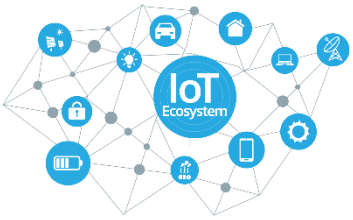
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN
PBL5 - KỸ THUẬT MÁY TÍNH



NHÀ THÔNG MINH KẾT HỢP ĐỊNH DANH PHÂN QUYỀN GIỌNG NÓI

GIẢNG VIÊN HƯỚNG DẪN: TS. Ninh Khánh Duy

STT NHÓM: 01 HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN ĐỒ ÁN
Trần Đức Trí	21Nh11
Phạm Nguyễn Anh Phát	21Nh11
Lê Anh Tuấn	21Nh11
Lê Văn Tiến Đạt	21Nh11

ĐÀ NẴNG, 06/2024

LỜI CẢM ƠN

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô khoa Công nghệ thông tin, trường Đại học Bách khoa – Đại học Đà Nẵng đã tạo điều kiện và giúp đỡ chúng em trong quá trình thực hiện Dự án Kỹ Thuật Máy Tính. Đặc biệt, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến thầy Ninh Khánh Duy, người đã không ngừng định hướng, đồng hành và hướng dẫn chúng em từng bước. Trong quá trình thực hiện, chắc chắn không tránh khỏi những thiếu sót và vẫn còn những khía cạnh cần được cải thiện. Chúng em mong nhận được sự góp ý chân thành từ phía quý thầy cô và các bạn để sản phẩm của chúng em ngày càng hoàn thiện hơn. Chúng em xin chân thành cảm ơn.

TÓM TẮT ĐỒ ÁN

Hiện nay, nhu cầu điều khiển thiết bị trong gia đình từ xa ngày càng tăng cao. Thay vì thao tác thủ công, người dùng có thể điều khiển bằng giọng nói hoặc từ xa qua điện thoại, máy tính bảng. Tuy nhiên, việc xác thực người nói và phân quyền vẫn còn là thách thức lớn.

Đồ án của nhóm chúng em là hệ thống nhà thông minh kết hợp định danh và phân quyền giọng nói. Chúng em đã sử dụng mô hình trí tuệ nhân tạo để xác định người nói, Raspberry Pi 4 làm bộ điều khiển trung tâm, các thiết bị như loa bluetooth, ESP32 và màn hình TFT để điều khiển và hiển thị trạng thái các thiết bị trong gia đình.

Sau quá trình nghiên cứu và thử nghiệm, hệ thống đã hoạt động mượt mà và đem lại kết quả tốt. Chúng em sẽ tiếp tục phát triển để giải quyết những thách thức còn tồn đọng và nâng cao tính ổn định và hoàn thiện của sản phẩm.

Đồ án của nhóm chúng em hướng đến mục tiêu mang lại trải nghiệm nhà thông minh an toàn và tiện lợi, đồng thời bảo vệ quyền riêng tư và bảo mật thông tin cho người dùng.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên	Các nhiệm vụ	Đánh giá
Trần Đức Trí	Thiết kế mô hình IOT	Đã Hoàn Thành
	Nhận diện giọng nói bằng MFCC + LSTM	Đã Hoàn Thành
	Xây dựng Website quản lý thành viên	Đã Hoàn Thành
	Xây dựng thuật toán điều khiển phần cứng	Đã Hoàn Thành
	Triển khai mô hình AI lên phần cứng	Đã Hoàn Thành
	Lắp ráp phần cứng	Đã Hoàn Thành
Phạm Nguyễn Anh Phát	Thiết kế mô hình IOT	Đã Hoàn Thành
	Nhận diện giọng nói bằng MFCC + CNN	Đã Hoàn Thành
	Xây dựng Website quản lý thành viên	Đã Hoàn Thành
	Xây dựng thuật toán điều khiển phần cứng	Đã Hoàn Thành
	Triển khai mô hình AI lên phần cứng	Đã Hoàn Thành
	Lắp ráp phần cứng	Đã Hoàn Thành
Lê Anh Tuấn	Thiết kế mô hình IOT	Đã Hoàn Thành
	Nhận diện câu nói bằng Tranformer	Đã Hoàn Thành
	Nhận diện câu nói bằng Wave2vec	Đã Hoàn Thành
	Triển khai mô hình AI lên phần cứng	Đã Hoàn Thành
	Lắp ráp phần cứng	Đã Hoàn Thành
Lê Văn Tiến Đạt	Thiết kế mô hình IOT	Đã Hoàn Thành
	Gán nhãn từ loại Pos tagging	Đã Hoàn Thành
	Xây dựng biểu thức chính quy	Đã Hoàn Thành
	Triển khai giao diện màn hình	Đã Hoàn Thành
	Triển khai kết nối không dây	Đã Hoàn Thành
	Lắp ráp phần cứng	Đã Hoàn Thành

MỤC LỤC

DANH SÁCH HÌNH ẢNH

DANH SÁCH BẢNG BIỂU

1. Giới thiệu

Tổng quan

Hiện nay, các hệ thống nhà thông minh kết hợp định danh phân quyền đang phát triển mạnh, từ các nền tảng như Amazon Alexa, Google Home đến Bkav SmartHome. Những hệ thống này tích hợp điều khiển thiết bị và quản lý truy cập, cho phép phân quyền theo vai trò. Tuy nhiên, những hệ thống này có nhược điểm lớn nhất là về mặt chi phí. Cụ thể, chi phí thiết lập và duy trì cao do yêu cầu phần cứng chuyên dụng như khóa cửa thông minh, cảm biến và bộ điều khiển trung tâm, vốn đắt đỏ và khó tích hợp với các thiết bị từ nhà cung cấp khác. Phần mềm điều khiển cũng đòi hỏi phải phức tạp và cập nhật thường xuyên, dẫn đến chi phí phát triển và bảo trì tăng. Việc cấu hình hệ thống và quản lý phân quyền cũng đòi hỏi kỹ năng kỹ thuật cao, gây khó khăn cho người dùng không chuyên và tăng chi phí cho hỗ trợ kỹ thuật. Hơn nữa, bảo trì định kỳ và nâng cấp hệ thống để đảm bảo tính tương thích và an toàn làm gia tăng thêm chi phí. Vậy nên nhóm thực hiện đề tài này để làm tốt hơn cũng như với chi phí thấp hơn.

Các vấn đề cần giải quyết

- Cần có các thiết bị phần cứng để thu thập dữ liệu âm thanh của người dùng
- Định danh giọng nói người dùng
- Xác định lời nói của người dùng, phân tích hành động và đối tượng tác động
- Cần một cách để người dùng có thể kiểm tra trạng thái các thiết bị trong nhà cũng như biết được kết quả của lời nói vừa rồi

Đề xuất các giải pháp tổng quan

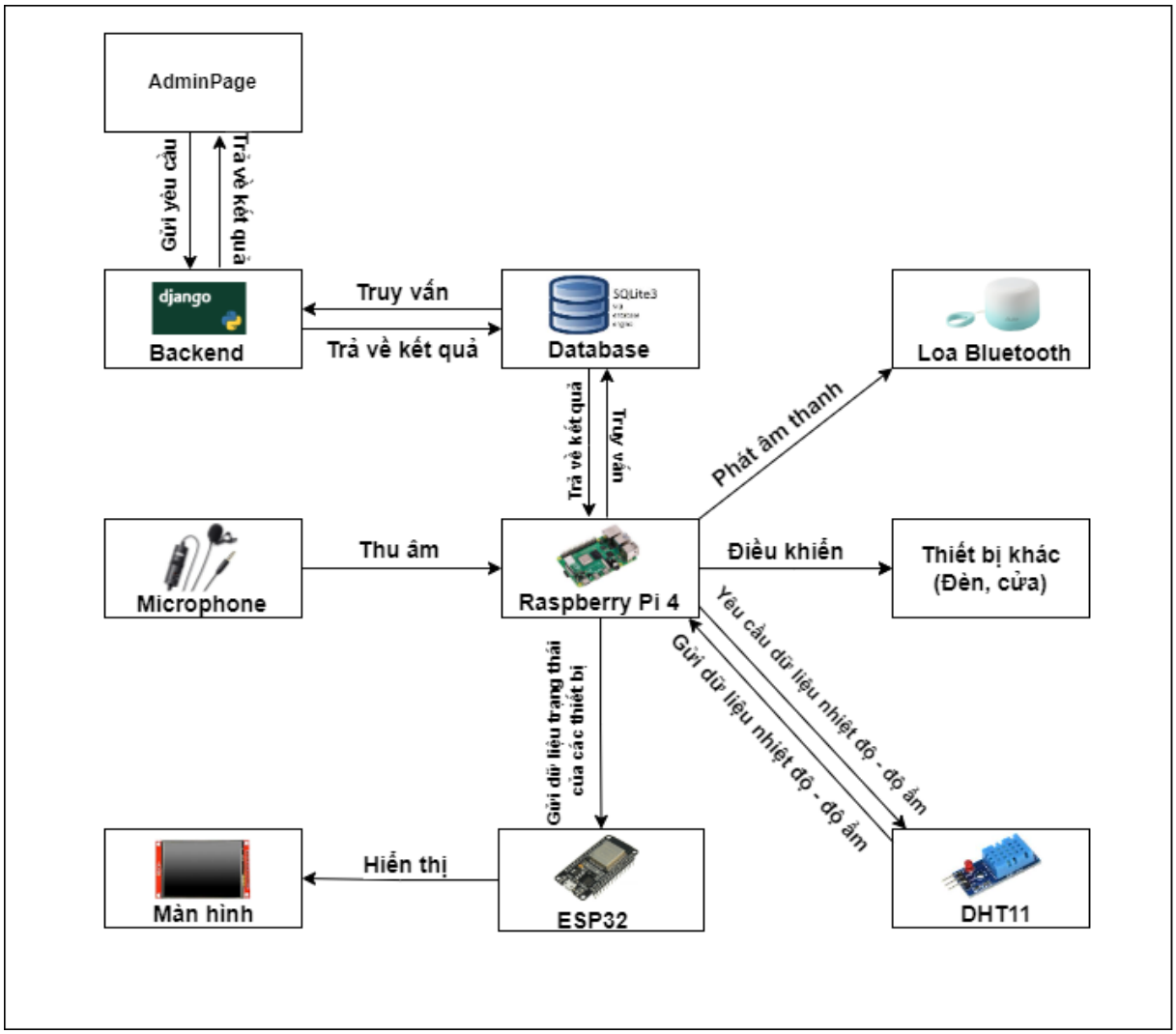
Bảng 1. Bảng đề xuất các giải pháp tổng quan

Vấn đề	Giải pháp đề xuất
Phần cứng	Raspberry Pi 4 Microphone BOYA M1 Module truyền phát ESP32 Màn hình TFT LCD
Định danh giọng nói	Xây dựng và huấn luyện model nhận diện giọng nói Thử nghiệm với các Model LSTM, CNN
Chuyển đổi lời nói thành văn bản	Thử nghiệm với các mô hình: Open AI Whisper, VinAI PhoWhisper,.. Thử nghiệm với Google Cloud Speech API
Ứng dụng	Xây dựng Website quản lý người dùng Kiểm tra kết quả phân quyền Kết nối với ESP32 và Loa Bluetooth để gởi kết quả cũng như trạng thái xử lý
Server	Viết bằng Django

2. Giải pháp

2.1. Giải pháp phần cứng và truyền thông

2.1.1. Sơ đồ tổng quan hệ thống



Hình 1. Sơ đồ tổng quan hệ thống IOT

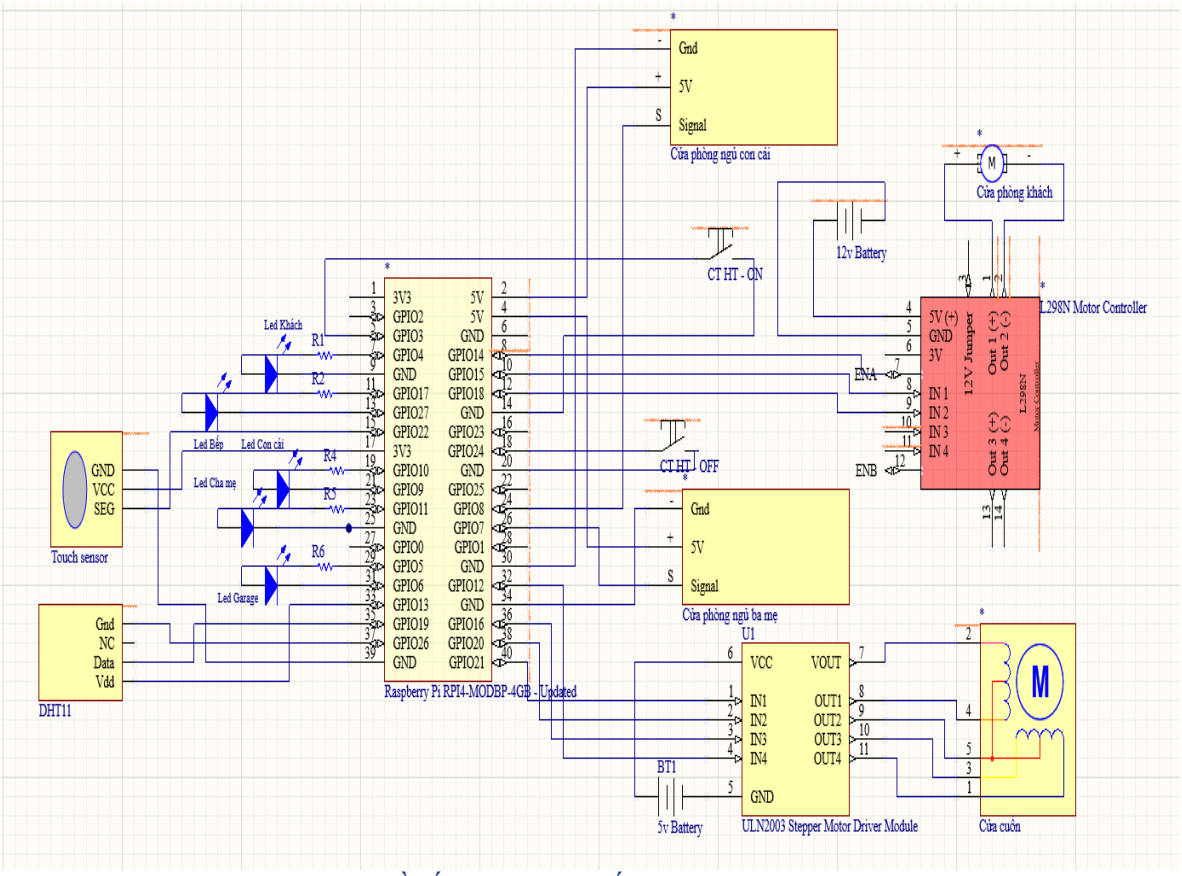
Hệ thống bao gồm Raspberry Pi 4 làm trung tâm, Microphone sử dụng để ghi âm và truyền dữ liệu ghi âm tới Raspberry Pi 4, sau đó trích xuất đặc trưng rồi so sánh với đặc trưng gốc lấy từ Database Server Django để gởi kết quả phân quyền về cho Raspberry Pi 4, các thiết bị như Đèn, DC Motor, Servo, Stepper motor, DHT11 thực hiện theo yêu cầu của Raspberry Pi 4, các thiết bị như Loa Bluetooth, ESP32 nhận yêu cầu hiển thị trực quan kết quả phân quyền cũng như trạng thái của thiết bị từ Rasperry Pi gởi về, với ESP32 là trạng thái của các thiết bị hiển thị thông qua màn hình TFT, Loa Bluetooth phát âm thanh kết quả phân quyền.

2.1.2. Sơ đồ kết nối các linh kiện phần cứng

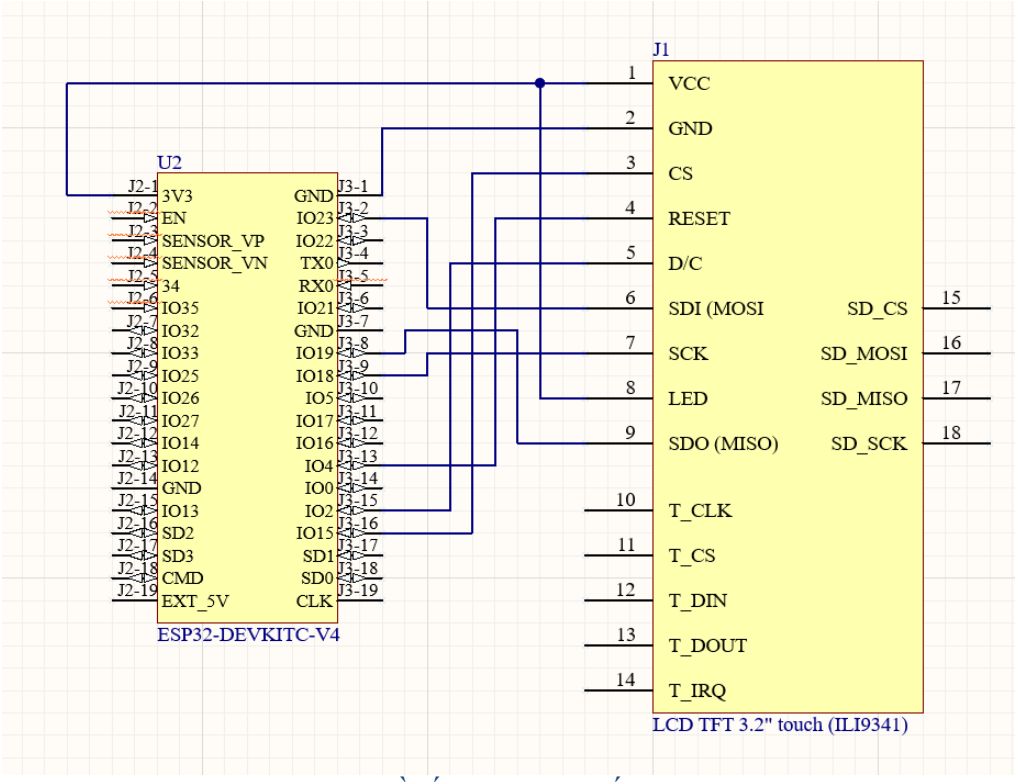
Sơ đồ lắp ráp mạch:

Các thiết bị phần cứng lắp đặt:

- Máy tính nhúng Raspberry Pi 4
- Động cơ Servo
- Module L298N
- Động cơ DC Motor
- Cảm biến chạm
- Công tắc hành trình
- Cảm biến đo độ ẩm nhiệt độ DHT11
- Động cơ bước 28BYJ48-5V
- Driver Điều Khiển Động Cơ Bước ULN2003
- Module thu phát Wifi BLE ESP32 CP2102 NodeMCU LuaNode32
- Màn Hình Cảm Ứng LCD TFT Touch Screen 3.2 Inch ILI9341 SPI Interface



Hình 2. Sơ đồ lắp mạch hệ thống với Rasperry Pi 4



Hình 3. Sơ đồ lắp mạch hệ thống với ESP32

- **Raspberry Pi 4 Model B**

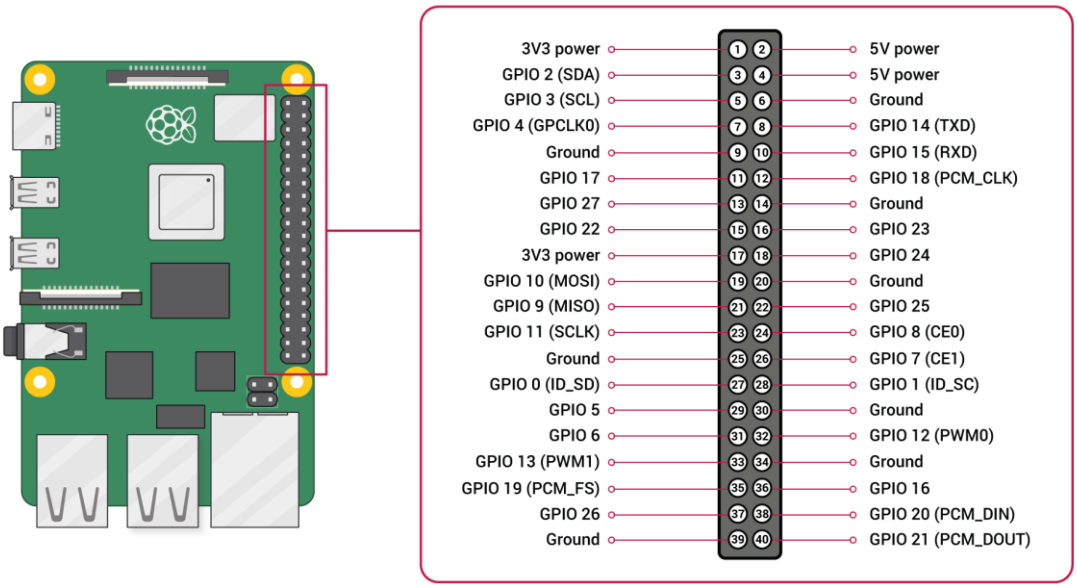
Hình 4. Raspberry Pi 4 Model B

Raspberry Pi 4 Model B là phiên bản mới nhất trong dòng máy tính nhỏ gọn nổi tiếng Raspberry Pi, được thiết kế để mang lại hiệu năng cao hơn và hỗ trợ nhiều tính năng tiên tiến cho các ứng dụng giáo dục, nghiên cứu, và phát triển nhúng. Với khả năng xử lý mạnh mẽ hơn, bộ nhớ RAM đa dạng, và khả năng kết nối phong phú, Raspberry Pi 4 Model B mở rộng đáng kể tiềm năng sáng tạo cho người dùng.

Các thông số kỹ thuật chính của máy tính nhúng:

- Bộ vi xử lý: Broadcom BCM2711, lõi tứ ARM Cortex-A72 (ARM v8) 64-bit, tốc độ 1.5 GHz.
- Bộ nhớ RAM: Tùy chọn 2 GB, 4 GB, hoặc 8 GB LPDDR4-3200 SDRAM.
- Đồ họa: Broadcom VideoCore VI, hỗ trợ OpenGL ES 3.1, 4Kp60 HEVC video decode.
- Lưu trữ: Khe cắm thẻ microSD (hỗ trợ UHS-I), USB 3.0, USB 2.0.
- Cổng USB: 2 x USB 3.0, 2 x USB 2.0.
- Kết nối mạng: Ethernet Gigabit, hỗ trợ PoE (yêu cầu bổ sung HAT PoE).
- Không dây: Wi-Fi 802.11 b/g/n/ac (2.4 GHz và 5.0 GHz), Bluetooth 5.0, BLE.
- Cổng màn hình: 2 x micro-HDMI, hỗ trợ độ phân giải lên đến 4Kp60.
- Cổng âm thanh: Jack âm thanh 3.5 mm (âm thanh và video composite), hỗ trợ HDMI audio.
- Nguồn điện: Cổng USB-C, cung cấp nguồn 5V/3A. [1]

Giao thức truyền thông và các chân giao tiếp:

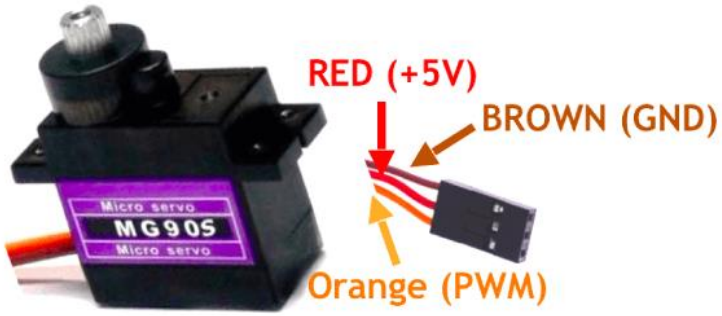


Hình 5. Sơ đồ chân của Rasperry Pi 4

- GPIO: 40 chân GPIO tiêu chuẩn, cung cấp giao diện để kết nối với các thiết bị phần cứng bên ngoài như cảm biến, đèn LED, hoặc module giao tiếp.
- UART: Universal Asynchronous Receiver/Transmitter, dùng để giao tiếp nối tiếp với các thiết bị khác như mô-đun GPS hoặc module không dây.
- SPI: Serial Peripheral Interface, một giao thức truyền thông nối tiếp tốc độ cao dùng để kết nối với các thiết bị như màn hình OLED hoặc cảm biến.
- I2C: Inter-Integrated Circuit, một giao thức truyền thông nối tiếp cho phép kết nối với các thiết bị như bộ giải mã ADC, cảm biến nhiệt độ, hoặc EEPROM.
- Ethernet: Hỗ trợ truyền thông mạng có dây với tốc độ Gigabit Ethernet, giúp tăng tốc độ truyền dữ liệu và độ tin cậy khi kết nối mạng.
- USB: Các cổng USB 3.0 và USB 2.0 cho phép kết nối với nhiều loại thiết bị ngoại vi như ổ cứng ngoài, bàn phím, chuột, và thiết bị lưu trữ USB.

Động cơ Servo MG90S

Động cơ servo MG90S là một loại servo micro hiệu suất cao, nổi tiếng với kích thước nhỏ gọn và lực xoắn mạnh mẽ nhờ sử dụng các bánh răng kim loại. Được ưa chuộng trong các ứng dụng yêu cầu điều khiển vị trí và chuyển động chính xác, MG90S là lựa chọn phổ biến trong các mô hình RC (radio-controlled), robot, và các dự án DIY.



Hình 6. Động cơ Servo MG90S

Dưới đây là thông số kỹ thuật của động cơ Servo MG90S:

- Điện áp hoạt động: 4.8V - 6.0V.
- Tốc độ hoạt động:
  - 4.8V: 0.11 giây/60 độ.

  - 6.0V: 0.10 giây/60 độ.
- Lực xoắn:
  - 4.8V: 1.8 kg.cm (25 oz.in).
  - 6.0V: 2.2 kg.cm (31 oz.in).
- Góc quay: 180 độ (tối đa).
- Kích thước: 22.8 x 12.2 x 28.5 mm.
- Trọng lượng: 13.4 g.
- Loại bánh răng: Bánh răng kim loại.
- Đầu ra trục: 21 răng, đường kính trục: 4.8 mm.
- Loại động cơ: Động cơ lõi chổi than.

Giao tiếp điều khiển:

- **1 ms:** Góc quay 0 độ.
- **1.5 ms:** Góc quay 90 độ.
- **2 ms:** Góc quay 180 độ.

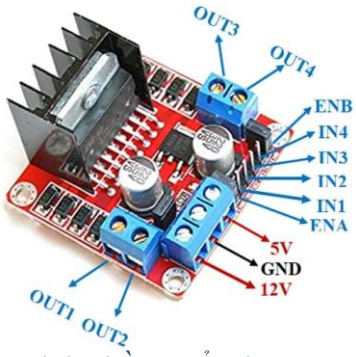
Chu kỳ xung PWM thường là 20 ms.

Kết nối và chân giao tiếp:

- Dây nguồn (Power - đỏ): Kết nối với nguồn cung cấp, điện áp từ 4.8V đến 6.0V.
- Dây nối đất (Ground - nâu): Kết nối đến điểm nối đất của hệ thống.
- Dây tín hiệu (Signal - cam): Nhận tín hiệu PWM từ bộ điều khiển để xác định vị trí của động cơ. [2]

Module điều khiển động cơ L298N

Module điều khiển động cơ L298N là một thiết bị thông dụng được sử dụng để điều khiển động cơ DC và động cơ bước trong các dự án điện tử và robot. Nó sử dụng chip L298N, một cầu H kép mạnh mẽ cho phép điều khiển độc lập hai động cơ DC hoặc một động cơ bước. Module này hỗ trợ nhiều loại động cơ và cung cấp khả năng điều khiển tốc độ và chiều quay một cách linh hoạt.



Hình 7. Module điều khiển động cơ L298N

Dưới đây là một vài thông số cơ bản về nó:

- **Thông số điện:**
  - Điện áp đầu vào cho động cơ (Vms): 5V đến 35V
  - Điện áp logic: 5V
  - Dòng điện đầu ra: Lên đến 2A mỗi cầu H (tổng cộng 4A với cả hai cầu H)
  - Dòng điện cực đại: 3A mỗi cầu H (trong thời gian ngắn)
- **Đặc điểm vật lý:**
  - Kích thước: Khoảng 43 mm x 43 mm x 27 mm
  - Trọng lượng: Khoảng 30 grams
- **Giao tiếp và điều khiển:**
  - Điều khiển tốc độ: PWM (Pulse Width Modulation)
  - Điều khiển chiều quay: Sử dụng các chân Input (IN1, IN2, IN3, IN4) để xác định chiều quay của động cơ
- **Chân điều khiển:**
  - ENA: Điều khiển tốc độ động cơ 1
  - ENB: Điều khiển tốc độ động cơ 2
  - IN1, IN2: Điều khiển chiều quay động cơ 1
  - IN3, IN4: Điều khiển chiều quay động cơ 2
- **Các thông số khác:**
  - Bảo vệ quá nhiệt: Tích hợp bảo vệ nhiệt trên chip L298N
  - Diode bảo vệ: Có các diode bảo vệ chống lại dòng ngược từ động cơ
  - Tản nhiệt: Tích hợp tản nhiệt để giảm nhiệt độ trong quá trình hoạt động [3]

- **Động Cơ DC Giảm Tốc Vàng**

Động cơ DC giảm tốc Vàng (Yellow DC Gear Motor) là một loại động cơ DC tích hợp hộp số giảm tốc, thường được sử dụng trong các ứng dụng yêu cầu lực xoắn cao và tốc độ quay thấp. Động cơ này phổ biến trong các dự án robot di động, xe điều khiển từ xa, và các hệ thống truyền động cơ bản do tính kinh tế, dễ sử dụng, và hiệu suất đáng tin cậy.

Hình 8. Động cơ DC giảm tốc vàng

Động cơ DC giảm tốc Vàng kết hợp một động cơ DC nhỏ với một hộp số giảm tốc tích hợp. Hộp số giảm tốc giúp giảm tốc độ quay của động cơ và tăng lực xoắn, cho phép động cơ cung cấp sức mạnh cần thiết để di chuyển các tải trọng nặng hoặc vượt qua các trở ngại lớn.

Nguyên lý hoạt động cơ bản của động cơ này dựa trên việc chuyển đổi năng lượng điện từ nguồn DC thành chuyển động quay của trục động cơ. Hộp số giảm tốc thay đổi tỷ lệ quay, biến tốc độ cao và lực xoắn thấp của động cơ thành tốc độ thấp và lực xoắn cao hơn ở trục đầu ra.

Dưới đây là thông số kỹ thuật của nó:

- Điện áp hoạt động: 3V - 12V DC.
- Tốc độ quay (ở 6V):
- Tỷ lệ giảm tốc 1:48: 200 RPM.
- Tỷ lệ giảm tốc 1:120: 100 RPM.
- Lực xoắn (ở 6V):
- Tỷ lệ giảm tốc 1:48: 1.1 kg.cm.
- Tỷ lệ giảm tốc 1:120: 2.5 kg.cm.

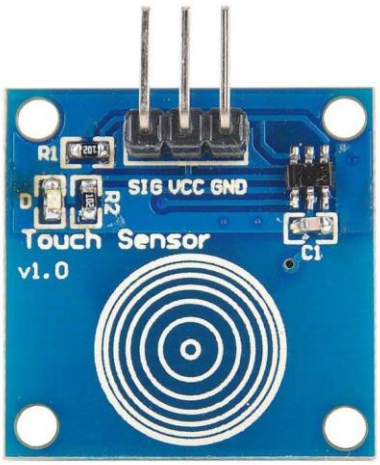
- Dòng điện không tải: Khoảng 70 mA (ở 6V).
- Dòng điện tải tối đa: Khoảng 350 mA (ở 6V).
- Kích thước hộp số: 70 mm x 22 mm x 18 mm.
- Đường kính trục: 5 mm.
- Trọng lượng: Khoảng 30 g. [4]

Giao thức điều khiển và các chân giao tiếp

- Dây nguồn (Power): Kết nối với cực dương của nguồn cung cấp điện (thường màu đỏ).
- Dây nối đất (Ground): Kết nối với cực âm của nguồn cung cấp điện (thường màu đen hoặc xanh dương).

Cảm Ứng Chạm Điện Dung TTP223

Module cảm ứng chạm điện dung TTP223 là một module cảm biến chạm đơn giản và dễ sử dụng, sử dụng IC TTP223 để phát hiện sự chạm. Module này được ứng dụng rộng rãi trong các dự án DIY, các hệ thống điều khiển chạm đơn giản, và các thiết bị điện tử tiêu dùng. Với khả năng phát hiện chạm nhanh chóng và chính xác, TTP223 là lựa chọn lý tưởng cho các ứng dụng cần điều khiển bằng cảm ứng chạm.



Hình 9. Cảm Ứng Chạm Điện Dung TTP223

Thông số kỹ thuật của cảm ứng chạm điện dung như sau:

- Điện áp hoạt động: 2.0V - 5.5V DC.
- Dòng tiêu thụ:
- Chế độ nhanh: 1.5 mA.
- Chế độ chậm: 3 uA.
- Đầu ra:
- Mức cao khi có chạm (3.3V hoặc 5V tùy theo nguồn cấp).
- Mức thấp khi không có chạm.
- Khoảng cách phát hiện: 0 - 5 mm.
- Kích thước: 11 mm x 10 mm.
- Loại đầu ra: Digital (số).

Chức năng và đặc điểm của nó:

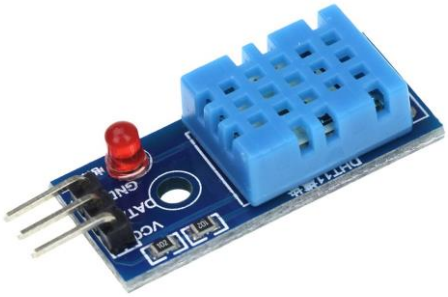
- Dễ dàng tích hợp: Chỉ cần kết nối ba dây (VCC, GND, OUT) với bộ điều khiển.
- Độ nhạy cao: Có thể phát hiện chạm qua các lớp vật liệu mỏng như giấy hoặc nhựa.
- Tiêu thụ năng lượng thấp: Chế độ tiết kiệm năng lượng khi không hoạt động.
- Tín hiệu ổn định: Đầu ra số ổn định, dễ dàng đọc và xử lý.
- Chế độ hoạt động: Có thể cấu hình để hoạt động ở chế độ bình thường hoặc chế độ tiết kiệm năng lượng. [5]

Sơ đồ chân và kết nối:

- VCC: Cấp nguồn cho module, kết nối với nguồn 2.0V - 5.5V.
- GND: Chân nối đất, kết nối với điểm nối đất của hệ thống.
- SIG: Chân đầu ra tín hiệu số, kết nối với chân đầu vào của vi điều khiển.

Cảm Biến Đo Độ Ẩm và Nhiệt Độ DHT11

DHT11 là một cảm biến đo nhiệt độ và độ ẩm giá rẻ, được thiết kế cho các ứng dụng trong các dự án DIY, các hệ thống tự động hóa gia đình, và các thiết bị giám sát môi trường. DHT11 tích hợp cả cảm biến độ ẩm và nhiệt độ trong một gói nhỏ gọn, cho phép dễ dàng đo lường và thu thập dữ liệu môi trường với độ chính xác chấp nhận được cho các ứng dụng thông thường.



Hình 10.Cảm Biến Đo Độ Ẩm và Nhiệt Độ DHT11

Thông số kỹ thuật của thiết bị:

- Nguồn cung cấp: 3.3V - 5.5V DC.
- Dải đo nhiệt độ: 0°C - 50°C.
- Sai số đo nhiệt độ: ±2°C.
- Dải đo độ ẩm: 20% - 90% RH.
- Sai số đo độ ẩm: ±5% RH.
- Chu kỳ đo: 1 giây (tối thiểu).
- Giao diện: Một dây (single wire), sử dụng giao thức giao tiếp độc quyền.
- Kích thước: 15.5 mm x 12 mm x 5.5 mm.
- Trọng lượng: Khoảng 2.4 g. [6]

DHT11 gồm 2 thành phần chính:

- Cảm biến nhiệt độ: Sử dụng nhiệt điện trở (thermistor) để đo nhiệt độ không khí xung quanh.

- Cảm biến độ ẩm: Sử dụng điện trở polymer để đo độ ẩm tương đối (RH) trong không khí. Cảm biến thay đổi trở kháng dựa trên lượng hơi nước trong không khí, chuyển đổi thành tín hiệu điện.

DHT11 gồm có 3 chân:

- VCC: Chân cấp nguồn cho cảm biến (3.3V - 5.5V DC).
- GND: Chân nối đất.
- DATA: Chân truyền dữ liệu số (kết nối với chân I/O của vi điều khiển).

Động Cơ Bước 28BYJ-48-5V và Driver ULN2003

Động cơ bước 28BYJ-48-5V là một loại động cơ bước nhỏ gọn, giá rẻ, thường được sử dụng trong các ứng dụng điện tử tiêu dùng và dự án DIY. Động cơ này hoạt động dựa trên nguyên lý điều khiển từng bước nhỏ, cho phép định vị chính xác và kiểm soát tốc độ quay. Để điều khiển động cơ bước này, người dùng thường sử dụng driver điều khiển ULN2003, một mạch tích hợp có khả năng điều khiển các cuộn dây của động cơ bằng cách chuyển đổi các tín hiệu từ bộ điều khiển thành dòng điện điều khiển.

Hình 11.Động Cơ Bước 28BYJ-48-5V và Driver ULN2003

Dưới đây là thông số thiết kế của cụm thiết bị

❖ Động cơ bước 28BYJ-48-5V

- Điện áp định mức: 5V DC.
- Tỷ số truyền: 64:1.
- Bước góc: 5.625°/64 (sau tỷ số truyền).
- Số bước mỗi vòng: 2048 bước.

- Dòng điện định mức: 240 mA.
- Điện trở cuộn dây: Khoảng 50Ω.
- Mô-men xoắn tĩnh: ≥ 34.3 mN·m (120 Hz).
- Kích thước: 28 mm x 28 mm x 25 mm.
- Khối lượng: Khoảng 30 g. [7]

❖ Driver điều khiển ULN2003

- Điện áp điều khiển: 5V - 12V DC.
- Dòng điện đầu ra: Tối đa 500 mA cho mỗi kênh.
- Số kênh: 7 (có thể điều khiển 4 cuộn dây của động cơ bước).
- Bảo vệ chống quá áp: Có (flyback diodes).
- Giao diện điều khiển: Tín hiệu số từ bộ điều khiển (thường từ vi điều khiển hoặc Arduino).
- Kích thước: 41 mm x 31 mm. [8]

Cấu tạo và nguyên lý hoạt động:

❖ Động cơ bước 28BYJ-48-5V

28BYJ-48 là một loại động cơ bước đơn cực, có cấu trúc gồm bốn cuộn dây (A, A', B, B') nối với các đầu ra. Khi dòng điện chạy qua các cuộn dây theo thứ tự, nó tạo ra các xung lực từ, đẩy rotor quay theo từng bước. Động cơ bước này có tỷ số truyền 64:1 nhờ hộp số bên trong, cho phép điều khiển vị trí rotor với độ chính xác cao. [7]

❖ Driver điều khiển ULN2003

ULN2003 là một mạch tích hợp gồm bảy transistor Darlington, cho phép điều khiển các cuộn dây của động cơ bước. Các transistor này chuyển đổi tín hiệu điều khiển từ bộ điều khiển thành dòng điện đủ mạnh để kích hoạt cuộn dây của động cơ. ULN2003 cũng bao gồm các diode bảo vệ để ngăn chặn dòng điện ngược khi cuộn dây ngắt kết nối. [8]

Sơ đồ chân kết nối:

Bảng 2. Động cơ bước 28BYJ-48-5V

Chân	Màu dây	Chức năng
1	Cam	Coil A
2	Vàng	Coil B
3	Hồng	Coil C
4	Xanh	Coil D
5	Đỏ	VCC

Bảng 3. Driver điều khiển ULN2003

Chân ULN2003	Chức năng	Kết nối với động cơ
IN1	Điều khiển Coil A	Coil A
IN2	Điều khiển Coil B	Coil B
IN3	Điều khiển Coil C	Coil C
IN4	Điều khiển Coil D	Coil D
COM	Nguồn VCC cho động cơ	VCC
GND	Nối đất	Nối đất

- **Module RF Thu Phát Wi-Fi Bluetooth ESP32**

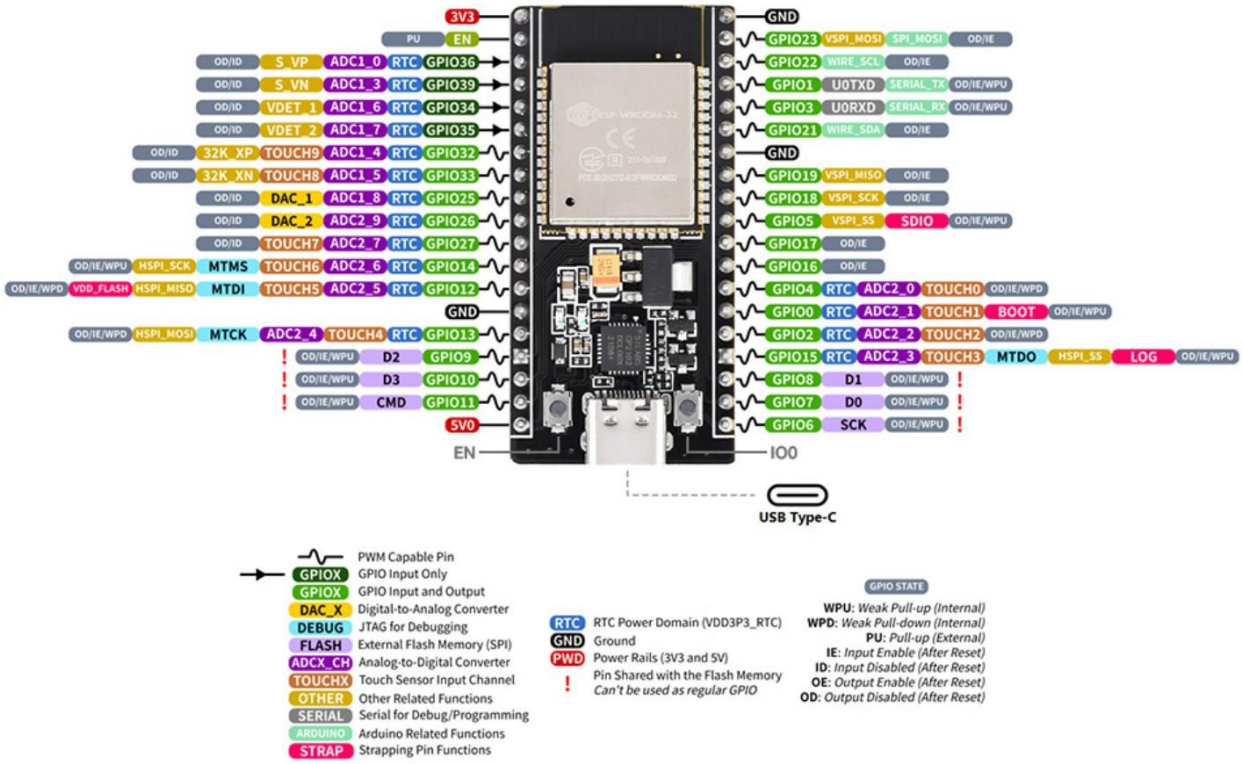
ESP32 là một module thu phát tích hợp mạnh mẽ với khả năng kết nối Wi-Fi và Bluetooth, được phát triển bởi Espressif Systems. ESP32 là một giải pháp lý tưởng cho các ứng dụng IoT (Internet of Things), hệ thống điều khiển thông minh, và các dự án DIY cần kết nối mạng không dây. Với khả năng xử lý mạnh mẽ, hỗ trợ nhiều giao thức truyền thông, và giá thành hợp lý, ESP32 đã trở thành một trong những lựa chọn phổ biến nhất cho các nhà phát triển và người đam mê điện tử.

Dưới đây là những thông số kỹ thuật của ESP32:

- CPU:
- Dual-core Xtensa® LX6 32-bit.
- Tần số xung nhịp: Lên đến 240 MHz.
- RAM: 520 KB SRAM.
- Flash: 4 MB (hoặc hơn tùy biến thể).
- Wi-Fi:
- 802.11 b/g/n (2.4 GHz).
- Hỗ trợ chế độ Station (STA), Soft Access Point (AP), và AP + STA.
- Bluetooth:
- V4.2 BR/EDR và BLE (Bluetooth Low Energy).
- GPIO: 34 chân I/O.
- ADC: 18 kênh, độ phân giải 12-bit.

- DAC: 2 kênh, độ phân giải 8-bit.
- PWM: Hỗ trợ trên tất cả các chân I/O.
- Giao thức truyền thông: UART, I2C, SPI, I2S, CAN, IR.
- Bộ quản lý năng lượng: Hỗ trợ chế độ tiết kiệm năng lượng (Deep Sleep, Light Sleep).
- Bảo mật: AES, SHA-2, RSA, ECC, Random Number Generator (RNG).
- Điện áp hoạt động: 2.2V - 3.6V DC.
- Kích thước: 18 mm x 25.5 mm (thay đổi tùy biến thể). [9]

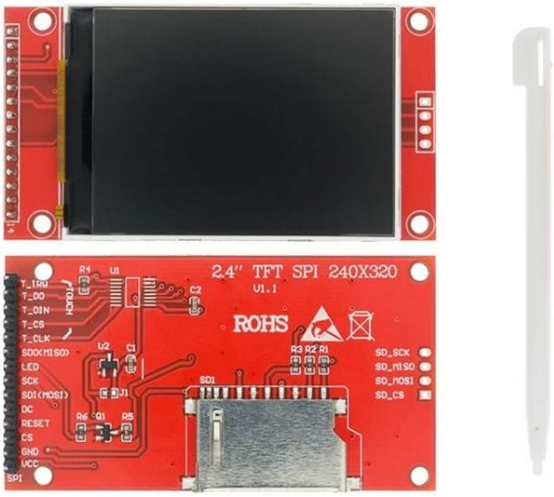
Sơ đồ chân của ESP32:



Hình 12. Sơ đồ chân của ESP32

Màn Hình Cảm Ứng LCD TFT Touch Screen 3.2 Inch ILI9341 SPI Interface

Màn hình cảm ứng LCD TFT 3.2 inch sử dụng driver ILI9341 là một giải pháp hiển thị phổ biến trong các ứng dụng nhúng, cung cấp giao diện đồ họa màu sắc sống động và khả năng tương tác cảm ứng. Với giao diện SPI (Serial Peripheral Interface), màn hình này dễ dàng tích hợp vào các dự án điện tử với các vi điều khiển như Arduino, Raspberry Pi, và ESP32. Đây là lựa chọn lý tưởng cho các ứng dụng yêu cầu hiển thị đồ họa và cảm ứng như giao diện người dùng, hệ thống điều khiển, và các thiết bị thông minh.



Hình 13.Màn Hình Cảm Ứng LCD TFT Touch Screen 3.2

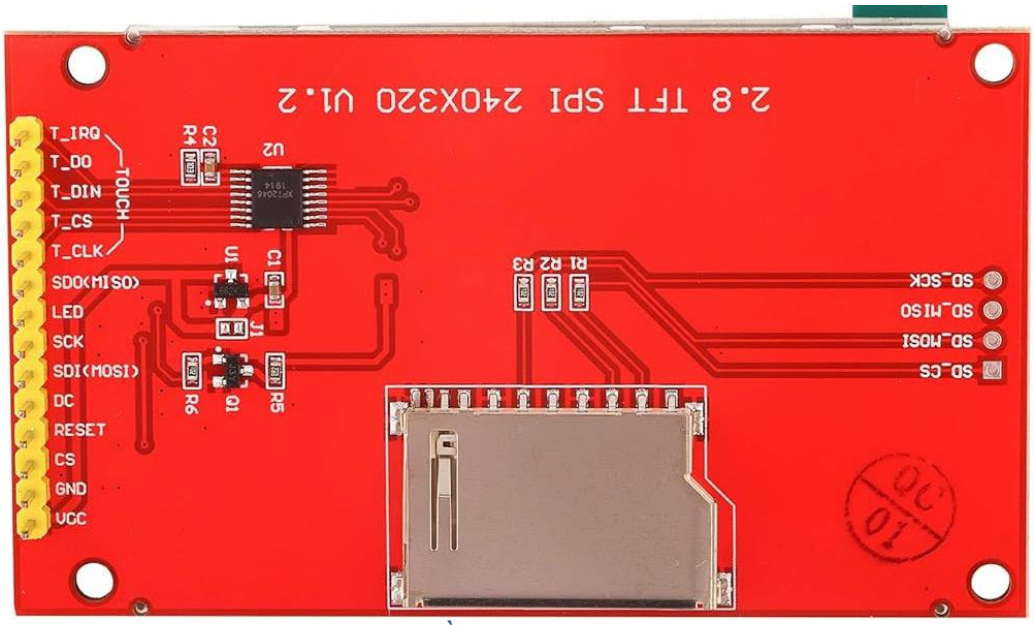
Dưới đây là thông số về màn hình này:

- Kích thước: 3.2 inch
- Độ phân giải: 240 x 320 pixels
- Driver: ILI9341
- Giao diện: SPI (Serial Peripheral Interface)
- Số màu hiển thị: 262K/65K
- Điện áp hoạt động: 3.3V/5V
- Đèn nền: LED
- Cảm ứng: Điện trở (Resistive) hoặc Điện dung (Capacitive) tùy biến thể
- Kích thước màn hình hiển thị: 48.6 mm x 64.8 mm
- Góc nhìn: 80°/80°/80°/80° (trái/phải/trên/dưới)
- Tần số làm mới: 60 Hz

- Điện năng tiêu thụ: Tối đa 300mW [10]

Cấu tạo và nguyên lý hoạt động: Màn hình TFT (Thin Film Transistor) sử dụng công nghệ transistor màng mỏng để điều khiển từng pixel trên màn hình, cho phép hiển thị màu sắc rực rỡ và chi tiết rõ ràng. Bộ điều khiển ILI9341 quản lý việc truyền dữ liệu từ vi điều khiển đến màn hình thông qua giao diện SPI. Giao diện SPI sử dụng một tập hợp các chân kết nối để truyền dữ liệu serial, giúp giảm số lượng chân I/O cần thiết.

Sơ đồ chân và kết nối:



Hình 14. Sơ đồ chân của màn hình LCD

- GND: Chân nối đất của màn hình.
- VCC: Chân nguồn, có thể sử dụng 3.3V hoặc 5V.
- CS (Chip Select): Chọn chip điều khiển màn hình khi giao tiếp SPI.
- RESET: Đặt lại bộ điều khiển ILI9341.
- DC (Data/Command): Chân này chuyển đổi giữa chế độ dữ liệu và chế độ lệnh. Ở chế độ lệnh, vi điều khiển gửi các lệnh điều khiển đến màn hình, còn ở chế độ dữ liệu, nó gửi các dữ liệu đồ họa cần hiển thị.
- SDI/MOSI (Serial Data Input/Master Out Slave In): Đầu vào dữ liệu cho giao tiếp SPI.
- SCK (Serial Clock): Đồng hồ SPI, cung cấp xung nhịp cho việc truyền dữ liệu.
- LED: Điều khiển đèn nền màn hình, có thể được điều chỉnh bằng PWM để thay đổi độ sáng.

- SDO/MISO (Serial Data Output/Master In Slave Out): Đầu ra dữ liệu cho giao tiếp SPI, thường ít sử dụng vì màn hình chủ yếu là nhận dữ liệu.
- T_CLK, T_CS, T_DIN, T_DO, T_IRQ: Các chân liên quan đến cảm ứng, sử dụng cho việc đọc tín hiệu cảm ứng từ màn hình. [10]

Loa Bluetooth AVA+ MiniPod Y23

Loa Bluetooth AVA+ MiniPod Y23 là một thiết bị âm thanh không dây được thiết kế với mục tiêu cung cấp giải pháp nghe nhạc linh hoạt và tiện lợi. Sản phẩm này kết hợp giữa tính năng di động cao và khả năng tái tạo âm thanh chất lượng, phù hợp với nhu cầu giải trí đa dạng của người tiêu dùng hiện đại.

Hình 15. Loa Bluetooth AVA+ MiniPod Y23

Dưới đây là một vài thông tin về thông số kỹ thuật của nó:

- Kích thước: Ngang 7.55 cm - Sâu 7.55 cm - Cao 6.15 cm - Nặng 0.16 kg
- Công suất: 3W
- Thời gian sử dụng: từ 4 đến 5 giờ (Sạc khoảng 2 đến 3 giờ)
- Kết nối không dây: Bluetooth 5.1
- Khoảng cách tối đa: 10m [11]

- **Bảng kê chi phí tổng cộng của đồ án**

Bảng 4. Bảng kê chi phí dự án

Mã Nhóm Linh Kiện	Tên Nhóm Linh Kiện	Số Tiền
1	Dây cắm mạch (Jumper wire)	141.500
2	Breadboard	23.500
3	Cảm biến nhiệt độ, độ ẩm	129.000
4	Động cơ DC, L298N, Linh kiện phục vụ lắp cửa trượt	206.000
5	Rasperry Pi 4 8GB + Adapter chuyển đổi	3.603.000
6	Microphone Boya M1 + Adapter chuyển đổi	490.000
7	Động cơ bước + Driver	74.000
8	Đèn Led	40.000
9	Module thu phát ESP32 + Màn hình TFT	434.000
10	Loa Bluetooth	153.000
11	Công tắc hành trình, cảm biến chạm	30.000
12	Các linh kiện cung cấp điện	121.000
13	Động cơ Servo	50.000
14	Bìa mica, bìa cứng, gỗ ép	745.000
15	Các linh kiện phục vụ lắp ráp nhà thông minh	434.000
TỔNG SỐ TIỀN		**6.674.000**

2.1.3. Truyền thông

- **Django Framework**

Django là một framework web cao cấp được viết bằng Python, nổi bật với khả năng phát triển nhanh chóng và sạch sẽ, cùng với sự hỗ trợ mạnh mẽ cho các dự án từ nhỏ đến lớn. Được phát hành lần đầu vào năm 2005, Django đã trở thành lựa chọn phổ biến cho các nhà phát triển web nhờ vào bộ công cụ mạnh mẽ và dễ sử dụng, tập trung vào các nguyên tắc thiết kế như DRY (Don't Repeat Yourself) và nguyên lý thiết kế đơn giản nhưng đầy đủ tính năng.

Dưới đây là các đặc điểm nổi bật của Django:

- Định Nghĩa Mô Hình (ORM): Django đi kèm với một hệ thống ORM (Object-Relational Mapping) mạnh mẽ, cho phép các nhà phát triển làm việc với cơ sở dữ liệu bằng cách sử dụng các mô hình Python thay vì viết các câu lệnh SQL thủ công. Điều này giúp mã nguồn trở nên trực quan và dễ bảo trì.
- Hệ Thống URL: Hệ thống định tuyến URL của Django rất linh hoạt, cho phép xác định các URL của ứng dụng một cách rõ ràng và dễ dàng. Hệ thống này hỗ trợ việc thiết lập URL

với các quy tắc động và cung cấp công cụ mạnh mẽ cho việc tạo và quản lý các URL phức tạp.

- Quản Lý Admin: Một trong những tính năng nổi bật của Django là giao diện quản trị tự động. Giao diện này được tạo ra từ các mô hình dữ liệu của bạn và cung cấp một hệ thống quản lý nội dung ngay lập tức, giúp việc quản lý dữ liệu trở nên dễ dàng mà không cần nhiều cấu hình. Và đây cũng là ưu điểm mà nhóm tận dụng triệt để nhất

Bluetooth

Bluetooth là một tiêu chuẩn công nghệ không dây cho phép trao đổi dữ liệu trong khoảng cách ngắn bằng cách sử dụng sóng vô tuyến UHF thuộc dải tần ISM. Bluetooth hỗ trợ nhiều cấu hình ứng dụng, bao gồm cả A2DP (Advanced Audio Distribution Profile) cho truyền tải âm thanh. Trong hệ thống sử dụng Raspberry Pi, Bluetooth có thể được cấu hình để kết nối với các thiết bị âm thanh như loa Bluetooth, cho phép truyền tải âm thanh không dây. Điều này được thực hiện thông qua việc thiết lập kết nối ghép cặp (pairing) và truyền luồng âm thanh số, giúp các hệ thống nhúng dễ dàng phát triển các ứng dụng âm thanh không dây tiện lợi và hiệu quả.

Wi-Fi (Wireless Fidelity)

Wi-Fi, viết tắt của Wireless Fidelity, là một bộ các giao thức truyền thông mạng không dây thuộc tiêu chuẩn IEEE 802.11, cung cấp khả năng kết nối thiết bị điện tử tới mạng lưới không dây (WLAN). Wi-Fi cho phép các thiết bị, như máy tính, điện thoại thông minh, và các thiết bị nhúng như Raspberry Pi, kết nối và trao đổi dữ liệu với mạng cục bộ hoặc internet mà không cần dây dẫn. Với tốc độ truyền dữ liệu cao, khả năng xử lý tín hiệu mạnh mẽ, và khả năng mở rộng mạng, Wi-Fi hỗ trợ đa dạng các ứng dụng từ truyền tải dữ liệu thông thường tới truyền thông đa phương tiện phức tạp. Trong bối cảnh sử dụng Raspberry Pi, Wi-Fi thường được sử dụng để kết nối thiết bị với mạng gia đình hoặc mạng cục bộ, hỗ trợ các ứng dụng IoT, truyền thông, và giải trí.

HTTP (Hypertext Transfer Protocol)

Hypertext Transfer Protocol (HTTP) là một giao thức tầng ứng dụng trong mô hình OSI, được thiết kế để truyền tải các tài liệu siêu văn bản trên World Wide Web. HTTP hoạt động dựa trên mô hình yêu cầu-phản hồi, nơi máy khách gửi yêu cầu tới máy chủ và nhận về tài liệu, bao gồm cả văn bản, hình ảnh, và video. Với các phương thức chính như GET, POST, PUT, DELETE, HTTP cung cấp cách thức giao tiếp chuẩn hóa cho các ứng dụng web và API. Trong

các ứng dụng với Raspberry Pi, HTTP có thể được sử dụng để thiết lập các máy chủ web nhỏ, cho phép điều khiển và giám sát thiết bị từ xa thông qua giao diện web hoặc API RESTful.

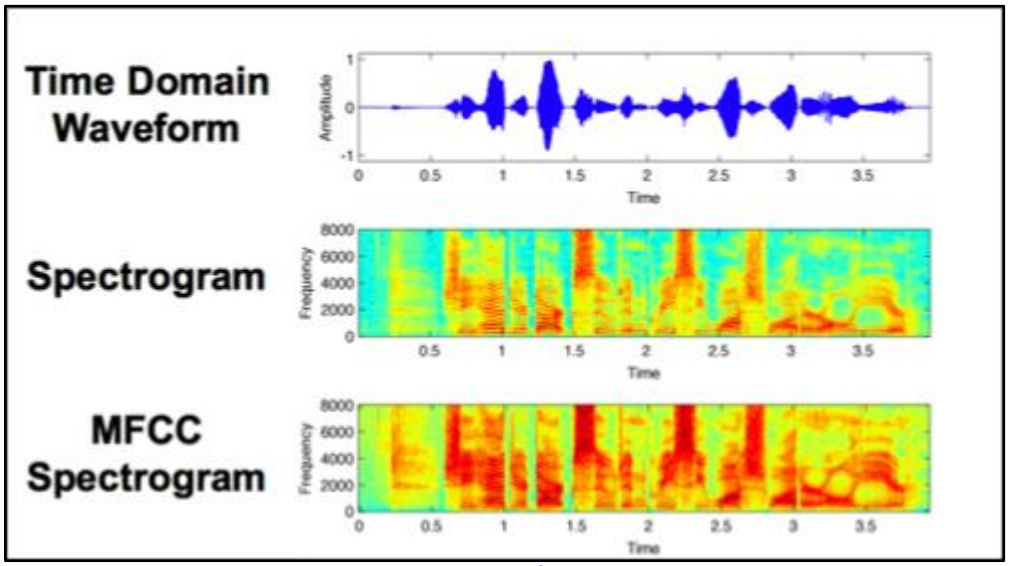
- **SPI (Serial Peripheral Interface)**

Serial Peripheral Interface (SPI) là một giao thức truyền thông đồng bộ, được thiết kế để truyền dữ liệu giữa một thiết bị chính (master) và một hoặc nhiều thiết bị phụ (slave). SPI sử dụng bốn đường tín hiệu chính: MISO (Master In Slave Out), MOSI (Master Out Slave In), SCLK (Serial Clock), và SS (Slave Select), để điều phối việc truyền nhận dữ liệu. Với khả năng truyền dữ liệu song song tốc độ cao, SPI thường được sử dụng trong các hệ thống nhúng để kết nối vi điều khiển với các thiết bị ngoại vi như cảm biến, màn hình, và bộ nhớ. Trên Raspberry Pi, SPI thường được sử dụng để giao tiếp với các màn hình TFT, cảm biến và các module mở rộng.

2.2. Giải pháp TTNT/KHDL

2.2.1 Giải pháp nhận diện giọng nói

- **Trích xuất đặc trưng**



Hình 16. Phổ MFCC

- Trong quá trình Trích xuất đặc trưng, chúng ta muốn chuyển đổi các tệp âm thanh từ định dạng FLAC thành dạng đặc trưng mà mô hình của chúng ta có thể xử lý hiệu quả. Nhóm đã sử dụng MFCC làm vector đặc trưng vì sức mạnh của nó trong việc xử lý âm thanh, đặc biệt là trong các ứng dụng nhận dạng người nói.
- "*Điều quan trọng là MFCC là một biểu diễn của âm thanh dựa trên các đặc trưng phổ của nó, được thiết kế để mô phỏng cách con người nghe tiếng nói*" [12] . Với

MFCC, chúng ta có thể chuyển đổi tín hiệu âm thanh thành một loạt các vectơ đặc trưng, mỗi vectơ đại diện cho một khung thời gian của tín hiệu. ⇒ MFCC (Mel-Frequency Cepstral Coefficients) là một biểu diễn đặc trưng được tạo ra từ tín hiệu âm thanh trên miền thời gian.

- Nếu âm thanh có hai kênh (stereo), hàm librosa.to_mono được sử dụng để chuyển đổi nó thành dạng đơn kênh (mono). Điều này đảm bảo rằng chúng ta chỉ xử lý một kênh của âm thanh, giúp giảm bớt phức tạp và tiêu tốn bộ nhớ. Nếu tốc độ mẫu của âm thanh không phải là 16kHz, hàm librosa.resample được sử dụng để tái mẫu âm thanh, đảm bảo rằng tất cả các âm thanh đều có cùng một tốc độ mẫu (16kHz) [13] để đồng bộ hóa quá trình trích xuất đặc trưng. N_MFCC = 80 , điều này chỉ định rằng chúng ta muốn trích xuất 80 hệ số Mel-Frequency Cepstral Coefficients (MFCC) từ mỗi khung thời gian của tín hiệu âm thanh (vector 80 chiều cho mỗi khung thời gian), 40-80 là con số được khuyến nghị cho việc trích xuất.

Tăng cường dữ liệu

- SpecAugment tương tự như các kỹ thuật tăng cường dữ liệu được sử dụng trong lĩnh vực xử lý hình ảnh, nhưng được thiết kế đặc biệt để áp dụng cho dữ liệu âm thanh, đặc biệt là cho các tác vụ liên quan đến xử lý giọng nói hoặc âm thanh. Bằng cách biến đổi các đặc trưng âm thanh, SpecAugment giúp mô hình học được các đặc điểm quan trọng và trở nên robust hơn đối với các biến đổi và nhiễu trong dữ liệu âm thanh.

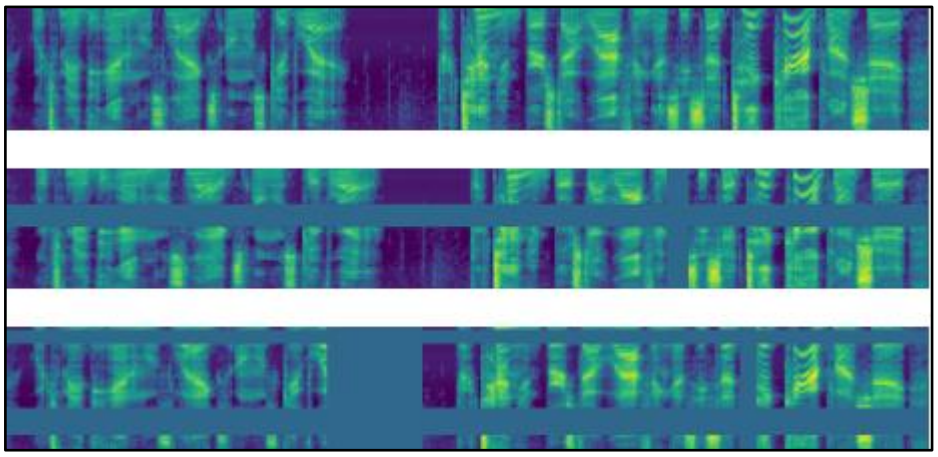
- SpecAugment là một phương pháp tăng cường dữ liệu được áp dụng trực tiếp vào đầu vào đặc trưng của mạng nơ-ron (ví dụ: các hệ số filter bank). Chính sách tăng cường này bao gồm việc biến đổi đặc trưng bằng cách uốn cong các đặc trưng, che phủ các khối kênh tần số, và che phủ các khối bước thời gian. SpecAugment nhằm xây dựng một chính sách tăng cường tác động trực tiếp lên log mel spectrogram, giúp mạng học các đặc trưng hữu ích.

Các kỹ thuật SpecAugment được sử dụng gồm có:

1. Time Warping (Biến đổi thời gian): Áp dụng thông qua hàm sparse_image_warp của TensorFlow. Với một log mel spectrogram có τ bước thời gian, nó được xem xét như một hình ảnh trong đó trục thời gian là ngang và trục tần số là dọc. Một điểm ngẫu nhiên dọc theo đường ngang đi qua trung tâm của hình ảnh trong khoảng

thời gian (W, τ−W) sẽ được uốn cong về phải hoặc về trái một khoảng w được chọn từ phân phối đồng đều từ 0 đến tham số uốn cong thời gian W.

2. Frequency Masking (Che phủ tần số): Áp dụng sao cho f kênh tần số mel liên tiếp [f0, f0+f) bị che phủ, trong đó f được chọn từ phân phối đồng đều từ 0 đến tham số che phủ tần số F, và f0 được chọn từ [0, ν−f). Ở đây, ν là số kênh tần số mel.
3. Time Masking (Che phủ thời gian): Áp dụng sao cho t bước thời gian liên tiếp [t0, t0+t) bị che phủ, trong đó t được chọn từ phân phối đồng đều từ 0 đến tham số che phủ thời gian T, và t0 được chọn từ [0, τ−t). Ở đây, τ là số bước thời gian [14].

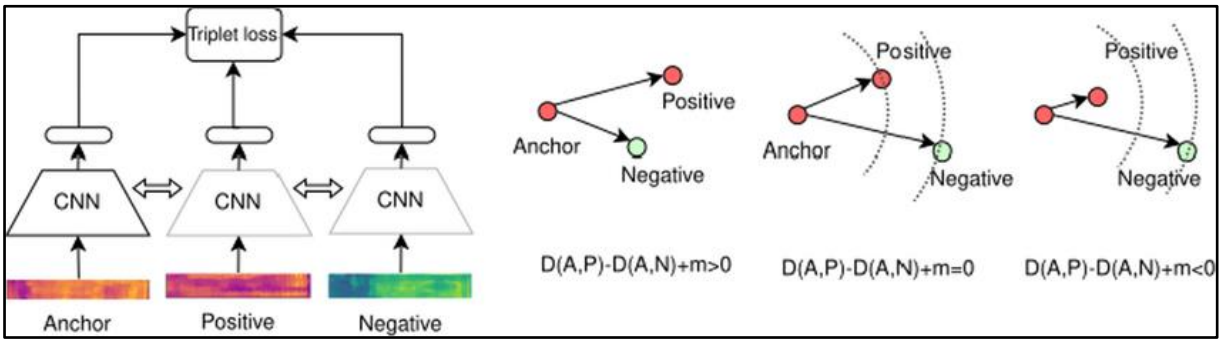


Hình 17. Kỹ thuật SpecAugment

- **Hàm mất mát: Triplet Loss**

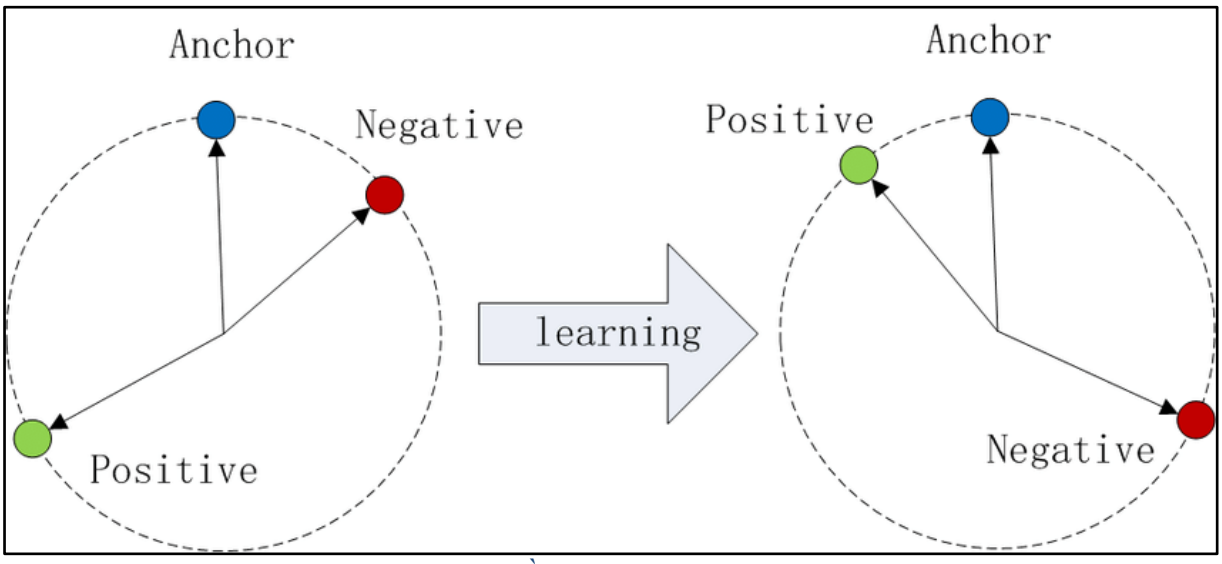
Triplet loss là một kỹ thuật phổ biến được sử dụng trong nhận dạng người nói, trong đó mô hình được huấn luyện để học cách phân biệt giữa các người nói dựa trên các đặc trưng của họ. Kỹ thuật này cũng được áp dụng rộng rãi trong các lĩnh vực khác như nhận dạng khuôn mặt, xác định đối tượng trong ảnh, etc.

Trong triplet loss, đầu vào của chúng ta bao gồm ba phân đoạn: một anchor (mẫu gốc), một positive (mẫu tương đồng), và một negative (mẫu không tương đồng). Hai phân đoạn anchor và positive thuộc về cùng một người nói, trong khi phân đoạn negative thuộc về một người nói khác. Mục tiêu của triplet loss là đảm bảo rằng khoảng cách giữa anchor và positive là nhỏ nhất có thể, trong khi khoảng cách giữa anchor và negative là lớn nhất có thể. Điều này đảm bảo rằng mô hình học được cách phân biệt giữa các người nói dựa trên các đặc trưng của họ.



Hình 18. Ví dụ triplet loss

Mục tiêu chính của Triplet Loss là đảm bảo rằng các cặp không giống nhau (dissimilar) cách xa nhau ít nhất một giá trị biên (margin) so với các cặp giống nhau (similar). Nói cách khác, tối thiểu hóa khoảng cách giữa Anchor và Positive, đồng thời tối đa hóa khoảng cách giữa Anchor và Negative. Điều này đồng nghĩa với việc mô hình cố gắng giảm sự khác biệt giữa các mẫu tương đồng (Positive) trong không gian nhúng, trong khi tăng cường sự khác biệt giữa các mẫu không tương đồng (Negative).



Hình 19. Sơ đồ hoạt động của triplet loss

Trong Triplet Loss, quá trình sử dụng các nhóm ba mục gọi là triplets, mỗi triplet bao gồm:

1. Anchor (mẫu gốc): Là một mẫu cố định với một danh tính nhất định.

2. Positive (mẫu tương đồng): Là một mẫu gần với anchor, thường là một mẫu cùng lớp hoặc cùng phân loại.

3. Negative (mẫu không tương đồng): Là một mẫu xa với anchor, không cùng lớp hoặc phân loại với anchor

Công thức tính Triplet loss như sau:

$$L(A, P, N) = max(D(A, P) - D(A, N) + m, 0)$$

Hình 20. Công thức triplet loss

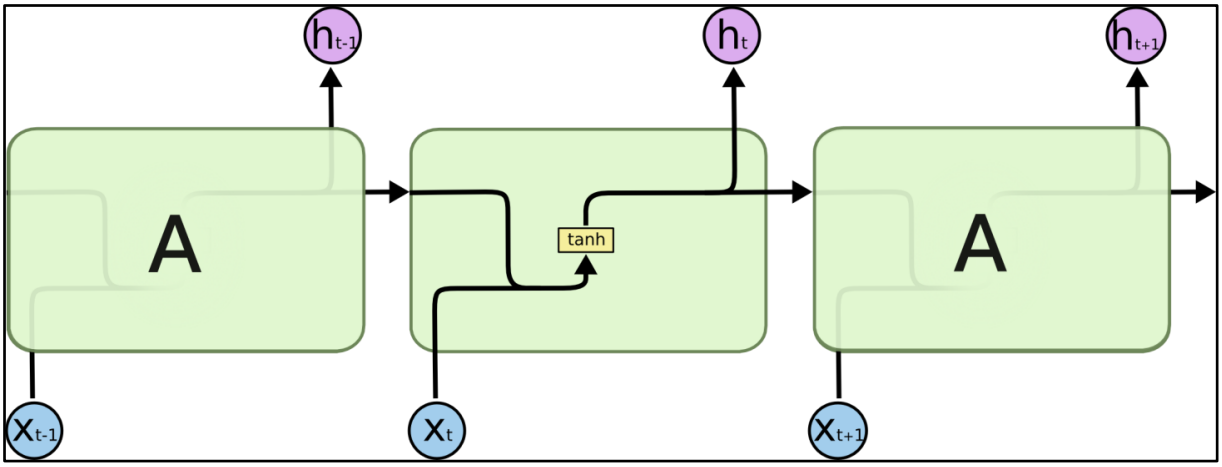
Trong đó:

- **L(A,P,N)**: Là giá trị loss cho bộ ba (triplet) gồm anchor (A), positive (P), và negative (N).
- **D(A,P)**: Là khoảng cách giữa vector đặc trưng của anchor (A) và positive (P). Khoảng cách này thường được tính bằng khoảng cách Euclidean hoặc khoảng cách cosine.
- **D(A,N)**: Là khoảng cách giữa vector đặc trưng của anchor (A) và negative (N).
- **m**: Là một giá trị biên (margin) dương. Giá trị này được thêm vào để đảm bảo rằng khoảng cách giữa anchor và positive nhỏ hơn khoảng cách giữa anchor và negative ít nhất là mmm.

Kiến trúc mạng Long Short Term Memory:

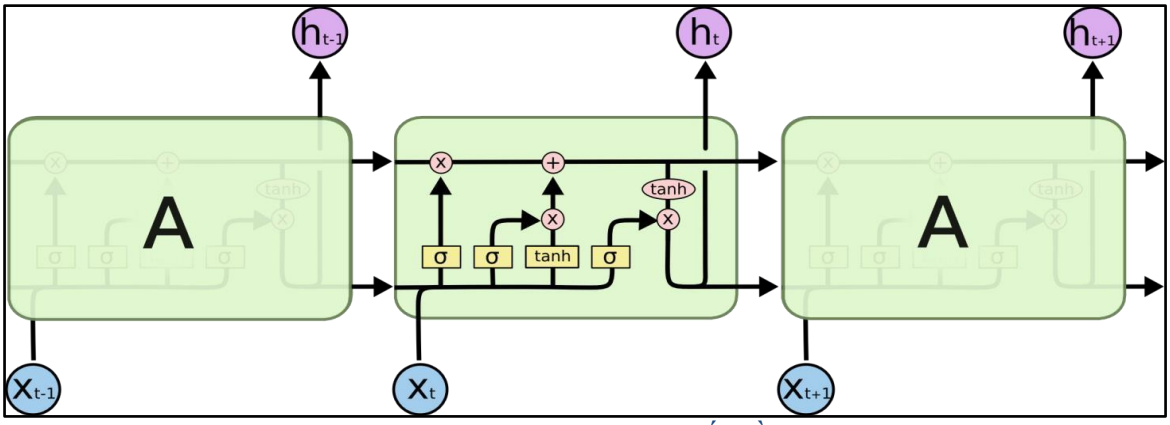
Mạng Bộ Nhớ Dài Ngắn Hạn – Long Short Term Memory (LSTM) – là một loại mạng nơ-ron hồi quy đặc biệt, có khả năng học các phụ thuộc dài hạn. LSTM được thiết kế rõ ràng để tránh vấn đề phụ thuộc dài hạn. Việc ghi nhớ thông tin trong thời gian dài gần như là hành vi mặc định của chúng.

Tất cả các mạng nơ-ron hồi quy đều có dạng một chuỗi các mô-đun lặp lại của mạng nơ-ron. Trong các mạng nơ-ron hồi quy tiêu chuẩn, mô-đun lặp lại này sẽ có cấu trúc rất đơn giản, chẳng hạn như một tầng tanh đơn lẻ.



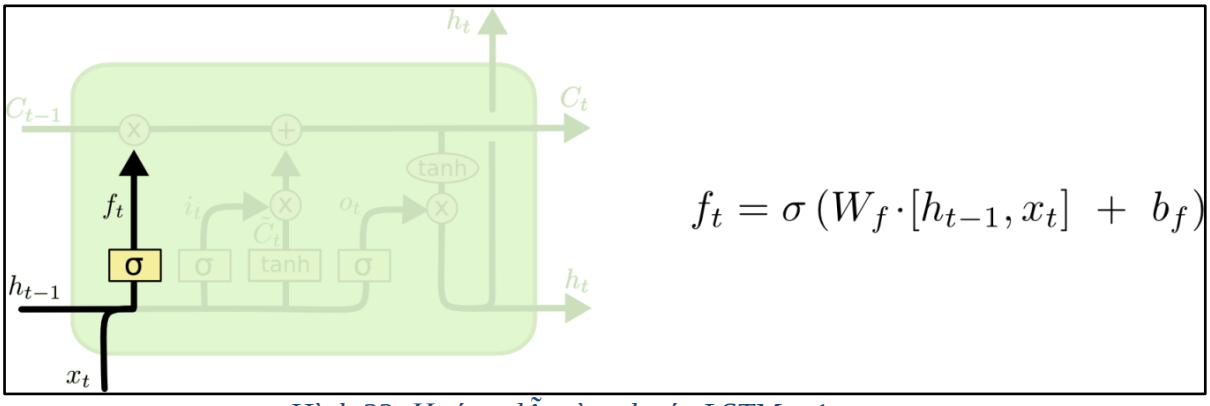
Hình 21. Mô-đun lặp lại trong một mạng nơ-ron hồi quy (RNN) tiêu chuẩn

LSTM cũng có cấu trúc dạng chuỗi như vậy, nhưng mô-đun lặp lại có một cấu trúc khác biệt. Thay vì chỉ có một tầng mạng nơ-ron, nó có bốn tầng, tương tác theo một cách rất đặc biệt.



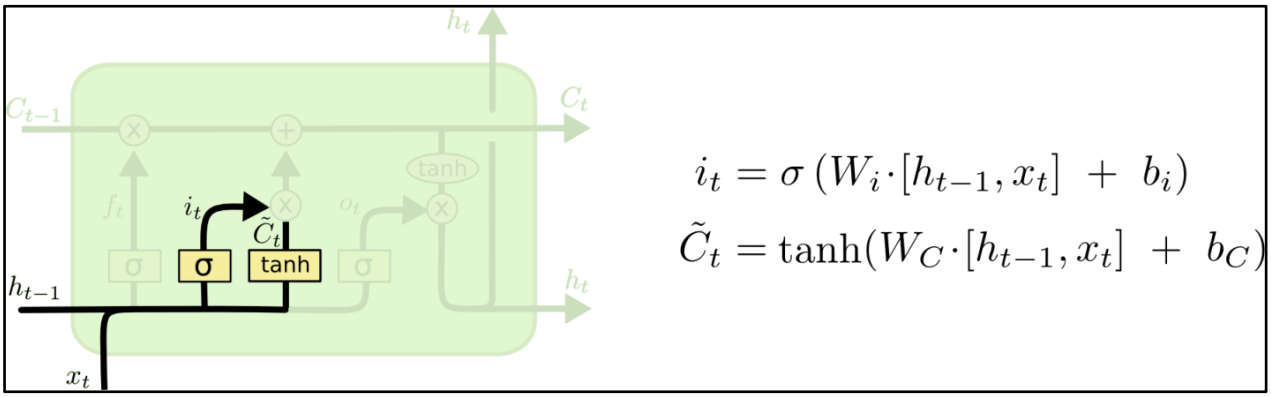
Hình 22. Mô-đun lặp lại trong một LSTM chứa bốn tầng tương tác với nhau

Bước đầu tiên trong LSTM của chúng ta là quyết định thông tin nào cần loại bỏ khỏi trạng thái ô nhớ (cell state). Quyết định này được thực hiện bởi một tầng sigmoid, gọi là "tầng cổng quên" (forget gate layer). Nó xem xét h_{t-1} và x_t, và đưa ra một giá trị trong khoảng từ 0 đến 1 cho mỗi phần tử trong trạng thái ô nhớ C_{t-1}. Giá trị 1 biểu thị "hoàn toàn giữ lại điều này" trong khi giá trị 0 biểu thị "hoàn toàn loại bỏ điều này."



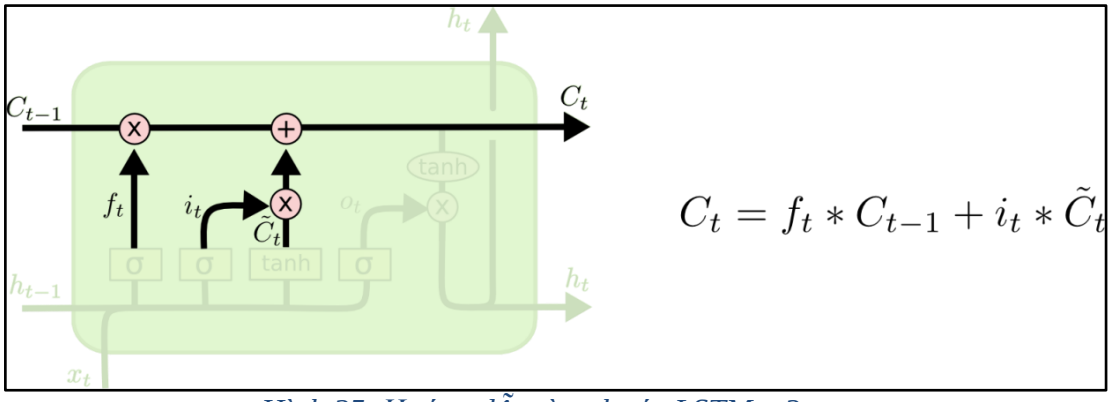
Hình 23. Hướng dẫn từng bước LSTM – 1

Bước tiếp theo là quyết định thông tin mới nào chúng ta sẽ lưu trữ trong trạng thái ô nhớ. Quá trình này bao gồm hai phần. Đầu tiên, một tầng sigmoid gọi là "tầng cổng đầu vào" (input gate layer) quyết định những giá trị nào sẽ được cập nhật. Sau đó, một tầng tanh tạo ra một vector các giá trị ứng cử viên mới, $\tilde{C}_t$, có thể được thêm vào trạng thái.



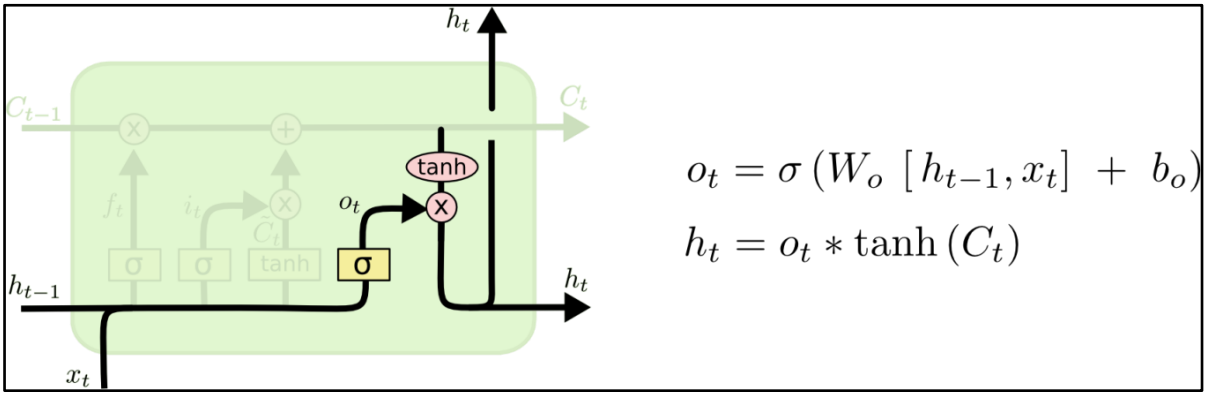
Hình 24. Hướng dẫn từng bước LSTM – 2

Trong bước tiếp theo, chúng ta sẽ kết hợp hai phần này để tạo ra một cập nhật cho trạng thái ô nhớ. Đã đến lúc cập nhật trạng thái ô nhớ cũ là C_{t-1}, thành trạng thái ô nhớ mới là C_t. Các bước trước đó đã quyết định phải làm gì. Chúng ta nhân trạng thái cũ với f_t, để loại bỏ những thông tin mà chúng ta đã quyết định quên trước đó. Sau đó, chúng ta cộng $i_t * C_t$. Đây là các giá trị ứng cử viên mới, được điều chỉnh theo mức độ chúng ta đã quyết định cập nhật mỗi giá trị trạng thái.



Hình 25. Hướng dẫn từng bước LSTM – 3

Cuối cùng, chúng ta cần quyết định những gì sẽ xuất ra. Đầu ra này sẽ dựa trên trạng thái ô nhớ của chúng ta, nhưng sẽ là một phiên bản đã được lọc. Trước tiên, chúng ta chạy một tầng sigmoid để quyết định phần nào của trạng thái ô nhớ sẽ được xuất ra. Sau đó, chúng ta đưa trạng thái ô nhớ qua hàm tanh (để đẩy các giá trị vào khoảng từ -1 đến 1) và nhân nó với đầu ra của tầng cổng sigmoid, sao cho chúng ta chỉ xuất ra những phần mà chúng ta đã quyết định.



Hình 26. Hướng dẫn từng bước LSTM – 4

- **Cửa sổ trượt cho nhận diện giọng nói**

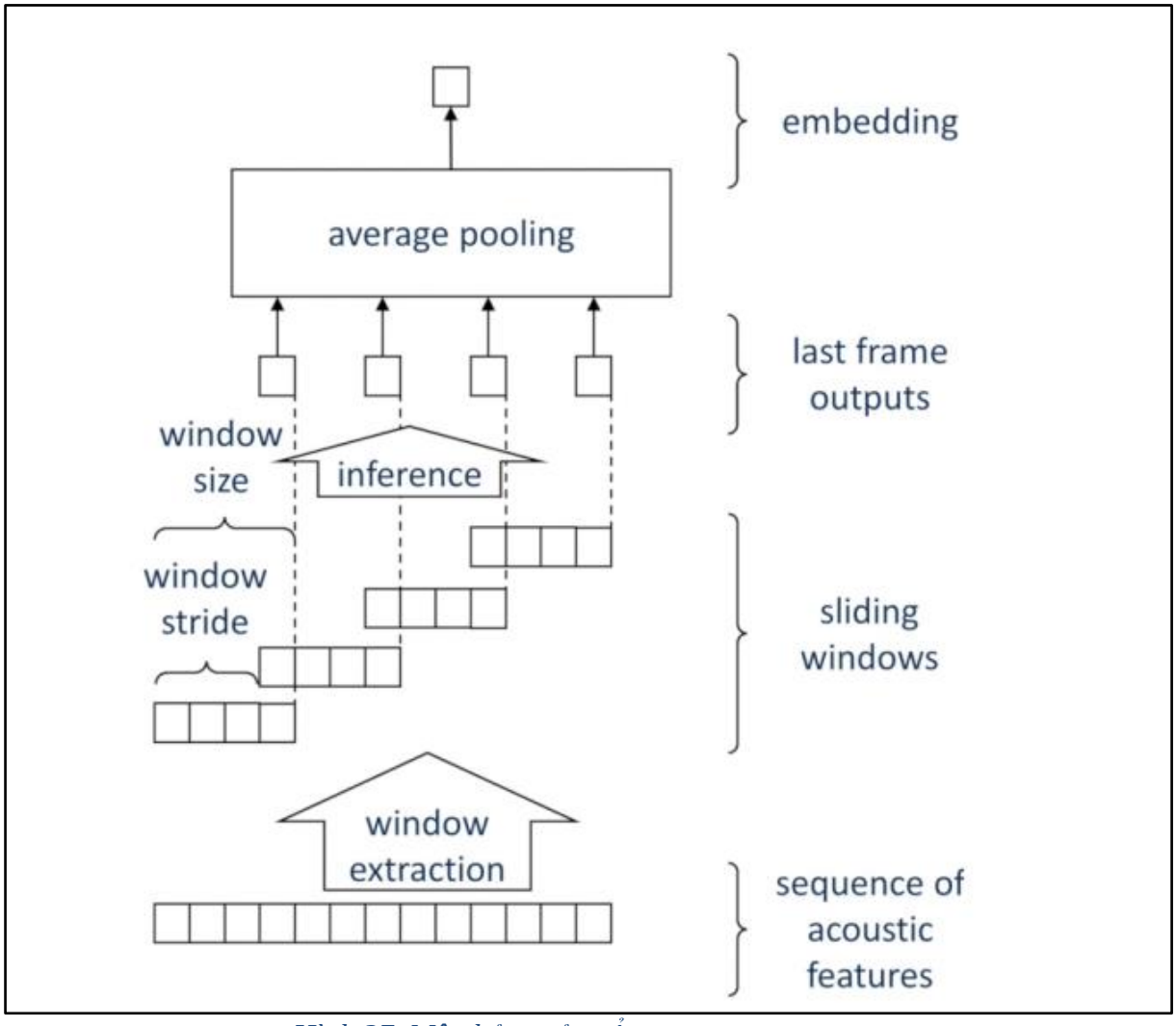
  - Phương pháp suy luận cửa sổ trượt, còn được gọi là kỹ thuật cửa sổ trượt, là một phương pháp được sử dụng trong ngữ cảnh của mô hình chuỗi, xử lý ngôn ngữ tự nhiên và các nhiệm vụ thị giác máy tính. Nó bao gồm việc áp dụng một mô hình vào dữ liệu đầu vào theo cách thức cửa sổ trượt, trong đó mô hình xử lý các phân đoạn con hoặc các cửa sổ chồng chéo của dữ liệu đầu vào để đưa ra dự đoán hoặc trích xuất thông tin liên quan.

  - Trong trường hợp của các nhiệm vụ mô hình chuỗi, như tạo ngôn ngữ hoặc phân tích tình cảm, chuỗi đầu vào được chia thành các cửa sổ chồng chéo nhỏ hơn. Sau đó, các cửa sổ này được cung cấp vào mô hình một cách tuần tự, và mô hình tạo ra dự đoán hoặc thực hiện tính toán trên mỗi cửa sổ. Bằng cách trượt cửa sổ qua toàn bộ chuỗi, dự đoán hoặc tính toán có thể được thực hiện cho mỗi cửa sổ, cho phép mô hình nắm bắt các phụ thuộc và mẫu tại các vị trí khác nhau trong chuỗi.

  - Suy luận cửa sổ trượt có thể hữu ích khi xử lý các chuỗi hoặc dữ liệu có độ dài biến đổi, vì nó cho phép mô hình xử lý dữ liệu một cách cục bộ và có ý thức về ngữ cảnh. Tuy nhiên, điều này có thể đưa vào sự quá tải tính toán, vì mô hình cần xử lý các cửa sổ chồng chéo nhiều lần. Do đó, quan trọng là phải đạt được sự cân bằng giữa kích thước cửa sổ, sự chồng chéo và tài nguyên tính toán để đạt được sự đánh đổi mong muốn giữa độ chính xác và hiệu suất.

  - Trong phạm vi dự án, nhóm đã chọn SEQ_LEN = 100, đại diện cho độ dài của một mẫu âm thanh, và SLIDING_WINDOW_STEP = 50, đại diện cho bước nhảy của cửa sổ trượt là 50 mẫu. Việc này giúp đảm bảo rằng số mẫu được lấy không quá ngắn, đồng thời đảm bảo có đủ đặc trưng giọng nói của mỗi người, từ đó tăng độ

chính xác khi dự đoán.



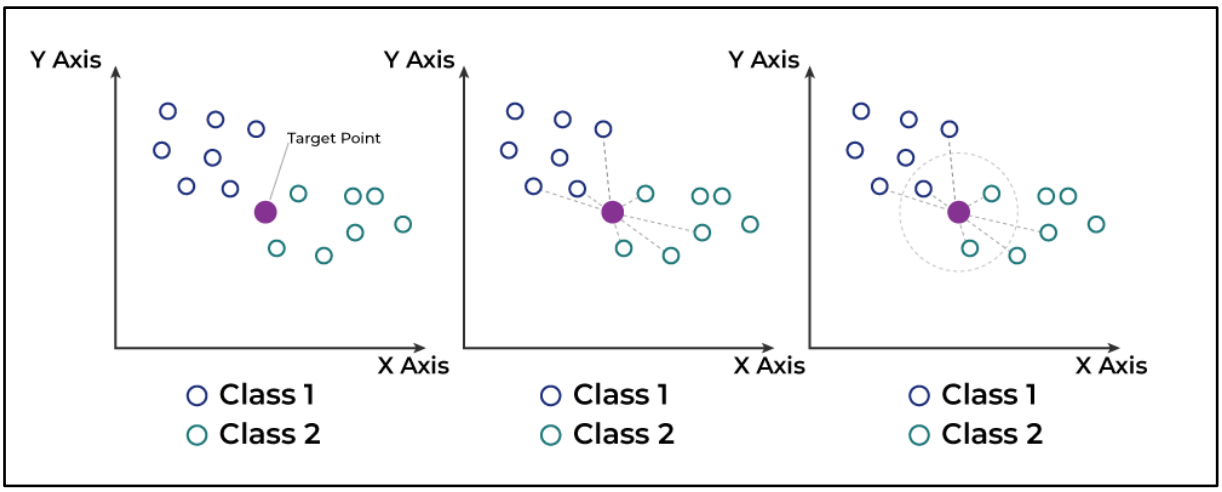
Hình 27. Mô phỏng cửa sổ trượt

- **Kỹ thuật xử lý nhận diện giọng nói**

Việc nhận dạng người nói sử dụng đặc trưng MFCC dễ dàng bị ảnh hưởng bởi chất lượng của microphone và khoảng cách giữa microphone và người nói, tiếng ồn và môi trường thu âm [15]. Vì vậy, những nghiên cứu và thực nghiệm cho thấy ưu tiên hàng đầu trong việc trích xuất đặc trưng MFCC của audio người nói không phải là thời lượng audio mà là số mẫu phải đa dạng và được thu trong nhiều ngữ cảnh và môi trường khác nhau [16].

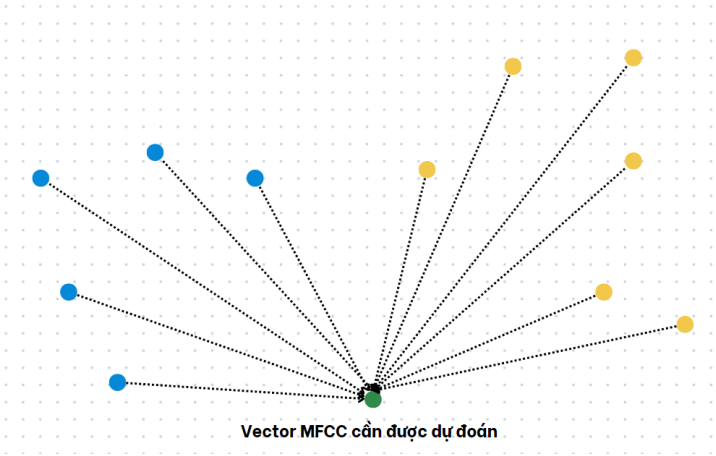
Vì vậy, nếu chỉ sử dụng một vector để làm vector gốc cho người dùng, kết quả sẽ chỉ có độ chính xác cao trên tập test với cùng điều kiện ghi âm nhưng sẽ có độ chính xác thấp với điều kiện ghi âm ngoài trời với sự có mặt của tiếng ồn và nhiễu. Vì vậy, để tăng tính ổn định và hiệu suất của model dự đoán, thay vì sử dụng duy nhất một vector để làm gốc cho mỗi người dùng để dự đoán, nhóm đã cắt nhỏ audio đó thành các phần nhỏ để tạo tính đa dạng cũng như các

audio được thu âm trong nhiều điều kiện khác nhau chứ không chỉ riêng trong môi trường phòng kín hay môi trường ngoài trời .



Hình 28. Mô phỏng thuật toán KNN

Nhóm đã tiến hành thực nghiệm và thống kê sử dụng các kỹ thuật máy học như KNN với tập các vector gốc của người dùng, tuy nhiên kết quả nhận diện còn thấp khi dễ dàng bị nhận diện sai khi một vector outlier của một người khác vô tình nằm gần người đang được nhận diện dẫn đến sai xót trong việc dự đoán. Vì vậy, nhóm đã tiến hành tính trung bình cosine similarity của vector MFCC cần được dự đoán đến tất cả các vector gốc của từng người dùng và lựa chọn ra người dùng có trung bình cosine similarity là nhỏ nhất.



Hình 29. Mô phỏng thuật toán nhận diện giọng nói

2.2.2 Giải pháp nhận diện lời nói

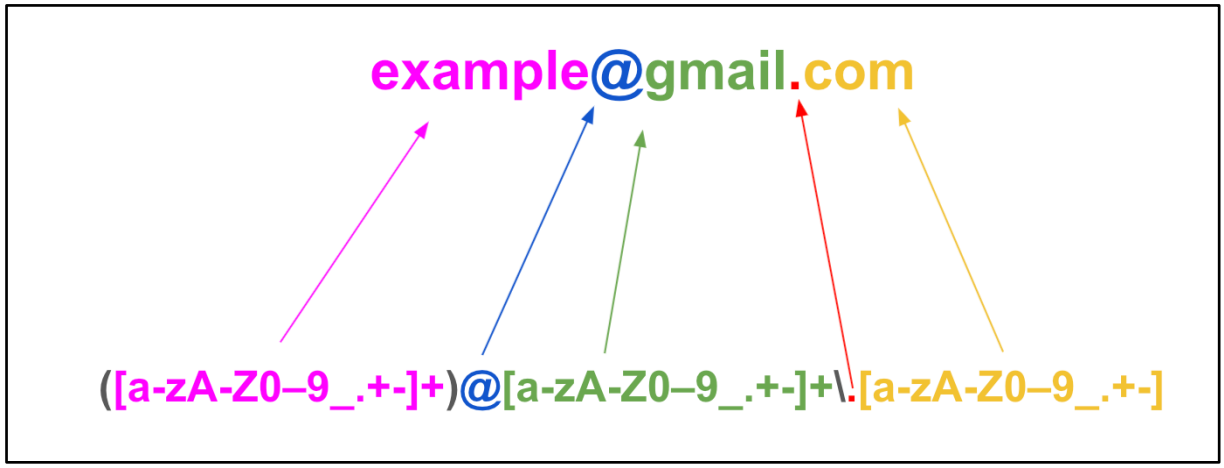
Hình 30. Google Speech-to-Text API

Google Speech-to-Text API là một dịch vụ mạnh mẽ do Google cung cấp, giúp chuyển đổi giọng nói thành văn bản một cách nhanh chóng và chính xác.

Tính năng chính:

- **Chuyển đổi thời gian thực**: Google Speech-to-Text API có khả năng chuyển đổi giọng nói thành văn bản gần như ngay lập tức, hỗ trợ các ứng dụng đòi hỏi thời gian thực như trợ lý ảo và dịch vụ chăm sóc khách hàng.
- **Đa ngôn ngữ**: API hỗ trợ nhiều ngôn ngữ và giọng địa phương khác nhau, bao gồm cả tiếng Việt, giúp người dùng trên khắp thế giới có thể sử dụng dễ dàng.
- **Nhận dạng giọng nói chính xác**:Sử dụng các mô hình học máy tiên tiến, API cung cấp độ chính xác cao trong việc nhận dạng giọng nói, kể cả trong điều kiện tiếng ồn hoặc giọng nói không rõ ràng.
- **Tùy chọn tùy chỉnh**: Người dùng có thể tùy chỉnh mô hình nhận dạng để phù hợp với từ vựng chuyên ngành hoặc ngữ cảnh cụ thể của họ, như các thuật ngữ y tế hoặc kỹ thuật.
- **Định dạng văn bản tự động**: API có khả năng tự động thêm dấu câu, nhận diện đoạn hội thoại và định dạng văn bản một cách chính xác, giúp tạo ra văn bản dễ đọc và hiểu hơn.

2.2.3 Giải pháp nhận dạng từ loại



Hình 31. Ví dụ về Regular Expression

Regular expressions (Regex) là một chuỗi ký tự đặc biệt được sử dụng để tìm kiếm và so khớp các mẫu trong văn bản. Những mẫu này có thể là các chuỗi ký tự cụ thể hoặc một tập hợp các ký tự có cấu trúc nhất định.

Regex cho phép thực hiện các thao tác như tìm kiếm, thay thế, trích xuất và kiểm tra tính hợp lệ của dữ liệu dựa trên các mẫu đã định nghĩa trước. Cụ thể:

- Tìm kiếm (Search): Tìm các chuỗi ký tự hoặc mẫu cụ thể trong văn bản.
- Thay thế (Replace): Thay thế các chuỗi ký tự phù hợp với mẫu bằng một chuỗi khác.
- Trích xuất (Extract): Trích xuất các phần của văn bản phù hợp với mẫu đã xác định.
- Kiểm tra tính hợp lệ (Validation): Kiểm tra xem một chuỗi có tuân theo một định dạng hoặc quy tắc nhất định không.

Regex được ứng dụng rộng rãi trong nhiều tác vụ xử lý văn bản, chẳng hạn như:

- Kiểm tra địa chỉ email hợp lệ: Xác định xem địa chỉ email có tuân thủ các quy tắc định dạng chuẩn không.
- Tìm các từ bắt đầu bằng chữ cái viết hoa: Xác định các từ có chữ cái đầu tiên viết hoa trong văn bản.
- Loại bỏ ký tự không mong muốn: Loại bỏ các ký tự không mong muốn như khoảng trắng thừa, dấu chấm câu, hoặc các ký tự đặc biệt từ một chuỗi.

Regex sử dụng một số ký tự đặc biệt để biểu diễn các mẫu cụ thể:

- Dấu chấm (.): Đại diện cho bất kỳ ký tự nào ngoại trừ dấu xuống dòng.

- Dấu mũ (^): Đại diện cho đầu chuỗi.
- Dấu đô-la ($): Đại diện cho cuối chuỗi.
- Dấu ngoặc vuông ([]): Biểu thị tập hợp các ký tự, trong đó bất kỳ ký tự nào trong tập hợp đều có thể khớp.

Regex cho phép bạn xác định các quy tắc để phù hợp với các mẫu cụ thể:

- Dấu cộng (+): Biểu thị rằng ký tự hoặc nhóm ký tự phải xuất hiện một hoặc nhiều lần.
- Dấu sao (*): Biểu thị rằng ký tự hoặc nhóm ký tự phải xuất hiện không hoặc nhiều lần.
- Dấu ngoặc nhọn ({}): Biểu thị số lần xuất hiện cụ thể của ký tự hoặc nhóm ký tự.

Regex không chỉ giúp tìm kiếm các mẫu trong văn bản mà còn cho phép thay thế các chuỗi phù hợp với mẫu đó bằng một chuỗi khác.

Một số cú pháp cơ bản của Regular Expression cũng như một vài ví dụ cụ thể ở dưới đây:

Bảng 5. Bảng cú pháp cơ bản của Regular Expression

Ký hiệu	Ý nghĩa	Ví dụ	Kết quả
'.'	Bất kì kí tự nào, ngoại trừ '\n'	c.t	'cat', 'cut',…
'*'	Kí tự trước nó được lặp lại ít nhất 0 lần	ab*	'a', 'ab', 'abb', …
'+'	Kí tự trước nó được lập lại ít nhất 1 lần	ab+	'ab', 'abb', 'abbb',… But not 'a'
'?'	Kí tự trước đó có thể được khớp 1 lần hoặc không	'ab?c'	'abc', 'ac'
'[]'	Chỉ định 1 tập hợp các ký tự có thể khớp	'[abc]'	'a', 'b', 'c'
'^'	Điểm bắt đầu của chuỗi	'^Dat'	'Dat' ở đầu chuỗi
'$'	Điểm kết thúc của chuỗi	'man city$'	'man city' ở cuối chuỗi
\s	Bất kì ký tự có khoảng trắng	\sa	[space]a, \ta, \na
\S	Bất kì kí tự nào không phải khoản trắng	\SF	aF, bF,…. But not \tF, \nF,…
\b	Từ biên	ion\b	connection, transistion,…
\B	Bất kì vị trí nào không phải là biên	\BX\B	EXCAPE, EXCLUSIVE,… . Bất kì từ nào có X ở giữa

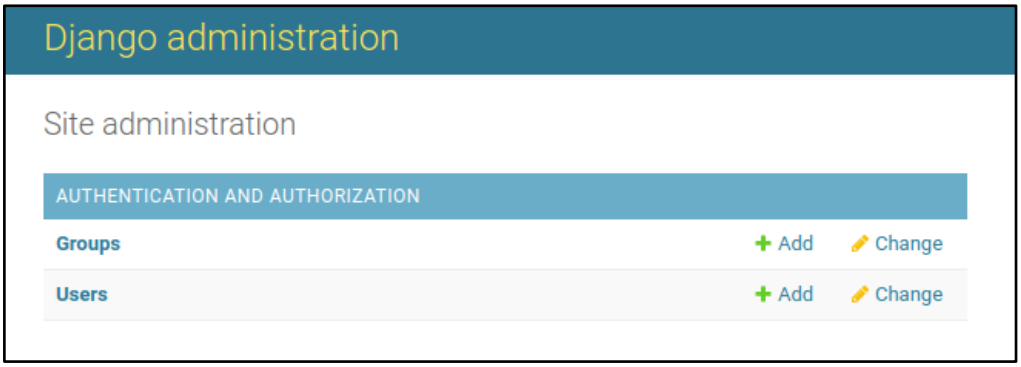
2.3. Giải pháp phần mềm

2.3.1 Phát triển bài toán

Xây dựng một website cho phép người dùng đại diện của căn hộ, với vai trò admin, quản lý các thành viên trong gia đình. Mục đích chính là cung cấp công cụ để admin có thể tạo, thêm, chỉnh sửa, hoặc xóa thông tin thành viên một cách dễ dàng và hiệu quả. Đặc biệt, khi tạo một thành viên mới, người dùng cần tải lên các file âm thanh của chính thành viên đó. Hệ thống sẽ trích xuất đặc trưng MFCC (Mel-Frequency Cepstral Coefficients) từ file âm thanh này và lưu trữ vector embedding vào cơ sở dữ liệu, giúp việc nhận dạng và so sánh âm thanh sau này được thực hiện nhanh chóng và chính xác.

2.3.2 Công nghệ sử dụng

Django admin: quản lý các thành viên, thiết bị và phân quyền trong gia đình thông qua giao diện admin có sẵn do Django cung cấp.

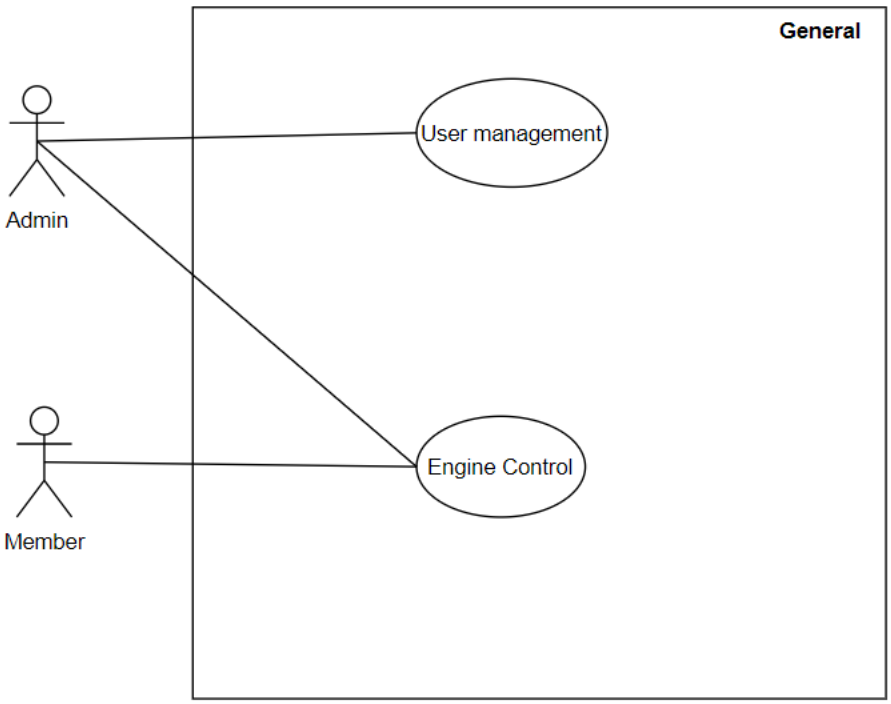


Hình 32. Giao diện Django Administrator

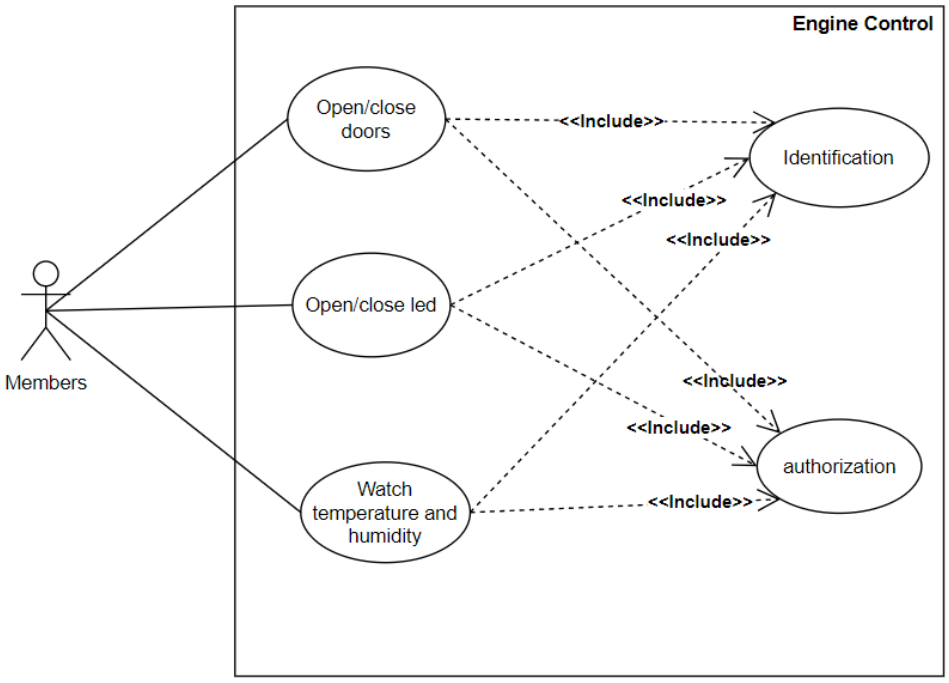
DbSQLite3: Lưu thông tin các thành viên trong gia đình, và phân quyền của chính họ. Ngoài ra vì chỉ sử dụng nội bộ nên SQLite nhẹ và dễ sử dụng, phù hợp với dự án.

Hình 33. Hệ quản trị cơ sở dữ liệu SQLite3

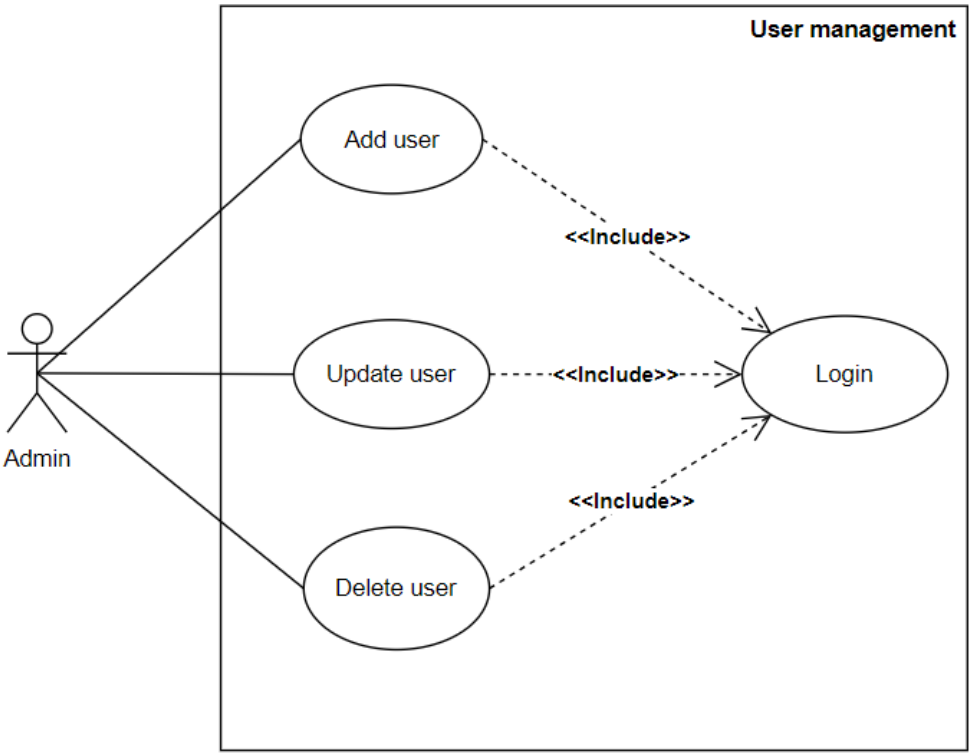
2.3.3 Biểu đồ usecase hệ thống



Hình 34. Sơ đồ Usecase tổng quan

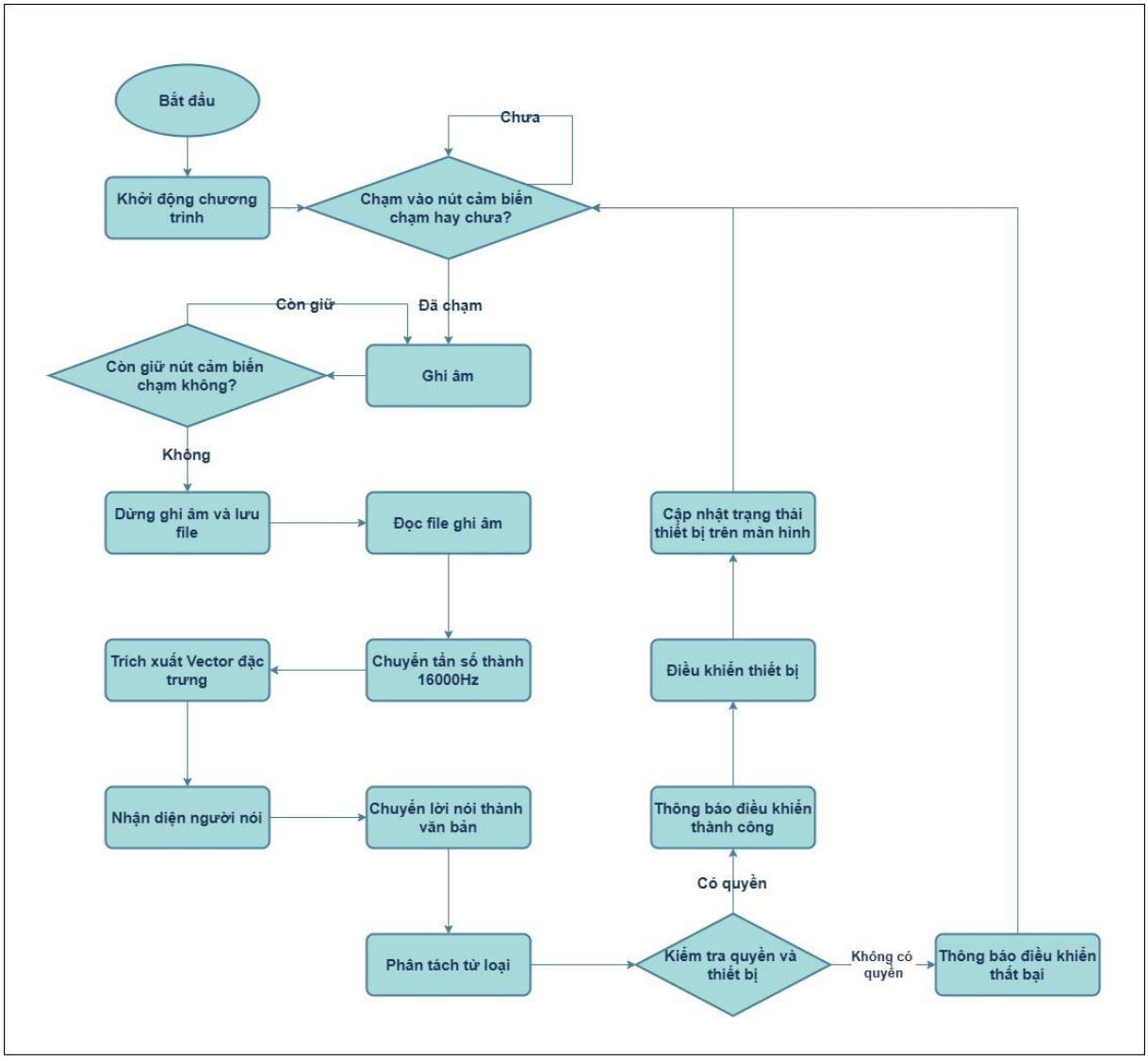


Hình 35. Sơ đồ Usecase Điều khiển động cơ



Hình 36. Sơ đồ Usecase Quản lý người dùng của Administrator

2.3.4 Sơ đồ khối hệ thống



Hình 37. Sơ đồ hệ thống.

3. Kết quả

3.1 Nhận diện giọng nói

3.1.1 Tập dữ liệu

Đối với bài toán nhận diện giọng nói, nhóm sử dụng tập dữ liệu LibriSpeech là một bộ sưu tập khoảng 1.000 giờ sách nói thuộc dự án LibriVox. LibriSpeech được phát triển bởi OpenSLR với tất cả dữ liệu được thu thập bởi sinh viên nghiên cứu của ông. Phần lớn sách nói đến từ dự án Gutenberg. [17]

Dữ liệu huấn luyện được chia thành 3 phần: bộ 100 giờ, bộ 360 giờ và bộ 500 giờ. Trong khi đó, dữ liệu phát triển và kiểm tra được chia thành các hạng mục 'clean' và 'other', tùy thuộc vào mức độ dễ hay khó mà các hệ thống Nhận dạng Giọng nói Tự động sẽ phải đối mặt. Mỗi bộ dữ liệu phát triển và kiểm tra dài khoảng 5 giờ âm thanh. Tập dữ liệu này cũng cung cấp các mô hình ngôn ngữ n-gram và các văn bản tương ứng được trích xuất từ sách của dự án Gutenberg, bao gồm 803 triệu tokens và 977 nghìn từ độc nhất. LibriSpeech được sử dụng trong nhiều ứng dụng như nhận dạng người nói và xác minh người nói tự động. OpenSLR (Nguồn tài nguyên ngôn ngữ và giọng nói mở) có 93 SLRs trong lĩnh vực phần mềm, âm thanh, nhạc, giọng nói, và tập dữ liệu văn bản mở để tải về. Tập dữ liệu LibriSpeech là SLR12, ghi âm giọng nói tiếng Anh. Định dạng file dữ liệu là FLAC (Free Lossless Audio Codec) không làm mất chất lượng hoặc mất bất kỳ dữ liệu âm thanh gốc nào.

Bảng 6. Thống kê tập dữ liệu LibriSpeech

subset	hours	per-spk minutes	female spkrs	male spkrs	total spkrs
dev-clean	5.4	8	20	20	40
test-clean	5.4	8	20	20	40
dev-other	5.3	10	16	17	33
test-other	5.1	10	17	16	33
train-clean-100	100.6	25	125	126	251
train-clean-360	363.6	25	439	482	921
train-other-500	496.7	30	564	602	1166

Sau khi huấn luyện mô hình LSTM với tập dữ liệu LibriSpeech nêu trên với tập train-

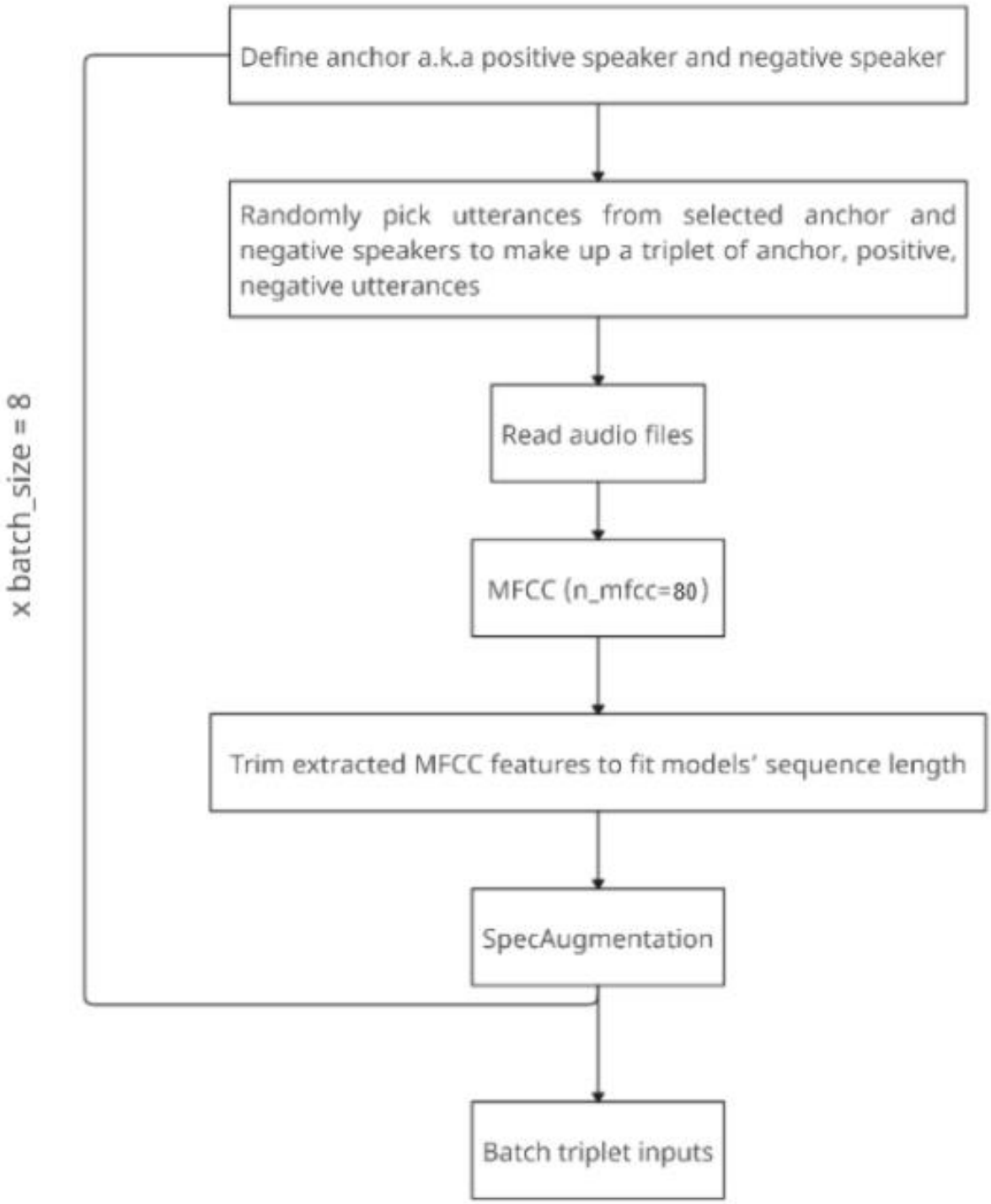
clean-360 với 363.6 giờ đọc sách, nhóm tiến hành xây dựng bộ dữ liệu để làm vector đối sánh cho từng người nói. Ban đầu bộ dữ liệu để tạo vector đối sánh cho từng người nói được xây dựng là một audio dài 5 phút nhưng qua thực nghiệm và thống kê không mang lại tích cực, vì vậy nhóm đã chia nhỏ ra thành nhiều audio nhỏ với độ dài từ 1 giây đến 3 giây và các thực nghiệm cho thấy kết quả trả về được cải thiện rất nhiều (Nhóm đã đề cập chi tiết giải pháp ở phần 2.2)

Về bộ dữ liệu tập test, nhóm tự xây dựng 2 bộ dữ liệu kiểm thử gồm có:

- Bộ kiểm thử trong điều kiện lý tưởng bao gồm 300 audio tương ứng với 3 người nói được chia cùng với bộ dữ liệu dùng để tạo ra vector đối sánh từ một audio dài duy nhất với mỗi người nói. Các audio trong bộ kiểm thử này và audio dùng để đối sánh nhìn chung có cùng giọng điệu, tốc độ đọc và điều kiện môi trường giống nhau.
- Bộ kiểm thử được thu âm thực tế cùng với nhà thông minh bao gồm 120 audio tương ứng với 3 người nói với audio cho từng người nói được thu âm ở điều kiện ngoài trời. Các audio trong bộ kiểm thử này nhìn chung đa dạng về giọng điệu, tốc độ đọc và điều kiện môi trường có nhiễu và tiếng ồn.

3.1.2 Huấn luyện mô hình

Sau khi có được các bộ dữ liệu và phân chia dữ liệu theo tỉ lệ phù hợp, nhóm tiến hành huấn luyện mô hình LSTM để giải quyết bài toán nhận diện giọng nói.



Hình 38. Quy trình huấn luyện mô hình

Trong mỗi bước của giai đoạn huấn luyện, một batch đầu vào với kích thước batch bằng 8 sẽ được nạp vào các mô hình. Mỗi một trong tám phần tử đầu vào này là một bộ ba bao gồm các đặc trưng của anchor, các đặc trưng tích cực và các đặc trưng tiêu cực, đã được cắt tỉa để tuân thủ độ dài chuỗi được chỉ định của các mô hình sử dụng.

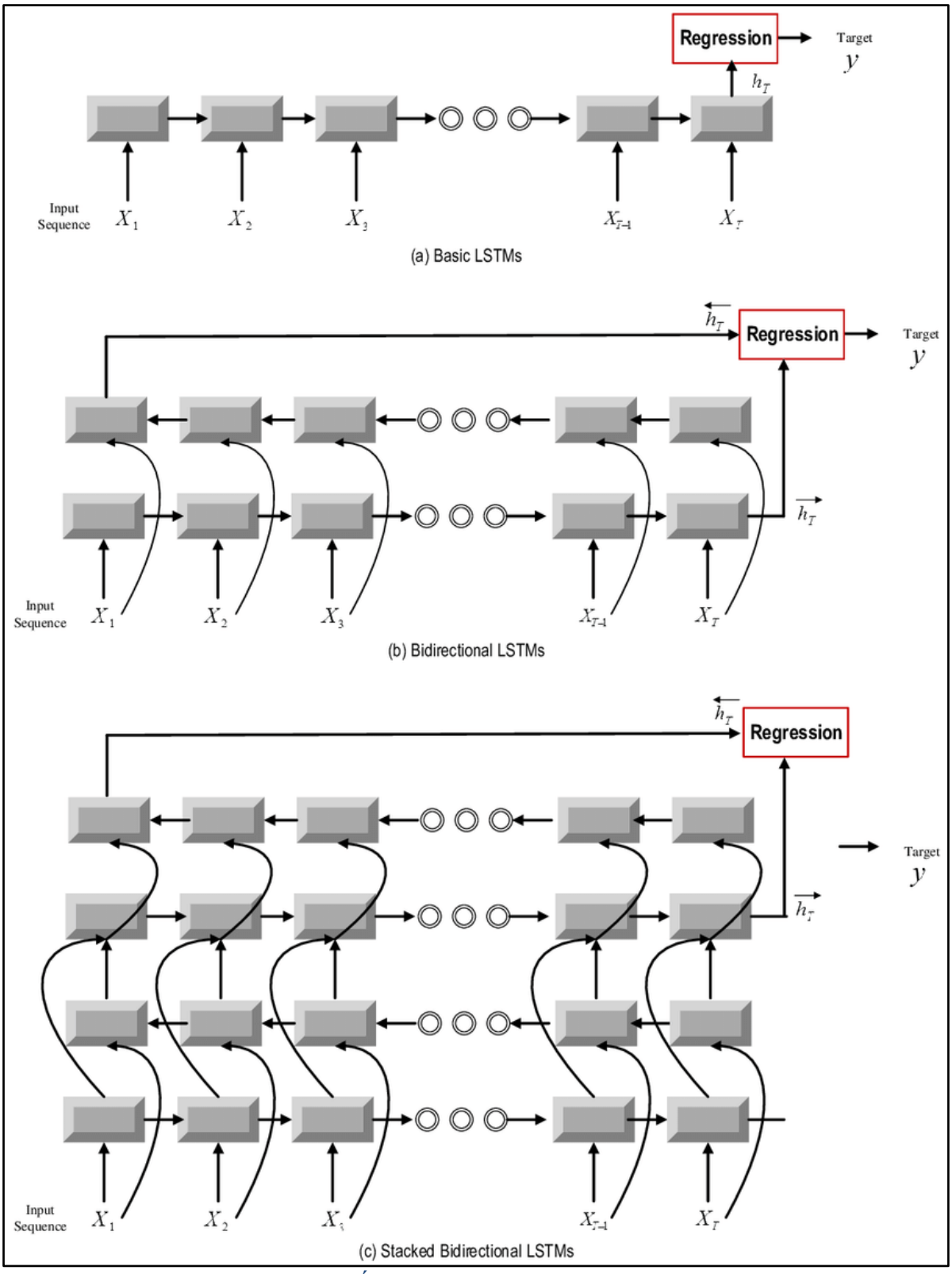
Đầu tiên, từ tập dữ liệu mục tiêu (LibriSpeech), nhóm tạo một từ điển mà khóa là mã định danh của người nói và giá trị là đường dẫn lưu trữ các phát ngôn của họ. Từ danh sách này, nhóm ngẫu nhiên chọn hai người nói: một người làm anchor/tích cực và một người làm tiêu cực. Bộ ba này s5ẽ là sự kết hợp của ba đường dẫn tệp phát ngôn từ hai người nói.

Nhóm đọc tín hiệu âm thanh trong bộ ba, chuyển đổi chúng từ analog sang số với tần số lấy mẫu 16kHz. Sau đó, tôi lấy các hệ số MFCC bằng cách gọi hàm librosa.feature.mfcc với tham số 'n_mfcc' bằng 80. Đầu vào sẽ được biến đổi thành ma trận kích thước 80x100 trước khi nạp vào mô hình. Nếu tệp âm thanh sau khi trích xuất có kích thước lớn hơn 80x100, nó sẽ được cắt xuống; nếu nhỏ hơn, bộ ba sẽ bị loại bỏ và chọn lại ngẫu nhiên.

Sau khi cắt tỉa, nhóm áp dụng SpecAugmentation (che thời gian và tần số) lên 30% dữ liệu huấn luyện. Các bước tiền xử lý này được lặp lại để tạo ra batch đầu vào kích thước (8 x 3) x 80 x 100, sau đó được sử dụng để huấn luyện các mô hình nhận dạng giọng nói.

Mô hình được sử dụng trong quá trình huấn luyện là mô hình Long Short-Term Memory (LSTM) với các đặc điểm sau:

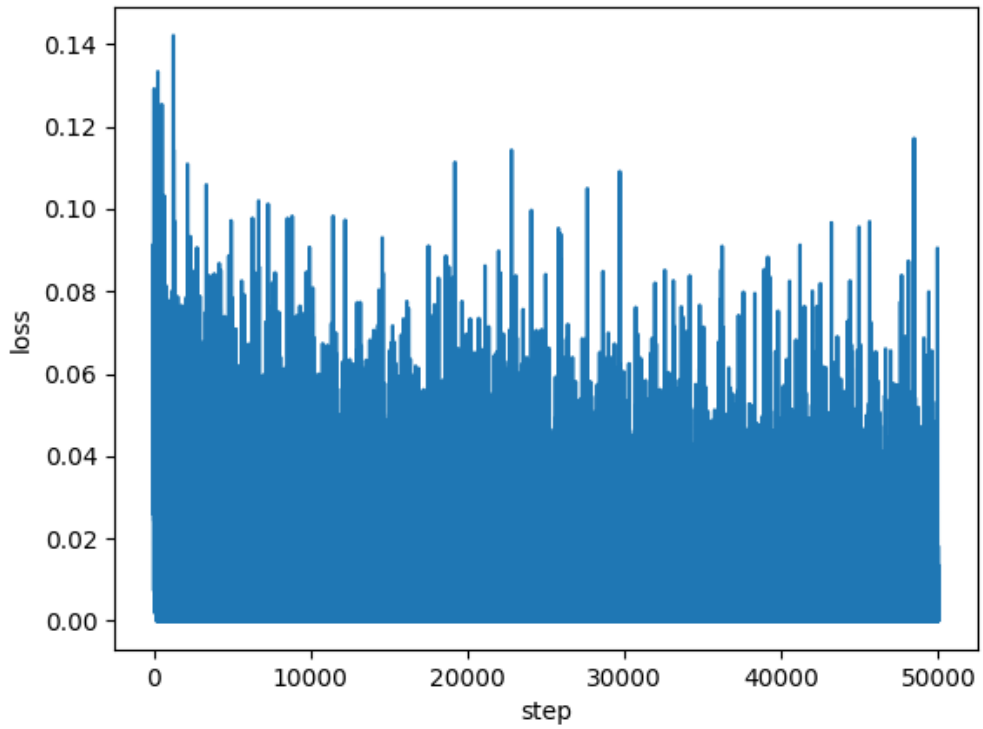
- **Kích thước ẩn của LSTM**: 64
- **Số lượng lớp LSTM**: 3
- **LSTM hai chiều**: Có



Hình 39. Kiến trúc LSTM 3 lớp với bidirectional

Sau quá trình huấn luyện với 50,000 bước trong 18 giờ trên máy tính cá nhân 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 1.38 GHz, mô hình đầu ra với 273408 tham số đã dần cải thiện khả năng phân biệt giữa các giọng nói khác nhau. Triplet Loss được sử dụng để tối ưu hóa khoảng cách giữa các mẫu giọng nói, với ba thành phần chính: anchor, positive và negative. Mục tiêu là giảm khoảng cách giữa anchor và positive trong khi tăng khoảng cách giữa anchor và negative.

Nhìn vào biểu đồ quá trình huấn luyện mô hình, ta có thể thấy rằng giá trị loss giảm dần theo thời gian, cho thấy mô hình đang dần học được các đặc trưng quan trọng và khoảng cách giữa các mẫu đang được tối ưu hóa. Mặc dù vẫn còn một số dao động vì trong mỗi bước huấn luyện, các triplet được lấy random, xu hướng tổng thể là giảm, chứng tỏ mô hình đang tiến bộ trong việc nhận diện giọng nói một cách hiệu quả (thực tế cho thấy ở 10000 bước huấn luyện cuối cùng, tỷ lệ loss đạt giá trị 0 là 82%)

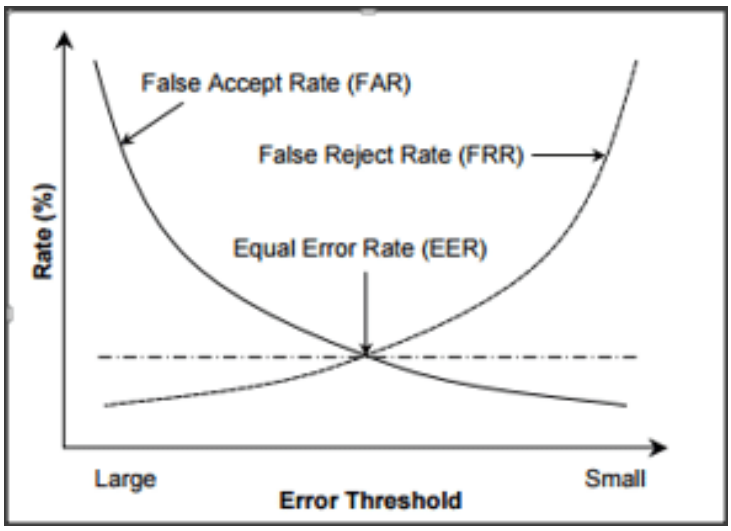


Hình 40. Kết quả Loss của việc Huấn luyện

3.1.3 Kết quả nhận diện

- **Kết quả định tính:**
  - Kết quả dự đoán dễ dàng bị ảnh hưởng bởi chất lượng của micro thu âm. Thực tế cho thấy khi sử dụng vector đối sánh được thu âm bằng laptop cá nhân và audio kiểm thử được thu âm bằng raspberry cho ra kết quả sai lệch hoàn toàn.
  - Kết quả dễ bị ảnh hưởng bởi điều kiện môi trường có tiếng ồn. Kết quả khi kiểm thử ở môi trường yên lặng (trong phòng kín) và ở quán cafe có sự khác biệt tương đối.
  - Giữa những chất giọng vùng miền tương đối giống nhau, model vẫn chưa có kết quả chính xác cao. Cụ thể người nói Phát (quê quán Quảng Nam) dễ bị nhầm lẫn sang Trí (quê quán Đà Nẵng).
  - Việc chia nhỏ audio để làm vector đối sánh đem lại hiệu quả khác biệt.
  - Việc lặp lại audio kiểm thử để tăng mẫu thử có cải thiện độ chính xác.
  - Việc kiểm tra người lạ bằng ngưỡng EER là không tốt khi cosine similarity các lần trả về không có sự ổn định cao.
- **Kết quả định lượng:**

Để đánh giá hiệu suất của mô hình nhận diện người nói, nhóm sử dụng thang đo Equal Error Rate (EER). Đây là một thang đo uy tín và được sử dụng rộng rãi trong lĩnh vực nhận diện người nói [18].



Hình 41. Mô tả thang đo EER

- **Trong bài toán nhận diện người nói:**

Giả sử chúng ta có một hệ thống nhận dạng người nói được đào tạo để phân biệt giữa hai người nói: A và B.

1. False Acceptance Rate (FAR): Đây là tỷ lệ của số lượng mẫu không chính xác của người B được chấp nhận như là người A so với tổng số mẫu của người B trong tập dữ liệu kiểm tra.

2. False Rejection Rate (FRR): Đây là tỷ lệ của số lượng mẫu của người A bị từ chối nhưng thực tế là của người A so với tổng số mẫu của người A trong tập dữ liệu kiểm tra.

3. Equal Error Rate (EER): Đây là điểm trên đồ thị ROC curve nơi tỷ lệ FAR và tỷ lệ FRR là bằng nhau. Nó là một thước đo quan trọng của hiệu suất của hệ thống nhận dạng. EER thường được hiểu là điểm mà khi bạn điều chỉnh ngưỡng quyết định, tỷ lệ lỗi chấp nhận và từ chối sẽ là bằng nhau.

4. Equal Error Rate (EER): Đây là tỷ lệ lỗi giữa tỷ lệ False Acceptance Rate (FAR) và False Rejection Rate (FRR) khi chúng bằng nhau. Nó thường được hiểu là điểm trên đồ thị ROC curve nơi tỷ lệ FAR và tỷ lệ FRR là bằng nhau. EER là một thước đo quan trọng của hiệu suất của hệ thống nhận dạng. Nó được đo lường dưới dạng phần trăm và thấp nhất là tốt nhất, vì nó chỉ ra rằng tỷ lệ lỗi chấp nhận và từ chối là gần như bằng nhau.

5. EER Threshold: Đây là ngưỡng quyết định tương ứng với EER, nơi mà tỷ lệ FAR

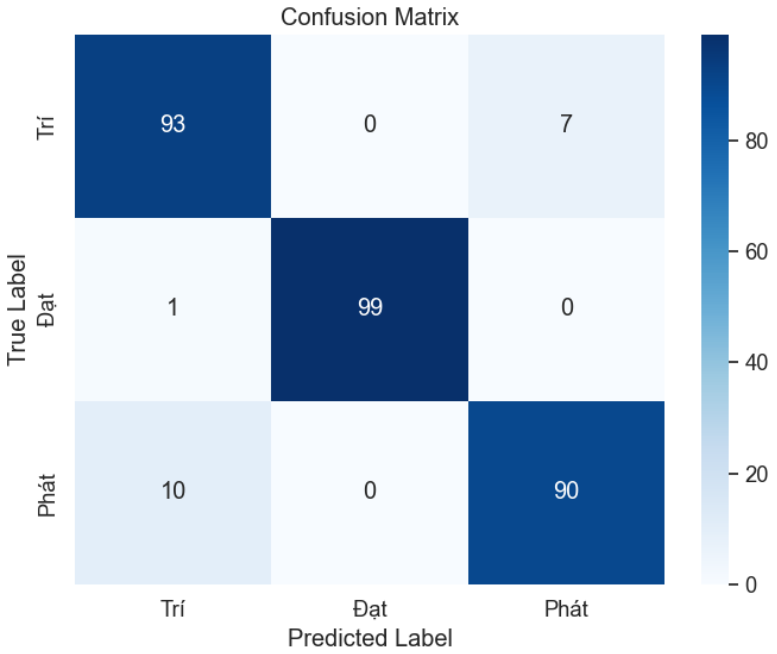
và FRR là bằng nhau. Nó là giá trị ngưỡng mà khi áp dụng cho quyết định của hệ thống nhận dạng, tỷ lệ lỗi chấp nhận và từ chối là gần như bằng nhau. EER Threshold là giá trị quyết định mà khi bạn vượt qua nó, mẫu sẽ được chấp nhận, và khi bạn dưới nó, mẫu sẽ bị từ chối.

Cụ thể, với thang đo đánh giá là EER nêu trên, mô hình đạt giá trị:

- **EER = 0.073**: Đây là tỷ lệ lỗi bằng nhau, tức là tỷ lệ mà tỷ lệ false acceptance rate (FAR) và false rejection rate (FRR) là bằng nhau. EER càng thấp thì mô hình càng có khả năng phân loại chính xác hơn. Trong trường hợp này, EER = 0.073 cho thấy mô hình có mức độ chính xác khá cao khi phân loại.
- **EER Threshold = 0.629**: Đây là ngưỡng (threshold) được sử dụng để tính toán EER. Ngưỡng này cho biết giá trị của độ tin cậy mà mô hình sử dụng để quyết định xác định liệu một mẫu là positive hay negative. Trong trường hợp này, ngưỡng 0.629 cho phép mô hình quyết định một cách khả quan giữa FAR và FRR để đạt được EER là 0.073.

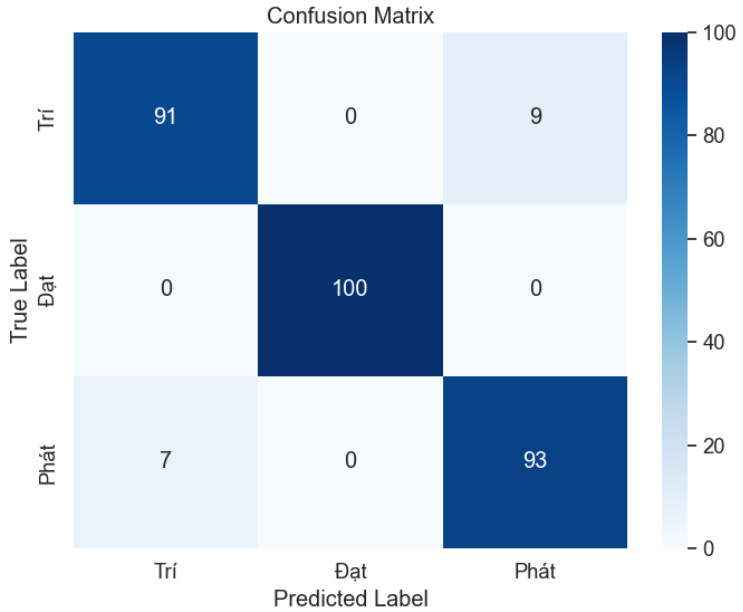
Dựa vào thang đo EER nêu trên cho thấy mô hình có hiệu suất tốt trong việc phân loại và đánh giá mẫu dữ liệu, với mức độ chính xác và sự phân loại cân bằng giữa các loại lỗi (FAR và FRR) khá cao. Đây là một kết quả khả quan trong việc đánh giá hiệu suất của mô hình nhận dạng.

- Đối với tập kiểm thử trong điều kiện lý tưởng bao gồm 300 audio tương ứng với 3 người nói được chia cùng với bộ dữ liệu dùng để tạo ra vector đối sánh từ một audio dài duy nhất với mỗi người nói, ma trận nhầm lẫn đầu ra có dạng:



Hình 42. Ma trận nhầm lẫn kiểm thử trong điều kiện lý tưởng

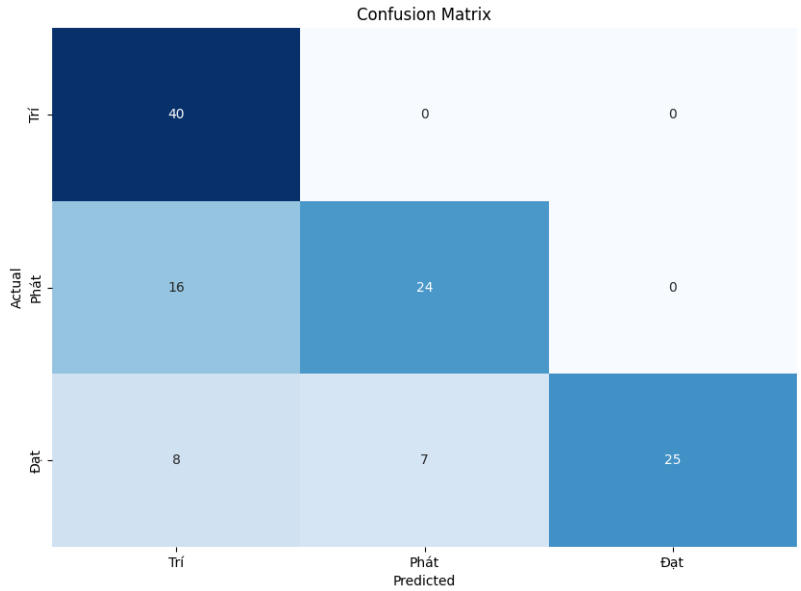
Nhìn chung cho thấy mô hình đạt được độ chính xác rất cao trong điều kiện lý tưởng khi các audio trong bộ kiểm thử này và audio dùng để đối sánh nhìn chung có cùng giọng điệu, tốc độ đọc và điều kiện môi trường giống nhau với độ chính xác 94%.



Hình 43. Ma trận nhầm lẫn kiểm thử trong điều kiện lý tưởng với lặp audio

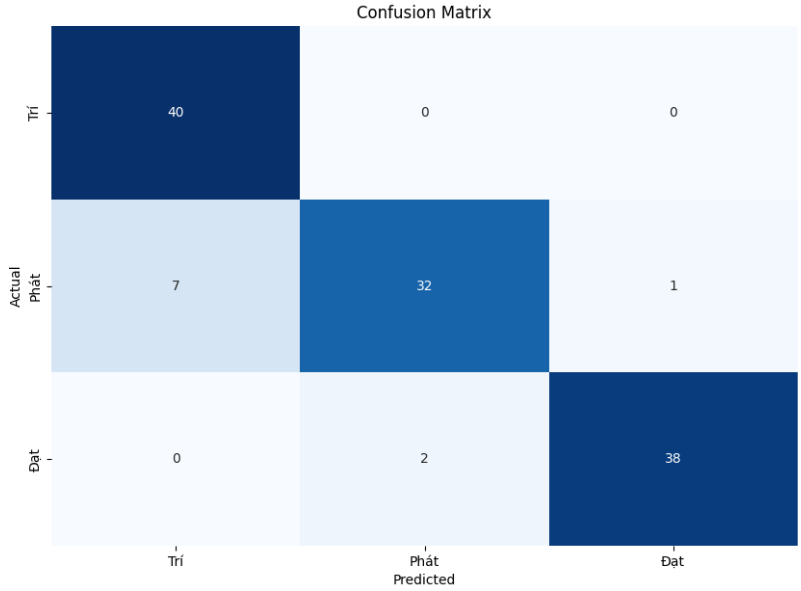
Khi lặp lại audio kiểm thử để tăng số lượng mẫu với N_TIMES_DUPLICATE=5 thì kết quả được cải thiện với độ chính xác là 94.67 %.

Đối với bộ kiểm thử được thu âm thực tế cùng với nhà thông minh bao gồm 120 audio tương ứng với 3 người nói với audio cho từng người nói được thu âm ở điều kiện ngoài trời, ta thấy có sự chênh lệch khá rõ ràng trong độ chính xác. Cụ thể, nếu chỉ sử dụng một audio dài để tạo vector đối sánh, độ chính xác chỉ đạt 74.17% với ma trận nhầm lẫn có dạng như sau:



Hình 44. Ma trận nhầm lẫn kiểm thử trong điều kiện thực tế với 1 file audio

Tuy nhiên khi sử dụng kỹ thuật chia nhỏ audio thành các phần nhỏ để tăng tính đa dạng và tổng quát kết hợp với kỹ thuật tính trung bình cosine similarity, có thể thấy độ chính xác được tăng rõ rệt lên 91.67% với ma trận nhầm lẫn có dạng như sau:



Hình 45. Ma trận nhầm lẫn kiểm thử trong điều kiện thực tế với tập nhiều audio

Bên cạnh thang đo EER và độ chính xác, yếu tố thời gian suy luận cũng là một tiêu chí vô cùng quan trọng. Bảng thống kê cho thấy tốc độ xử lý nhận diện giọng nói gần như đạt yêu cầu thời gian thực để đưa vào trong thực tiễn nhà thông minh.

Bảng 7. Kết quả đo tốc độ thực thi (Raspberry Pi 4 Model B: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz RAM 8GB)

Thời lượng audio kiểm thử (s)	**Số audio kiểm thử**	**Tổng thời gian (s)**	**Thời gian trung bình (s)**
2	300	57.63	0.19
3	300	57.75	0.19
5	300	60.51	0.20
10	300	64.47	0.21

Thời gian được khảo sát nêu trên đa phần là dành cho giai đoạn tính khoảng cách giữa vector cần được dự đoán và tập các vector đối sánh. Thực nghiệm cho thấy thời gian model LSTM nhận audio cần được dự đoán đầu vào và trả về embedding vector là xấp xỉ 0.0 giây, đạt yêu cầu xử lý real time của hệ thống.

3.2 Nhận diện lời nói

3.2.1 Mô hình Whisper

Whisper là một mô hình chuyển giọng nói thành văn bản (speech-to-text) tiên tiến được phát triển bởi OpenAI. Nó nổi bật với khả năng xử lý nhiều ngôn ngữ và cung cấp độ chính xác cao ngay cả trong các điều kiện âm thanh phức tạp.

- **Tính năng chính của Whisper:**

  - **Đa ngôn ngữ**: Whisper hỗ trợ nhiều ngôn ngữ khác nhau, giúp dễ dàng tích hợp và sử dụng trên toàn cầu.
  - **Nhận dạng giọng nói chính xác**: Sử dụng các kỹ thuật học sâu tiên tiến, Whisper có thể chuyển đổi giọng nói thành văn bản với độ chính xác cao, kể cả trong điều kiện tiếng ồn nền.
  - **Khả năng chuyển đổi thời gian thực**: Whisper có thể xử lý và chuyển đổi giọng nói thành văn bản gần như ngay lập tức, hỗ trợ các ứng dụng thời gian thực như trợ lý ảo và dịch vụ trực tuyến.
  - **Tùy chọn tùy chỉnh**: Người dùng có thể tùy chỉnh mô hình để phù hợp với các nhu cầu cụ thể, như từ vựng chuyên ngành hoặc ngữ cảnh cụ thể.
  - **Tự động định dạng văn bản**: Whisper có khả năng tự động thêm dấu câu và định dạng văn bản, tạo ra kết quả dễ đọc và dễ hiểu.

- **Kết quả:**

STT	Model	Detection	Insertion	Total Time(s)
1	Tiny Whisper	0,762946	0,335356	121,304846
2	Base Whisper	0,860437	0,221278	39,25188
3	Small Whisper	0,8839	0,173948	54,972825
4	Medium Whisper	0,90979	0,173948	54,972825
5	Large Whisper	0,911408	0,149272	123,028449
6	Large Whisper v2	0,904126	0,171521	125,639401
7	Large Whisper v3	0,911408	0,149272	123,609138

Hình 46. Kết quả kiểm thử trên dữ liệu điều khiển nhà cửa sử dụng Google Colab

- **Nhận xét:**
    - Kết quả nhận diện của các mô hình nhỏ(Tiny, Base, Small) trên tập dữ liệu kiểm thử có độ chính xác không cao.
    - Kết quả nhận diện của các mô hình lớn hơn(Medium, Large) trên tập dữ liệu kiểm thử khá tốt, tuy nhiên số lượng tham số quá lớn khiến mô hình có thời gian xử lý khá lâu.

3.2.2 Mô hình PhoWhisper

PhoWhisper là một mô hình chuyển đổi giọng nói thành văn bản dựa trên Whisper, được thiết kế đặc biệt để hỗ trợ tiếng Việt một cách chính xác và hiệu quả hơn. Mô hình này tận dụng các ưu điểm của Whisper nhưng được tối ưu hóa cho ngữ cảnh và đặc thù ngôn ngữ tiếng Việt. Các tính năng chính của PhoWhisper tương tự với mô hình Whisper. Điểm mạnh của PhoWhisper so với Whisper chính là hỗ trợ tối ưu mạnh trên dữ liệu tiếng Việt. Tuy nhiên PhoWhisper lại không có hỗ trợ đa ngôn ngữ. Do đó nó chỉ phù hợp với các ứng dụng sử dụng tiếng Việt.

- **Kết quả:**

STT	Model	Detection	Insertion	Total Time(s)
1	Tiny PhoWhisper	0,947006	0,124191	33,487694
2	Base PhoWhisper	0,94822	0,115291	25,329878
3	Small PhoWhisper	0,968447	0,097492	41,617583
4	Medium PhoWhisper	0,965615	0,095874	86,714818
5	Large PhoWhisper	0,977346	0,091424	140,098449

Hình 47. Kết quả kiểm thử trên dữ liệu điều khiển nhà cửa sử dụng Google Colab

- **Nhận xét:**

- Kết quả nhận diện của tất cả mô hình đều khá tốt (trên 94%), cao nhất là mô hình Large PhoWhisper với kết quả 97,7%.
- Kết quả nhận diện của PhoWhisper trên tất cả các mô hình đều tốt hơn các mô hình Whisper.
- Tuy nhiên khi thực hiện kiểm thử trên Raspberry Pi 4 trung bình mất khoảng 10s cho 1 đoạn âm thanh khoảng 2s sử dụng mô hình Base PhoWhisper.

3.2.3 Google Speech-to-Text API

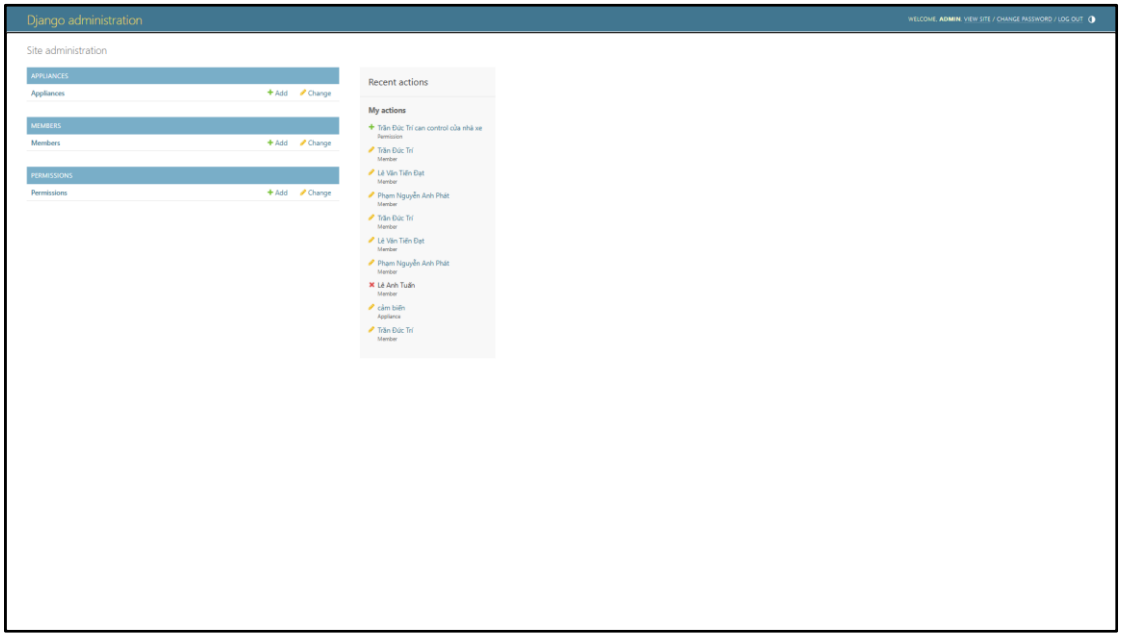
Kết quả:

- Tỉ lệ nhận diện được từ xấp xỉ 99% trên tập dữ liệu điều khiển nhà cửa.

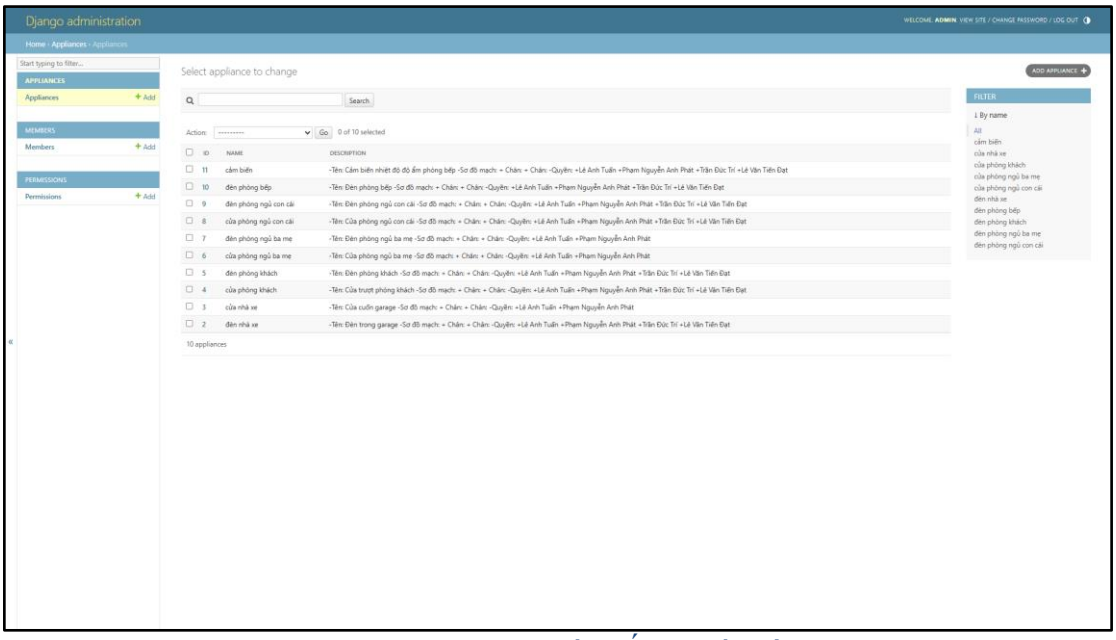
Nhận xét:

- Kết quả nhận diện chính xác và nhanh chóng mất khoảng 0.5s cho 1 đoạn âm thanh dài 2-3s.
- Tuy nhiên cần phải kết nối Wi-Fi để gọi API thực hiện.

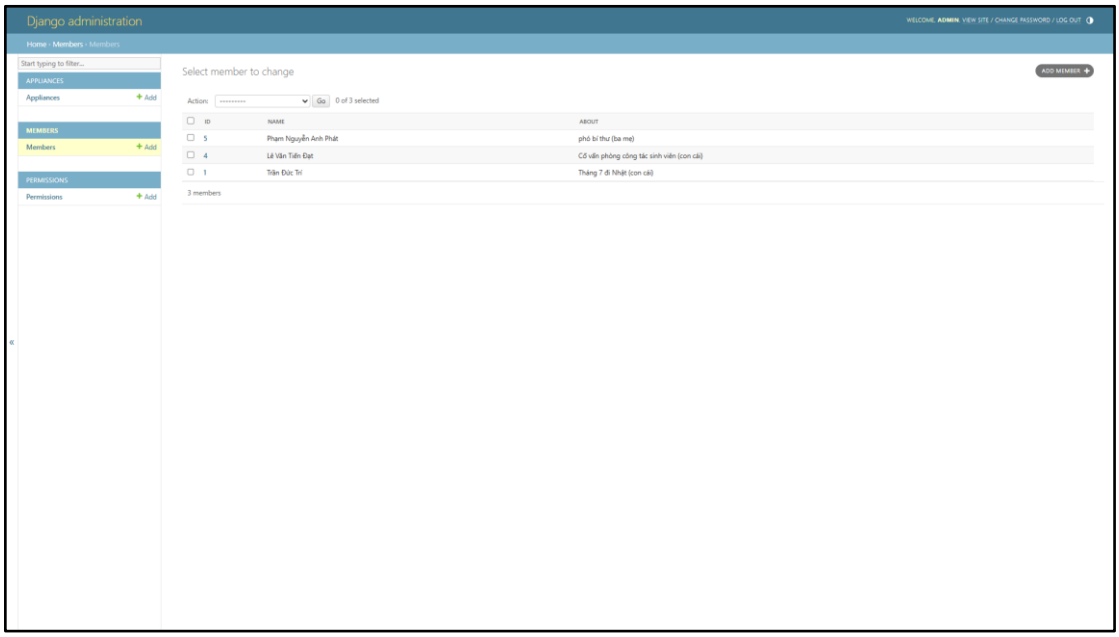
3.4. Website quản lý người dùng



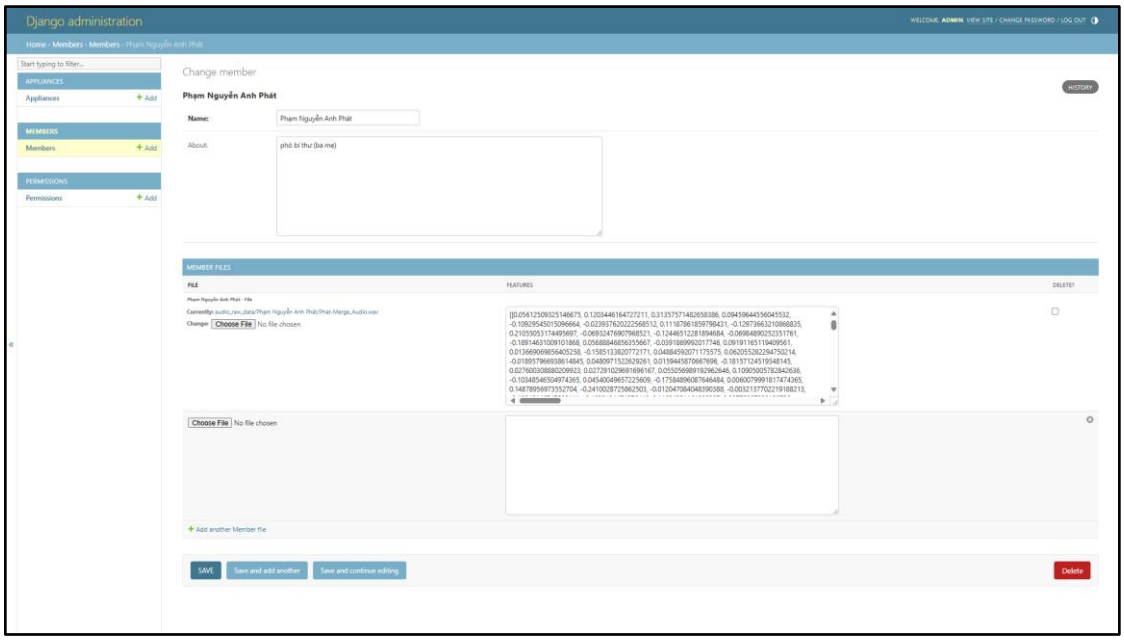
Hình 48. Giao diện quản lý người dùng



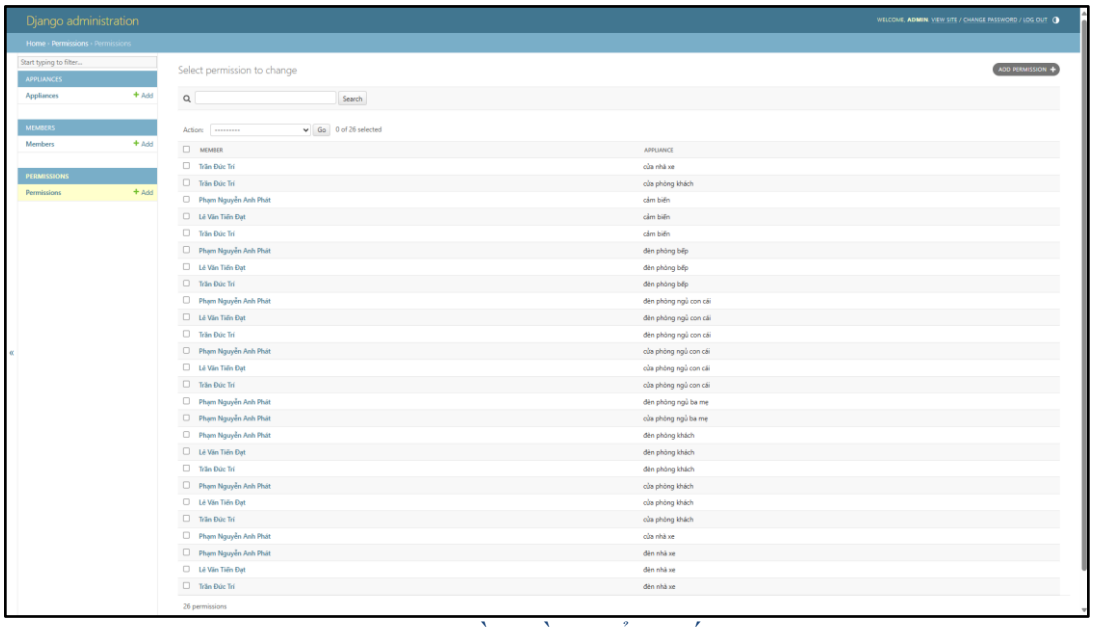
Hình 49. Giao diện quản lý thiết bị nhà thông minh



Hình 50. Giao diện quản lý thành viên trong nhà thông minh



Hình 51. Giao diện thêm audio người nói cho thành viên trong nhà thông minh



Hình 52. Giao diện quản lý quyền điều khiển thiết bị trong nhà thông minh

Nhìn chung, website quản lý người dùng, các thiết bị điện tử và quyền điều khiển thiết bị có giao diện tương đối thân thiện với người dùng, tốc độ phản hồi nhanh khi cập nhật audio để làm vector đối sánh cho người dùng. Tuy nhiên, giao diện vẫn chưa đẹp mắt và cần phát triển thêm.

4. Kết luận

4.1 Đánh giá

- **Phần cứng**: Hệ thống phần cứng được tích hợp đầy đủ các thành phần IoT, thiết kế tinh tế và phù hợp với kiến trúc nhà ở và căn hộ tại Việt Nam. Hệ thống có khả năng cải tiến và mở rộng để bổ sung nhiều thiết bị theo nhu cầu sử dụng.
- **Chức năng nhận diện giọng nói**: Mô-đun nhận diện giọng nói, dù chưa hoàn thiện ở chức năng phát hiện người lạ, đã cho kết quả khá tốt với độ chính xác cao và thời gian phản hồi nhanh chóng, đáp ứng yêu cầu của đồ án.
- **Chức năng nhận diện lời nói**: Sử dụng API từ Google, hệ thống đạt độ chính xác cao trong việc nhận dạng lời nói và tốc độ phản hồi nhanh. Dù tốc độ có thể bị ảnh hưởng bởi đường truyền Internet, nhưng tổng thể vẫn đáp ứng tốt các yêu cầu ban đầu của đồ án.
- **Website**: Giao diện thân thiện và dễ sử dụng với người dùng. Các chức năng hoạt động ổn định, không gặp lỗi, hoàn toàn đáp ứng các yêu cầu của đồ án.

4.2 Hướng phát triển

4.2.1. Nhận diện giọng nói

- Hiện tại, mô hình phát hiện giọng nói của nhóm vẫn chưa đạt được độ chính xác cao. Để cải thiện, chúng tôi sẽ mở rộng dữ liệu huấn luyện với nhiều môi trường và địa điểm đa dạng hơn. Đồng thời, nhóm cũng sẽ tập trung vào việc xử lý nhận diện giọng nói của người lạ, những người không thuộc thành viên gia đình, để tăng cường tính bảo mật và độ tin cậy của hệ thống.
- Nhóm sẽ tích hợp thêm tính năng nhận diện giọng nói Deepfake để ngăn chặn mọi âm mưu giả mạo giọng nói, bảo vệ ngôi nhà khỏi nguy cơ bị đột nhập. Công nghệ tiên tiến này cho phép hệ thống phân biệt được giọng nói thật và giả, giúp phát hiện và ngăn chặn kịp thời các hành vi xâm nhập bất hợp pháp. Bằng cách này, chúng tôi không chỉ nâng cao bảo mật hai lớp đảm bảo tính bảo mật tối đa cho người dùng, mang lại sự yên tâm tuyệt đối trong mọi tình huống

4.2.2. Nhận diện lời nói

Nhóm sẽ phát triển song song hai giải pháp nhận diện lời nói:

- **Sử dụng API nhận diện lời nói của Google**: Đảm bảo khả năng nhận diện chính xác và nhanh chóng khi có kết nối internet.
- **Phát triển mô hình nhận diện nội bộ**: Đảm bảo hệ thống vẫn hoạt động hiệu quả ngay cả khi không có kết nối internet, nhằm tăng cường độ ổn định và khả năng đáp ứng trong mọi tình huống.

4.2.3. Website quản lý người dùng

Nhóm sẽ cải tiến giao diện website quản lý người dùng, hướng đến thiết kế thân thiện và dễ sử dụng hơn. Giao diện mới sẽ giúp người dùng dễ dàng thao tác và quản lý các cài đặt liên quan đến hệ thống.

4.2.4. Phần cứng

- **Sử dụng micro chất lượng cao**: Đảm bảo thu âm giọng nói với chất lượng tốt nhất, giúp tăng cường độ chính xác của hệ thống nhận diện giọng nói.
- **Tích hợp nguồn điện năng lượng mặt trời**: Làm nguồn dự phòng trong trường hợp hệ thống điện chính bị ngắt, đảm bảo hệ thống luôn hoạt động ổn định và liên tục.

Với những cải tiến này, nhóm hy vọng sẽ mang lại trải nghiệm người dùng tốt hơn và nâng cao độ chính xác của hệ thống nhận diện giọng nói.

5. Danh mục tài liệu tham khảo

[1] "Raspberry Pi 4 Tech Specs," Raspberry Pi Foundation, [Trực tuyến]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/. [Đã truy cập 10 06 2024].

[2] "MG90S – Metal Gear Micro Servo Motor," Components101, 30 3 2019. [Trực tuyến]. Available: https://components101.com/motors/mg90s-metal-gear-servo-motor. [Đã truy cập 8 6 2024].

[3] "L298N Motor Driver Module," Components101, 13 4 2021. [Trực tuyến]. Available: https://components101.com/modules/l293n-motor-driver-module. [Đã truy cập 2 6 2024].

[4] "Yellow DC Motor 3-12VDC 2 Flats Shaft," Wiltronics, [Trực tuyến]. Available: https://www.wiltronics.com.au/product/10137/yellow-motor-3-12vdc-2-flats-shaft/. [Đã truy cập 8 6 2024].

[5] Swagatam, "How to Use TTP223 Capacitive Touch Switch Module," Homemade Circuit Projects, 1 2 2024. [Trực tuyến]. Available: https://www.homemade-circuits.com/ttp223-capacitive-touch-module-explained/. [Đã truy cập 8 6 2024].

[6] Components101, "DHT11–Temperature and Humidity Sensor," Components101, 16 7 2021. [Trực tuyến]. Available: https://components101.com/sensors/dht11-temperature-sensor. [Đã truy cập 8 6 2024].

[7] Components101, "28BYJ-48 - 5V Stepper Motor," Components101, 3 5 2021. [Trực tuyến]. Available: https://components101.com/motors/28byj-48-stepper-motor. [Đã truy cập 8 6 2024].

[8] WatElectronics, "ULN2003 MotorDriver IC : PinOut, Specifications, Interfacing & Its Applications," WatElectronics.com, 12 6 2023. [Trực tuyến]. Available: https://www.watelectronics.com/uln2003-motor-driver-ic/. [Đã truy cập 8 6 2024].

[9] *ESP32 Series*, Espressif Systems, 2024.

[10] "3.2inch SPI Module ILI9341 SKU:MSP3218," LCD Wiki, 12 7 2019. [Trực tuyến]. Available: http://www.lcdwiki.com/3.2inch_SPI_Module_ILI9341_SKU:MSP3218. [Đã truy cập 8 6 2024].

[11] "Loa Bluetooth AVA+ MiniPod Y23," Thegioididong, [Trực tuyến]. Available: https://www.thegioididong.com/loa-laptop/loa-bluetooth-ava-plus-minipod-y23. [Đã truy cập 7 6 2024].

[12] "Mel Frequency Cepstral Coefficient (MFCC) trong ASR (Automatic Speech Recognition)," Hamhochoi, 26 08 2020. [Trực tuyến]. Available: https://dothanhblog.wordpress.com/2020/08/26/mel-frequency-cepstral-coefficient-mfcc-trong-asr-automatic-speech-recognition/. [Đã truy cập 05 04 2024].

[13] PEX TEAM, "MFCCs: Engineering features from sound," PEX, 07 08 2020. [Trực tuyến]. Available: https://pex.com/blog/machine-learning-mfccs-engineering-features-from-sound/. [Đã truy cập 9 4 2024].

[14] D. S. P. W. C. Y. Z. C.-C. C. B. Z. E. D. C. và Q. V. L. , "SpecAugment: A Simple Data Augmentation Method," *arXiv,* tập Proc. Interspeech 2019, số arXiv: 1904.08779, pp. 2613-2617, 2019.

[15] N. S. R. K. và R. S. , "MFCC and Prosodic Feature Extraction Techniques: A Comparative Study," *International Journal of Computer Applications,* tập 54, số 1, 2012.

[16] pichenettes, "stackexchange.com," 12 5 2015. [Trực tuyến]. Available: https://dsp.stackexchange.com/questions/18720/determine-the-time-length-of-audio-training-samples. [Đã truy cập 12 6 2024].

[17] TensorFlow Datasets, "TensorFlow Datasets," [Trực tuyến]. Available: https://www.tensorflow.org/datasets/catalog/librispeech. [Đã truy cập 2024 6 13].

[18] M. F.-Z. và E. M.-M. , "State-of-the-art in speaker recognition," *arXiv preprint arXiv:2202.12705,* 2022.