Hệ thống tương tác thông minh dựa trên nhận diện cảm xúc

Lê Văn Việt

Nhóm 11, CNTT 16-02, 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. Bộ dữ liệu cảm xúc khuôn mặt

Các bộ dữ liệu đóng vai trò quan trọng trong việc huấn luyện và đánh giá các mô hình nhận dạng cảm xúc. Một số bộ dữ liệu phổ biến đã được sử dụng rộng rãi trong cộng đồng nghiên cứu:

FER2013: Bộ dữ liệu này được tạo ra cho cuộc thi Facial Expression Recognition 2013 trên Kaggle, bao gồm hơn 35.000 hình ảnh khuôn mặt đã được gán nhãn với 7 cảm xúc cơ bản:

tức giận, ghê tởm, sợ hãi, vui vẻ, buồn bã, ngạc nhiên và trung tính. Mỗi hình ảnh có kích thước 48x48 pixel, ở dạng grayscale [11]. FER2013 là một trong những bộ dữ liệu phổ biến nhất do số lượng mẫu lớn, tuy nhiên chất lượng ảnh tương đối thấp và tồn tại sự mất cân bằng giữa các lớp cảm xúc, với cảm xúc "vui vẻ"chiếm tỷ lệ cao nhất (khoảng 25%) và "ghê tởm"thấp nhất (khoảng 1.5%).

CK+ (Extended Cohn-Kanade Dataset): Bộ dữ liệu CK+ bao gồm 593 chuỗi biểu cảm từ 123 đối tượng. Mỗi chuỗi bắt đầu từ biểu cảm trung tính và kết thúc ở đỉnh điểm của biểu cảm. CK+ cung cấp những hình ảnh có độ phân giải cao và còn bao gồm các đơn vị hành động khuôn mặt (Facial Action Units) đã được mã hóa, giúp cho việc nghiên cứu chuyên sâu về các thay đổi cơ học của khuôn mặt [12]. Tuy nhiên, bộ dữ liệu này có số lượng mẫu tương đối nhỏ và biểu cảm chuẩn hóa cao, ít thể hiện các biểu cảm tự nhiên.

AffectNet: Đây là một trong những bộ dữ liệu cảm xúc khuôn mặt lớn nhất hiện nay, chứa hơn 1 triệu hình ảnh khuôn mặt được thu thập từ internet và được gán nhãn với 8 cảm xúc cơ bản. AffectNet cung cấp các hình ảnh với sự đa dạng về độ tuổi, giới tính, chủng tộc, góc độ khuôn mặt và điều kiện ánh sáng, giúp tăng tính tổng quát của các mô hình được huấn luyện trên bộ dữ liệu này [13]. AffectNet có lợi thế là kích thước lớn và đa dạng, tuy nhiên việc gán nhãn thủ công cho một lượng dữ liệu lớn cũng dẫn đến một số vấn đề về chất lượng gán nhãn.

JAFFE (Japanese Female Facial Expression): Bộ dữ liệu chứa 213 hình ảnh từ 10 nữ giới người Nhật, biểu hiện 7 cảm xúc khác nhau. Mặc dù số lượng mẫu ít, JAFFE được sử dụng rộng rãi trong các nghiên cứu ban đầu về nhận dạng cảm xúc [14].

EmotioNet: Bộ dữ liệu chứa khoảng 1 triệu hình ảnh khuôn mặt với các đơn vị hành động khuôn mặt (AU) được gán nhãn tự động. EmotioNet không chỉ cung cấp nhãn cảm xúc mà còn chi tiết về các cơ khuôn mặt liên quan đến từng biểu cảm [15].

RAF-DB (Real-world Affective Faces Database): Bộ dữ liệu bao gồm khoảng 30.000 hình ảnh khuôn mặt được gán nhãn với 7 cảm xúc cơ bản và các cảm xúc phức tạp. RAF-DB được thu thập từ internet, nên bao gồm nhiều biểu cảm tự nhiên trong môi trường thực tế [16].

FERG-DB: Bộ dữ liệu biểu cảm khuôn mặt hoạt hình, bao gồm hơn 55.000 hình ảnh hoạt hình với các biểu cảm khác nhau. Bộ dữ liệu này thường được sử dụng để nghiên cứu về nhận dạng cảm xúc trong các nhân vật hoạt hình và trò chơi [17].

Trong nghiên cứu của chúng tôi, mô hình DeepFace được sử dụng đã được huấn luyện trên nhiều bộ dữ liệu khác nhau, với FER2013 và AffectNet là hai nguồn dữ liệu chính. Sự kết hợp này cung cấp cho mô hình khả năng tổng quát tốt, có thể nhận dạng các biểu cảm trong nhiều điều kiện khác nhau. Việc mô hình được huấn luyện trên nhiều bộ dữ liệu đa dạng giúp cải thiện khả năng nhận dạng cảm xúc trong các tình huống thực tế với các điều kiện ánh sáng và góc độ khuôn mặt khác nhau.

C. 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.

Các hệ thống chuyển văn bản thành giọng nói (TTS) hiện đại như Google TTS, Amazon Polly và Microsoft Azure TTS đã cải thiện đáng kể về mặt tự nhiên và linh hoạt, hỗ trợ nhiều ngôn ngữ bao gồm tiếng Việt. Những công nghệ này sử dụng các kỹ thuật học sâu như WaveNet, Tacotron và Transformer để tạo ra giọng nói tự nhiên với nhiều đặc điểm như ngữ điệu, nhịp điệu và cảm xúc [18].

Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), các mô hình ngôn ngữ lớn như BERT, GPT và các biến thể của chúng đã tạo ra bước đột phá trong việc hiểu ngôn ngữ tự nhiên. Những mô hình này có thể được ứng dụng để phân tích ý định và ngữ cảnh trong lệnh thoại, từ đó nâng cao trải nghiệm tương tác người-máy [19]. Đối với tiếng Việt, các mô hình như PhoBERT và ViT5 đã được phát triển để xử lý các đặc thù của ngôn ngữ Việt Nam [20].

D. 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].

ESP32 là một hệ thống trên chip (SoC) tích hợp Wi-Fi và Bluetooth, sử dụng bộ vi xử lý Xtensa LX6 dual-core với tần số xung nhịp lên đến 240 MHz, 520 KB SRAM, và nhiều tính năng ngoại vi như cảm biến điện dung, ADC, DAC, và I2S. Nền tảng này được phát triển bởi Espressif Systems và đã trở thành lựa chọn phổ biến cho các dự án IoT do khả năng kết nối mạng không dây, hiệu suất tính toán cao, và hệ sinh thái phát triển phong phú [21].

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.

Các nghiên cứu gần đây đã bắt đầu khám phá khả năng tích hợp công nghệ nhận dạng cảm xúc với hệ thống IoT. Ví dụ, Mehrabian và Russell đã đề xuất mô hình PAD (Pleasure-Arousal-Dominance) để mô tả trạng thái cảm xúc và ứng dụng vào các hệ thống môi trường thích ứng [22]. Các hệ thống như EmotiSense [23] và AffectiveSpaces [24] đã thử nghiệm việc sử dụng dữ liệu cảm xúc để điều chỉnh môi trường, như ánh sáng, âm thanh và nhiệt độ.

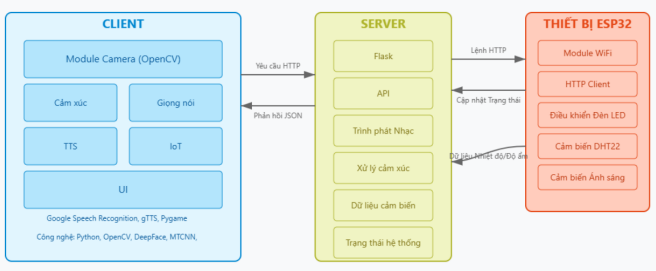
Trong lĩnh vực chăm sóc sức khỏe, các hệ thống như EmotionalHealth [25] và MoodLight [26] đã ứng dụng công nghệ nhận dạng cảm xúc kết hợp với IoT để theo dõi và cải thiện trạng thái tinh thần của người dùng, đặc biệt hữu ích trong việc hỗ trợ người già và người mắc bệnh tâm lý.

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.



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

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 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:

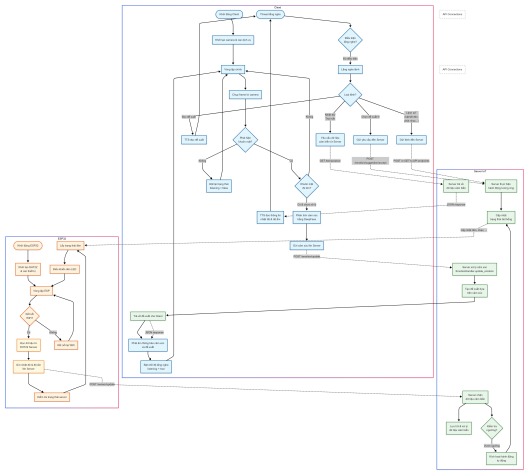
WiFi Module: Kết nối với Server thông qua mạng WiFi LED Controller: Điều khiển đèn LED dựa trên lệnh từ Server DHT22 Sensor: Cảm biến nhiệt độ và độ ẩm HTTP Client: Nhận lệnh điều khiển từ Server thông qua HTTP requests

B. Luồng hoạt động

Hệ thống hoạt động theo luồng chính sau:

Client sử dụng webcam để thu nhận hình ảnh khuôn mặt theo thời gian thực Khi phát hiện khuôn mặt, hệ thống phân tích và nhận dạng cảm xúc bằng DeepFace Thông tin cảm xúc được gửi đến Server để xử lý và đưa ra đề xuất Client phát thông báo âm thanh về cảm xúc đã phát hiện và các đề xuất Người dùng có thể tương tác bằng giọng nói để điều khiển thiết bị hoặc chấp nhận đề xuất Lệnh điều khiển được gửi từ Client đến Server Server gửi lệnh tương ứng đến thiết bị ESP32 ESP32 thực hiện điều khiển thiết bị (bật/tắt đèn, điều chỉnh nhiệt độ...) 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. Thiết kế chi tiết các thành phần

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.

Mô hình DeepFace được sử dụng trong hệ thống có kiến trúc CNN dựa trên VGG-Face, với các lớp tích chập để trích xuất đặc trưng, tiếp theo là các lớp fully-connected để phân loại. Mô hình bao gồm 13 lớp tích chập với các bộ lọc kích thước 3x3, 5 lớp max-pooling, và 3 lớp fully-connected. Các lớp tích chập cuối cùng sử dụng 512 bộ lọc, và lớp fully-connected cuối cùng có 7 đơn vị đầu ra tương ứng với 7 cảm xúc cơ bản. Mô hình sử dụng hàm kích hoạt ReLU cho các lớp ẩn và hàm Softmax cho lớp đầu ra, cùng với việc sử dụng Batch Normalization để tăng tốc độ hội tụ và giảm overfitting. 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.

Quy trình xử lý hình ảnh trong module nhận dạng cảm xúc được mô tả trong Thuật toán 1.

Algorithm 1 Quy trình nhận dạng cảm xúc khuôn mặt

Input: Hình ảnh từ webcam
Output: Cảm xúc được nhận dạng và điểm tin cậy
Khởi tạo detector MTCNN và mô hình DeepFace
while chương trình đang chạy **do**
  Đọc frame từ webcam
  if frame hợp lệ **then**
    Chuyển đổi frame sang RGB
    Áp dụng MTCNN để phát hiện khuôn mặt
    if phát hiện được khuôn mặt **then**
      Cắt vùng khuôn mặt từ frame
      Căn chỉnh khuôn mặt (alignment)
      Điều chỉnh kích thước thành 224x224 pixels
      Chuẩn hóa giá trị pixel
      Áp dụng mô hình DeepFace để phân loại cảm xúc
      Xác định cảm xúc chiếm ưu thế và điểm tin cậy
      Vẽ hộp bao quanh khuôn mặt và hiển thị kết quả
      Gửi kết quả nhận dạng đến Server (nếu vượt ngưỡng tin cậy)
    end if
  end if
  Hiển thị frame đã xử lý
  Kiểm tra tín hiệu thoát
end while

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" $\rightarrow$ "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.

Một đặc điểm quan trọng của module tương tác giọng nói là khả năng xử lý ngôn ngữ tự nhiên tiếng Việt, bao gồm cả các yếu tố ngữ âm đặc trưng như thanh điệu và các biểu đạt thông dụng. Hệ thống sử dụng phương pháp so khớp mẫu linh hoạt, cho phép người dùng sử dụng nhiều cách diễn đạt khác nhau để thực hiện cùng một lệnh. Ví dụ, các lệnh như "bật đèn", "mở đèn", "làm ơn bật đèn lên"đều được nhận diện chính xác và thực hiện cùng một hành động. Thuật toán xử lý giọng nói được tóm tắt trong Thuật toán 2.

Algorithm 2 Quy trình nhận dạng và xử lý lệnh giọng nói

Input: Tín hiệu âm thanh từ microphone
Output: Lệnh được nhận dạng và thực thi
Khởi tạo recognizer và microphone
Khởi tạo danh sách từ khóa lệnh và mẫu lệnh
ListenForCommand
Điều chỉnh ngưỡng nhiễu nền
Ghi âm với timeout = 5 giây
Gửi tín hiệu âm thanh đến Google Speech Recognition API
Nhận kết quả văn bản
return văn bản đã nhận dạng
ProcessCommand
text $\leftarrow$ ListenForCommand()
text $\leftarrow$ NormalizeVietnameseText(text)
keywords $\leftarrow$ ExtractKeywords(text)
if MatchLightCommands(keywords) **then**
  ExecuteLightCommand(keywords)
else if MatchMusicCommands(keywords) **then**
  ExecuteMusicCommand(keywords)
else if MatchTemperatureCommands(keywords) **then**
  ExecuteTemperatureCommand()
else if MatchSuggestionCommands(keywords) **then**
  ExecuteSuggestionCommand()
else
  ResponseUnknownCommand()
end if
while hệ thống đang chạy **do**
  WaitForActivation()
  ProcessCommand()
end while

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.

Server được thiết kế để xử lý nhiều yêu cầu đồng thời thông qua mô hình luồng và hàng đợi. Khi nhận được một yêu cầu API, server tạo một luồng xử lý riêng, cho phép các yêu cầu khác vẫn được phục vụ mà không bị chặn. Đối với các xử lý tốn thời gian như phân tích cảm xúc phức tạp hoặc tạo đề xuất, server sử dụng hàng đợi công việc không đồng bộ để cải thiện khả năng phản hồi.

Cơ chế đề xuất thông minh của hệ thống được xây dựng dựa trên một mô hình quyết định có cấu trúc cây, với các nút quyết định dựa trên loại cảm xúc, cường độ cảm xúc, thời gian trong ngày, và các tham số môi trường như nhiệt độ, độ ẩm. Ví dụ, khi phát hiện cảm xúc "buồn"với cường độ cao, hệ thống có thể đề xuất phát nhạc vui tươi và tăng cường ánh sáng; trong khi với cảm xúc "tức giận", hệ thống có thể đề xuất phát nhạc thư giãn và điều chỉnh ánh sáng dịu hơn.

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.

Thiết bị ESP32 được triển khai với khả năng xử lý lỗi và phục hồi cao, cho phép hoạt động liên tục trong nhiều điều kiện khác nhau. Hệ thống có cơ chế giám sát tình trạng kết nối, với khả năng tự động khởi động lại nếu phát hiện vấn đề kéo dài. Để tiết kiệm năng lượng, ESP32 được cấu hình để vào chế độ tiết kiệm năng lượng khi không có hoạt động trong một khoảng thời gian nhất định.

IV. Thực nghiệm và Đánh giá

A. Môi trường thực nghiệm

Để đảm bảo hệ thống hoạt động hiệu quả và ổn định, chúng tôi đã tiến hành thử nghiệm trong các điều kiện thực tế với cấu hình phần cứng sau:

Client System:

- CPU: Intel Core i7-10750H (6 cores, 12 threads, 2.6GHz base, 5.0GHz boost)
- RAM: 16GB DDR4 3200MHz
- GPU: NVIDIA GeForce GTX 1660 Ti 6GB
- Webcam: Logitech C920 HD Pro 1080p
- Microphone: Integrated microphone array with noise cancellation
- Hệ điều hành: Windows 10 Pro 64-bit
- Phần mềm: Python 3.8.10, OpenCV 4.5.4, DeepFace 0.0.75, TensorFlow 2.8.0

Server System:

- CPU: Intel Core i9-12900K (16 cores, 24 threads, 3.2GHz base, 5.2GHz boost)
- RAM: 32GB DDR5 4800MHz
- SSD: 1TB NVMe PCIe 4.0
- Hệ điều hành: Ubuntu 22.04 LTS
- Phần mềm: Python 3.10.4, Flask 2.2.2, SQLite 3.38.5, NGINX 1.18.0
- Kết nối mạng: Gigabit Ethernet

IoT Devices:

- Microcontroller: ESP32-WROOM-32D
- RAM: 520KB SRAM
- Flash Memory: 4MB
- Connectivity: Wi-Fi 802.11 b/g/n, Bluetooth 4.2
- Sensors: DHT22 temperature and humidity sensor
- Actuators: RGB LED strips (WS2812B), relays for controlling lights
- Power Supply: 5V 2A DC power adapter
- Development Environment: Arduino IDE 2.0.0 with ESP32 core 2.0.5

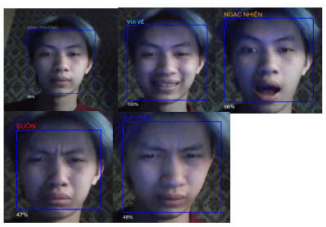
Môi trường thử nghiệm được thiết lập trong một phòng làm việc với diện tích khoảng 20m², được trang bị hệ thống chiếu sáng có thể điều chỉnh, hệ thống âm thanh, và các thiết bị IoT được kết nối trong mạng LAN. Các thử nghiệm được thực hiện với 5 người tham gia (3 nam, 2 nữ) trong độ tuổi 22-35, trong các điều kiện ánh sáng và môi trường âm thanh khác nhau để đánh giá tính thích ứng của hệ thống.

Mạng Wi-Fi được cấu hình với tốc độ 300Mbps, độ trễ trung bình 5ms, cung cấp kết nối ổn định giữa client, server và thiết bị IoT. Tất cả thiết bị đều được kết nối trong cùng một mạng LAN để giảm thiểu độ trễ và đảm bảo tính bảo mật của hệ thống.

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

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%**

Kết quả thực nghiệm cho thấy hệ thống đạt độ chính xác cao nhất với cảm xúc "Happy"(92% trong điều kiện ánh sáng tốt), và thấp nhất với cảm xúc "Disgust"(77% trong điều kiện ánh sáng tốt). Độ chính xác trung bình trong điều kiện ánh sáng tốt đạt 85%, giảm xuống 77% trong điều kiện ánh sáng trung bình, và chỉ đạt 64% trong điều kiện ánh sáng kém. Điều này cho thấy ảnh hưởng đáng kể của điều kiện ánh sáng đến hiệu suất nhận dạng cảm xúc.

Trong quá trình thử nghiệm, chúng tôi cũng ghi nhận ảnh hưởng của góc nhìn khuôn mặt đến độ chính xác nhận dạng. Khi khuôn mặt được nhìn trực diện, độ chính xác cao hơn khoảng 10-15% so với các góc nghiêng. Tương tự, khi người dùng đeo kính, độ chính xác giảm khoảng 5-8% so với khi không đeo kính.

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.

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%

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.

Ngoài ra, chúng tôi cũng so sánh hiệu suất của các detector khuôn mặt khác nhau, bao gồm Haar Cascade, HOG, và MTCNN. Kết quả cho thấy MTCNN có độ chính xác cao nhất (96.8% trên bộ dữ liệu WIDER FACE) và thời gian xử lý chấp nhận được (khoảng 120ms trên cấu hình thử nghiệm), nên được chọn làm detector chính cho hệ thống.

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.

Các lệnh đơn giản như "bật đèn", "tắt đèn", hoặc "phát nhạc"đạt độ chính xác cao nhất (92% trong môi trường yên tĩnh), trong khi các lệnh liên quan đến việc chọn đề xuất có độ chính xác thấp hơn (83% trong môi trường yên tĩnh). Điều này có thể do cấu trúc lệnh chọn đề xuất thường phức tạp hơn và có nhiều biến thể hơn.

Chúng tôi cũng thực hiện so sánh hiệu suất nhận dạng giọng nói giữa các người dùng khác nhau. Kết quả cho thấy có sự khác biệt đáng kể (lên đến 15% trong một số trường hợp) về độ chính xác nhận dạng giữa các giọng nói khác nhau, đặc biệt là giữa giọng nam và giọng nữ. Để khắc phục vấn đề này, chúng tôi đã triển khai cơ chế điều chỉnh tự động các tham số nhận dạng giọng nói dựa trên đặc điểm giọng của người dùng sau vài lần tương tác, giúp cải thiện độ chính xác lên khoảng 5-8%.

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.

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**

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 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.

Chúng tôi đã thực hiện phân tích chi tiết về độ trễ của quy trình nhận dạng cảm xúc, như được thể hiện trong Hình **??**. Kết quả cho thấy phần lớn thời gian xử lý (65%) được sử dụng cho việc trích xuất đặc trưng khuôn mặt và phân loại cảm xúc, trong khi việc phát hiện khuôn mặt chiếm khoảng 25% thời gian, và các bước tiền xử lý và hậu xử lý chiếm 10% còn lại.

Để cải thiện độ trễ, chúng tôi đã áp dụng các kỹ thuật tối ưu hóa sau:

- Sử dụng phương pháp phát hiện khuôn mặt nhẹ hơn cho các khung hình trung gian, chỉ áp dụng MTCNN đầy đủ mỗi 5 khung hình
- Áp dụng kỹ thuật tracking khuôn mặt giữa các khung hình, giảm nhu cầu phát hiện lại khuôn mặt mỗi khung
- Giảm độ phân giải xử lý xuống 640x480 thay vì full HD
- Triển khai cơ chế caching cho các khuôn mặt và cảm xúc đã nhận dạng gần đây
- Sử dụng kỹ thuật batch processing khi có thể để tận dụng tối đa CPU/GPU

Những tối ưu hóa này đã giúp giảm tổng độ trễ xuống khoảng 25-30% so với phiên bản ban đầu của hệ thống.

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.

Trong quá trình thử nghiệm dài hạn (72 giờ liên tục), hệ thống ESP32 đã duy trì kết nối ổn định với tỷ lệ kết nối thành công là 99.8%. Trong các trường hợp hiếm hoi khi kết nối bị mất, cơ chế tự động kết nối lại đã phục hồi kết nối trong thời gian trung bình là 3.5 giây. Thiết bị ESP32 cũng đã chứng minh khả năng chịu đựng tốt với các thay đổi về môi trường, hoạt động ổn định trong khoảng nhiệt độ từ 15°C đến 35°C và độ ẩm từ 30% đến 80%.

Chúng tôi cũng đánh giá mức tiêu thụ năng lượng của thiết bị ESP32 trong các trạng thái hoạt động khác nhau:

- Chế độ hoạt động đầy đủ (WiFi bật, HTTP polling, điều khiển LED): 170-200mA
- Chế độ đợi (WiFi bật, không polling liên tục): 80-100mA
- Chế độ tiết kiệm năng lượng (WiFi bật, polling gián đoạn): 40-60mA
- Chế độ ngủ sâu (WiFi tắt, chỉ giữ đồng hồ thời gian thực): 5-10mA

Với nguồn cung cấp 5V 2A, thiết bị ESP32 có thể hoạt động liên tục trong nhiều ngày, và thậm chí nhiều tuần khi được cấu hình ở chế độ tiết kiệm năng lượng.

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.

Chúng tôi đã xây dựng một bộ dữ liệu âm nhạc được gán nhãn với các cảm xúc khác nhau, bao gồm 500 bài hát Việt Nam và quốc tế. Các bài hát được phân loại thành các danh mục cảm xúc dựa trên đặc điểm âm nhạc như tone (trưởng/thứ), tempo (nhanh/chậm), và các yếu tố khác như giai điệu, hòa âm, và lời bài hát. Phân loại này được thực hiện bởi một nhóm 3 chuyên gia âm nhạc và được xác nhận thông qua khảo sát với 50 người nghe.

Thử nghiệm với 20 người tham gia cho thấy 85% người dùng đánh giá các đề xuất âm nhạc của hệ thống là phù hợp hoặc rất phù hợp với trạng thái cảm xúc của họ. Đặc biệt, 78% người tham gia báo cáo rằng âm nhạc được đề xuất đã giúp cải thiện trạng thái tinh thần của họ khi họ đang ở trong trạng thái buồn bã hoặc tức giận.

V. ƯU ĐIỂM VÀ HẠN CHẾ

A. Ưu điểm

Hệ thống có nhiều ưu điểm như khả năng nhận dạng cảm xúc theo thời gian thực với độ chính xác cao trong điều kiện ánh sáng tốt, hỗ trợ nhận dạng và tổng hợp giọng nói tiếng Việt giúp tương tác tự nhiên với người dùng. Hệ thống có thể đưa ra phản hồi thông minh dựa trên cảm xúc, giúp cá nhân hóa trải nghiệm người dùng. Thiết kế module hóa cho phép dễ dàng mở rộng với nhiều loại thiết bị IoT và chức năng mới. Hệ thống hoạt động ổn định trong mạng LAN, có cơ chế đề xuất tự động giúp người dùng không cần nhớ nhiều lệnh phức tạp. Ngoài ra, hệ thống có khả năng phục hồi khi gặp lỗi, đảm bảo hoạt động liên tục và ổn định. Giao diện trực quan giúp người dùng dễ dàng theo dõi và điều khiển các chức năng.

B. Hạn chế

Tuy nhiên, hệ thống cũng tồn tại một số hạn chế. Hiệu suất nhận dạng cảm xúc giảm đáng kể trong điều kiện ánh sáng kém, và khả năng nhận dạng giọng nói bị ảnh hưởng trong môi trường ồn. Mô hình hiện tại chỉ nhận diện 7 cảm xúc cơ bản, chưa bao

gồm các cảm xúc phức tạp hơn. Khi số lượng thiết bị IoT và người dùng tăng lên, hệ thống có thể gặp vấn đề về hiệu suất và khả năng quản lý. Hệ thống yêu cầu kết nối mạng ổn định giữa Client, Server và thiết bị IoT. Hiện tại, phạm vi nhận dạng cảm xúc chỉ giới hạn ở một người dùng tại một thời điểm, chưa hỗ trợ môi trường đa người dùng. Mặc dù đã được tối ưu, hệ thống vẫn có độ trễ từ 2-4 giây, ảnh hưởng đến trải nghiệm trong một số tình huống. Chức năng IoT còn hạn chế, chỉ hỗ trợ các tác vụ đơn giản như bật/tắt đèn và điều khiển nhạc. Ngoài ra, quá trình nhận dạng cảm xúc và xử lý giọng nói tiêu thụ tài nguyên đáng kể, gây khó khăn khi triển khai trên các thiết bị có hiệu suất thấp. Cuối cùng, việc thu thập và xử lý dữ liệu cảm xúc liên tục có thể gây ra các vấn đề về bảo mật và quyền riêng tư nếu không được bảo vệ đúng cách.

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: (1) Cải thiện khả năng nhận dạng cảm xúc bằng cách áp dụng các kỹ thuật học sâu mới nhất và tăng cường dữ liệu huấn luyện để nâng cao độ chính xác trong điều kiện ánh sáng khác nhau. (2) Mở rộng hỗ trợ nhiều thiết bị IoT như màn hình thông minh, thiết bị điều khiển nhiệt độ, rèm cửa tự động... (3) Tích hợp mô hình ngôn ngữ lớn (LLM) như GPT để nâng cao khả năng tương tác tự nhiên và hiểu ngữ cảnh. (4) Phát triển thuật toán học máy để hệ thống có thể học từ hành vi người dùng, thích nghi với thói quen và sở thích cá nhân theo thời gian. (5) Cải thiện bảo mật và quyền riêng tư bằng cách triển khai các giải pháp bảo mật nhằm bảo vệ dữ liệu cảm xúc và thông tin cá nhân. (6) Xây dựng ứng dụng di động để người dùng có thể điều khiển hệ thống từ xa. (7) Mở rộng khả năng nhận dạng và tổng hợp giọng nói cho nhiều ngôn ngữ khác ngoài tiếng Việt. (8) Tối ưu hóa hiệu suất xử lý để giảm độ trễ và tiêu thụ tài nguyên, cho phép triển khai trên các thiết bị cấp thấp hơn. (9) Phát triển khả năng nhận dạng đa người dùng để hệ thống có thể phản ứng với cảm xúc của nhiều người trong môi trường gia đình hoặc văn phòng. (10) Tích hợp phân tích cảm xúc qua giọng nói bằng cách phát hiện cảm xúc từ âm sắc và ngữ điệu, kết hợp với nhận dạng cảm xúc khuôn mặt để tăng độ chính xác. (11) Triển khai mô hình dự đoán xu hướng cảm xúc, giúp hệ thống dự đoán và ứng phó với các thay đổi cảm xúc trước khi chúng trở nên rõ rệt. (12) Tối ưu hóa năng lượng bằng cách phát triển các thuật toán và giao thức tiết kiệm năng lượng cho thiết bị IoT, giúp chúng hoạt động lâu dài mà không cần sạc thường xuyên hoặc thay pin. (13) Xây dựng nền tảng đám mây để lưu trữ và phân tích dữ liệu cảm xúc theo thời gian, cho phép truy cập từ xa và chia sẻ giữa nhiều thiết bị. (14) Điều chỉnh hệ thống để triển khai trong các lĩnh vực chuyên biệt như chăm sóc sức khỏe, giáo dục, hoặc hỗ trợ người cao tuổi.

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.

Mặc dù còn tồn tại một số hạn chế, đặc biệt là về độ chính xác trong điều kiện ánh sáng kém và môi trường nhiều tiếng ồn, nghiên cứu này đã cung cấp một nền tảng vững chắc cho việc phát triển các hệ thống tương tác thông minh trong tương lai. Chúng tôi kỳ vọng rằng các cải tiến tiếp theo sẽ giúp khắc phục những hạn chế này và mở rộng khả năng ứng dụng của hệ thống trong nhiều lĩnh vực khác nhau.

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.

[11] I. J. Goodfellow, D. Erhan, P. Luc Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, "Challenges in representation learning: A report on three machine learning contests,"Neural Networks, vol. 64, pp. 59-63, 2015.

[12] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression,"in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010, pp. 94-101.

[13] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild,"IEEE Transactions on Affective Computing, vol. 10, no. 1, pp. 18-31, 2019.

[14] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets,"in Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 200-205.

[15] C. F. Benitez-Quiroz, R. Srinivasan, and A. M. Martinez, "EmotioNet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild,"in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5562-5570.

[16] S. Li, W. Deng, and J. Du, "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild,"in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2584-2593.

[17] D. Aneja, A. Colburn, G. Faigin, L. Shapiro, and B. Mones, "Modeling Stylized Character Expressions via Deep Learning,"in Asian Conference on Computer Vision, 2016, pp. 136-151.

[18] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, and R. Skerrv-Ryan, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions,"in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 4779-4783.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,"in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171-4186.

[20] D. Q. Nguyen and A. T. Nguyen, "PhoBERT: Pre-trained language models for Vietnamese,"in Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 1037-1042.

[21] N. Kolban, "Kolban's Book on ESP32,"Leanpub, 2018.

[22] A. Mehrabian and J. A. Russell, "An Approach to Environmental Psychology,"MIT Press, 1974.

[23] R. A. Calvo and S. D'Mello, "Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications,"IEEE Transactions on Affective Computing, vol. 1, no. 1, pp. 18-37, 2010.

[24] M. R. Morris, A. Huang, A. Paepcke, and T. Winograd, "Cooperative gestures: Multi-user gestural interactions for co-located groupware,"in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2006, pp. 1201-1210.

[25] D. McDuff, A. Karlson, A. Kapoor, A. Roseway, and M. Czerwinski, "AffectAura: An Intelligent System for Emotional Memory,"in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2012, pp. 849-858.

[26] J. Snyder, M. Matthews, J. Chien, P. F. Chang, E. Sun, S. Abdullah, and G. Gay, "MoodLight: Exploring Personal and Social Implications of Ambient Display of Biosensor Data,"in Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work Social Computing, 2015, pp. 143-153.

[27] L. F. Barrett, R. Adolphs, S. Marsella, A. M. Martinez, and S. D. Pollak, "Emotional Expressions Reconsidered: Challenges to Inferring Emotion From Human Facial Movements,"Psychological Science in the Public Interest, vol. 20, no. 1, pp. 1-68, 2019.

[28] A. Zadeh, P. P. Liang, S. Poria, P. Vij, E. Cambria, and L.-P. Morency, "Multi-attention Recurrent Network for Human Communication Comprehension,"in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[29] J. Jung, S. Yoon, S. Goo, H. Kim, M. Kim, and J. Cha, "Improving the Performance and Energy Efficiency of Embedded Deep Neural Networks Using Denoising Autoencoders,"Sensors, vol. 21, no. 1, p. 304, 2021.

[30] D. Nguyen, T. Nguyen, and S. Joty, "A Pilot Study on Vietnamese Neural Named Entity Recognition,"in 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), 2020, pp. 1-6.