**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ 2**

-----------🙞🙜🕮🙞🙜----------



**BÁO CÁO MÔN HỌC**

**ĐỀ TÀI: THIẾT BỊ ĐIỀU KHIỂN THIẾT BỊ GIA DỤNG BẰNG GIỌNG NÓI**

**Môn học: Xây dựng các hệ thống nhúng**

**Giảng viên: Nguyễn Trọng Kiên**

**Sinh viên thực hiện: Nhóm 13**

**Nguyễn Anh Tuấn**

**Lương Thành Lợi**

**Nguyễn Thành Trung**

**Ngô Cao Hy**

**Võ Anh Kiệt**

**Lê Minh Thông**

*Thành phố Hồ Chí Minh, ngày 04 tháng 06 năm 2025*

**MỤC LỤC**

[LỜI CẢM ƠN 2](#_Toc199870084)

[CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI 3](#_Toc199870085)

[1.Đặt vấn đề 3](#_Toc199870086)

[2.Mục tiêu đề tài 3](#_Toc199870087)

[3. Nội dung đề tài 4](#_Toc199870088)

[CHƯƠNG II: CƠ SỞ LÍ THUYẾT 5](#_Toc199870089)

[1. Tổng quan nội dung 5](#_Toc199870090)

[2. Giới thiệu phần ứng 5](#_Toc199870091)

[3. Giới thiệu phần mềm 7](#_Toc199870092)

[CHƯƠNG III: THIẾT KẾ HỆ THỐNG VÀ KẾT QUẢ 9](#_Toc199870093)

[1.Nguyên lí hoạt động 9](#_Toc199870094)

[2.Quy trình cấu hình trên ESP32 13](#_Toc199870095)

[3.Sơ đồ phương pháp tính Spectogram 15](#_Toc199870096)

[4.Sơ đồ CNN hệ thống 17](#_Toc199870097)

[5.Kiến trúc hoat động web 20](#_Toc199870098)

[CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 21](#_Toc199870099)

[1. Kết luận 21](#_Toc199870100)

[2. Hướng phát triển 22](#_Toc199870101)

# LỜI CẢM ƠN

Lời đầu tiên, nhóm em xin gửi lời cảm ơn chân thành đến Thầy Nguyễn Trọng Kiên. Trong quá trình học tập và nghiên cứu bộ môn Xây dựng các hệ thống nhúng, chúng em đã nhận được sự hướng dẫn, giảng dạy tận tâm và nhiệt huyết từ thầy. Thầy đã giúp chúng em hiểu rõ hơn về những kiến thức quan trọng và thực tế trong lĩnh vực quản lý dự án, từ đó tích lũy thêm nhiều kinh nghiệm bổ ích và kỹ năng cần thiết.

Từ những kiến thức mà thầy truyền đạt, nhóm em đã áp dụng vào việc hoàn thành bài tiểu luận cuối kỳ của môn học. Tuy nhiên, do vẫn còn những hạn chế về kinh nghiệm và hiểu biết trong lĩnh vực này, bài tiểu luận của chúng em khó tránh khỏi những thiếu sót. Rất mong thầy xem xét, đánh giá và góp ý để bài tiểu luận của chúng em được hoàn thiện hơn.

Nhóm em kính chúc thầy luôn dồi dào sức khỏe, thành công trong sự nghiệp giảng dạy, và tiếp tục dìu dắt các thế hệ sinh viên đến với những thành tựu mới trên con đường học tập và phát triển bản thân. Nhóm em xin chân thành cảm ơn

# CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

## Đặt vấn đề

Sự phát triển mạnh mẽ của công nghệ thông tin, đặc biệt là Internet vạn vật (IoT) và trí tuệ nhân tạo (AI), đang mở ra nhiều cơ hội mới trong việc nâng cao chất lượng cuộc sống của con người. Trong đó, việc điều khiển thiết bị bằng giọng nói được xem là một xu hướng nổi bật, giúp con người tương tác tự nhiên và thuận tiện hơn với các thiết bị điện tử xung quanh.

Việc điều khiển thiết bị gia dụng bằng giọng nói mang lại nhiều lợi ích thiết thực, đặc biệt là trong môi trường gia đình thông minh (smart home). Người dùng có thể bật tắt đèn, điều khiển quạt, hoặc các thiết bị điện tử khác chỉ bằng cách nói các lệnh đơn giản, thay vì phải sử dụng công tắc vật lý hay ứng dụng điện thoại. Điều này không chỉ tiện lợi trong sinh hoạt thường ngày mà còn là giải pháp hữu ích cho người khuyết tật, người cao tuổi, hay những người gặp khó khăn trong việc di chuyển.

Với sự phổ biến của các vi điều khiển tích hợp như ESP32, cùng các cảm biến âm thanh kỹ thuật số như INMP441 sử dụng giao tiếp I2S, việc hiện thực hóa một hệ thống điều khiển thiết bị bằng giọng nói trở nên khả thi với chi phí thấp và mức độ tùy chỉnh cao.

Từ đó, nhóm chúng em quyết định chọn thực hiện đề tài “Điều khiển thiết bị gia dụng bằng giọng nói sử dụng ESP32 và cảm biến âm thanh INMP441 I2S”.

## Mục tiêu đề tài

Mục tiêu chính của đề tài là xây dựng một hệ thống điều khiển thiết bị gia dụng bằng giọng nói đơn giản, hoạt động độc lập và hiệu quả. Cụ thể:

Xây dựng hệ thống nhận dạng giọng nói cơ bản để nhận biết một số lệnh đơn giản như “bật đèn”, “tắt đèn”, “quạt”,...

Ứng dụng kỹ thuật xử lý tín hiệu âm thanh, biến dữ liệu âm thanh đầu vào thành phổ tần (spectrogram) để phục vụ cho việc phân tích.

Áp dụng mạng nơ-ron tích chập (CNN) để nhận diện và phân loại các lệnh giọng nói.

Triển khai thực tế hệ thống trên phần cứng ESP32-CAM kết hợp với cảm biến INMP441, từ đó điều khiển thiết bị điện qua relay hoặc mô-đun điều khiển.

## 3. Nội dung đề tài

Để đạt được các mục tiêu trên, đề tài sẽ tập trung thực hiện các nội dung sau:

Nghiên cứu phần cứng và phần mềm liên quan, bao gồm các vi điều khiển ESP32, cảm biến âm thanh INMP441 và các thư viện/phần mềm hỗ trợ xử lý âm thanh.

Thiết kế và hiện thực hệ thống thu âm, phân tích âm thanh, xây dựng mô hình phân loại tín hiệu, kết hợp giữa phần cứng và phần mềm để tạo ra hệ thống hoàn chỉnh.

Thử nghiệm hệ thống điều khiển thiết bị gia dụng, đánh giá hiệu quả hoạt động thực tế, đo độ chính xác trong nhận dạng và mức độ phản hồi của hệ thống.

# CHƯƠNG II: CƠ SỞ LÍ THUYẾT

## 1. Tổng quan nội dung

Đề tài tập trung vào việc xây dựng một hệ thống nhúng có khả năng thu nhận và xử lý lệnh giọng nói của người dùng. Quá trình xử lý bao gồm việc ghi âm, biến đổi tín hiệu âm thanh thành dạng phổ tần (spectrogram), và sử dụng mô hình học sâu (CNN) để phân loại các lệnh. Kết quả phân loại sẽ được dùng để điều khiển thiết bị gia dụng tương ứng.

Hệ thống được chia làm ba phần chính:

Phần cứng thu âm và điều khiển thiết bị: sử dụng cảm biến INMP441 để ghi âm và vi điều khiển ESP32-CAM để xử lý tín hiệu và điều khiển thiết bị.

Phần mềm xử lý tín hiệu âm thanh: tín hiệu âm thanh được cắt nhỏ và chuyển sang phổ tần (spectrogram) bằng thuật toán FFT (Fast Fourier Transform).

Mô hình học sâu CNN: spectrogram được đưa vào mạng CNN để phân loại lệnh và xuất tín hiệu điều khiển thiết bị tương ứng.

## 2. Giới thiệu phần ứng

a. ESP32

Sơ lược: ESP32 là một module tích hợp vi điều khiển ESP32 với khả năng kết nối WiFi và Bluetooth, đi kèm camera OV2640, phù hợp với các ứng dụng giám sát hoặc xử lý ảnh.

Cấu hình chi tiết:

Vi xử lý: ESP32-D0WD lõi kép, tốc độ 240 MHz

RAM: 520KB SRAM

Bộ nhớ flash: 4MB

Kết nối: WiFi 802.11 b/g/n, Bluetooth 4.2

Hỗ trợ thẻ nhớ microSD

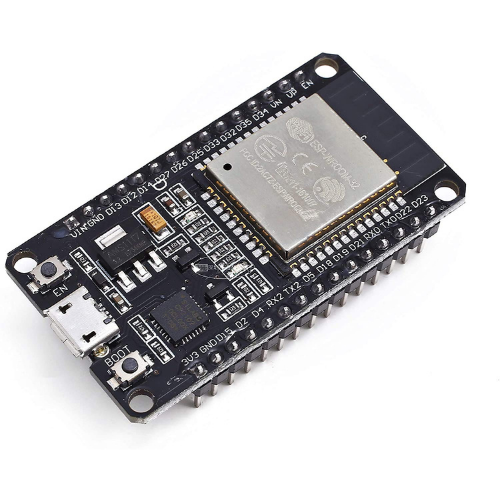
Thông số kỹ thuật:

Điện áp hoạt động: 3.3V

Dòng điện tiêu thụ thấp, tiết kiệm năng lượng

Số chân GPIO: 26 chân

Kích thước: 27mm x 40.5mm



*Hình 1 :ESP32*

b. Cảm biến âm thanh INMP441 (I2S)

Sơ lược: INMP441 là cảm biến âm thanh kỹ thuật số MEMS, giao tiếp I2S, cho phép truyền tín hiệu âm thanh số chất lượng cao, lý tưởng cho các ứng dụng nhận dạng giọng nói.

Cấu hình chi tiết:

Giao tiếp: I2S

Độ phân giải tín hiệu: 24-bit

Hướng thu: đa hướng (omnidirectional), thu âm đồng đều từ mọi phía

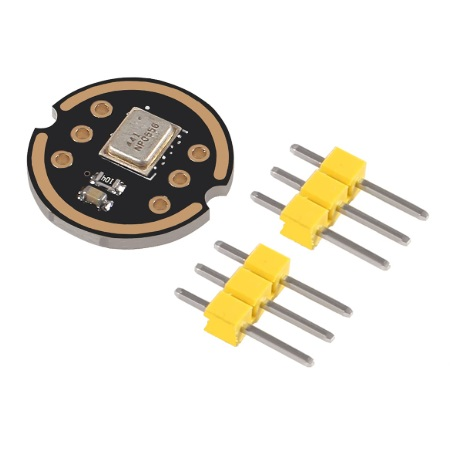
Thông số kỹ thuật:

Điện áp hoạt động: 3.3V

Tần số đáp ứng: 60Hz – 15kHz

Dải động: 120dB

Tín hiệu đầu ra: dạng số (digital I2S)



*Hình 2: Cảm biến âm thanh INMP441*

## 3. Giới thiệu phần mềm

CNN (Convolutional Neural Network)

CNN là một loại mạng nơ-ron tích chập, phổ biến trong các bài toán xử lý ảnh như nhận diện khuôn mặt, phân loại vật thể. Trong đề tài này, CNN được ứng dụng để phân loại lệnh giọng nói dựa trên ảnh spectrogram — tức biểu diễn dạng ảnh của tín hiệu âm thanh.

Quy trình xử lý phần mềm gồm 4 bước chính:

Bước 1: Thu âm

Sử dụng INMP441 để thu tín hiệu âm thanh giọng nói người dùng.

Bước 2: Xử lý tín hiệu

Tín hiệu âm thanh được cắt thành từng đoạn nhỏ.

Mỗi đoạn được biến đổi thành ảnh phổ tần (spectrogram) bằng thuật toán FFT (Fast Fourier Transform), thể hiện sự thay đổi tần số theo thời gian.

Bước 3: Phân loại bằng CNN

Ảnh spectrogram được đưa vào mô hình CNN đã được huấn luyện để phân loại các lệnh như: "bật đèn", "tắt quạt", "mở tivi",...

Bước 4: Điều khiển thiết bị

Kết quả phân loại được xử lý để gửi tín hiệu đến các thiết bị thông qua chân GPIO hoặc module relay, thực hiện hành động tương ứng.

# CHƯƠNG III: THIẾT KẾ HỆ THỐNG VÀ KẾT QUẢ

## Nguyên lí hoạt động

Hệ thống bao gồm hai thành phần người dùng chính:

-Người dùng : là người dùng phát âm lệnh điều khiển thiết bị gia dụng bằng giọng nói.

-Người Giám sát : người giám sát điều khiển trạng thái các thiết bị qua frontend web.

Hệ thống vận hành theo các bước chính sau:

*Bước 1: Ghi nhận dữ liệu âm thanh*

Người dùng phát ra lệnh giọng nói như bật đèn hoặc tắt quạt.

Cảm biến microphone INMP441 trên ESP32 thu thập dữ liệu âm thanh ở tần số lấy mẫu 16kHz.

Dữ liệu âm thanh được thu thập liên tục theo từng đoạn và lưu vào một vùng đệm trên ESP32.

*Bước 2: Chuẩn bị dữ liệu đầu vào cho mô hình AI*

Khi vùng đệm tích lũy đủ dữ liệu âm thanh cho một cửa sổ xử lý, chẳng hạn 3200 mẫu tương ứng 200ms, hệ thống sẽ chuyển dữ liệu sang một luồng xử lý riêng biệt, gọi là Thread 2 hoặc Task Process Audio Window, để thực hiện tiền xử lý.

Dữ liệu âm thanh thô được chuyển đổi sang định dạng float và chuẩn hóa.

Hệ thống kiểm tra mức năng lượng của âm thanh để xác định xem có giọng nói hiện diện hay không. Nếu không có giọng nói, quá trình suy luận sẽ bị bỏ qua để tiết kiệm tài nguyên.

*Bước 3: Tiền xử lý dữ liệu và dự đoán bằng mô hình CNN*

Luồng xử lý Thread 2 thực hiện các bước tiền xử lý tín hiệu số trên dữ liệu âm thanh:

Chia cửa sổ âm thanh thành các phân đoạn chồng lấn.

Áp dụng cửa sổ Hann cho mỗi phân đoạn.

Thực hiện phép biến đổi Fourier nhanh FFT trên mỗi phân đoạn để chuyển đổi dữ liệu từ miền thời gian sang miền tần số.

Tính toán phổ năng lượng và tạo ra ma trận Spectrogram với kích thước 257x11 làm đầu vào cho mô hình AI.

Ma trận Spectrogram này được đưa vào mô hình học sâu CNN đã được tích hợp sẵn trên ESP32 để dự đoán.

Mô hình xuất ra kết quả phân loại, gọi là best\_predicted\_class, là một trong các lệnh đã được huấn luyện như bat, den, hai, mot, ba, tat.

*Bước 4: Điều khiển thiết bị gia dụng*

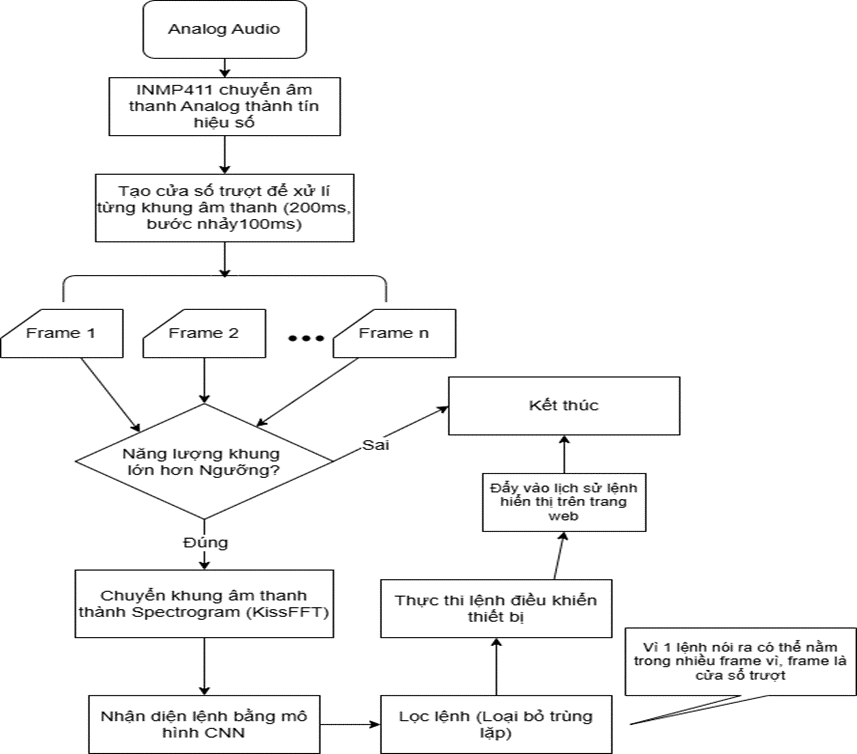
Dựa trên kết quả dự đoán từ mô hình CNN, ESP32 sẽ kích hoạt tín hiệu điều khiển tương ứng đến thiết bị gia dụng.

*Bước 5: Hiển thị trạng thái và log hoạt động*

Giao diện web hoặc ứng dụng di động Frontend sẽ tự động gửi yêu cầu truy xuất dữ liệu từ Backend.

Hiển thị danh sách các lệnh đã được nhận diện và các hành động điều khiển thiết bị tương ứng.

Cho phép người quản lý theo dõi trạng thái hiện tại của các thiết bị hoặc xem lại lịch sử các lệnh đã thực thi.



*Hình 3 :Sơ đồ tổng quan hệ thống*

Quy trình xử lý (chi tiết từng bước)

*[Step 1]: Hành động từ người dùng*

Người dùng phát lệnh giọng nói.

Thread 1 trên ESP32, gọi là Task Read I2S, thu thập dữ liệu âm thanh từ microphone.

Ghi dữ liệu âm thanh với mẫu 16-bit vào một buffer sau mỗi chu kỳ lấy mẫu, ví dụ 50ms.

*[Step 2]: Khi có đủ dữ liệu âm thanh cho một cửa sổ xử lý, chẳng hạn 3200 mẫu tương ứng 200ms, thì tạo Data buffer.*

Thread 1 dừng ghi tạm thời và nhường quyền xử lý cho Thread 2.

*[Step 3]: Thread 2 trên ESP32, gọi là Task Process Audio Window, thực hiện tiền xử lý và dự đoán*

Chuẩn hóa dữ liệu âm thanh: Chuyển đổi dữ liệu âm thanh từ định dạng int16\_t sang float trong khoảng từ -1.0 đến 1.0.

Kiểm tra hiện diện giọng nói: Nếu không có giọng nói, bỏ qua bước suy luận và chuyển sang cửa sổ tiếp theo.

Tính toán Spectrogram: Dữ liệu âm thanh được xử lý bằng FFT để tạo ra ma trận Spectrogram, là input\_data cho mô hình AI.

Chạy mô hình AI: Ma trận Spectrogram được đưa vào mô hình CNN đã được tích hợp trên ESP32.

Nhận kết quả dự đoán: Mô hình trả về các xác suất cho từng lớp lệnh. Hệ thống xác định best\_predicted\_class là lớp có xác suất cao nhất và độ tin cậy.

*[Step 4]: Điều khiển thiết bị gia dụng*

Dựa trên best\_predicted\_class, ví dụ bat den, tat quat, mo tivi:ESP32 kích hoạt tín hiệu điều khiển qua các chân GPIO

*[Step 5]: Backend nhận kết quả và lưu log*

*[Step 6]: Frontend hiển thị log hoạt động*

## Quy trình cấu hình trên ESP32

A screenshot of a computer

Description automatically generated

*Hình 4:Sơ đồ mô tả luồn thực thi hệ thống*

ESP32 được sử dụng để thu nhận và xử lý dữ liệu âm thanh từ microphone, nhận diện lệnh giọng nói và điều khiển thiết bị gia dụng. Hệ thống chia thành ba nhiệm vụ chính (Task), được phân bổ trên hai lõi để tối ưu hóa hiệu suất và đảm bảo xử lý thời gian thực.

Thành phần chính trong sơ đồ

*Core 1:*

Task 1: Lấy âm thanh từ IINMP441 (8kHz, mono, 16bit PCM)

Task 1 chạy trên Core 0 và chịu trách nhiệm thu nhận dữ liệu âm thanh từ microphone IINMP441.

Tần số lấy mẫu: 8Hz, định dạng mono, 16-bit PCM .

Dữ liệu âm thanh được ghi vào một hàng đợi (audioQueue) để chuyển tiếp sangTask 2.

Task 1 hoạt động liên tục, đảm bảo dữ liệu âm thanh được thu nhận không bị gián đoạn.

*Core 0:*

Task 2: Nhận âm thanh từ audioQueue, tạo spectrogram (mỗi khung 200ms, step 100ms)

Task 2 chạy trên Core 0 và nhận dữ liệu âm thanh từ audioQueue do Task 1 cung cấp.

Dữ liệu âm thanh được chia thành các khung (frame) với độ dài mỗi khung là 200ms và bước trượt (step) là 100ms, đảm bảo các khung có sự chồng lấn để không bỏ sót thông tin.

Task 2 thực hiện tiền xử lý tín hiệu âm thanh, bao gồm tính toán phổ năng lượng và tạo ra ma trận Spectrogram (dữ liệu đầu vào cho mô hình nhận diện giọng nói).

Spectrogram sau khi tạo được đưa vào một hàng đợi khác (commandQueue) để Task 3 xử lý tiếp.

*Core 0:*

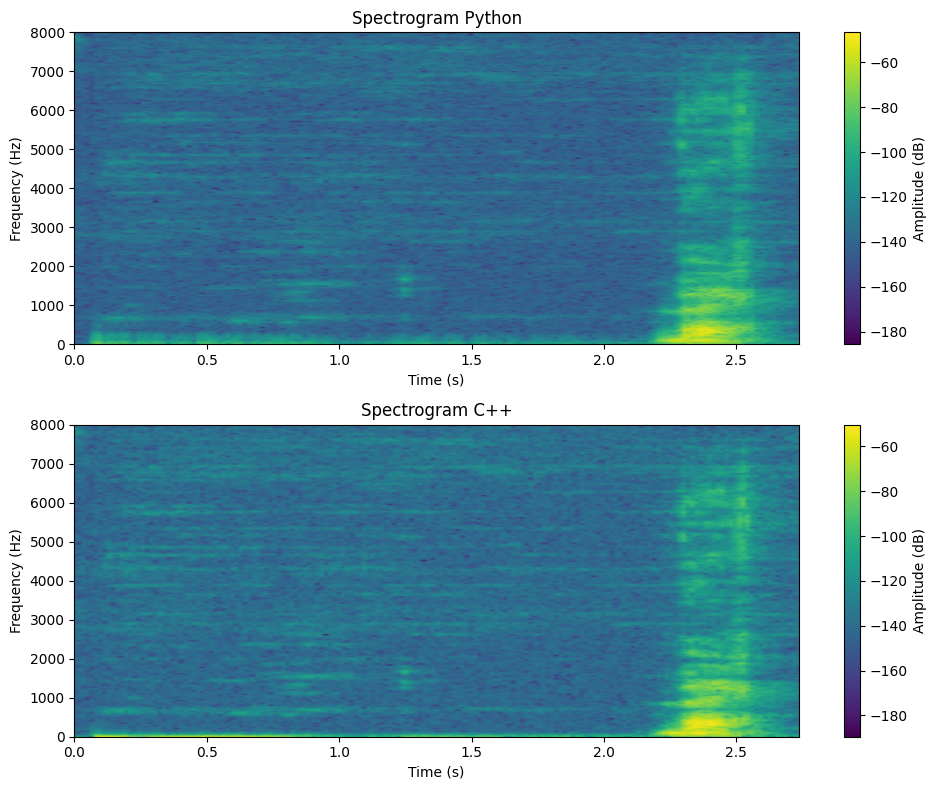
Task 3: Lấy dữ liệu thực thi từ commandQueue, thực thi lệnh với 74HC595

Task 3 cũng chạy trên Core 1, nhận dữ liệu từ

Task 3 sử dụng mô hình AI (CNN) để nhận diện lệnh từ Spectrogram, ví dụ như "bat den", "tat quat".

Sau khi nhận diện, Task 3 gửi tín hiệu điều khiển đến thiết bị gia dụng thông qua IC 74HC595.

## Sơ đồ phương pháp tính Spectogram



*Hình 5: Sơ đồ Spectrogram*

Kết quả sơ đồ Spectrogram được tính toán từ hai phương pháp, Python và KissFFT, cho thấy sự tương đồng rõ rệt trong việc biểu diễn phổ năng lượng của tín hiệu âm thanh qua trục thời gian và tần số. Cả hai biểu đồ đều thể hiện các vùng năng lượng tập trung chủ yếu ở dải tần số thấp (dưới 2000 Hz), với các đỉnh năng lượng nổi bật trong khoảng thời gian từ 1,0 đến 1,5 giây, phản ánh các đặc trưng quan trọng của tín hiệu giọng nói. Màu sắc trên thang đo cường độ (từ -180 dB đến -60 dB) cho thấy sự phân bố năng lượng đồng đều, với các vùng sáng (cường độ cao) tương ứng với các thành phần âm thanh chính.

Tuy nhiên, sự khác biệt nhỏ giữa hai phương pháp có thể được nhận thấy ở một số khu vực, đặc biệt tại tần số thấp (dưới 1000 Hz), nơi độ chênh lệch cường độ lên đến ±2 dB. Điều này có thể do sự khác biệt trong cách thực hiện FFT hoặc quy trình chuẩn hóa dữ liệu giữa Python và KissFFT. Dù vậy, độ chênh lệch này không đáng kể (chủ yếu nằm trong khoảng ±0,5 dB ở phần lớn các điểm), chứng tỏ cả hai phương pháp đều mang lại độ tin cậy cao trong việc phân tích tín hiệu. Kết quả này khẳng định tính khả thi của việc sử dụng KissFFT trên các thiết bị nhúng như ESP32, phù hợp với các ứng dụng nhận diện giọng nói thời gian thực.

A purple square with white text

Description automatically generated

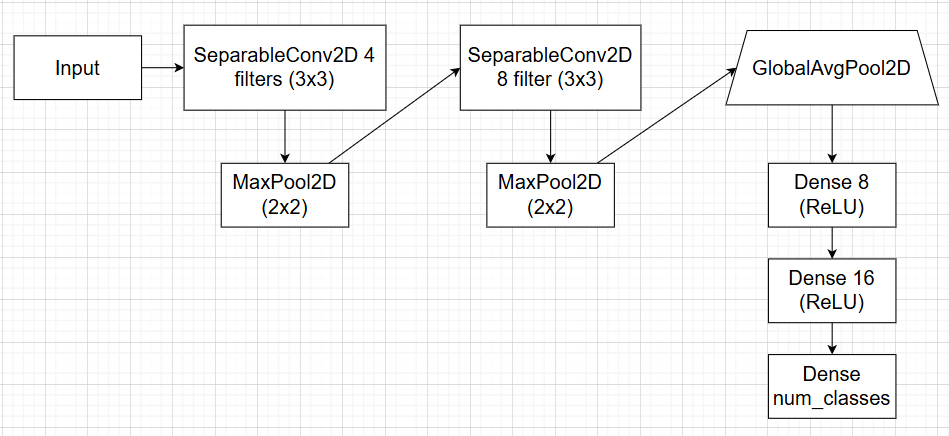
*Hình 6 :Sơ đồ so sánh hai điểm Spectrogram*

Kết quả so sánh từng điểm của hai Spectrogram (tính bằng Python và KissFFT) được thể hiện qua biểu đồ "Absolute Difference between Spectrograms". Biểu đồ này biểu diễn độ chênh lệch tuyệt đối (trong đơn vị dB) giữa hai ma trận Spectrogram, với trục thời gian từ 0 đến 2,5 giây và trục tần số từ 0 đến 8000 Hz. Thang màu thể hiện độ chênh lệch từ 0 (màu tím đậm) đến 1,2 dB (màu vàng).

Quan sát biểu đồ, phần lớn các điểm trên ma trận có độ chênh lệch rất nhỏ, gần bằng 0 dB, được thể hiện qua màu tím đậm chiếm ưu thế trên toàn bộ biểu đồ. Điều này cho thấy hai phương pháp tính toán Spectrogram (Python và KissFFT) có sự tương đồng cao về mặt kết quả. Tuy nhiên, một số vùng nhỏ, đặc biệt trong khoảng thời gian từ 1,0 đến 1,5 giây và tại dải tần số thấp (dưới 1000 Hz), xuất hiện độ chênh lệch đáng chú ý, với giá trị tối đa khoảng 1,2 dB (màu vàng). Sự khác biệt này có thể xuất phát từ các yếu tố như độ chính xác của phép tính FFT, cách chuẩn hóa dữ liệu hoặc các tham số xử lý tín hiệu giữa hai phương pháp.

Nhìn chung, độ chênh lệch nhỏ và không đáng kể ở hầu hết các điểm cho thấy cả hai phương pháp đều đáng tin cậy và có thể sử dụng thay thế cho nhau trong các ứng dụng thực tế, đặc biệt là trong hệ thống nhúng như ESP32 để nhận diện giọng nói. Kết quả này khẳng định tính hiệu quả của KissFFT trong việc tính toán Spectrogram trên thiết bị có tài nguyên hạn chế, với độ chính xác tương đương Python.

## Sơ đồ CNN hệ thống



*Hình 7:Sơ đồ huấn luyện CNN của hệ thống*

Mô hình CNN được xây dựng với các lớp liên tiếp để xử lý dữ liệu đầu vào là ma trận Spectrogram có kích thước 257x11x1 . Các lớp trong mô hình được thiết kế như sau:

*Lớp đầu vào*

Dữ liệu đầu vào là ma trận Spectrogram có kích thước 257x11x1, biểu diễn phổ năng lượng theo thời gian của tín hiệu âm thanh. Đây là đầu vào cho toàn bộ mạng nơ-ron tích chập.

*Lớp SeparableConv2D 4 filter(3X3)*

Lớp tích chập đầu tiên là SeparableConv2D với 4 bộ lọc và kernel 3x3, cho phép giảm số lượng tham số so với Conv2D thông thường nhờ vào việc tách tích chập không gian và tích chập chiều sâu. Kết quả sau lớp này là ma trận đặc trưng kích thước 255x9x4.

*Lớp MaxPool2D*

Tiếp theo, lớp MaxPool2D với kích thước cửa sổ 2x2 được sử dụng để giảm kích thước không gian của ma trận đầu ra, giúp giảm độ phức tạp tính toán và tránh hiện tượng quá khớp. Kích thước đầu ra sau lớp này là 127x4x4.

*Lớp SeparableConv2D 8 filter(3X3)*

Một lớp SeparableConv2D thứ hai với 8 bộ lọc, kích thước kernel 3x3, được áp dụng để tiếp tục trích xuất đặc trưng. Đầu ra của lớp này là ma trận 125x2x8.

*Lớp MaxPool2D*

Lớp MaxPool2D thứ hai với cửa sổ 2x2 được sử dụng để giảm kích thước không gian xuống còn 62x1x8.

*Lớp GlobalAvgPool2D*

Lớp GlobalAvgPool2D được áp dụng để lấy giá trị trung bình toàn cục trên các chiều không gian, giảm kích thước đầu ra xuống còn 62 (từ 62x1x8). Lớp này giúp giảm số lượng tham số và tổng quát hóa đặc trưng.

*Lớp Dense*

Một lớp Dense với 8 nơ-ron và hàm kích hoạt ReLU được sử dụng để học các đặc trưng cấp cao hơn.

*Lớp Dense*

Lớp Dense thứ hai với 16 nơ-ron và hàm kích hoạt ReLU tiếp tục xử lý các đặc trưng trước khi đưa vào lớp đầu ra.

*Lớp Dense đầu ra (num\_classes, Softmax)*

Lớp Dense cuối cùng có số nơ-ron bằng số lớp cần phân loại (num\_classes), sử dụng hàm kích hoạt Softmax để xuất ra xác suất cho từng lớp lệnh giọng nói (ví dụ: "bat", "tat", "den").

Quá trình huấn luyện

*Chuẩn bị dữ liệu*

Dữ liệu huấn luyện bao gồm các ma trận Spectrogram được tạo từ các file âm thanh chứa lệnh giọng nói. Mỗi Spectrogram được gắn nhãn tương ứng với một lớp lệnh cụ thể.

*Tiền xử lý*

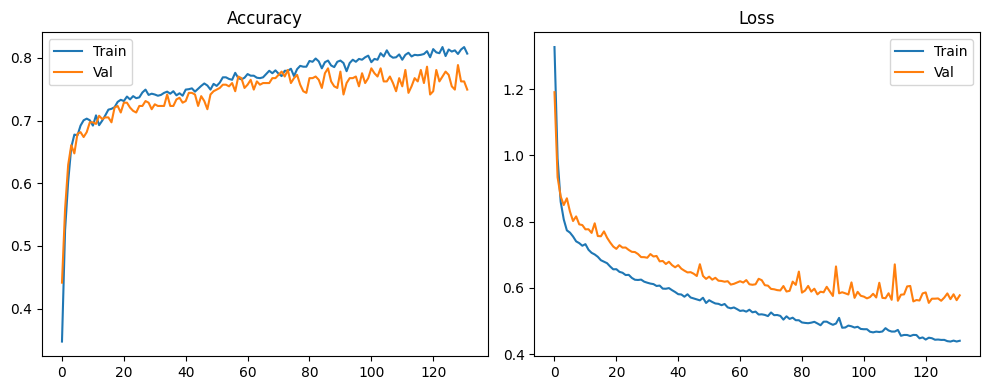
Các ma trận Spectrogram được chuẩn hóa để đảm bảo giá trị nằm trong khoảng phù hợp giúp mô hình hội tụ nhanh hơn trong quá trình huấn luyện.

*Huấn luyện mô hình*

Mô hình được huấn luyện bằng thuật toán tối ưu hóa với hàm mất mát Categorical Crossentropy, phù hợp cho bài toán phân loại đa lớp.

Quá trình huấn luyện diễn ra trong nhiều epoch, với mỗi epoch bao gồm việc truyền dữ liệu qua các lớp của mạng, tính toán mất mát, và cập nhật trọng số dựa trên gradient.

Lớp Dropout được sử dụng trong quá trình huấn luyện để tăng khả năng tổng quát hóa, giảm nguy cơ quá khớp trên tập dữ liệu huấn luyện.



*Hình 8: Kết quả huấn luyện CNN của hệ thống*

*Độ chính xác (Accuracy)*

Độ chính xác trên tập huấn luyện (màu xanh) và tập kiểm tra (màu cam) cho thấy xu hướng tăng đều qua các epoch, phản ánh khả năng học tốt các đặc trưng của mô hình từ dữ liệu đầu vào.

Giai đoạn đầu (0–30 epoch): Độ chính xác tăng nhanh cho cả hai tập dữ liệu, chứng tỏ mô hình học tốt các đặc trưng ban đầu từ Spectrogram.

Từ khoảng epoch 30 trở đi: Độ chính xác trên tập huấn luyện tiếp tục tăng đều và chạm mốc gần 0.8 ở cuối quá trình huấn luyện (epoch 130), trong khi độ chính xác trên tập kiểm tra dao động quanh mức 0.7–0.75.

Sự chênh lệch nhẹ giữa Train và Val ở giai đoạn sau nhưng không đáng kể, cho thấy mô hình có khả năng tổng quát hóa tương đối tốt và chưa có dấu hiệu rõ ràng của quá khớp.

*Hàm mất mát (Loss)*

Trong 25 epoch đầu, loss giảm nhanh từ mức hơn 1.2 xuống dưới 0.8, thể hiện quá trình học nhanh giai đoạn đầu.

Sau epoch 30, loss tiếp tục giảm ổn định trên tập huấn luyện và đạt khoảng 0.45 ở cuối quá trình.

Trên tập kiểm tra, giá trị loss dao động nhẹ quanh mức 0.6 – 0.7, không có xu hướng tăng trở lại

Biểu đồ loss cho thấy mô hình học tốt và duy trì sự ổn định, không có dấu hiệu học quá mức .

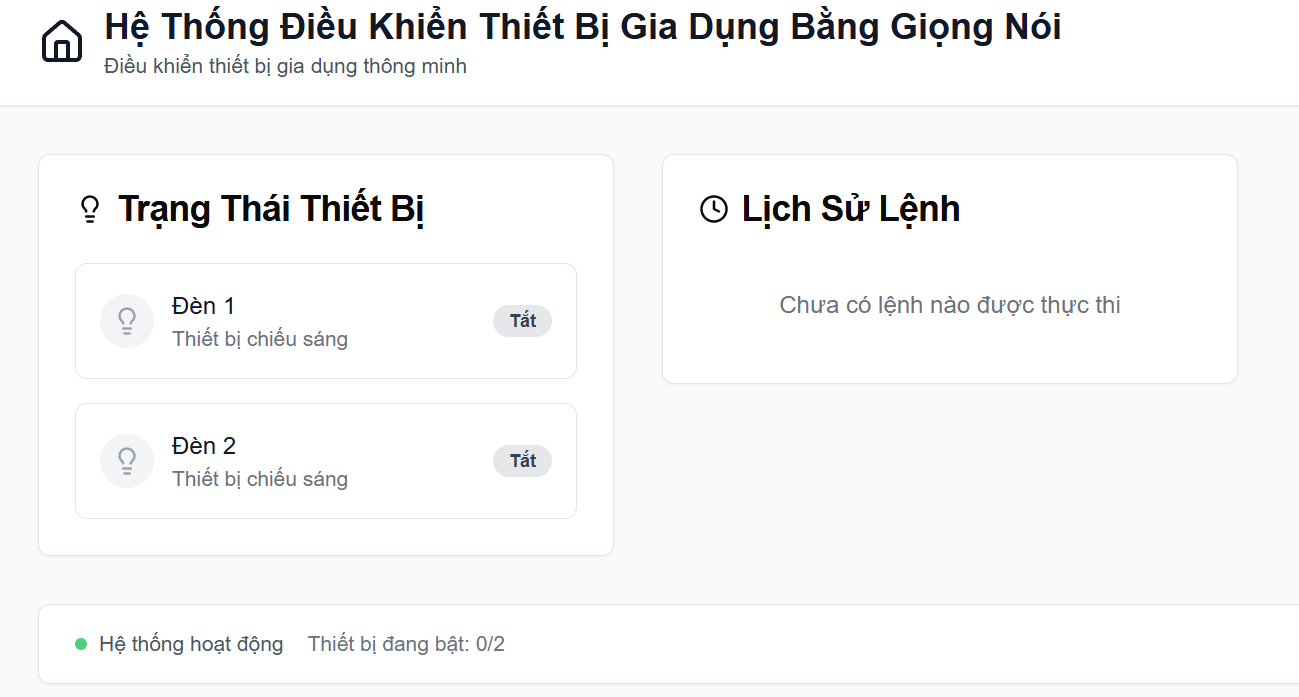
*Nhận xét tổng quan*

Mô hình CNN sử dụng các lớp SeparableConv2D, MaxPooling2D, và GlobalAveragePooling2D đã học hiệu quả từ dữ liệu Spectrogram.

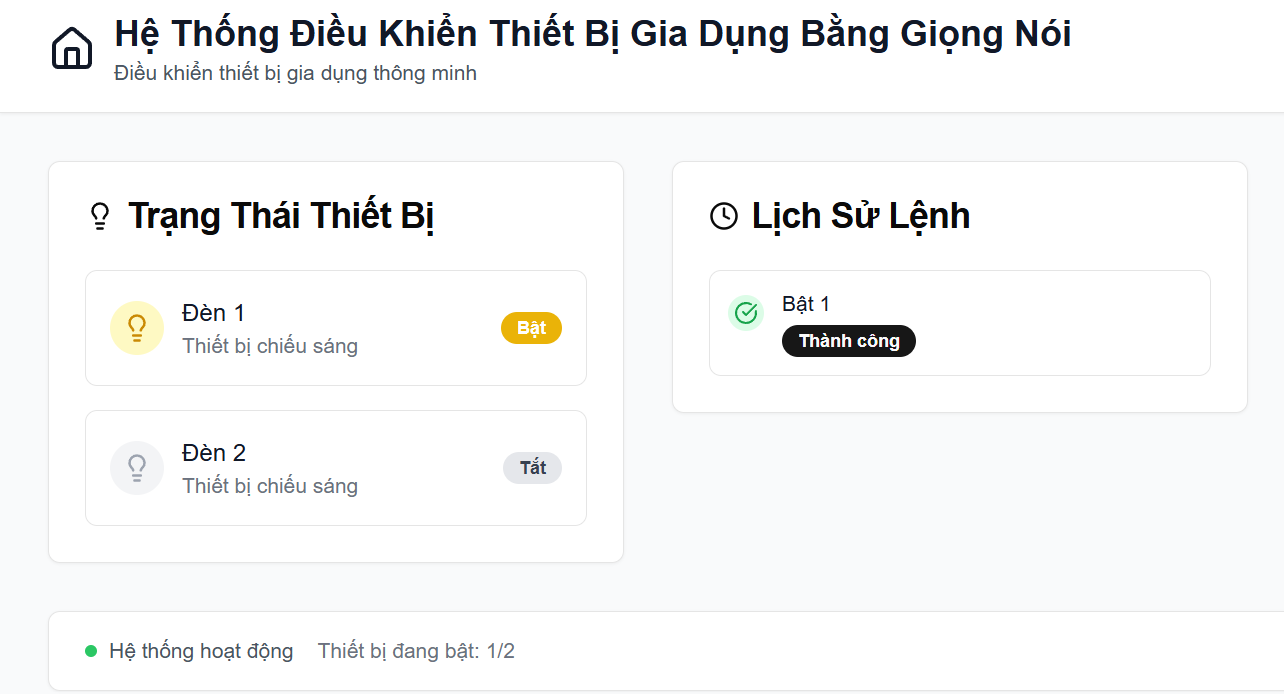
Độ chính xác cuối cùng trên tập kiểm tra đạt khoảng 74–75%, với loss ổn định chứng minh hiệu quả của mô hình trong việc trích xuất đặc trưng và phân loại lệnh giọng nó.

Kết quả huấn luyện xác nhận mô hình đã sẵn sàng để triển khai trong các ứng dụng điều khiển thiết bị thông minh bằng giọng nói, đặc biệt phù hợp với môi trường tài nguyên hạn chế như ESP32.

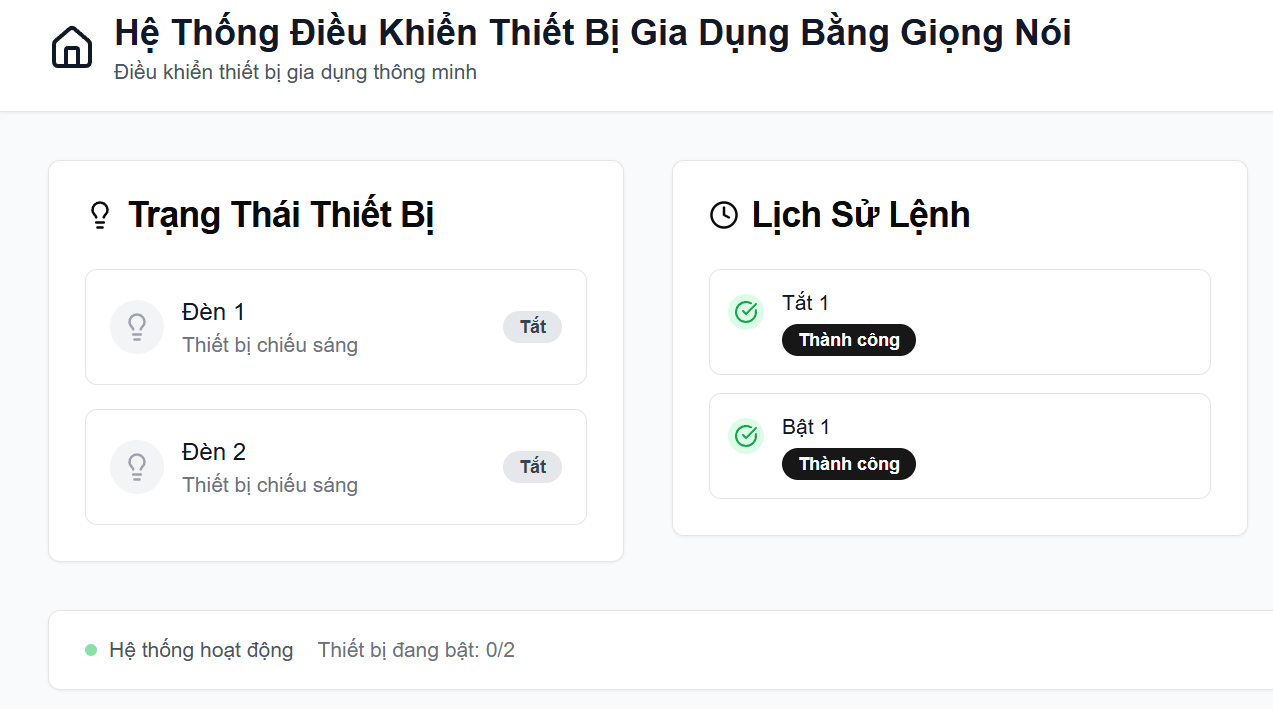
## Kiến trúc hoat động web



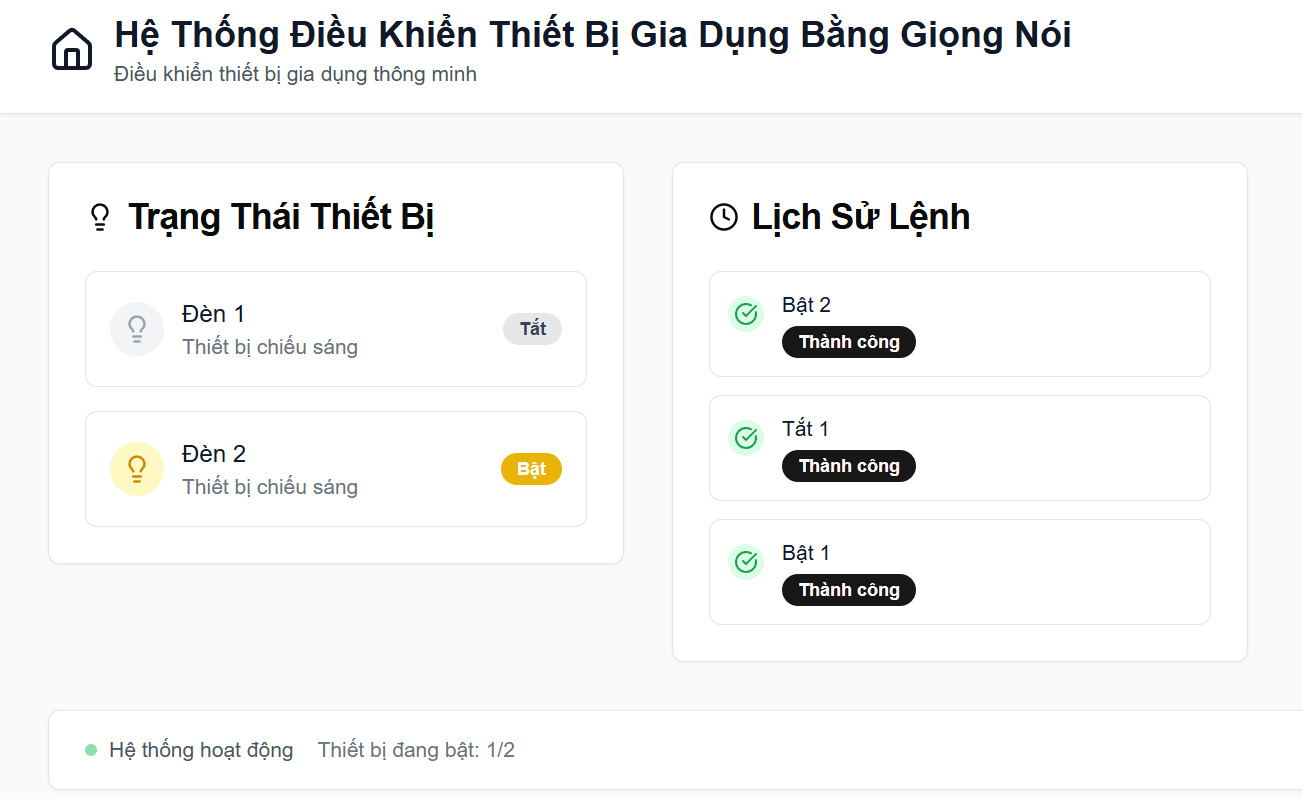
* Trang chủ của website điều khiển thiết bị gia dụng bằng giọng nói.
* Các thiết bị đèn ban đầu sẽ có trạng thái là Tắt và số lượng thiết bị đang bật là 0/2.
* Danh sách Lịch Sử Lệnh trống.

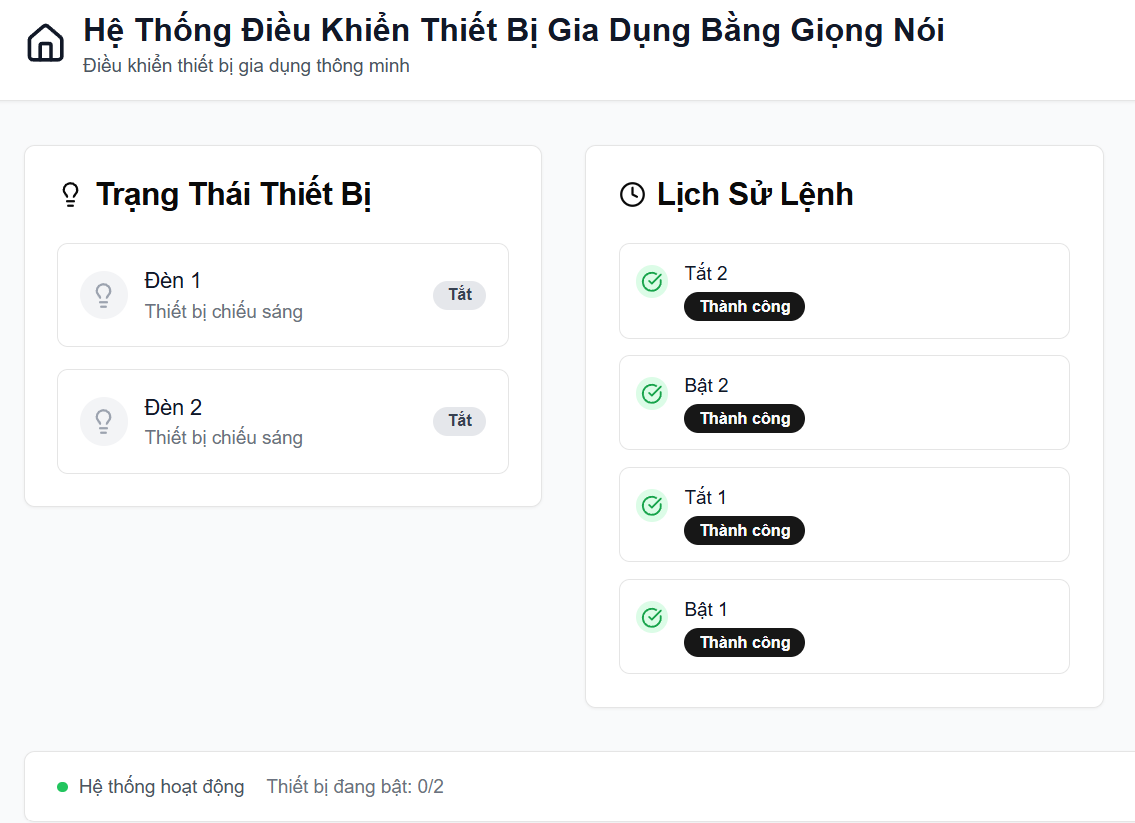


* Khi web nhận được tín hiệu bật đèn 1 từ server, giao diện trang chủ sẽ biểu thị sự thay đổi trạng thái bật của đèn.
* Số lượng thiết bị đang bật thay đổi theo số lượng đèn đang có trạng thái “Bật” trên trang.
* Đồng thời tín hiệu lệnh nhận được sẽ được hiển thị trên danh sách Lịch Sử Lệnh.



* Tương tự, khi web nhận được tín hiệu tắt đèn 1 từ server, giao diện trang chủ sẽ biểu thị sự thay đổi trạng thái thành trạng thái tắt của đèn, tín hiệu lệnh mới sẽ được thêm vào danh sách Lịch Sử Lệnh và số lượng thiết bị đang bật cũng được cập nhật để khớp với số đèn đang bật.





* Tương tự, khi web nhận được tín hiệu bật đèn 2 từ server, giao diện trang chủ sẽ biểu thị sự thay đổi trạng thái bật / tắt của đèn, tín hiệu lệnh mới sẽ được thêm vào danh sách Lịch Sử Lệnh và số lượng thiết bị đang bật cũng được cập nhật để khớp với số đèn đang bật.

# CHƯƠNG IV: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. Kết luận

Đề tài "Thiết bị điều khiển thiết bị gia dụng bằng giọng nói " đã thành công trong việc xây dựng một hệ thống nhận diện và phản hồi lệnh giọng nói cục bộ, mang lại giải pháp điều khiển thông minh và tiện lợi cho các thiết bị gia dụng. Hệ thống đã chứng minh khả năng tích hợp công nghệ AI trên phần cứng nhúng với tài nguyên hạn chế.

a. Ưu điểm của đề tài

Tính cục bộ (Edge AI): Hệ thống thực hiện suy luận (inference) trực tiếp trên ESP32 mà không cần kết nối Internet liên tục để gửi dữ liệu lên đám mây. Điều này giúp giảm độ trễ (latency), tăng cường quyền riêng tư dữ liệu, và đảm bảo hoạt động ổn định ngay cả khi không có mạng.

Chi phí thấp: Sử dụng vi điều khiển ESP32 và các module phần cứng phổ biến, giá thành phải chăng, giúp giảm tổng chi phí triển khai hệ thống so với các giải pháp thương mại.

Hiệu quả năng lượng: ESP32 là một vi điều khiển tiêu thụ năng lượng thấp, phù hợp cho các ứng dụng hoạt động liên tục hoặc chạy bằng pin (với một số tối ưu hóa).

Dễ dàng tùy biến và mở rộng: Mã nguồn mở và tính linh hoạt của nền tảng Arduino/ESP-IDF cho phép dễ dàng tùy chỉnh các lệnh giọng nói, tích hợp với các loại thiết bị gia dụng khác nhau, và mở rộng thêm các tính năng thông minh.

Thân thiện với người dùng: Cung cấp một phương thức tương tác tự nhiên và tiện lợi hơn so với việc điều khiển thủ công hoặc qua ứng dụng, đặc biệt hữu ích cho người cao tuổi, người khuyết tật, hoặc khi tay đang bận.

Ứng dụng thực tiễn cao: Giải quyết nhu cầu thiết yếu trong tự động hóa nhà thông minh, cải thiện chất lượng cuộc sống và sự tiện nghi.

Học tập và nghiên cứu: Đề tài cung cấp một nền tảng tuyệt vời để nghiên cứu và thực hành về xử lý tín hiệu số, học máy trên thiết bị nhúng (TinyML), và phát triển IoT.

b. Nhược điểm của đề tài

Độ chính xác phụ thuộc vào dữ liệu huấn luyện: Hiệu suất nhận diện giọng nói bị ảnh hưởng lớn bởi chất lượng, số lượng, và sự đa dạng của dữ liệu âm thanh dùng để huấn luyện mô hình. Môi trường nhiều tiếng ồn hoặc giọng nói khác lạ có thể làm giảm độ chính xác.

Hạn chế về từ vựng: Mô hình CNN được huấn luyện cho một tập hợp các lệnh cố định. Việc thêm mới hoặc thay đổi lệnh đòi hỏi phải huấn luyện lại và cập nhật mô hình lên thiết bị.

Yêu cầu tài nguyên: Mặc dù đã tối ưu hóa, việc chạy mô hình CNN vẫn yêu cầu một lượng RAM đáng kể trên ESP32, có thể hạn chế khả năng mở rộng cho các mô hình phức tạp hơn hoặc các tác vụ khác.

Phức tạp trong tiền xử lý: Giai đoạn tiền xử lý tín hiệu âm đòi hỏi kiến thức về xử lý tín hiệu số và có thể tốn tài nguyên tính toán.

Chưa có khả năng hiểu ngữ cảnh/đối thoại: Hệ thống chỉ nhận diện các lệnh độc lập, không có khả năng hiểu các câu nói phức tạp, duy trì ngữ cảnh hội thoại, hay phân biệt giọng nói của nhiều người.

Thiếu phản hồi đa dạng: Hiện tại chỉ có phản hồi qua điều khiển thiết bị. Thiếu các phản hồi bằng giọng nói hoặc hình ảnh từ chính thiết bị ESP32 để xác nhận lệnh.

Khả năng chịu lỗi: Hệ thống có thể bị nhầm lẫn bởi các âm thanh nền hoặc giọng nói không phải là lệnh, dẫn đến việc điều khiển sai thiết bị.

## 2. Hướng phát triển

*Nâng cao độ chính xác và khả năng thích nghi của mô hình*

Thu thập thêm dữ liệu huấn luyện đa dạng hơn, bao gồm nhiều giọng nói khác nhau và các môi trường tiếng ồn đa dạng, nhằm cải thiện độ chính xác và khả năng chịu lỗi của mô hình.

Tiến hành nghiên cứu các kiến trúc CNN hoặc các mô hình AI tiên tiến hơn (như RNN/LSTM) phù hợp với xử lý chuỗi thời gian, nhằm nâng cao hiệu suất nhận diện giọng nói trên thiết bị nhúng như ESP32.

Áp dụng các kỹ thuật tăng cường dữ liệu trong quá trình huấn luyện để mở rộng tập dữ liệu, từ đó tăng cường khả năng tổng quát hóa của mô hình.

*Tích hợp khả năng điều khiển thiết bị đa dạng*

Mở rộng tính năng điều khiển các thiết bị thông minh thông qua các giao thức IoT phổ biến như MQTT, HomeKit, Zigbee, và Z-Wave, với sự hỗ trợ của gateway hoặc module tương thích.

Phát triển module điều khiển hồng ngoại để cho phép điều khiển các thiết bị điện tử truyền thống (như TV, điều hòa) bằng giọng nói, nâng cao tính tương thích với các thiết bị cũ.

*Cải thiện trải nghiệm người dùng*

Tích hợp đèn LED hoặc màn hình nhỏ trên thiết bị ESP32 để hiển thị trạng thái hoạt động hoặc các lệnh đã được nhận diện, tăng cường tính trực quan.

Triển khai tính năng kích hoạt bằng từ khóa, cho phép thiết bị chỉ bắt đầu lắng nghe sau khi nhận diện một từ khóa cụ thể, nhằm tiết kiệm năng lượng và bảo vệ quyền riêng tư.

*Tối ưu hóa năng lượng*

Nghiên cứu và áp dụng các chế độ tiết kiệm năng lượng của ESP32 (như Deep Sleep, Light Sleep) kết hợp với cơ chế kích hoạt bằng từ khóa, nhằm kéo dài thời lượng pin cho các thiết bị di động sử dụng hệ thống.

*Mở rộng khả năng hiểu biết*

Tiến hành nghiên cứu các phương pháp để mô hình có thể nhận diện các biến thể nhỏ của lệnh hoặc các cấu trúc câu đơn giản hơn (như "bật đèn ngủ", "đèn phòng khách bật"), nâng cao tính linh hoạt.

Tích hợp khả năng phân biệt giọng nói, cho phép hệ thống chỉ phản hồi với các lệnh từ người dùng được ủy quyền, tăng cường tính bảo mật.

Những định hướng này sẽ góp phần hoàn thiện hệ thống, nâng cao hiệu suất, và đáp ứng tốt hơn nhu cầu thực tế của người dùng trong các ứng dụng điều khiển thiết bị gia dụng bằng giọng nói.