**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**🙟🕮🙝**

**A logo of hands holding a book and a flame

Description automatically generated**

**TIỂU LUẬN KẾT THÚC HỌC PHẦN**

**ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**Mã học phần : PROJ215879**

**TÊN ĐỒ ÁN :**

**XÂY DỰNG TRỢ LÝ ẢO ĐIỀU KHIỂN BẰNG GIỌNG NÓI VÀ CỬ CHỈ**

**Nhóm sinh viên thực hiện: Nhóm 6**

|  |  |
| --- | --- |
| 1. **Dương Đình Vũ** | **22110269** |
| 1. **Phan Hùng Anh** | **22110102** |

|  |
| --- |
| **Tp.HCM, tháng 11 năm 2024** |

**Giảng viên hướng dẫn: ThS. Trần Nhật Quang**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**🙟🕮🙝**

**A logo of hands holding a book and a flame

Description automatically generated**

**TIỂU LUẬN KẾT THÚC HỌC PHẦN**

**ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**Mã học phần: PROJ215879**

**TÊN ĐỒ ÁN**

**XÂY DỰNG TRỢ LÝ ẢO ĐIỀU KHIỂN BẰNG GIỌNG NÓI VÀ CỬ CHỈ**

**Nhóm sinh viên thực hiện: Nhóm 6**

|  |  |
| --- | --- |
| 1. **Dương Đình Vũ** | **22110269** |
| 1. **Phan Hùng Anh** | **22110102** |

|  |
| --- |
| **Tp.HCM, tháng 11 năm 2024** |

**Giảng viên hướng dẫn: ThS. Trần Nhật Quang**

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Ngày……….Tháng………Năm……

**Giáo Viên Hướng Dẫn**

(Ký và ghi rõ họ tên)

**LỜI CẢM ƠN**

Nhóm chúng em xin chân thành bày tỏ lòng biết ơn sâu sắc đến thầy Trần Nhật Quang, giảng viên khoa Công Nghệ Thông Tin tại trường Đại học SPKT. Sự hướng dẫn tận tình và tâm huyết của thầy không chỉ trang bị cho chúng em những kiến thức quý báu mà còn mang lại sự hỗ trợ quý giá xuyên suốt quá trình học tập.

Chúng em thực sự cảm kích trước sự nhiệt tình chia sẻ tri thức và sự đồng hành của thầy trong hành trình hoàn thành môn học và bài tiểu luận này. Đặc biệt, chúng em cũng xin gửi lời cảm ơn chân thành đến khoa và nhà trường, nhờ vào sự ủng hộ và hỗ trợ từ các thầy cô và bạn bè mà chúng em có thể tiếp thu những kiến thức một cách sâu sắc hơn.

Trong quá trình hoàn thiện bài tiểu luận, chúng em nhận thức rằng vẫn còn những thiếu sót. Vì vậy, chúng em rất trân trọng và mong nhận được những ý kiến đóng góp từ thầy, điều này sẽ giúp chúng em nâng cao kiến thức và kỹ năng của bản thân.

Xin chân thành cảm ơn thầy trong suốt quá trình vừa qua. Mong rằng nhận được sự đồng hành và hỗ trợ từ thầy trong tương lai. Kính chúc thầy sức khỏe và thành công trong sự nghiệp giảng dạy!

**Chúng em xin chân thành cảm ơn!**

MỤC LỤC

[A. MỞ ĐẦU 1](#_Toc183730241)

[1.Lý do chọn đề tài. 1](#_Toc183730242)

[2. Phương pháp nghiên cứu. 1](#_Toc183730243)

[B. NỘI DUNG 2](#_Toc183730244)

[CHƯƠNG 1: ĐẶT TẢ 2](#_Toc183730245)

[CHƯƠNG 2: PHÂN CÔNG NHIỆM VỤ 3](#_Toc183730246)

[CHƯƠNG 3: THIẾT KẾ 4](#_Toc183730247)

[3.1 Thuật toán 4](#_Toc183730248)

[3.2. Thiết kế lớp 6](#_Toc183730249)

[3.3. Thiết kế giao diện. 26](#_Toc183730250)

[CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ 31](#_Toc183730251)

[4.1. Các thư viện cần phải cài đặt 31](#_Toc183730252)

[4.2 Kiểm thử 34](#_Toc183730253)

[CHƯƠNG 5: KẾT LUẬN 40](#_Toc183730254)

[C. TÀI LIỆU THAM KHẢO 40](#_Toc183730255)

# **A. MỞ ĐẦU**

1. **Lý do chọn đề tài.**

Sự phát triển của công nghệ hiện đại và nhu cầu sử dụng ngày càng cao của người dùng. Trợ lý ảo, kết hợp với khả năng điều khiển bằng giọng nói và cử chỉ, không chỉ đáp ứng xu hướng phát triển của trí tuệ nhân tạo và thị giác máy tính mà còn đem lại sự tiện lợi và thân thiện trong tương tác. Đề tài này có tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực như chăm sóc sức khỏe, giáo dục, giải trí và nhà thông minh, giúp cải thiện trải nghiệm của người dùng trong cuộc sống hàng ngày. Ngoài ra, việc nghiên cứu và phát triển trợ lý ảo mang lại cơ hội nâng cao kỹ năng chuyên môn về AI, học máy, xử lý ngôn ngữ tự nhiên và xử lý hình ảnh, đây đều là các lĩnh vực quan trọng và đang có nhu cầu cao trong ngành công nghệ. Hơn nữa, đề tài này cũng mang tính sáng tạo khi cho phép người dùng tương tác với hệ thống một cách tự nhiên và dễ dàng thông qua giọng nói và cử chỉ, thay vì các thao tác truyền thống. Với khả năng mở rộng và nâng cấp trong tương lai, chúng em mong muốn rằng trợ lý ảo này không chỉ đáp ứng tốt nhu cầu hiện tại mà còn tạo ra tiền đề cho những cải tiến mới, mang đến trải nghiệm tiện ích và hiện đại hơn cho người dùng.

## **2. Phương pháp nghiên cứu.**

- Phương pháp nghiên cứu lý thuyết

- Phương pháp nghiên cứu ứng dụng trí tuệ nhân tạo (AI)

- Phương pháp thử nghiệm và mô phỏng

- Nghiên cứu nhận diện cử chỉ tay và thị giác máy tính

# **B. NỘI DUNG**

## **CHƯƠNG 1: ĐẶT TẢ**

Đề tài "Xây dựng trợ lý ảo điều khiển bằng giọng nói và cử chỉ" tập trung vào việc phát triển một phần mềm giúp người dùng tương tác với máy tính thông qua hai phương thức chính: giọng nói và cử chỉ tay. Phần mềm này sẽ đặc biệt hữu ích trong các ngữ cảnh mà người dùng cần điều khiển máy tính một cách linh hoạt. Đặc biệt, nó mang lại lợi ích lớn cho những người khuyết tật, giúp họ có thể điều khiển máy tính dễ dàng mà không cần chuột hay bàn phím. Các lệnh giọng nói sẽ được nhận dạng thông qua micro, trong khi cử chỉ tay sẽ được phân tích từ hình ảnh thu được từ camera. Đầu vào của hệ thống là tín hiệu âm thanh (giọng nói) và hình ảnh (cử chỉ tay). Dựa trên các tín hiệu này, trợ lý ảo có thể thực hiện nhiều tác vụ khác nhau, như phát nhạc, tạm dừng, chuyển bài hát, mở ứng dụng, hoặc điều chỉnh âm lượng.

Trong các tình huống sử dụng điển hình, người dùng có thể ra lệnh bằng giọng nói như “Chơi nhạc” để bắt đầu phát một bài hát hoặc sử dụng cử chỉ tay như vẫy tay để chuyển bài. Nếu người dùng muốn mở nhanh một ứng dụng, họ có thể ra lệnh “Mở Google và tìm kiếm” và trợ lý ảo sẽ thực hiện ngay lập tức. Phần mềm cũng hỗ trợ tìm kiếm trên Google hoặc YouTube bằng cách ra lệnh tìm kiếm giọng nói, như “Tìm kiếm thời tiết hôm nay”. Khi cần điều chỉnh âm lượng, người dùng chỉ cần giơ ngón tay cái lên hoặc ra lệnh bằng giọng nói để tăng hoặc giảm âm lượng.

Giao diện dự kiến của phần mềm sẽ được thiết kế đơn giản và trực quan, tập trung vào việc hiển thị thông tin về các lệnh mà trợ lý ảo đang thực hiện. Một số thông tin như trạng thái của hệ thống (đang nghe lệnh, đang xử lý cử chỉ) hay kết quả của các lệnh (bài hát đang phát, ứng dụng đã mở) sẽ được hiển thị rõ ràng. Ngoài ra, giao diện cũng cung cấp một cửa sổ nhỏ hiển thị video từ camera để người dùng có thể kiểm tra các cử chỉ của mình. Tất cả các yếu tố giao diện được thiết kế để đảm bảo trải nghiệm người dùng mượt mà, tiện lợi và dễ theo dõi, giúp tối ưu hóa sự tương tác với trợ lý ảo.

## **CHƯƠNG 2: PHÂN CÔNG NHIỆM VỤ**

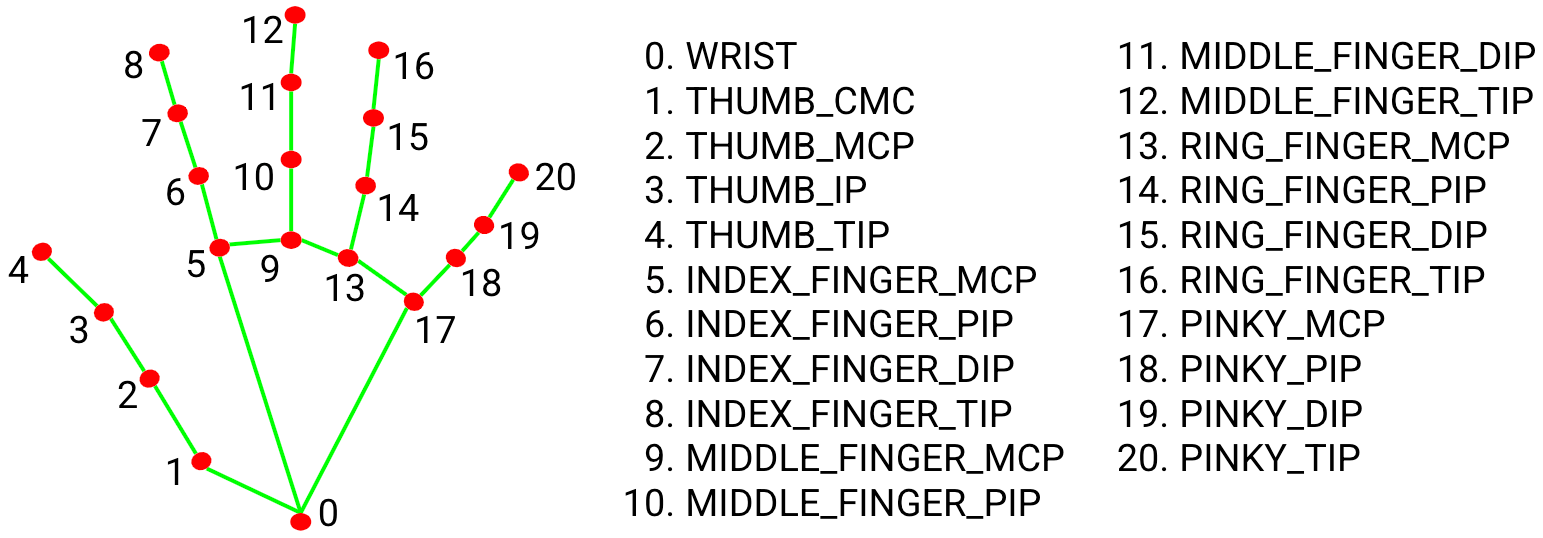
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên SV | Phân công nhiệm vụ | % Đóng góp |
| 01 | Phan Hùng Anh | - Viết hàm nhận diện cử chỉ  - UI  - Biên soạn Word | 50% |
| 02 | Dương Đình Vũ | - Viết hàm nhận diện giọng nói  - UI  - Chỉnh sửa Word | 50% |

## **CHƯƠNG 3: THIẾT KẾ**

### **3.1. Thuật toán**

Thuật toán voice\_assitant của trợ lý ảo sử dụng một chuỗi các bước tuần tự để nhận diện lệnh bằng giọng nói, xử lý nội dung lệnh và thực hiện các hành động tương ứng. Trước tiên, hệ thống khởi tạo một mô-đun nhận diện giọng nói, lắng nghe và chuyển đổi âm thanh từ người dùng thành văn bản. Sau khi có văn bản lệnh, thuật toán tiến hành so sánh chuỗi lệnh này với các từ khóa hoặc câu mẫu đã được định nghĩa trước. Dựa trên sự tương đồng giữa chuỗi lệnh và các từ khóa, trợ lý ảo sẽ xác định được yêu cầu của người dùng, như mở ứng dụng, kiểm tra thời tiết, tìm kiếm trên Internet, hoặc phát nhạc từ YouTube.Thuật toán này sử dụng mô hình điều kiện if-else hoặc chuỗi so khớp từ khóa để xác định lệnh, đồng thời tích hợp các thư viện hoặc API từ bên ngoài để mở rộng chức năng như tìm kiếm thông tin trên Internet hoặc truy cập dữ liệu thời tiết.

Thuật toán nhận diện cử chỉ trong sử dụng một chuỗi các bước để nhận diện và phản hồi các cử chỉ tay. Đầu tiên, nó khởi tạo các thư viện MediaPipe [1] và OpenCV [2] để xử lý hình ảnh từ camera, nhận dạng và theo dõi vị trí của các ngón tay. Trong mỗi khung hình, thuật toán phân tích vị trí của từng đầu ngón tay và tính toán khoảng cách giữa chúng. Cử chỉ tay được nhận dạng dựa trên các ngưỡng khoảng cách giữa ngón cái và các ngón còn lại, từ đó xác định các hành động cần thực hiện. Ví dụ, nếu khoảng cách giữa ngón cái và ngón trỏ nhỏ hơn ngưỡng nhất định, hệ thống hiểu rằng đây là lệnh để thay đổi âm lượng, với khoảng cách ngắn sẽ giảm âm lượng và khoảng cách dài sẽ tăng âm lượng. Khi nhận diện cử chỉ "vuốt", hệ thống xác định tốc độ và hướng của ngón trỏ để di chuyển sang lệnh "Next" hoặc "Previous". Ngoài ra, nếu ngón cái bị gập, hệ thống sẽ dừng chuột và hiển thị thông báo. Khi không có ngón cái bị gập, hệ thống sẽ điều khiển chuột dựa trên vị trí của ngón trỏ. Các thao tác như "click trái" được nhận dạng khi ngón cái và ngón trỏ cách nhau một khoảng cách nhất định và ngón giữa giữ vị trí đặc biệt.



Hình ảnh tọa độ bàn tay của thư viện MediaPipe

Thuật toán nhận diện cảm xúc của người dùng thiết kế để xây dựng một hệ thống nhận diện cảm xúc từ hình ảnh khuôn mặt sử dụng mạng nơ-ron tích chập (CNN) [3]. Dữ liệu được đọc từ tệp fer2013.csv [4] , chứa các ảnh được biểu diễn dưới dạng chuỗi pixel và nhãn cảm xúc tương ứng. Đầu tiên, dữ liệu được tiền xử lý bằng cách chuyển đổi chuỗi pixel thành mảng NumPy, chuẩn hóa giá trị pixel (0-255 về 0-1), và định hình lại thành các ảnh xám kích thước 48x48x1. Nhãn cảm xúc được mã hóa thành dạng one-hot encoding với 7 lớp. Sau đó, dữ liệu được chia thành tập huấn luyện và kiểm tra với tỷ lệ 80-20. Mạng CNN được xây dựng với ba lớp tích chập (Conv2D) để trích xuất đặc trưng, mỗi lớp đều đi kèm với lớp gộp (MaxPooling2D) để giảm kích thước và lớp loại bỏ (Dropout) để giảm hiện tượng overfitting. Mạng kết thúc bằng lớp phẳng (Flatten) và hai lớp dày đặc (Dense), với lớp đầu tiên sử dụng hàm kích hoạt relu và lớp cuối cùng sử dụng softmax để dự đoán xác suất cho 7 nhãn cảm xúc. Mô hình được biên dịch với thuật toán tối ưu hóa adam, hàm mất mát categorical\_crossentropy, và chỉ số đánh giá accuracy. Sau khi huấn luyện trong 25 epoch với batch size 64, mô hình được đánh giá trên tập kiểm tra, và độ chính xác cuối cùng được in ra. Đồ thị biểu diễn quá trình huấn luyện và kiểm tra, gồm độ chính xác và độ mất mát qua các epoch, cũng được vẽ để phân tích hiệu năng. Cuối cùng, mô hình được lưu dưới dạng tệp .h5 để sử dụng sau này.

### **3.2. Thiết kế lớp**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 01 | Util | -Dùng để thực hiện một số hàm tính toán như:  -Dùng để thực hiên tính độ dài khoảng cách của các điểm (x1,y1) và (x2,y2)  -Dùng để thực hiện tính góc dựa vào 3 điểm (a,b,c) | Phan Hùng Anh |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **get\_angle(a, b, c)**  Input: a, b  Output: số đo góc ABC  Pseudo code: | Dùng để thực hiện tính góc dựa vào 3 điểm (a,b,c) | Util.py(3) | Phan Hùng Anh |
| 02 | **get\_distance(landmark\_list)**  Input: danh sách các ngón tay trong danh sách  Output: khoảng cách giữa hai điểm(ngón tay)  Pseudo code: | Dùng để tính toán và trả về khoảng cách giữa hai điểm(ngón tay) trong danh sách **landmark\_list** | Util.py(9) | Phan Hùng Anh |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 02 | **voice\_assistant** | -Thu nhập thông tin âm thanh từ người dùng, dựa vào đó thực hiện các chức năng như: Phát âm, Nhân diện giọng nói, chào hỏi, mở Google và tìm kiếm, đặt báo thức,…. | Dương Đình Vũ, Phan Hùng Anh |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **speak(t, app):**  Input: Text t  Output: không có | **Dùng để chuyển văn bản thành giọng nói.**  **Khi chuyền văn bản t, hàm dùng thư viện gTTS chuyển đổi văn bản thành giọng nói.** | voice\_assistant(34) | Dương Đinh Vũ,  Phan Hùng Anh |
| 02 | **get\_voice(timeout=15):**  Input: Thời gian chờ 15 giây  Output: Văn bản text | Hàm có chức năng nhận diện và chuyển đổi giọng nói của người dùng thành văn bản.  Sử dụng micro làm nguồn thu âm,sau khi ghi nhận âm thanh, hàm sử dụng Google Speech Recognition để nhận dạng giọng nói và chuyển đổi âm thanh thành văn bản | voice\_assistant(57) | Dương Đình Vũ |
| 03 | **stop(app):**  Input: Không có  Output: Không có | Hàm có chức năng chào tạm biệt người dùng bằng cách gọi hàm speak() để phát ra "Hẹn gặp lại bạn nhé!". | voice\_assistant(86) | Dương Đình Vũ |
| 04 | **get\_text(app):**  Input: Không có  Output: Không có | Hàm có chức năng thu thập dữ liệu âm thanh từ giọng nói người dùng và chuyển đổi thành văn bản. Nó thử thu giọng nói tối đa 3 lần, và nếu không nhận được dữ liệu, sẽ yêu cầu người dùng lặp lại. Nếu sau 3 lần không thành công, hàm sẽ chờ 10 giây rồi gọi hàm stop() để chào tạm biệt | voice\_assistant(113) | Dương Đình Vũ |
| 05 | **talk(name,app):**  Input: Tên người dùng  Output: Không có | Hàm có chức năng tương tác với người dùng bằng cách chào họ tùy thuộc vào thời gian trong ngày. Nếu trước 12 giờ, hàm sẽ chúc "Chào buổi sáng" và nếu sau 18 giờ, sẽ chúc "Chào buổi tối". Sau khi chào, hàm hỏi "Bạn có khỏe không?" và chờ phản hồi từ người dùng. Nếu người dùng trả lời "có", hàm sẽ đáp lại "Thật là tốt!", còn nếu không, hàm sẽ khuyên người dùng nên nghỉ ngơi. | voice\_assistant(129) | Dương Đình Vũ |
| 06 | **open\_website(text,app):**  Input: Văn bản text  Output: True/False | Hàm có chức năng mở một trang web dựa trên yêu cầu của người dùng. Nó tìm cụm từ "mở" trong văn bản đầu vào, trích xuất tên miền, và tạo URL bằng cách thêm"<https://www>." vào trước tên miền đó. để mở trang web trong trình duyệt. Nếu thành công, nó thông báo rằng trang web đã được mở; nếu không, trả về False. | voice\_assistant(203) | Dương Đình Vũ |
| 07 | **google\_search(text,app):**  Input: văn bản text  Output: không có  Pseudo code | Hàm có chức năng thực hiện tìm kiếm trên Google dựa vào đầu vào từ người dùng. Nó trích xuất cụm từ tìm kiếm từ văn bản, sau đó khởi tạo WebDriver của Chrome để mở trang Google, nhập từ khóa tìm kiếm vào ô tìm kiếm và nhấn Enter. Trình duyệt sẽ mở và giữ lại cho đến khi người dùng nói "ok" để đóng trình duyệt. Trong quá trình này, hàm còn thông báo cho người dùng biết kết quả tìm kiếm đã hoàn thành. | voice\_assistant(213) | Dương Đình Vũ |
| 08 | **play\_youtube(app)**  Input: không có  Output: không có  Pseudo code | Hàm có chức năng yêu cầu người dùng chọn một bài hát và sau đó sử dụng thư viện YouTube Search để tìm kiếm bài hát đó trên YouTube. Khi tìm thấy kết quả, hàm sẽ mở liên kết bài hát đầu tiên trong trình duyệt và thông báo cho người dùng rằng bài hát đã được mở để thưởng thức. | voice\_assistant(225) | Dương Đình Vũ |
| 09 | **weather(app):**  Input: không có  Output: không có  Pseudo code | Hàm yêu cầu người dùng cung cấp tên thành phố mà họ muốn xem thời tiết. Sau đó, nó sử dụng API thời tiết từ OpenWeatherMap[5] để lấy thông tin thời tiết cho thành phố đó, bao gồm nhiệt độ, áp suất không khí, độ ẩm, thời gian mặt trời mọc và lặn, cùng với mô tả thời tiết. Nếu thông tin thời tiết được lấy thành công, hàm sẽ phát âm thanh thông báo chi tiết thời tiết hiện tại cho người dùng. Nếu không tìm thấy thành phố, hàm sẽ thông báo rằng không tìm thấy thành phố đó. | voice\_assistant(246) | Dương Đình Vũ |
| 10 | **capture\_screen():**  Input: Không có  Output: Không có | Hàm có chức năng chụp ảnh màn hình, lưu vào thư mục screenshots với tên file bao gồm thời gian chụp để tránh trùng lặp. Nếu thư mục chưa tồn tại, hàm sẽ tự động tạo. Sau khi lưu ảnh thành công, hàm thông báo đường dẫn file và xác nhận với người dùng bằng giọng nói. | voice\_assistant(412) | Phan Hùng Anh |
| 11 | **shutdown\_computer():**  Input: Không có  Output: Không có | Hàm có tác dụng tắt máy nếu người dùng gọi | voice\_assistant(354) | Dương Đình Vũ |
| 12 | **open\_app\_by\_voice(app)**  Input: Không có  Output: Không có | Hàm open\_app\_by\_voice() có mục đích mở các ứng dụng trên máy tính theo lệnh giọng nói của người dùng. Đầu tiên, hàm yêu cầu người dùng nói tên ứng dụng muốn mở. Sau khi nhận được đầu vào giọng nói, hàm kiểm tra xem người dùng có yêu cầu mở các ứng dụng phổ biến như Google Chrome, Microsoft Word, Microsoft Excel hay Notepad không. Nếu tìm thấy ứng dụng tương ứng, hàm sẽ mở ứng dụng đó và thông báo cho người dùng biết. Nếu không tìm thấy ứng dụng yêu cầu, hàm sẽ thông báo lỗi và yêu cầu người dùng thử lại. | voice\_assistant(282) | Dương Đình Vũ |
| 13 | **validate\_alarm\_time(alarm\_time)**  Input: Một chuỗi alarm\_time (dạng chuỗi ký tự), đại diện cho thời gian báo thức. Output:  Trả về một tuple với 2 giá trị:  **True và đối tượng thời gian** nếu alarm\_time hợp lệ (đúng định dạng HH:MM).  **False và None** nếu alarm\_time không hợp lệ (không đúng định dạng hoặc không thể phân tích được). | Hàm validate\_alarm\_time() có mục đích kiểm tra và phân tích thời gian báo thức đầu vào. Hàm nhận vào một chuỗi thời gian (alarm\_time) và xác minh xem nó có đúng định dạng HH:MM hay không. Nếu thời gian hợp lệ, hàm sẽ chuyển đổi chuỗi này thành một đối tượng thời gian (datetime.time) và trả về giá trị True cùng với thời gian đã được phân tích. Nếu định dạng không đúng hoặc không hợp lệ, hàm sẽ trả về False và None để thông báo lỗi. | voice\_assistant(300) | Dương  Đình Vũ |
| 14 | **set\_alarm(alarm\_time,app):**  Input: alarm\_time một chuỗi đại diện cho thời gian báo thức (theo định dạng HH:MM).  Output: Không có | Hàm set\_alarm được thiết kế để cài đặt và quản lý báo thức cho người dùng. Khi nhận được thời gian báo thức từ người dùng, hàm sẽ kiểm tra tính hợp lệ của thời gian đầu vào, nếu thời gian hợp lệ, nó sẽ thông báo cho người dùng biết báo thức đã được cài đặt và bắt đầu đếm giờ. Mỗi 30 giây, hàm sẽ kiểm tra thời gian hiện tại và so sánh với thời gian báo thức đã cài. Khi đến giờ báo thức, âm thanh báo thức sẽ được phát và người dùng sẽ nhận được thông báo. Đồng thời, người dùng có thể tắt báo thức bằng cách nhấn phím Enter, khi đó âm thanh báo thức sẽ dừng và hàm sẽ thông báo báo thức đã tắt. Nếu thời gian báo thức không hợp lệ, hàm sẽ yêu cầu người dùng nhập lại thời gian đúng định dạng. | voice\_assistant(315) | Dương Đình Vũ |
| 15 | **sleep():**  Input: Không có  Output: Không có | Hàm sleep có mục đích chuyển trợ lý ảo vào trạng thái ngủ (chế độ chờ) và thông báo cho người dùng rằng trợ lý đã đi ngủ, đồng thời cho phép người dùng đánh thức trợ lý bằng cách chạm bàn phím. | voice\_assistant(91) | Dương Đình Vũ |
| 16 | **wake\_up():**  Input: Không có  Output:  Nếu trợ lý đang ở trạng thái ngủ (is\_sleeping == True), hàm sẽ:  Cập nhật giá trị của biến toàn cục is\_sleeping thành False, đánh dấu trợ lý đã thức dậy.  Phát ra thông báo giọng nói: "tui đã thức dậy rùi!" | Hàm wake\_up có mục đích đánh thức trợ lý ảo khi người dùng tương tác và thay đổi trạng thái từ "ngủ" sang "thức". Khi trợ lý được đánh thức, hàm sẽ thông báo bằng giọng nói rằng nó đã thức dậy. | voice\_assistant(96) | Dương Đình Vũ |
| 17 | **call\_sen(app):**  Input: Không có  Output: Không có | Hàm call\_sen là trung tâm của chức năng trợ lý ảo, giúp giao tiếp và thực hiện các lệnh của người dùng thông qua giọng nói. Khi bắt đầu, trợ lý sẽ chào hỏi người dùng và yêu cầu họ nhập tên. Sau khi nhận được tên, trợ lý sẽ tiếp tục lắng nghe các lệnh và thực hiện các yêu cầu như dừng hoạt động, trò chuyện, mở ứng dụng, tìm kiếm trên Google, chơi nhạc, cung cấp thông tin thời tiết, chụp màn hình hoặc đặt báo thức. Nếu người dùng yêu cầu đặt báo thức, trợ lý sẽ yêu cầu người dùng nhập thời gian và xác nhận tính hợp lệ của thời gian đó. Nếu thời gian hợp lệ, báo thức sẽ được cài đặt. Trong khi trợ lý hoạt động, nó sẽ liên tục kiểm tra và xử lý các lệnh của người dùng, đồng thời lắng nghe hoạt động từ bàn phím và chuột để có thể thức dậy nếu đang ở chế độ ngủ. | voice\_assistant(359) | Dương Đình Vũ,  Phan Hùng Anh |
| 18 | **on\_keyboard\_activity(key, app):**  Input: phím bấm từ bàn phím  Output: Không có | Nếu người dùng nhấn bất kỳ phím nào thì hàm sẽ gọi hàm wakeup(app): | voice\_assistant(102) | Dương Đình Vũ |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 03 | **gesture\_control** | - Chức năng thu thập thông tin hình ảnh từ camera rồi từ đó nhận diện cử chỉ tay  - Dựa vào cử chỉ của các ngón tay, thực hiện các như di chuyển chuột theo bàn tay, điều chỉnh âm lượng, click chuột, chuyển Tab,…. | Phan Hùng Anh |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **process\_frame(frame):**  Input: Khung hình video gốc.  Output: Khung hình video đã được lật, vẽ các điểm đặc trưng của bàn tay, và có thể đã phát hiện cử chỉ.  Pseudo code: | Hàm có chức năng xử lý khung hình video để nhận diện bàn tay và cử chỉ bằng MediaPipe. Nó lật khung hình, chuyển đổi màu từ BGR sang RGB, sau đó phát hiện các điểm đặc trưng của bàn tay. Nếu phát hiện bàn tay, các điểm này được vẽ lên khung hình và lưu vào danh sách tọa độ. Cuối cùng, hàm gọi detect\_gesture để nhận diện cử chỉ và trả về khung hình đã xử lý. | gesture\_control(165) | Phan Hùng Anh |
| 02 | **find\_finger\_tip(processed):**  Input: kết quả đầu ra từ quá trình xử lý khung hình thông qua chứa thông tin về các điểm đặc trưng của bàn tay.  Output:  Nếu có bàn tay: Trả về tọa độ của đầu ngón trỏ (index\_finger\_tip) và đầu ngón cái (thumb\_tip) đầu ngón giữa (middle finger), dưới dạng các đối tượng với thuộc tính x, y, z.  Nếu không có bàn tay: Trả về None, None.  Pseudo code: | **Hàm có chức năng tìm kiếm và trả về vị trí của đầu ngón trỏ và đầu ngón cái từ kết quả xử lý bàn tay trong khung hình. Đầu tiên, hàm kiểm tra xem kết quả xử lý có phát hiện bàn tay nào hay không. Nếu có, hàm sẽ lấy danh sách các điểm đặc trưng của bàn tay đầu tiên được phát hiện. Sau đó, nó trích xuất tọa độ của đầu ngón trỏ (INDEX\_FINGER\_TIP) và đầu ngón cái (THUMB\_TIP),**  **đầu ngón giữa(MIDDLE\_FINGER) từ danh sách này. Nếu tìm thấy cả hai đầu ngón tay, hàm trả về tọa độ của chúng, bao gồm các giá trị x, y, và z. Trong trường hợp không phát hiện bàn tay, hàm sẽ trả về giá trị None cho cả hai ngón.** | gesture\_control(143) | Phan Hùng Anh |
| 03 | **ngontaygap(landmark\_list):**  Input: Danh sách các tọa độ điểm đặc trưng (landmarks) của bàn tay (landmark\_list) Output:  **True**: Nếu ngón cái được xác định là gập vào lòng bàn tay (khoảng cách giữa đầu ngón cái và gốc bàn tay nhỏ hơn 150).  **False**: Nếu ngón cái không gập (khoảng cách lớn hơn 150). | Hàm có chức năng kiểm tra xem ngón cái có đang gập vào lòng bàn tay hay không, dựa trên khoảng cách giữa đầu ngón cái và gốc bàn tay. Đầu tiên, hàm lấy tọa độ của đầu ngón cái (INDEX 4) và gốc bàn tay (INDEX 0) từ danh sách các điểm đặc trưng của bàn tay. Sau đó, nó sử dụng hàm util.get\_distance để tính khoảng cách giữa hai điểm này. Nếu khoảng cách nhỏ hơn 150 đơn vị (giả sử là pixel), hàm kết luận rằng ngón cái đang gập vào lòng bàn tay và trả về True. Ngược lại, nếu khoảng cách lớn hơn, ngón cái được coi là không gập và hàm trả về False. | gesture\_control(152) | Phan Hùng Anh |
| 04 | **move\_mouse(index\_finger\_tip, thumb\_folded)**  Input: Vị trí đầu ngón trỏ, trạng thái tay có gập vào không  Output: Không có | Hàm có chức năng di chuyển con trỏ chuột đến vị trí của đầu ngón trỏ, với điều kiện ngón cái không gập. Nó kiểm tra tính hợp lệ của vị trí ngón trỏ và tính toán tọa độ chuột dựa trên kích thước màn hình. Nếu điều kiện thỏa mãn, hàm sử dụng pyautogui.moveTo để di chuyển con trỏ đến tọa độ đã tính. | gesture\_control(158) | Phan Hùng Anh |
| 05 | **is\_left\_click(landmark\_list, thumb\_index\_dist)**  Input:  landmark\_list: Danh sách các tọa độ của các điểm đặc trưng  ngón tay trên bàn tay  thumb\_index\_dist: Khoảng cách giữa ngón cái và ngón trỏ  Output: True/False | Hàm có chức năng xác định xem người dùng có thực hiện thao tác nhấp chuột trái hay không bằng cách kiểm tra các góc giữa các khớp ngón tay và khoảng cách giữa ngón cái và ngón trỏ. Nếu góc giữa ngón trỏ và ngón giữa nhỏ hơn 50 độ, góc giữa ngón nhẫn và ngón giữa lớn hơn 90 độ, và khoảng cách giữa ngón cái và ngón trỏ lớn hơn 50 đơn vị, hàm sẽ trả về True, cho thấy người dùng đang nhấp chuột trái. Nếu không, hàm trả về False. | gesture\_control(164) | Phan Hùng Anh |
| 06 | **dieu\_chinh\_am\_luong(thumb\_tip, index\_finger\_tip)**  Input:  thumb\_tip: Tọa độ của đầu ngón cái  index\_finger\_tip: Tọa độ của đầu ngón trỏ  Ouput: không có | Hàm có chức năng điều chỉnh âm lượng hệ thống dựa trên khoảng cách giữa đầu ngón cái và đầu ngón trỏ. Khi có tọa độ của cả hai ngón tay, hàm tính toán khoảng cách và truy cập âm lượng hiện tại. Nếu khoảng cách lớn hơn 200 đơn vị, âm lượng sẽ tăng thêm 0.1; nếu nhỏ hơn 100 đơn vị, âm lượng sẽ giảm đi 0.1. Hàm cũng in ra âm lượng mới sau mỗi lần điều chỉnh để người dùng dễ theo dõi. | gesture\_control(164) | Phan Hùng Anh |
| 07 | **nhan\_dien\_vuot(index\_finger\_tip, hand\_label)** Input:  index\_finger\_tip: Tọa độ của đầu ngón trỏ  hand\_label: Tay đang sử dụng ("Left" hoặc "Right").  Output:  "Next": Nếu cử chỉ vuốt sang phải được phát hiện.  "Previous": Nếu cử chỉ vuốt sang trái được phát hiện.  None: Nếu không có cử chỉ vuốt nào được phát hiện. | Hàm có chức năng phát hiện cử chỉ vuốt từ ngón trỏ để xác định thao tác "tiếp theo" hoặc "trước đó". Khi được gọi, hàm lưu tọa độ x và thời gian hiện tại, sau đó tính toán thời gian chênh lệch để xác định tốc độ di chuyển của ngón tay. Nếu tốc độ vượt ngưỡng (1.5), hàm sẽ trả về "Next" cho vuốt sang phải hoặc "Previous" cho vuốt sang trái, tùy thuộc vào tay đang sử dụng. Nếu không có cử chỉ vuốt nào, hàm sẽ lưu vị trí và thời gian để sử dụng cho lần gọi tiếp theo. | gesture\_control(62) | Phan Hùng Anh |
| 08 | **dieu\_chinh\_man\_hinh(thumb\_tip, middle\_finger\_tip):**  Input:  thumb\_tip: Tọa độ ngón cái cái  middle\_tip: Tọa độ ngón giữa  Output: Không có | Hàm có chức năng điều màn hình dựa trên khoảng cách giữa đầu ngón cái và đầu ngón giữa. Khi có tọa độ của cả hai ngón tay, hàm tính toán khoảng cách và điều chỉnh kích thước màn hình | gesture\_control(39) | Phan Hùng Anh |
| 09 | **nhan\_dien\_cu\_chi(frame, landmark\_list, processed):**  **Input**  +frame: Khung hình đầu vào từ camera.  +landmark\_list: Danh sách các điểm đặc trưng của bàn tay (các khớp ngón tay).  +processed: Kết quả xử lý từ MediaPipe chứa thông tin nhận diện bàn tay (gồm loại tay, vị trí bàn tay). Output: Không có | Hàm nhan\_dien\_cu\_chi xử lý khung hình (frame) và danh sách tọa độ bàn tay (landmark\_list) để nhận diện các cử chỉ tay (gập ngón, điều chỉnh âm lượng, di chuyển chuột, click chuột, và vuốt chuyển tab). | gesture\_control(171) | Phan Hùng Anh |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 04 | **nhandiencamxuc** | Lớp này nhận diện cảm xúc từ khuôn mặt trong ảnh bằng cách sử dụng mô hình đã huấn luyện | Dương Đình Vũ, Phan Hùng Anh |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | Input: Một frame ảnh đầu vào.  Output:  + Một chuỗi đại diện cho cảm xúc được dự đoán của khuôn mặt (ví dụ: "Happy", "Sad", v.v.) nếu phát hiện khuôn mặt.  + Trả về None nếu không có khuôn mặt nào được phát hiện trong ảnh. | Hàm này có nhiệm vụ nhận diện khuôn mặt trong ảnh, sau đó dự đoán cảm xúc của khuôn mặt đó bằng mô hình đã huấn luyện. Kết quả trả về là tên cảm xúc tương ứng, hỗ trợ trong các ứng dụng phân tích hoặc nhận diện cảm xúc. | nhandiencamxuc(15) | Dương Đình Vũ,  Phan Hùng Anh |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 05 | **ChatBotUI** | + Tạo giao diện chat giữa người dùng và trợ lý ảo.  + Hiển thị các tin nhắn dưới dạng bong bóng (chat bubble) từ bot hoặc người dùng.  Tự động đọc, cập nhật tin nhắn từ file chat.txt. | Phan Hùng Anh |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **calculate\_text\_height(self, text, width):**  Input:  text: nội dung văn bản cần được hiển thị.  width: Chiều rộng giới hạn của vùng hiển thị văn bản, tính theo pixel.  Output:  Chiều cao cần thiết (tính bằng pixel) để hiển thị đầy đủ toàn bộ văn bản, bao gồm cả khoảng cách giữa các dòng. | Hàm được thiết kế để tính toán chiều cao (height) cần thiết để hiển thị một đoạn văn bản (text) trong một không gian với độ rộng (width) nhất định. Điều này đặc biệt hữu ích trong việc xây dựng giao diện người dùng, nơi các đoạn văn bản phải được hiển thị đầy đủ trong một vùng cố định. | UI(55) | Phan Hùng Anh |
| 02 | **create\_chat\_bubble(self, container, text, is\_bot)**  Input:  +container:nơi chat sẽ được chèn vào.  +text: Nội dung văn bản cần hiển thị.  +is\_bot (bool):  True: Tin nhắn do bot gửi (màu sắc sẽ khác).  False: Tin nhắn do người dùng gửi.  Output:  Một widget dạng frame chứa văn bản. | Hàm được thiết kế để tạo một bong bóng chat (chat bubble) hiển thị nội dung tin nhắn trong giao diện người dùng. Bong bóng chat này có thể được tùy chỉnh về kích thước, màu sắc dựa trên vai trò của người gửi (bot hoặc người dùng). | UI(66) | Phan Hùng Anh |
| 03 | **create\_message\_bubble(self, text, is\_bot=True):**  Input  text:Nội dung tin nhắn cần hiển thị.  is\_bot: Xác định tin nhắn là của bot hay người dùng:  True: Tin nhắn của bot.  False: Tin nhắn của người dùng.  Output  Hàm không trả về giá trị (None) vàthêm một tin nhắn (dạng khung) vào giao diện trò chuyện. | Hàm có chức năng tạo và hiển thị một tin nhắn trên giao diện chat, định dạng phù hợp cho bot hoặc người dùng (kèm biểu tượng nếu có), lưu tin nhắn vào danh sách quản lý, và tự động cuộn xuống cuối giao diện. | UI(90) | Phan Hùng Anh |
| 04 | **clear\_chat(self):**  Input: Không có  Output: Không có | Hàm dùng để xóa toàn bộ tin nhắn khỏi giao diện chat và đặt lại danh sách lưu trữ tin nhắn về trạng thái rỗng. | UI(129) | Phan Hùng Anh |
| 05 | **load\_messages(self):**  Input: Không có  Output: Không có | Hàm có chức năng đọc tin nhắn từ file, hiển thị tin nhắn mới lên giao diện và thông báo lỗi nếu file không tồn tại. | UI(136) | Phan Hùng Anh |
| 06 | **check\_for\_updates(self):**  Input: Không có  Output: Không có | Hàm có tính năng tự cập nhật nội dung từ chat.txt vào hộp thoại text box | UI(159) | Phan Hùng Anh |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 05 | **AssistantApp** | Tạo giao diện trợ lý ảo với GIF động. Nó tải và hiển thị GIF, điều khiển việc phát và dừng GIF, và cập nhật GIF liên tục khi được yêu cầu. | Dương Đình Vũ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **load\_gif\_frames(self, gif):**  **Input**: Một đối tượng gif (file GIF).  **Output**: Danh sách các frame của GIF dưới dạng PhotoImage | Hàm được thiết kế để tính toán chiều cao (height) cần thiết để hiển thị một đoạn văn bản (text) trong một không gian với độ rộng (width) nhất định. Điều này đặc biệt hữu ích trong việc xây dựng giao diện người dùng, nơi các đoạn văn bản phải được hiển thị đầy đủ trong một vùng cố định. | UI(195) | Dương Đình Vũ |
| 02 | **animate(self):**  Input: Không có  Output: Không có | Điều khiển quá trình hoạt hình của GIF, thay đổi frame mỗi 100ms để tạo hiệu ứng hoạt hình. | UI(207) | Dương Đình Vũ |
| 03 | **start\_gif(self)**  Input: Không có  Output: Không có | Bắt đầu chạy gif | UI(215) | Dương Đình Vũ |
| 04 | **stop\_gif(self):**  Input: Không có  Output: Không có | Dừng gif | UI(218) | Dương Đình Vũ |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 05 | **CameraUI** | Tạo giao diện trợ lý ảo với GIF động. Nó tải và hiển thị GIF, điều khiển việc phát và dừng GIF, và cập nhật GIF liên tục khi được yêu cầu. | Phan Hùng Anh |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **update\_video(self):**  Input: Không có  Output: Không có | Hàm cập nhật hình ảnh video từ camera và hiển thị liên tục lên giao diện người dùng | UI(243) | Phan Hùng Anh |
| 02 | **start\_gesture\_recognition(self)**  Input: Không có  Output: Không có | Chạy một luồng riêng mở camera | UI(257) | Phan Hùng Anh |

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên lớp | Mục đích | Tên các SV phụ trách |
| 06 | **MainUI** | Chạy giao diện của ba UI: ChatBotUI, AssitantApp, CameraUI | Phan Hùng Anh |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách |
| 01 | **run(self):**  Input: Không có  Output: Không có | Chạy giao diện của ba UI: ChatBotUI, AssitantApp, CameraUI | UI(283) | Phan Hùng Anh |
| 02 | **on\_close(self):**  Input: Không có  Output: Không có | Khi giao diện cửa số chính tắt, thì sẽ xóa hết nội dụng văn bản chat.txt | UI(286) | Phan Hùng  Anh |

### **3.3. Thiết kế giao diện.**

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Màn hình/Cửa sổ/Dialog | Mục đích | **Giải thích ngắn gọn các quyết định thiết kế khi thiết kế màn hình** |
| 1 | Cửa sổ gif trợ lý ảo    Giao diện trợ lý ảo khi đang nói | Thêm tính năng để Chatbox chuyển động khi trợ lý ảo đang nói chuyện với người dùng và dừng lại khi không nói. | Phan Hùng Anh, Dương Đình Vũ  Chatbox sẽ chuyển động khi đang nói chuyện với người dùng. Còn khi không nói thì chat sẽ dừng ­ |
| 2 | Cửa sổ giao diện camera    Giao diện camera  ­  Khi nhận diện được cử chỉ click chuột    Trợ lý ảo nhận diện cảm xúc người dùng    Tăng giảm âm lượng | Thêm tính năng nhận diện cử chỉ người dùng từ đó dựa vào cử chỉ thực hiện các hành đông như di chuyển chuột, chuyển tab, click chuột,tăng giảm âm lượng, phóng to, thu nhỏ màn hình, nhận diện cảm xúc | Phan Hùng Anh, Dương Đình Vũ  Người dùng có thể đưa lệnh cho chatbox bằng cử chỉ bàn tay như:  - Di chuyển chuột bằng cách di chuyển bàn tay  - Chuyển tab qua lại bằng vuốt qua tay phải qua bên trái hoặc bên phải  - Click chuột bằng cách gập đông thời ngón cái và ngón tay trỏ của bàn tay phải lại  - Tăng giảm âm lượng bằng cách thu hẹp phóng to khoảng cách ngón cái và ngón trỏ bàn tay trái  - Tăng giảm, phóng to màn hình bằng cách thu hẹp phóng to khoảng cách ngón cái và ngón giữa bàn tay trái  - Nhận diện cảm xúc |
| 3 | Cửa sổ giao diện hộp thoại chat    Cửa sổ giao diện hộp thoại chat    Trợ lý ảo trò chuyện với người dùng      Trợ lý ảo thông báo thời tiết    Trợ lý ảo mở Google và tìm kiếm    Kết quả tìm kiếm của trợ lý ảo    Trợ lý ảo chạy bài hát người dùng muốn    Bài hát người dùng muốn chơi    Người dùng đặt báo thức cho trợ lý ảo  ­ | Thêm tính năng lắng nghe người dùng rồi từ đó thực hiện các hành động như trò chuyện, xem thời tiết, mở google và tìm kiếm, … | Phan Hùng Anh, Dương Đình Vũ  Chat bot lắng nghe giọng nói của người dùng xong thực hiện các câu lệnh điều khiển theo quy định như  - Mở bài hát  - Xem thời tiết  - Trò chuyện với người dùng  - Mở google và tìm kiếm  - Đặt báo thức  - Mở ứng dụng  - Chụp màn hình,.. |

## **CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ**

### **4.1. Các thư viện cần phải cài đặt**

* **speech\_recognition**

Thư viện speech\_recognition giúp nhận diện giọng nói và chuyển đổi âm thanh thành văn bản trong Python. Nó hỗ trợ ghi âm từ micro, file âm thanh và sử dụng các API như Google, Sphinx để nhận diện nhiều ngôn ngữ[6]

Cách cài đặt: pip install SpeechRecognition

* **playsound**

Thư viện playsound trong Python cho phép phát âm thanh từ file một cách đơn giản. Nó hỗ trợ các định dạng âm thanh phổ biến như MP3, WAV, và có cú pháp sử dụng rất dễ dàng, chỉ cần chỉ định đường dẫn của file âm thanh. Thư viện này hoạt động trên cả Windows, macOS và Linux mà không yêu cầu thư viện bên ngoài.[7]

Cách cài đặt: pip install playsound

* **gtts**

Thư viện gTTS (Google Text-to-Speech) trong Python cho phép chuyển đổi văn bản thành giọng nói bằng cách sử dụng API của Google. Người dùng có thể tạo file âm thanh (MP3) từ văn bản với nhiều ngôn ngữ và giọng đọc khác nhau. Thư viện này rất dễ sử dụng và tiện lợi cho các ứng dụng như tạo trợ lý ảo, đọc văn bản tự động, hoặc các ứng dụng hỗ trợ tiếp cận.[8]

Cách cài đặt: pip install gtts

* **wikipedia**

Thư viện wikipedia trong Python cung cấp một giao diện để truy xuất và lấy thông tin từ Wikipedia. Nó cho phép người dùng tìm kiếm bài viết, lấy tóm tắt nội dung, tiêu đề, và các thông tin chi tiết khác từ Wikipedia một cách dễ dàng. Thư viện hỗ trợ nhiều ngôn ngữ và có thể xử lý các tìm kiếm không chính xác bằng cách đưa ra các đề xuất gần đúng.[9]

Cách cài đặt: pip install wikipedia

* **pygame**

Thư viện pygame trong Python là một công cụ mạnh mẽ để phát triển các trò chơi và ứng dụng đa phương tiện. Nó cung cấp các chức năng để quản lý đồ họa, âm thanh, và đầu vào từ bàn phím, chuột hoặc joystick. pygame được xây dựng trên thư viện SDL, giúp dễ dàng tạo ra các trò chơi 2D, xử lý sự kiện, phát âm thanh và vẽ hình ảnh một cách mượt mà trên nhiều nền tảng.[10]

Cách cài đặt: pip install pygame

* **youtube-search**

Thư viện youtube-search trong Python cho phép người dùng thực hiện tìm kiếm video trên YouTube một cách dễ dàng. Nó cung cấp các chức năng để lấy danh sách video dựa trên từ khóa tìm kiếm, bao gồm thông tin như tiêu đề, mô tả, URL và độ dài video. Thư viện này rất hữu ích cho các ứng dụng liên quan đến việc tìm kiếm và thu thập dữ liệu từ YouTube mà không cần phải tương tác trực tiếp với API của YouTube.[11]

Cách cài đặt: pip install webdriver-manager

* **open cv**

Thư viện OpenCV (Open Source Computer Vision Library) trong Python là một công cụ mạnh mẽ dành cho xử lý hình ảnh và thị giác máy tính. Nó cung cấp hàng ngàn hàm để thực hiện các tác vụ như nhận diện khuôn mặt, theo dõi đối tượng, phân đoạn ảnh, và xử lý video.OpenCV hỗ trợ nhiều định dạng hình ảnh và video, cho phép người dùng dễ dàng thao tác và phân tích dữ liệu hình ảnh trong thời gian thực.

Cách tải đặt: pip install opencv-python

* **mediapipe**

Thư viện MediaPipe trong Python là một framework mạnh mẽ dành cho xử lý video và hình ảnh, đặc biệt trong lĩnh vực thị giác máy tính. Nó cung cấp các mô hình học sâu được tối ưu hóa cho các tác vụ như nhận diện khuôn mặt, phát hiện cử chỉ tay, theo dõi cơ thể, và phân đoạn ảnh. MediaPipe cho phép người dùng dễ dàng tích hợp các tính năng này vào ứng dụng của mình với hiệu suất cao và thời gian xử lý nhanh.

Cách cài đặt: pip install mediapipe

* **pyautogui**

Thư viện pyautogui trong Python cho phép tự động hóa các tác vụ điều khiển chuột và bàn phím. Người dùng có thể sử dụng thư viện này để thực hiện các hành động như di chuyển chuột, nhấp chuột, gõ phím, chụp màn hình, và nhận diện các vị trí trên màn hình. pyautogui rất hữu ích cho việc tự động hóa quy trình, kiểm thử giao diện người dùng, và tạo ra các ứng dụng tương tác mà không cần can thiệp trực tiếp từ người dùng. Thư viện này hỗ trợ cả Windows, macOS và Linux.[11]

Cách cài đặt: pip install pyautogui

* **pynput**

Thư viện pynput là một thư viện Python được thiết kế để điều khiển và giám sát bàn phím và chuột. Với pynput, bạn có thể thực hiện các thao tác như giả lập phím bấm, di chuyển con trỏ chuột, nhấp chuột, hoặc ghi nhận các sự kiện từ người dùng.[12]

Cách cài đặt: pip install pynput

* **keras, tensorflow**

Keras là một API học sâu cấp cao, viết bằng Python, chạy trên nền tảng TensorFlow. Nó giúp đơn giản hóa quá trình xây dựng và huấn luyện các mô hình học sâu với các thành phần dễ sử dụng, mô-đun và có thể mở rộng.[13]

TensorFlow, được phát triển bởi Google, là một framework mã nguồn mở cho học máy, hỗ trợ xây dựng các mô hình học sâu và các thuật toán học máy khác. TensorFlow cung cấp các công cụ cho cả nghiên cứu và triển khai mô hình trong môi trường sản xuất.[14]

Cách cài đặt: pip install tensorflow

* **CustomTkinter**

**Thư viện CustomTkinter** là một framework mở rộng Tkinter, cung cấp giao diện hiện đại với thiết kế phẳng, hỗ trợ chủ đề sáng/tối và các widget tùy chỉnh như nút, thanh trượt, hộp nhập liệu.[15]

pip install customtkinter

* **PIL**

**Thư viện PIL (Python Imaging Library):** hỗ trợ xử lý hình ảnh trong Python, như mở, chỉnh sửa, và lưu ảnh ở nhiều định dạng khác nhau.[16]

pip install pillow

### **4.2 Kiểm thử**

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tình huống | Mục đích | Giải thích ngắn gọn quyết định lựa chọn tình huống và dữ liệu đầu vào |
| 1 | Khi không nói gì với trợ lý ảo    Khi không nói gì với trợ lý ảo | Kiểm thử phần mềm có bị lỗi khi không nói với trợ lý ảo | Trợ lý ảo sẽ thông báo “Chat box nghe không nghe rõ, bạn nói lại đi?” |
| 2 | Chọn một thành phố không có thật    Khi chọn thành phố không có thật | Kiểm tra ứng dụng có lỗi khi chọn thành phố sai | Khi không tìm thấy thành phố thì trợ lý ảo sẽ báo “Không tìm thấy thành phố” |
| 3 | Chọn thời gian đặt báo thức không đúng cú pháp HH:MM    Khi đặt báo thức sai cú pháp | Kiểm tra ứng dụng có lỗi khi đặt sai thời gian báo thức | Khi người dùng đặt sai báo thức thì trợ lý ảo sẽ thông báo “Thời gian không hợp lệ. Vui lòng nhập lại định dạng HH:MM” |
| 04 | Di chuyển chuột bằng 2 tay | Kiểm tra trợ lý ảo có nhầm lẫn cử chỉ khi dùng cả hai tay cùng lúc | Trợ lý ảo chỉ có thể nhận diện cử chỉ người dùng chỉ với một bàn tay |
| 05 | Phóng to, thu nhỏ màn hình, âm lượng bằng tay phải và ngược lại có thể­­­­­­ di chuột, click chuột bằng tay trái      Thử tăng, giảm âm lượng bằng tay phải  ­­­­  ­­­­  Click chuột bằng tay trái | Kiểm tra trợ lý ảo có bị nhầm lẫn giữa hai tay không | Trợ lý ảo không thể thực hiện hành động phóng to, thu nhỏ màn hình, âm lượng bằng tay phải, nhưng có thể di chuyển chuột, click chuột bằng cả hai tay trái hoặc phải |

Khảo sát phần mềm tương tự

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Ứng dụng | Mô tả | Hình ảnh |
| 1 | Chat GPT[17]  https://chatgpt.com/ | Ứng dụng cho phép trò chuyện với người dùng | Hình ảnh ứng dụng Chat GPT(1)    Hình ảnh ứng dụng Chat GPT(2)    Hình ảnh ứng dụng Chat GPT(3) |
| 2 | Gemini[18]  https://gemini.google.com/ | Ứng dụng cho phép trò chuyện với người dùng | Hình ảnh ứng dụng Gemini(1)    Hình ảnh ứng dụng Gemini(2)    Hình ảnh ứng dụng Gemini(2) |

## **CHƯƠNG 5: KẾT LUẬN**

Đề tài "Xây dựng trợ lý ảo điều khiển bằng giọng nói và cử chỉ" là một nghiên cứu có ý nghĩa quan trọng trong bối cảnh phát triển công nghệ hiện nay, khi mà trí tuệ nhân tạo (AI), nhận diện giọng nói và cử chỉ, cũng như các ứng dụng điều khiển thông minh đang ngày càng trở nên phổ biến. Việc kết hợp hai phương thức điều khiển — giọng nói và cử chỉ — trong một hệ thống trợ lý ảo không chỉ mang lại sự tiện lợi, mà còn tạo ra một trải nghiệm người dùng thân thiện và tối ưu hơn.

Đề tài không những mà còn mang lại nhiều cơ hội học hỏi và phát triển các kỹ năng quan trọng trong lĩnh vực công nghệ thông tin. Chúng em được tìm hiểu về trí tuệ nhân tạo và học máy, đặc biệt là các thuật toán nhận diện giọng nói và cử chỉ, cũng như cách áp dụng chúng để xây dựng các hệ thống thông minh. Thêm vào đó, các bạn sẽ học cách phát triển giao diện người dùng (UI/UX), tích hợp các công nghệ khác nhau để tạo ra một trợ lý ảo hoàn chỉnh, thân thiện với người sử dụng. Quá trình phát triển hệ thống này cũng giúp chúng em nâng cao kỹ năng giải quyết vấn đề, sáng tạo và làm việc nhóm, đồng thời rèn luyện khả năng quản lý dự án, teamwok, cách dùng git, kiểm thử,…

Nhược điểm:

* Độ chính xác còn chưa cao
* Thời gian phản hồi chưa được nhanh
* Có thể gây ra tình trạng giật lag

# **C. TÀI LIỆU THAM KHẢO**

[1] MediaPipe, Tác giả Google, Phát Hành 2012, Nguồn <https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer?hl=vi>

[2] OpenCV, Tác giả [Intel](https://en.wikipedia.org/wiki/Intel), [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage), Itseez, Được Cập Nhật Mới Nhất 4 June 2024, Nguồn

<https://opencv.org/>

[3] Mô hình CNN, Wikipedia, Nguồn GitHub

<https://cs231n.github.io/convolutional-networks/>

[4] FER2013 Dataset, Tác giả Manas Sambare, Nguồn Kaggle <https://www.kaggle.com/datasets/msambare/fer2013/>

[5] API openwheathermap, Tác giả OpenWeather Ltd, Phát Hành 2014, Nguồn

<http://api.openweathermap.org/data/2.5/weather?>

[6] speech\_recognition, Tác giả Anthony F. Schapira, Phát Hành 2011, Nguồn

<https://pypi.org/project/SpeechRecognition/>

[7] playsound, Tác giả Boris Smus, Phát hành 2012, Nguồn

<https://pypi.org/project/playsound/>

[8] gtts (Google Text-to-Speech), Tác giả: Google, Phát Hành 2015, Nguồn

<https://pypi.org/project/gTTS/>

[8] wikipedia, Tác giả Abhijit Joshi, Phát Hành 2015, Nguồn

<https://pypi.org/project/wikipedia/>

[9] pygame, Tác giả Pete Shinners và cộng đồng Pygame, Phát Hành 2000, Nguồn

<https://www.pygame.org/news>

[10] youtube-search, Tác giả Xeles, Phát Hành 2017, Nguồn

<https://pypi.org/project/youtube-search/>

[11] pyautogui, Tác giả Al Sweigart, Phát Hành 2008, Nguồn

<https://pyautogui.readthedocs.io/en/latest/>

[12] pynput, Tác giả Jonathan Wong, Phát Hành 2015, Nguồn

<https://pypi.org/project/pynput/>

[13] [14] keras, tensoflow, Tác giả Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Nguồn

<https://www.tensorflow.org/guide/keras?hl=vi>

[15] customtkinter, Tác giả Tom Schimansky, Phát Hành 2022, Nguồn GitHub

<https://github.com/TomSchimansky/CustomTkinter>

[16] pil (Python Imaging Library), Tác giả Fredrik Lundh, Phát Hành 1995, Nguồn

<https://pypi.org/project/pillow/>

[17] ChatGPT, Tác giả OpenAI, Phát Hành 2022, Nguồn

<https://chatgpt.com/>

[18] Gemini, Tác giả Google, Phát Hành 2023, Nguồn

<https://chatgpt.com/>