16] Nguyễn Thu Hà, “Cách tìm địa chỉ IP Raspberry Pi”, 2020, [quantrimang.com/](http://quantrimang.com/)

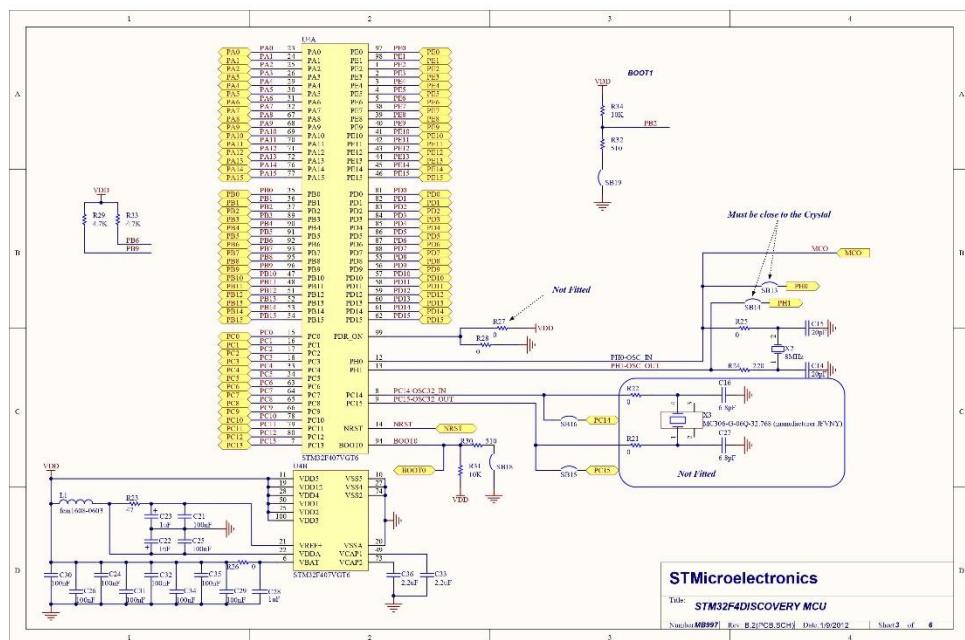
## 8. PHỤ LỤC

- Chi tiết sơ đồ nguyên lý STM32-DISCO:

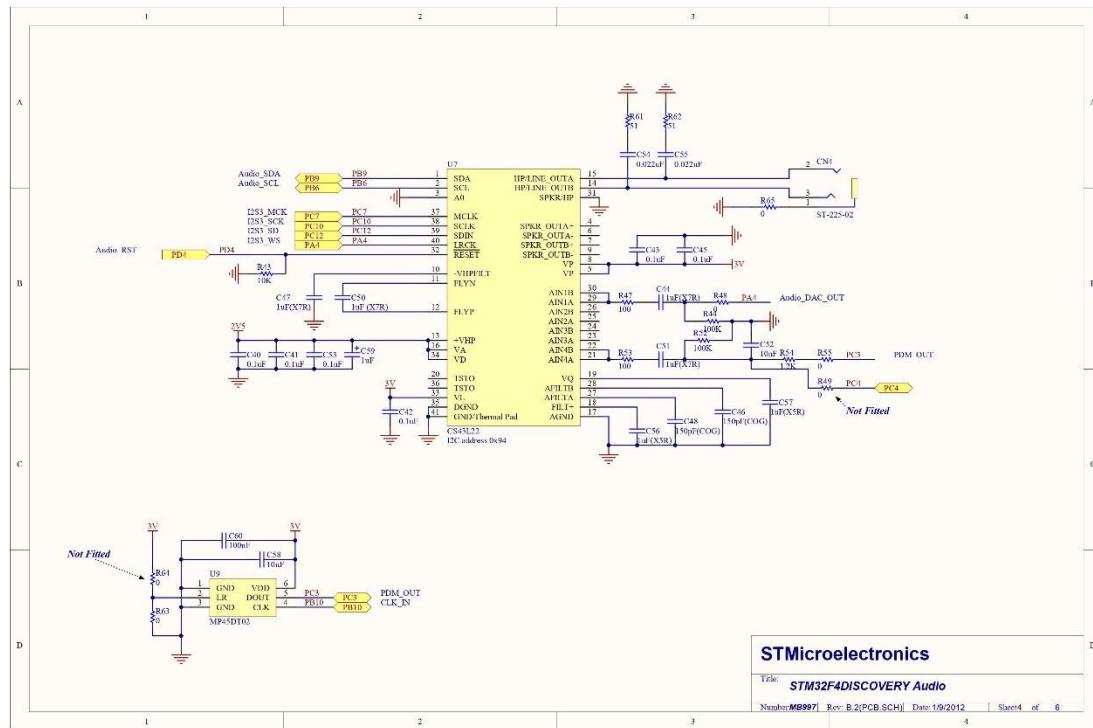
- ST-LINK:



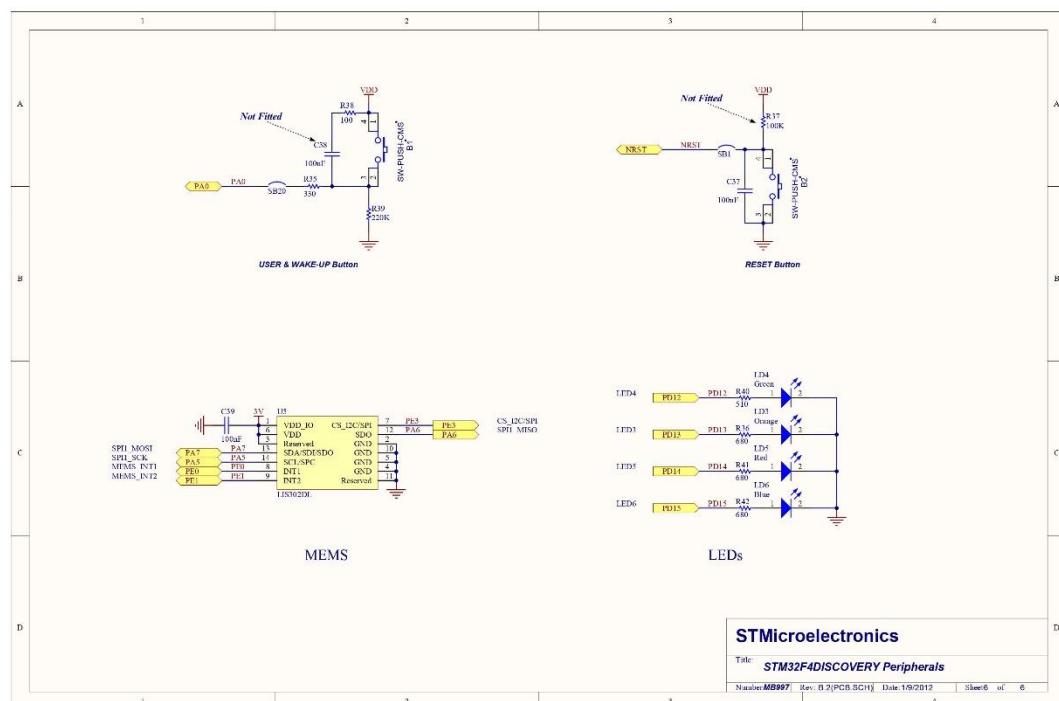
- MCU:



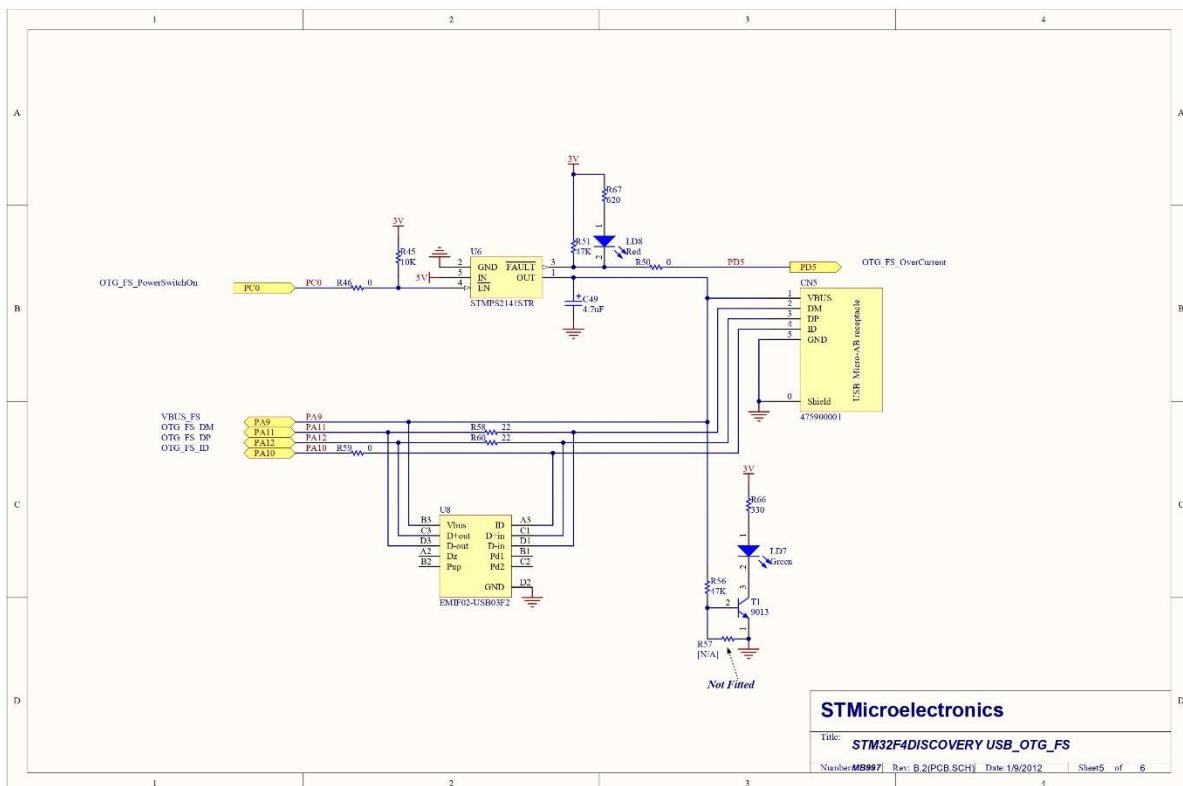
○ Audio:



○ Ngoại vi:

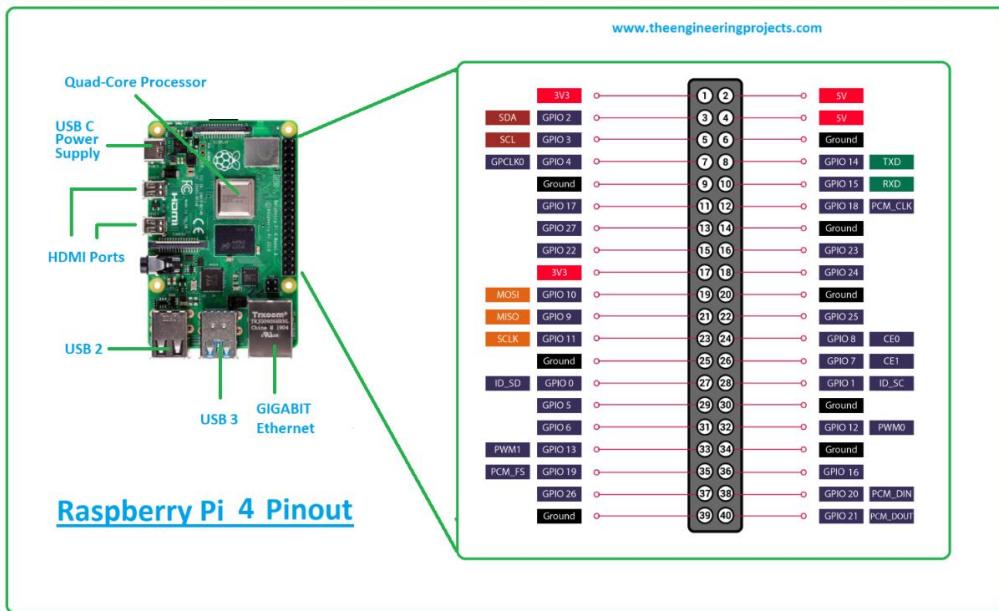


- USB OTG FS:



- Chi tiết sơ đồ nguyên lý Raspberry Pi 4:

o Sơ đồ chân:



o Sơ đồ nguyên lý:

