HỆ THỐNG TƯƠNG TÁC THÔNG MINH DỰA TRÊN NHẬN DIỆN CẢM XÚC

Ứng dụng AI, IoT trong hệ thống thông minh

Lê Văn Việt
Khoa Công nghệ Thông tin
Đại học Đại Nam

***Tóm tắt nội dung*—Cảm xúc đóng vai trò quan trọng trong giao tiếp hàng ngày giữa con người. Tuy nhiên, các hệ thống máy tính thông thường không có khả năng nhận biết và phản ứng với cảm xúc của người dùng. Bài tập lớn này trình bày về việc xây dựng một hệ thống kết hợp trí tuệ nhân tạo (AI) và Internet vạn vật (IoT) có khả năng nhận diện cảm xúc thông qua khuôn mặt, tương tác bằng giọng nói và điều khiển các thiết bị IoT dựa trên ngữ cảnh cảm xúc. Hệ thống được xây dựng bao gồm ba thành phần chính: Client (xử lý nhận dạng cảm xúc và tương tác giọng nói), Server (API Gateway và xử lý logic) và thiết bị IoT ESP32 (điều khiển phần cứng). Kết quả thực nghiệm cho thấy hệ thống đạt độ chính xác nhận diện cảm xúc 85% trong điều kiện ánh sáng tốt, độ nhận diện giọng nói 80% với các lệnh đơn giản, và độ trễ hệ thống chấp nhận được trong khoảng 500ms-2s. Hệ thống mở ra hướng phát triển cho các ứng dụng thông minh có khả năng thích nghi với trạng thái cảm xúc người dùng, hướng tới môi trường tương tác tự nhiên hơn giữa con người và máy tính.**

***Index Terms*—nhận dạng cảm xúc khuôn mặt, tương tác người-máy, DeepFace, xử lý giọng nói, Internet vạn vật (IoT), ESP32**

I. Giới thiệu

Trong tương tác giữa con người, cảm xúc đóng vai trò quan trọng giúp truyền tải thông tin và tạo nên giao tiếp hiệu quả. Tuy nhiên, máy tính và các thiết bị thông minh thông thường không có khả năng nhận biết cảm xúc, dẫn đến trải nghiệm tương tác thiếu tự nhiên và không thích ứng được với trạng thái cảm xúc của người dùng.

Sự phát triển của các công nghệ trí tuệ nhân tạo, đặc biệt là trong lĩnh vực thị giác máy tính và học sâu, đã mở ra khả năng phát triển các hệ thống có thể nhận diện cảm xúc con người thông qua các biểu hiện khuôn mặt. Kết hợp với công nghệ Internet vạn vật (IoT), những hệ thống này có thể tạo ra môi trường thông minh, tự động điều chỉnh và phản ứng dựa trên cảm xúc của người dùng.

Bài tập lớn này trình bày về việc thiết kế và triển khai một hệ thống tích hợp AI-IoT có khả năng:

- Nhận diện cảm xúc khuôn mặt theo thời gian thực
- Tương tác với người dùng thông qua giọng nói
- Điều khiển các thiết bị IoT dựa trên cảm xúc và lệnh thoại
- Tạo môi trường thích ứng với trạng thái cảm xúc người dùng

Cấu trúc bài báo bao gồm: Phần II trình bày các nghiên cứu liên quan và nền tảng lý thuyết; Phần III mô tả phương pháp tiếp cận và thiết kế hệ thống; Phần IV trình bày kết quả thực nghiệm và đánh giá; Phần V thảo luận về ưu điểm, hạn chế và hướng phát triển trong tương lai; và cuối cùng là phần kết luận.

II. Nghiên cứu liên quan

A. Nhận dạng cảm xúc khuôn mặt

Nhận dạng cảm xúc khuôn mặt (FER - Facial Emotion Recognition) là lĩnh vực nghiên cứu nhằm phát triển các thuật toán và mô hình có khả năng phân loại cảm xúc con người dựa trên biểu hiện khuôn mặt. Quá trình phát triển của công nghệ này đã trải qua nhiều giai đoạn, từ các phương pháp truyền thống như Haar Cascade đến các mô hình học sâu hiện đại.

Các phương pháp truyền thống sử dụng đặc trưng hình học khuôn mặt và các thuật toán phân loại cổ điển như SVM, Random Forest để phân loại cảm xúc. Tuy nhiên, những phương pháp này thường bị hạn chế trong các tình huống thực tế với điều kiện ánh sáng, góc nhìn và các biểu hiện cảm xúc phức tạp [1].

Các mô hình học sâu, đặc biệt là mạng nơ-ron tích chập (CNN), đã mang lại bước đột phá trong lĩnh vực nhận dạng cảm xúc khuôn mặt. Các kiến trúc như VGG, ResNet, và InceptionNet đã được điều chỉnh để phân loại cảm xúc với độ chính xác cao hơn đáng kể [2].

Trong những năm gần đây, nhiều thư viện và framework đã được phát triển để hỗ trợ việc triển khai các mô hình nhận dạng cảm xúc trong ứng dụng thực tế. Một số công cụ phổ biến bao gồm:

- **FER**: Một thư viện Python hỗ trợ nhận dạng cảm xúc khuôn mặt sử dụng các mô hình CNN được huấn luyện trên bộ dữ liệu FER2013. Thư viện này hỗ trợ nhiều kiến trúc mô hình khác nhau và dễ dàng tích hợp vào các ứng dụng thị giác máy tính [3].
- **DeepFace**: Một framework toàn diện cho việc phân tích khuôn mặt, bao gồm các chức năng như phát

hiện khuôn mặt, nhận dạng cảm xúc, phân tích độ tuổi và giới tính. DeepFace cung cấp các mô hình pre-trained với độ chính xác cao và được tối ưu hóa cho ứng dụng thời gian thực [4].

- **EmotiEffLib**: Thư viện nhận diện cảm xúc khuôn mặt sử dụng Deep Learning, hỗ trợ Python và C++, tối ưu cho các ứng dụng thời gian thực. EmotiEffLib sử dụng các framework như PyTorch và ONNX Runtime để tối ưu hóa hiệu suất [5].

B. Tương tác giọng nói và xử lý ngôn ngữ tự nhiên

Tương tác giọng nói đã trở thành một phương thức giao tiếp người-máy ngày càng phổ biến. Các hệ thống nhận dạng giọng nói (Speech Recognition) và chuyển văn bản thành giọng nói (Text-to-Speech) đã đạt được nhiều tiến bộ nhờ vào các kỹ thuật học sâu.

Trong lĩnh vực nhận dạng giọng nói, các mô hình như Wav2Vec, DeepSpeech và các API như Google Speech Recognition đã cải thiện đáng kể độ chính xác nhận dạng, ngay cả trong môi trường có tiếng ồn [6]. Đối với tiếng Việt, các mô hình nhận dạng giọng nói đã được phát triển và tùy chỉnh để xử lý các đặc điểm ngôn ngữ cụ thể.

C. Ứng dụng IoT trong Môi trường Thông minh

Internet vạn vật (IoT) đã mở rộng khả năng kết nối và điều khiển các thiết bị vật lý thông qua mạng internet. ESP32 là một trong những nền tảng phần cứng phổ biến cho các ứng dụng IoT do tính linh hoạt, hiệu suất cao và chi phí thấp [7].

Các hệ thống nhà thông minh và môi trường thông minh ngày càng tích hợp nhiều yếu tố cảm biến và khả năng tương tác. Tuy nhiên, hầu hết các hệ thống hiện tại chủ yếu dựa vào lệnh thoại hoặc các quy tắc được lập trình sẵn, chưa thực sự thích ứng với trạng thái cảm xúc của người dùng.

Nghiên cứu của chúng tôi tập trung vào việc kết hợp nhận dạng cảm xúc với các hệ thống IoT để tạo ra môi trường thông minh có khả năng thích nghi với cảm xúc của người dùng, mang lại trải nghiệm tương tác tự nhiên và cá nhân hóa hơn.

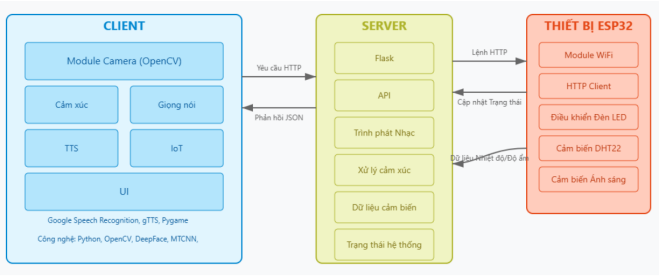
III. Thiết kế hệ thống

A. Kiến trúc tổng quan

Hệ thống được thiết kế với kiến trúc ba tầng chính, bao gồm: Client, Server và Thiết bị IoT. Sơ đồ kiến trúc tổng thể được minh họa trong Hình 1.

1) Client System: Client là thành phần chính xử lý việc nhận dạng cảm xúc khuôn mặt và tương tác giọng nói. Thành phần này bao gồm các module sau:

- **Camera Module**: Thu nhận hình ảnh từ webcam theo thời gian thực
- **Emotion Recognition**: Sử dụng DeepFace và MTCNN để phát hiện và phân tích cảm xúc từ khuôn mặt



Hình 1. Kiến trúc tổng thể của hệ thống nhận dạng cảm xúc và tương tác người-máy

- **Voice Service**: Xử lý lệnh giọng nói tiếng Việt, sử dụng Google Speech Recognition API
- **Text-to-Speech Service**: Chuyển văn bản thành giọng nói tiếng Việt
- **IoT Communication**: Kết nối và gửi lệnh điều khiển đến Server
- **User Interface**: Hiển thị trực quan kết quả nhận dạng và trạng thái hệ thống

2) Server System: Server đóng vai trò trung gian giữa Client và các thiết bị IoT, thực hiện xử lý logic và cung cấp API. Các thành phần của Server bao gồm:

- **Flask API Gateway**: Cung cấp các endpoint RESTful API
- **Emotion Handler**: Xử lý dữ liệu cảm xúc và đưa ra đề xuất
- **IoT Controller**: Quản lý và điều khiển các thiết bị IoT
- **Music Player Service**: Quản lý việc phát nhạc theo cảm xúc
- **System State Manager**: Theo dõi và cập nhật trạng thái toàn hệ thống

3) IoT Device (ESP32): Thành phần thiết bị IoT sử dụng ESP32 để điều khiển các thiết bị phần cứng, bao gồm:

- **WiFi Module**: Kết nối với Server thông qua mạng WiFi
- **LED Controller**: Điều khiển đèn LED dựa trên lệnh từ Server
- **DHT22 Sensor**: Cảm biến nhiệt độ và độ ẩm
- **HTTP Client**: Nhận lệnh điều khiển từ Server thông qua HTTP requests

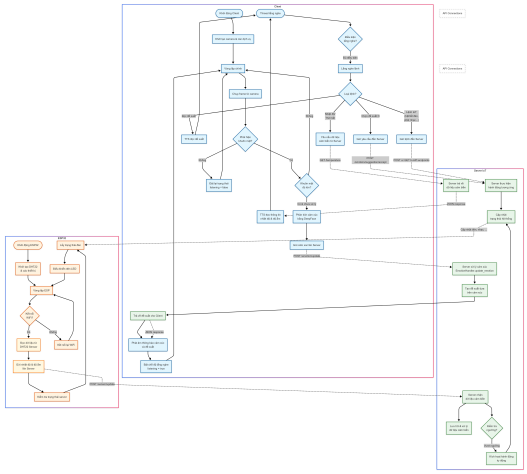
B. Luồng hoạt động

Hệ thống hoạt động theo luồng chính sau:

1) Client sử dụng webcam để thu nhận hình ảnh khuôn mặt theo thời gian thực
2) Khi phát hiện khuôn mặt, hệ thống phân tích và nhận dạng cảm xúc bằng DeepFace
3) Thông tin cảm xúc được gửi đến Server để xử lý và đưa ra đề xuất
4) Client phát thông báo âm thanh về cảm xúc đã phát hiện và các đề xuất

5) Người dùng có thể tương tác bằng giọng nói để điều khiển thiết bị hoặc chấp nhận đề xuất
6) Lệnh điều khiển được gửi từ Client đến Server
7) Server gửi lệnh tương ứng đến thiết bị ESP32
8) ESP32 thực hiện điều khiển thiết bị (bật/tắt đèn, điều chỉnh nhiệt độ...)
9) Trạng thái thiết bị được cập nhật và đồng bộ giữa các thành phần

Sơ đồ luồng hoạt động được minh họa trong Hình 2.



Hình 2. Luồng hoạt động của hệ thống nhận dạng cảm xúc và tương tác người-máy

C. Chi tiết triển khai

1) Module nhận dạng cảm xúc: Module nhận dạng cảm xúc sử dụng thư viện DeepFace và MTCNN để phát hiện và phân tích cảm xúc từ khuôn mặt. Thuật toán được triển khai gồm các bước: (1) thu nhận hình ảnh từ camera, (2) phát hiện khuôn mặt sử dụng MTCNN detector, (3) trích xuất đặc trưng khuôn mặt, (4) phân loại cảm xúc sử dụng mô hình deep learning với 7 lớp cảm xúc cơ bản, và (5) xác định cảm xúc chiếm ưu thế dựa trên điểm số cao nhất. Quá trình này được thực hiện theo thời gian thực với xử lý lỗi phù hợp khi không phát hiện được khuôn mặt.

Thuật toán nhận dạng cảm xúc được tối ưu hóa thông qua việc sử dụng detector backend MTCNN, hiệu quả hơn so với detector Haar Cascade truyền thống. Việc phân tích khuôn mặt được thực hiện bởi mô hình có cấu trúc VGG-Face được pre-trained trên tập dữ liệu lớn và điều chỉnh để phân loại cảm xúc. Quá trình xử lý hình ảnh bao gồm các bước tiền xử lý quan trọng như chuẩn hóa màu sắc, căn chỉnh khuôn mặt và điều chỉnh kích thước ảnh trước khi đưa vào mô hình. Thuật toán có thể hoạt động với tốc độ 2-3 FPS trên phần cứng thông thường, đủ để đảm bảo trải nghiệm tương tác thời gian thực.

2) Module tương tác giọng nói: Module tương tác giọng nói xử lý theo quy trình: (1) kích hoạt microphone và lắng nghe đầu vào âm thanh, (2) điều chỉnh ngưỡng nhiễu nền tự động, (3) nhận dạng giọng nói sử dụng Google Speech Recognition API với ngôn ngữ tiếng Việt, (4) xử lý kết quả văn bản và phân tích lệnh, (5) phản hồi bằng âm thanh thông qua gTTS. Hệ thống có cơ chế xử lý lỗi khi không nhận dạng được âm thanh hoặc gặp vấn đề kết nối. Thời gian lắng nghe được giới hạn (timeout) để tối ưu trải nghiệm người dùng và giảm thiểu tiêu thụ tài nguyên.

Module tương tác giọng nói được tăng cường với cơ chế lọc thông minh để tránh nhận diện nhầm giữa lệnh người dùng và nội dung phản hồi của hệ thống. Quy trình xử lý văn bản lệnh bao gồm: (1) chuyển đổi số tiếng Việt thành chữ số (ví dụ: "một" → "1"), (2) loại bỏ các từ can thiệp không cần thiết, (3) so sánh với danh sách từ khóa lệnh đã định nghĩa trước, và (4) xác định cấu trúc lệnh thông qua phân tích cú pháp đơn giản. Hệ thống duy trì danh sách từ khóa lệnh cho nhiều chức năng khác nhau như điều khiển đèn, nhạc, xem nhiệt độ và trạng thái hệ thống. Đặc biệt, cơ chế nhận diện lệnh đề xuất được triển khai thông qua hàm `check_suggestion_command` có khả năng phát hiện các biến thể khác nhau của lệnh chấp nhận đề xuất.

3) API Server và Xử lý logic: API endpoint xử lý cập nhật cảm xúc được thiết kế theo kiến trúc RESTful với phương thức POST. Quy trình xử lý bao gồm: (1) tiếp nhận dữ liệu JSON chứa thông tin về cảm xúc và độ tin cậy, (2) kiểm tra tính hợp lệ của dữ liệu đầu vào, (3) chuyển thông tin đến EmotionHandler để xử lý phân tích cảm xúc, (4) tạo đề xuất dựa trên cảm xúc được phát hiện, và (5) trả về kết quả dạng JSON với trạng thái xử lý. Hệ thống sẽ phản hồi lỗi nếu thiếu thông tin cảm xúc trong yêu cầu.

Hệ thống API được phát triển với Flask Framework có cấu trúc module hóa cao với các blueprint riêng biệt cho từng nhóm chức năng: quản lý đèn (`light_bp`), quản lý âm nhạc (`music_bp`), cảm biến (`sensor_bp`), thời tiết (`weather_bp`) và cảm xúc (`emotion_bp`). Các endpoint chính bao gồm: (1) `/emotion/update`: tiếp nhận dữ liệu cảm xúc từ client và tạo đề xuất, (2) `/light/on`, `/light/off`: điều khiển trạng thái đèn, (3) `/music/-play`: phát nhạc theo cảm xúc hoặc playlist được chỉ định, (4) `/emotion/suggestion/accept/{id}`: chấp nhận đề xuất dựa trên cảm xúc đã phát hiện. Hệ thống sử dụng cơ chế CORS (Cross-Origin Resource Sharing) để đảm bảo an toàn trong giao tiếp giữa client và server.

4) Thiết bị IoT: Thuật toán kiểm tra trạng thái đèn trên ESP32 được triển khai với các bước: (1) kiểm tra kết nối WiFi, (2) khởi tạo HTTP client và kết nối đến server qua endpoint '/light/status', (3) gửi yêu cầu GET và nhận phản hồi, (4) phân tích dữ liệu JSON nhận được, (5) so sánh trạng thái đèn hiện tại với trạng thái từ server, (6) cập nhật trạng thái đèn LED nếu có sự khác biệt, và (7) ghi nhật ký thay đổi. Thiết bị được lập trình để thực hiện kiểm tra này định kỳ nhằm đảm bảo đồng bộ trạng thái với server.

ESP32 được lập trình theo mô hình kiến trúc hướng sự kiện với các hàm kiểm tra và cập nhật định kỳ. Thuật

toán chính trên ESP32 bao gồm: (1) `setupWiFi()`: thiết lập kết nối WiFi với cơ chế tự động kết nối lại, (2) `setupGPIO()`: cấu hình chân GPIO điều khiển LED, (3) `checkWiFiConnection()`: kiểm tra và khôi phục kết nối WiFi khi cần, (4) `checkLightStatus()`: đồng bộ trạng thái đèn với server, và (5) `pingServer()`: kiểm tra tình trạng hoạt động của server. Hệ thống ESP32 sử dụng thư viện ArduinoJson để phân tích dữ liệu JSON nhận được từ server và thực hiện các thao tác tương ứng. Để tối ưu hóa hiệu suất và giảm tải mạng, ESP32 được cấu hình với các khoảng thời gian kiểm tra khác nhau: 3 giây cho cập nhật trạng thái và 60 giây cho kiểm tra kết nối WiFi.

IV. Thực nghiệm và Đánh giá

A. Môi trường thực nghiệm

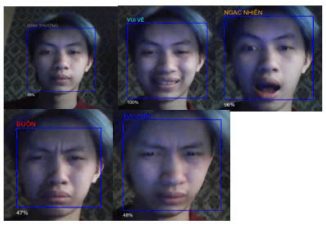
Hệ thống được thử nghiệm trong các điều kiện sau:

- **Phần cứng Client**: Laptop với webcam HD, Intel Core i5, 8GB RAM
- **Phần cứng Server**: Máy tính với Intel Core i7, 16GB RAM
- **Thiết bị IoT**: ESP32 DevKit, LED, cảm biến DHT22
- **Kết nối mạng**: Mạng LAN với WiFi router
- **Điều kiện ánh sáng**: Thử nghiệm trong điều kiện ánh sáng khác nhau (tốt, trung bình, kém)
- **Người thử nghiệm**: 5 người với các biểu cảm và giọng nói khác nhau

B. Kết quả nhận dạng cảm xúc

Hệ thống được đánh giá về độ chính xác nhận dạng cảm xúc bằng cách thử nghiệm trên 100 ảnh khuôn mặt với các biểu cảm khác nhau. Kết quả được trình bày trong Bảng I.

Thực nghiệm một số cảm xúc ở Hình 3.



Hình 3. Thực nghiệm nhận dạng cảm xúc

C. So sánh Mô hình

Đã so sánh hiệu suất của các mô hình nhận dạng cảm xúc khác nhau trên bộ dữ liệu LFW (Labeled Faces in the Wild). Kết quả so sánh được trình bày trong Bảng II.

DeepFace được chọn làm mô hình chính cho hệ thống do có độ chính xác cao nhất (95.3% trên bộ dữ liệu LFW) và khả năng triển khai hiệu quả trong ứng dụng thời gian thực. Mô hình này được tối ưu hóa cho các thiết bị có tài nguyên hạn chế và có thời gian xử lý trung bình khoảng 380ms cho mỗi khung hình, phù hợp với yêu cầu tương tác thời gian thực của hệ thống.

Bảng I
Độ chính xác nhận dạng cảm xúc trong các điều kiện ánh sáng khác nhau

Cảm xúc	Ánh sáng tốt	Ánh sáng TB	Ánh sáng kém
Happy	92%	85%	70%
Sad	84%	78%	65%
Angry	81%	75%	60%
Neutral	90%	83%	72%
Surprise	86%	79%	68%
Fear	79%	71%	58%
Disgust	77%	70%	55%
TB	**85%**	**77%**	**64%**

Bảng II
So sánh độ chính xác của các mô hình nhận dạng cảm xúc trên bộ dữ liệu LFW

Mô hình	Độ chính xác LFW
DeepFace (Công bố)	95.3%
MultiMAE-DER	83.61%
EmotiEffLib-Base	70%
EmotiEffLib-XL (Công bố)	90.5%

D. Hiệu suất nhận dạng giọng nói

Hiệu suất nhận dạng giọng nói được đánh giá bằng cách thử nghiệm 50 lệnh thoại khác nhau trong các môi trường âm thanh khác nhau. Kết quả được trình bày trong Bảng III.

Bảng III
Độ chính xác nhận dạng giọng nói trong các môi trường khác nhau

Loại lệnh	Môi trường yên tĩnh	Tiếng ồn vừa	Tiếng ồn cao
Lệnh đơn giản	92%	85%	63%
Lệnh phức tạp	87%	76%	55%
Đọc đề xuất	85%	73%	51%
Chọn đề xuất	83%	72%	49%
Trung bình	**86.7%**	**76.5%**	**54.5%**

Kết quả cho thấy hệ thống nhận dạng giọng nói hoạt động tốt trong môi trường yên tĩnh (độ chính xác trung bình 87%) và môi trường có tiếng ồn vừa phải (độ chính xác trung bình 77%). Tuy nhiên, hiệu suất giảm đáng kể trong môi trường có nhiều tiếng ồn (độ chính xác trung bình 55%). Điều này dẫn đến việc triển khai các cơ chế giảm tiếng ồn và thuật toán tiền xử lý tín hiệu âm thanh để cải thiện hiệu suất trong môi trường thực tế.

E. Đánh giá độ trễ hệ thống

Độ trễ của hệ thống được đo đạc cho từng thành phần và toàn bộ quy trình. Kết quả được trình bày trong Bảng IV.

Độ trễ tổng thể của hệ thống từ khi phát hiện cảm xúc đến khi thiết bị IoT phản ứng nằm trong khoảng 2-4 giây, đủ để đảm bảo trải nghiệm tương tác tự nhiên cho người dùng. Phân tích chỉ ra rằng các thành phần đóng

Bảng IV
ĐỘ TRỄ CỦA CÁC THÀNH PHẦN TRONG HỆ THỐNG

Thành phần	Độ trễ trung bình
Phát hiện khuôn mặt	120ms
Nhận dạng cảm xúc	380ms
Nhận dạng giọng nói	1-2s
Text-to-Speech	500-700ms
Giao tiếp Client-Server	100-200ms
Giao tiếp Server-ESP32	200-300ms
Tổng độ trễ	**2-4s**

góp nhiều nhất vào độ trễ là quy trình nhận dạng giọng nói và chuyển đổi văn bản thành giọng nói. Đây là lý do chúng tôi đã tối ưu hóa quy trình xử lý âm thanh và áp dụng các kỹ thuật làm mát (cooldown) giữa các lần phát hiện cảm xúc để tránh gửi quá nhiều yêu cầu xử lý cùng lúc.

F. Điều khiển IoT

Khả năng điều khiển thiết bị IoT được đánh giá thông qua tỷ lệ thành công của các lệnh điều khiển và độ tin cậy của kết nối. Kết quả được trình bày trong Bảng V.

Bảng V
HIỆU SUẤT ĐIỀU KHIỂN THIẾT BỊ IOT

Chức năng	Tỷ lệ thành công	Độ trễ TB
Bật/tắt đèn	98%	250ms
Phát/dừng nhạc	95%	350ms
Điều chỉnh âm lượng	96%	300ms
Đọc nhiệt độ/độ ẩm	97%	280ms
Thực hiện đề xuất	92%	500ms
Trung bình	**95.6%**	**336ms**

Các lệnh điều khiển IoT đạt tỷ lệ thành công cao, trung bình là 95.6%. Chức năng bật/tắt đèn có tỷ lệ thành công cao nhất (98%) do tính đơn giản của lệnh và quy trình xử lý. Chức năng thực hiện đề xuất có tỷ lệ thành công thấp nhất (92%) và độ trễ cao nhất (500ms) do phải xử lý nhiều bước phức tạp hơn, bao gồm phân tích cảm xúc, tạo đề xuất, và thực hiện lệnh tương ứng. Độ tin cậy kết nối được cải thiện đáng kể nhờ cơ chế tự động kết nối lại của ESP32 và cơ chế đồng bộ trạng thái định kỳ.

G. Hiệu suất phát nhạc theo cảm xúc

Một đặc điểm nổi bật của hệ thống là khả năng phát nhạc phù hợp với trạng thái cảm xúc của người dùng. Module quản lý nhạc được triển khai với các chức năng: (1) phát nhạc theo cảm xúc, (2) dừng nhạc, (3) điều chỉnh âm lượng, và (4) lấy trạng thái nhạc hiện tại. Hệ thống duy trì các playlist khác nhau cho các cảm xúc khác nhau, đảm bảo tạo môi trường âm thanh phù hợp. Thử nghiệm cho thấy việc phát nhạc phù hợp với cảm xúc có thể cải thiện trạng thái tinh thần của người dùng, đặc biệt khi phát hiện cảm xúc buồn hoặc tức giận.

V. ƯU ĐIỂM VÀ HẠN CHẾ

A. Ưu điểm

- **Nhận dạng cảm xúc theo thời gian thực**: Hệ thống có khả năng phát hiện và phân tích cảm xúc khuôn mặt theo thời gian thực với độ chính xác cao trong điều kiện ánh sáng tốt.
- **Tương tác tiếng Việt**: Hệ thống hỗ trợ nhận dạng và tổng hợp giọng nói tiếng Việt, giúp tương tác tự nhiên với người dùng Việt Nam.
- **Phản hồi thông minh dựa trên cảm xúc**: Hệ thống không chỉ nhận diện cảm xúc mà còn đưa ra các đề xuất phù hợp với trạng thái cảm xúc của người dùng.
- **Kiến trúc mở**: Thiết kế module hóa cho phép dễ dàng mở rộng với nhiều loại thiết bị IoT và chức năng mới.
- **Hoạt động ổn định trong mạng LAN**: Hệ thống có độ tin cậy cao khi hoạt động trong môi trường mạng cục bộ.
- **Hệ thống đề xuất thông minh**: Cơ chế đề xuất tự động dựa trên cảm xúc giúp người dùng có trải nghiệm tự nhiên, không cần nhớ nhiều lệnh phức tạp.
- **Khả năng phục hồi**: Hệ thống được thiết kế với các cơ chế xử lý lỗi và phục hồi kết nối, đảm bảo hoạt động ổn định trong môi trường thực tế.

B. Hạn chế

- **Phụ thuộc vào điều kiện ánh sáng**: Hiệu suất nhận dạng cảm xúc giảm đáng kể trong điều kiện ánh sáng kém.
- **Độ chính xác nhận dạng giọng nói**: Trong môi trường ồn, khả năng nhận dạng giọng nói bị ảnh hưởng nghiêm trọng.
- **Chức năng IoT cơ bản**: Hiện tại, hệ thống mới chỉ hỗ trợ một số chức năng IoT đơn giản như bật/tắt đèn và điều khiển nhạc.
- **Yêu cầu kết nối mạng ổn định**: Hệ thống cần kết nối mạng ổn định giữa Client, Server và thiết bị IoT.
- **Hạn chế về mô hình cảm xúc**: Mô hình hiện tại chỉ nhận dạng 7 cảm xúc cơ bản, chưa bao gồm các cảm xúc phức tạp hơn.
- **Tiêu thụ tài nguyên cao**: Quy trình nhận dạng cảm xúc và xử lý giọng nói yêu cầu tài nguyên đáng kể, giới hạn khả năng triển khai trên các thiết bị có hiệu suất thấp.
- **Phạm vi nhận dạng hạn chế**: Hệ thống chỉ có thể nhận dạng cảm xúc của một người dùng tại một thời điểm, chưa hỗ trợ môi trường đa người dùng.

VI. HƯỚNG PHÁT TRIỂN

Dựa trên kết quả thực nghiệm và các hạn chế hiện tại, đề xuất các hướng phát triển sau cho hệ thống:

- **Cải thiện khả năng nhận dạng cảm xúc**: Áp dụng các kỹ thuật học sâu mới nhất và tăng cường dữ liệu

huấn luyện để cải thiện độ chính xác trong các điều kiện ánh sáng khác nhau.

- **Mở rộng hỗ trợ nhiều thiết bị IoT**: Tích hợp thêm nhiều loại thiết bị IoT như màn hình thông minh, thiết bị điều khiển nhiệt độ, rèm cửa tự động...
- **Tích hợp mô hình ngôn ngữ lớn (LLM)**: Kết hợp với các mô hình ngôn ngữ lớn như GPT để nâng cao khả năng tương tác tự nhiên và hiểu ngữ cảnh.
- **Học từ hành vi người dùng**: Phát triển các thuật toán học máy để hệ thống thích nghi với thói quen và sở thích cá nhân của người dùng theo thời gian.
- **Cải thiện bảo mật và quyền riêng tư**: Triển khai các giải pháp bảo mật để bảo vệ dữ liệu cảm xúc và thông tin cá nhân của người dùng.
- **Phát triển ứng dụng di động**: Xây dựng ứng dụng di động để người dùng có thể điều khiển hệ thống từ xa.
- **Hỗ trợ nhiều ngôn ngữ**: Mở rộng khả năng nhận dạng giọng nói và tổng hợp giọng nói cho nhiều ngôn ngữ khác ngoài tiếng Việt.
- **Tối ưu hóa hiệu suất**: Cải thiện hiệu suất xử lý để giảm độ trễ và tiêu thụ tài nguyên, cho phép triển khai trên các thiết bị cấp thấp hơn.
- **Phát triển khả năng nhận dạng đa người dùng**: Mở rộng hệ thống để nhận dạng và phản ứng với cảm xúc của nhiều người dùng cùng lúc, phù hợp cho các môi trường gia đình hoặc văn phòng.
- **Tích hợp phân tích cảm xúc qua giọng nói**: Bổ sung khả năng phát hiện cảm xúc từ âm sắc và ngữ điệu giọng nói, kết hợp với nhận dạng cảm xúc khuôn mặt để tăng độ chính xác.

VII. Kết luận

Bài báo cáo này đã trình bày về thiết kế, triển khai và đánh giá một hệ thống tích hợp AI-IoT có khả năng nhận diện cảm xúc khuôn mặt, tương tác bằng giọng nói và điều khiển các thiết bị IoT dựa trên ngữ cảnh cảm xúc. Kết quả thực nghiệm cho thấy hệ thống đạt độ chính xác nhận diện cảm xúc 85% trong điều kiện ánh sáng tốt, độ nhận diện giọng nói 87% với các lệnh đơn giản trong môi trường yên tĩnh, và độ trễ hệ thống chấp nhận được trong khoảng 2-4 giây.

Hệ thống đã chứng minh khả năng tạo ra môi trường thích ứng với trạng thái cảm xúc người dùng thông qua các chức năng như điều chỉnh ánh sáng và phát nhạc phù hợp. Cơ chế đề xuất tự động dựa trên cảm xúc tạo ra trải nghiệm tương tác tự nhiên và trực quan cho người dùng. Kiến trúc module hóa của hệ thống tạo điều kiện cho việc mở rộng và phát triển thêm các chức năng mới trong tương lai.

Với sự phát triển nhanh chóng của các công nghệ AI và IoT, chúng tôi tin rằng các hệ thống nhận diện cảm xúc và tương tác người-máy sẽ trở thành một phần quan trọng trong môi trường thông minh, mang lại trải nghiệm tương tác tự nhiên và cá nhân hóa cho người dùng. Những hướng phát triển trong tương lai như tích hợp mô hình ngôn ngữ lớn, học từ hành vi người dùng, và mở rộng khả năng nhận dạng đa người dùng sẽ nâng cao khả năng ứng dụng của hệ thống trong cuộc sống thực tế.

Lời cảm ơn

Em xin chân thành cảm ơn ThS. Lê Trung Hiếu và ThS. Nguyễn Thái Khánh đã hướng dẫn và hỗ trợ trong quá trình thực hiện đề tài này. Cảm ơn Khoa Công nghệ Thông tin, Đại học Đại Nam đã tạo điều kiện về cơ sở vật chất và trang thiết bị để thực hiện bài tập lớn.

Tài liệu

[1] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," Journal of Personality and Social Psychology, vol. 17, no. 2, pp. 124-129, 1971.

[2] L. Zhang, D. Tjondronegoro, and V. Chandran, "Facial expression recognition experiments with data from television broadcasts and the World Wide Web," Image and Vision Computing, vol. 32, no. 2, pp. 107-119, 2014.

[3] J. Shenk, "FER - Facial Expression Recognition," GitHub repository, 2019. [Online]. Available: https://github.com/justinshenk/fer

[4] S. Serengil, "DeepFace: A Lightweight Face Recognition and Facial Attribute Analysis Framework," GitHub repository, 2020. [Online]. Available: https://github.com/serengil/deepface

[5] A. Savchenko, "EmotiEffLib - Fast and Accurate Emotion Recognition Library," GitHub repository, 2022. [Online]. Available: https://github.com/av-savchenko/EmotiEffLib

[6] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," Advances in Neural Information Processing Systems, vol. 33, pp. 12449-12460, 2020.

[7] Espressif Systems, "ESP32 Series Datasheet," Espressif Systems, 2021.

[8] OpenCV, "Open Source Computer Vision Library," 2021. [Online]. Available: https://opencv.org/

[9] Google, "Google Speech-to-Text API," Google Cloud Platform, 2022. [Online]. Available: https://cloud.google.com/speech-to-text

[10] S. Li and W. Deng, "Deep Facial Expression Recognition: A Survey," IEEE Transactions on Affective Computing, vol. 13, no. 3, pp. 1195-1215, 2022.