

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN**

Tel. (+84.0236) 3736949, Fax. (+84.0236) 3842771
Website: <http://dut.udn.vn/khoacntt>, E-mail: cntt@dut.udn.vn



**BÁO CÁO MÔN HỌC
MÔ HÌNH HÓA HÌNH HỌC**

**ĐỀ TÀI:
ĐIỀU KHIỂN ĐỐI TƯỢNG 3D BẰNG
NHẬN DẠNG CỬ CHỈ TAY**

SINH VIÊN THỰC HIỆN

Nguyễn Văn Mạnh	LỚP: 20T1	MSSV: 102200024
Nguyễn Văn Hoàng Phúc	LỚP: 20T1	MSSV: 102200028

GIẢNG VIÊN HƯỚNG DẪN: PGS.TS. Nguyễn Tấn Khôi

Đà Nẵng, 06/2024

MỤC LỤC

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT	3
1.1 CƠ SỞ LÝ THUYẾT	3
1.1.1 Lý thuyết về vật thể 3D và ứng dụng trong thực tế.....	3
1.1.2 Công nghệ WebGL	4
1.1.3 Thư viện Three.js.....	5
1.2 PHÁT BIỂU BÀI TOÁN	7
1.3 KẾT CHƯƠNG	8
CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG.....	9
2.1 PHÂN TÍCH THIẾT KẾ HỆ THỐNG	9
2.1.1 Kiến trúc hệ thống	9
2.1.2 Mô tả các module trong hệ thống	9
2.2 PHÂN TÍCH DỮ LIỆU.....	10
2.2.1 Thu thập các mẫu đối tượng 3D có sẵn	10
2.2.2 Xây dựng tập dữ liệu cử chỉ tay	11
2.3 PHÂN TÍCH CHỨC NĂNG.....	13
2.3.1 Chức năng lựa chọn mô hình.....	13
2.3.2 Chức năng hướng dẫn sử dụng.....	13
2.3.3 Chức năng điều khiển đối tượng bằng nút nhấn.....	13
2.3.4 Chức năng điều khiển đối tượng bằng nhận diện cử chỉ tay	13
2.4 THIẾT KẾ CƠ SỞ DỮ LIỆU	14
2.5 XÂY DỰNG HỆ THỐNG	14
2.5.1 Tổng quan hệ thống	14
2.5.2 Xây dựng và huấn luyện model nhận diện hành động	15
2.5.3 Tổ chức chương trình	20
2.6 KẾT CHƯƠNG	21
CHƯƠNG 3. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ	22
3.1 KẾT QUẢ TRÊN TẬP KIỂM THỬ.....	22
3.1.1 Kết quả huấn luyện	22
3.1.2 Kết quả kiểm thử	22
3.1.3 Bàn luận.....	24
3.2 KẾT QUẢ THỰC NGHIỆM	25
3.2.1 Nhận diện hành động.....	25
3.2.2 Dữ liệu hành động được lưu vào database	25
3.2.3 Chức năng hướng dẫn sử dụng	26
3.2.4 Chức năng lựa chọn đối tượng	27
3.2.5 Chức năng điều khiển đối tượng bằng nút nhấn.....	28
3.2.6 Chức năng điều khiển vật thể bằng cử chỉ tay.....	28
3.3 NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ	28

3.3.1	Chức năng socket.....	28
3.3.2	Load màu các đối tượng 3D	29
3.4	KẾT CHƯỠNG	29
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		30
A.	KẾT QUẢ ĐẠT ĐƯỢC.....	30
B.	KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN.....	30

DANH SÁCH HÌNH ẢNH

Hình 1.1 Hình ảnh quá trình in 3D	4
Hình 1.2 Các bước xử lý của WebGL	5
Hình 1.3 Cấu trúc để build 3D three.js app	6
Hình 2.1 Thu thập các mẫu đối tượng 3D có sẵn	11
Hình 2.2 Xây dựng tập dữ liệu cử chỉ tay	12
Hình 2.3 Diagram của cơ sở dữ liệu hệ thống	14
Hình 2.4 Tổng quan hệ thống	14
Hình 2.5 Pipeline đề xuất đối với hệ thống nhận diện DSL	15
Hình 2.6 Thứ tự và nhãn cho các điểm chính có trong tay của MediaPipe	16
Hình 2.7 Thứ tự và nhãn cho các điểm chính có trong pose	17
Hình 2.8 Trích xuất đặc trưng các bộ phận trên cơ thể (tay, khung xương)	18
Hình 2.9 Cấu trúc mô hình LSTM	19
Hình 2.10 Cấu trúc mô hình GRU	19
Hình 2.11. Tổ chức mã nguồn của hệ thống	21
Hình 2.12. Các tập tin OBJ liên quan đến đối tượng 3D	21
Hình 2.13. Các tập tin liên quan đến database và diagram	21
Hình 3.1 Đồ thị accuracy khi huấn luyện trên tập train	22
Hình 3.2 Đồ thị loss function khi huấn luyện trên tập train	22
Hình 3.3 Ma trận nhầm lẫn cho 14 loại cử chỉ tay	23
Hình 3.4 Nhận diện hành động	25
Hình 3.5 Dữ liệu hành động được lưu vào database	25
Hình 3.6 Chức năng hướng dẫn sử dụng	26
Hình 3.7 Chức năng lựa chọn đối tượng	27
Hình 3.8 Chức năng điều khiển vật thể bằng nút nhấn	28
Hình 3.9 Chức năng điều khiển vật thể bằng cử chỉ tay	28

DANH SÁCH BẢNG

Bảng 2.1 Kiến trúc hệ thống	9
Bảng 2.2 Mô hình đề xuất.....	20
Bảng 2.3 Bảng thống kê kết quả chạy trên các mô hình.....	23
Bảng 2.4 Bảng thống kê thời gian dự đoán trên tập kiểm thử (140 video).....	23

DANH SÁCH TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
WebGL	Web Graphics Library
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
3D	Three-Dimensional
AI	Artificial Intelligence
OBJ	Object File

MỞ ĐẦU

1. Tổng quan về đề tài

Trong một thế giới hiện đại, nơi công nghệ ngày càng tiên bộ, việc phát triển các phương pháp điều khiển đối tượng 3D bằng cử chỉ tay đã trở thành một mảnh ghép quan trọng trong lĩnh vực tương tác người-máy. Đây không chỉ là một lĩnh vực nghiên cứu hấp dẫn mà còn mang lại tiềm năng ứng dụng rộng lớn trong nhiều lĩnh vực khác nhau như thực tế ảo, y tế, giáo dục, và giải trí.

Công nghệ nhận dạng cử chỉ tay cho phép người dùng tương tác với đối tượng 3D một cách tự nhiên bằng cách sử dụng các cử chỉ và chuyển động của tay, thay vì phải sử dụng bàn phím, chuột hoặc các thiết bị điều khiển truyền thống. Điều này tạo ra một trải nghiệm tương tác mới mẻ và sinh động, giúp người dùng dễ dàng thao tác và điều khiển các đối tượng 3D một cách hiệu quả hơn.

Đề tài này không chỉ tập trung vào việc phát triển các thuật toán nhận dạng cử chỉ tay chính xác và hiệu quả mà còn đề xuất và thử nghiệm các phương pháp điều khiển đối tượng 3D dựa trên những cử chỉ tay đã được nhận dạng. Các ứng dụng tiềm năng của đề tài này có thể bao gồm điều khiển các đối tượng 3D trong môi trường thực tế ảo, điều khiển các robot trong sản xuất tự động, hoặc thậm chí điều khiển các thiết bị y tế trong phẫu thuật.

Việc nghiên cứu và phát triển các phương pháp điều khiển đối tượng 3D bằng nhận dạng cử chỉ tay không chỉ đóng vai trò quan trọng trong việc nâng cao trải nghiệm người dùng mà còn mở ra những cơ hội mới trong lĩnh vực tương tác người-máy và ứng dụng công nghệ 3D.

2. Mục đích và ý nghĩa của đề tài

2.1. Mục đích

Mục tiêu của dự án là phát triển một phần mềm điều khiển đối tượng 3D bằng nhận dạng cử chỉ tay. Cung cấp cho người dùng khả năng tương tác máy tính thông qua cử chỉ tay, tạo ra một trải nghiệm tương tác mới mẻ và hiệu quả.

2.2. Ý nghĩa

Việc phát triển phần mềm này không chỉ giúp cải thiện trải nghiệm người dùng khi tương tác với đối tượng 3D mà còn mở ra các tiềm năng ứng dụng rộng rãi trong nhiều lĩnh vực như thực tế ảo, giáo dục, y tế, và sản xuất tự động. Đồng thời, việc tạo ra một giao diện tương tác dựa trên cử chỉ tay còn giúp nâng cao sự tiện lợi và hiệu quả của các ứng dụng 3D, đồng thời khuyến khích sự sáng tạo và sự tham gia của người dùng.

3. Bố cục của đề án

Đề án bao gồm các nội dung sau:

Mở đầu

Chương 1: Cơ sở lý thuyết

Chương 2: Phân tích thiết kế và xây dựng hệ thống

Chương 3: Kết quả thực nghiệm và đánh giá kết quả

Kết luận và hướng phát triển.

Chương 1. CƠ SỞ LÝ THUYẾT

1.1 CƠ SỞ LÝ THUYẾT

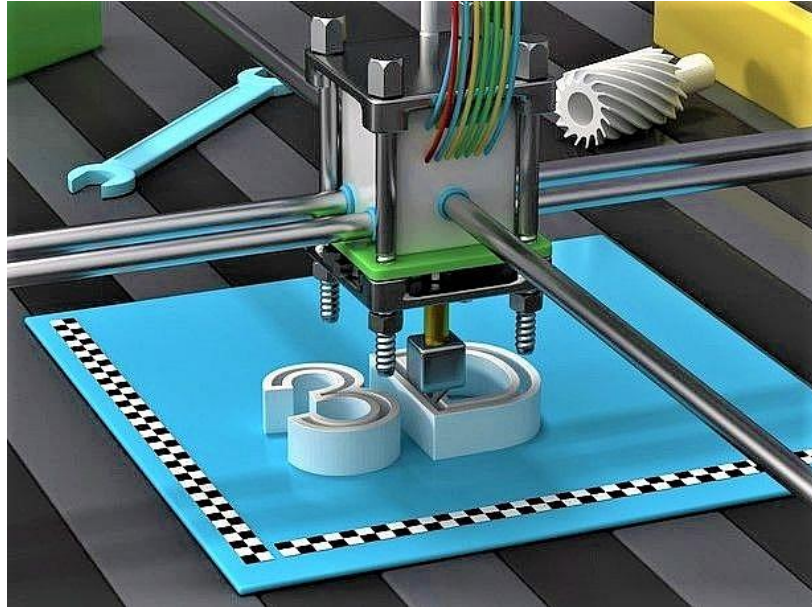
1.1.1 Lý thuyết về vật thể 3D và ứng dụng trong thực tế

Trên cơ sở của lý thuyết vật thể 3D và ứng dụng của nó trong thực tế, chúng ta có thể hiểu rõ hơn về cách mà các đối tượng ba chiều được biểu diễn và sử dụng trong nhiều lĩnh vực khác nhau. Vật thể 3D là một khái niệm quan trọng trong lĩnh vực đồ họa máy tính, mô phỏng, và thiết kế sản phẩm, cũng như trong các ngành công nghiệp như kiến trúc, y học, và giải trí số.

Trong thế giới của vật thể 3D, chúng ta thấy rằng mỗi đối tượng được biểu diễn dưới dạng một mô hình ba chiều, bao gồm chiều dài, chiều rộng và chiều cao. Các mô hình này có thể được tạo ra thông qua việc sử dụng các phần mềm đồ họa 3D như Blender, 3ds Max, hoặc Maya, trong đó các hình dạng và cấu trúc của các vật thể được tạo ra và chỉnh sửa.

Lý thuyết vật thể 3D, chúng ta cũng có các thành phần chính như sau:

- **Dữ liệu:** Đây là thông tin về các vật thể 3D, bao gồm cả hình dạng, kích thước và vị trí trong không gian. Dữ liệu này có thể được tạo ra từ nhiều nguồn khác nhau như quét 3D, mô hình hóa hoặc tạo ra bằng tay từ các công cụ thiết kế.
- **Phần mềm:** Các ứng dụng và công cụ đồ họa 3D cho phép tạo ra, chỉnh sửa và hiển thị các vật thể 3D. Các phần mềm phổ biến bao gồm Autodesk Fusion 360, SolidWorks và SketchUp.
- **Quy trình:** Các bước và phương pháp để tạo ra và xử lý các vật thể 3D, từ quy trình thiết kế đến quy trình sản xuất.
 - **Thiết kế:** Sử dụng phần mềm như Blender, Maya hoặc SketchUp để tạo mô hình 3D ban đầu.
 - **Tối ưu hóa:** Giảm số lượng đa giác mà không làm mất chi tiết quan trọng để cải thiện hiệu suất.
 - **Xuất file:** Xuất mô hình 3D sang định dạng OBJ hoặc các định dạng khác để sử dụng trong các ứng dụng hoặc game engines.
 - **In 3D:** Sử dụng máy in 3D để tạo mô hình vật lý từ tệp thiết kế.



Hình 1.1 Hình ảnh quá trình in 3D

- Con người: Những người sử dụng và quản lý dữ liệu vật thể 3D, bao gồm cả các nhà thiết kế, kỹ sư và nhà sản xuất.
 - Nhà thiết kế: Tạo mô hình 3D cho các sản phẩm mới, ví dụ như mô hình nội thất.
 - Kỹ sư: Phân tích và kiểm tra các khía cạnh kỹ thuật của sản phẩm bằng mô hình 3D.
 - Nhà sản xuất: Hướng dẫn quá trình sản xuất và lắp ráp bằng mô hình 3D.

Tổng hợp lại, lý thuyết về vật thể 3D là một phần quan trọng của nhiều lĩnh vực, từ công nghiệp đến giáo dục và giải trí, và hiểu biết về các khái niệm cơ bản và thành phần của nó giúp chúng ta áp dụng hiệu quả hơn trong thực tế.

- Công nghiệp: Cải thiện thiết kế sản phẩm và quy trình sản xuất, ví dụ như trong sản xuất máy bay.
- Giáo dục: Giảng dạy các khái niệm phức tạp trong khoa học và kỹ thuật, ví dụ như học giải phẫu trong y khoa.
- Giải trí: Tạo nền tảng cho phim hoạt hình, video game và thực tế ảo.

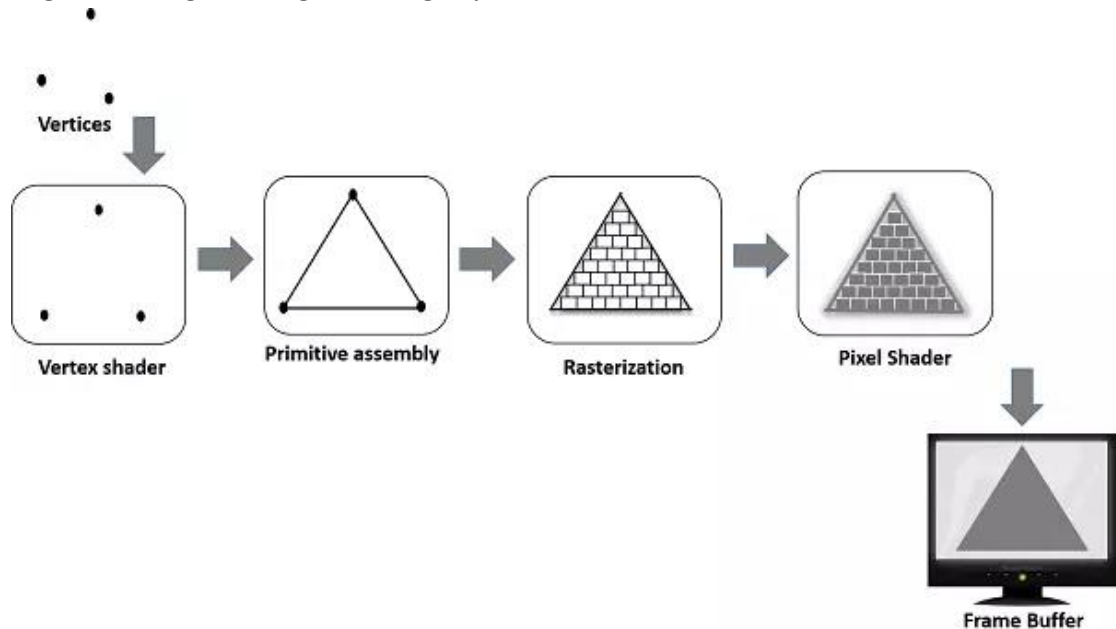
1.1.2 Công nghệ WebGL

WebGL là một công nghệ quan trọng trong lĩnh vực phát triển web, cho phép hiển thị đồ họa 3D trực tiếp trong trình duyệt mà không cần sử dụng các plugin bổ sung. Điều này mang lại lợi ích lớn cho việc tạo ra các trải nghiệm trực tuyến phong phú và tương tác.

Với WebGL, các nhà phát triển có khả năng tạo ra các ứng dụng đồ họa 3D động, trò chơi trực tuyến, các biểu đồ phức tạp và nhiều ứng dụng khác mà trước đây thường chỉ có thể được thực hiện thông qua phần mềm độc lập. WebGL cung cấp một cách tiếp cận mạnh mẽ cho việc hiển thị và tương tác với đối tượng 3D trên web, mở ra cánh cửa cho sự sáng tạo không giới hạn.

Tuy nhiên, việc làm quen với WebGL đòi hỏi một kiến thức cơ bản về lập trình JavaScript và đồ họa máy tính 3D. Các nhà phát triển cần phải hiểu rõ về các khái niệm như véc-tơ, ma trận, ánh sáng và vật liệu để tận dụng tối đa tiềm năng của công nghệ này.

Với sự phát triển không ngừng của WebGL và sự hỗ trợ từ các framework như Three.js, việc tạo ra các ứng dụng và trải nghiệm 3D trên web trở nên dễ dàng hơn bao giờ hết. Điều này thúc đẩy sự phát triển của các ứng dụng web động đỉnh cao và mang lại trải nghiệm người dùng tuyệt vời hơn.



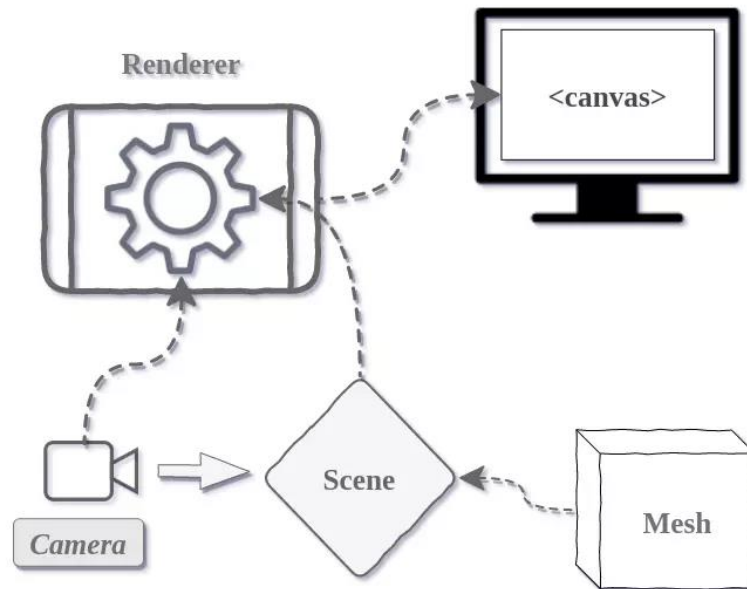
Hình 1.2 Các bước xử lý của WebGL

1.1.3 Thư viện Three.js

Three.js là một thư viện JavaScript mã nguồn mở được sử dụng để tạo và hiển thị đồ họa 3D trên trình duyệt web. Nó cung cấp một cách tiếp cận dễ dàng và mạnh mẽ để tạo ra các đối tượng 3D, kết cấu, ánh sáng và hiệu ứng trong không gian 3 chiều. Với Three.js, có thể tạo ra các trải nghiệm tương tác 3D hấp dẫn trên các trang web một cách dễ dàng.

Một trong những định dạng phổ biến để lưu trữ các mô hình 3D là định dạng OBJ. Định dạng OBJ là một định dạng văn bản đơn giản được sử dụng để mô tả các đối tượng 3D bằng cách định nghĩa các điểm, đoạn, mặt và các thông tin về vật liệu. Một tệp OBJ có thể bao gồm cả thông tin về hình dạng và vật liệu của mô hình.

Khi sử dụng Three.js, có thể tải và hiển thị các tệp OBJ 3D trong trình duyệt web một cách dễ dàng. Thư viện cung cấp các lớp và phương pháp để tải và hiển thị các mô hình OBJ, cũng như điều chỉnh các thuộc tính như vị trí, quay, phóng to thu nhỏ và ánh sáng của các đối tượng 3D này. Điều này mở ra nhiều cơ hội cho việc tạo ra các ứng dụng web tương tác và trải nghiệm thú vị sử dụng đồ họa 3D.



Hình 1.3 Cấu trúc để build 3D three.js app

Các bước sử dụng Three.js trong HTML và JavaScript thuần:

1. Bao gồm thư viện Three.js:

- Tải và thêm thư viện Three.js vào trang HTML của dự án. Có thể làm điều này bằng cách thêm thẻ `<script>` trong phần `<head>` hoặc `<body>` của tệp HTML và liên kết tới phiên bản CDN của Three.js.

2. Khởi tạo các thành phần cơ bản:

- Tạo một cảnh (scene) để chứa tất cả các đối tượng 3D.
- Tạo một camera để xác định góc nhìn của người dùng trong cảnh 3D.
- Tạo một renderer để hiển thị cảnh và camera trên trang web. Cần đặt kích thước cho renderer và thêm nó vào tài liệu HTML.

3. Tạo đối tượng 3D:

- Tạo hình học (geometry) cho đối tượng, ví dụ như khối lập phương (cube).
- Tạo vật liệu (material) để áp dụng cho hình học, giúp xác định màu sắc và giao diện của đối tượng.
- Tạo một mesh bằng cách kết hợp hình học và vật liệu, sau đó thêm nó vào cảnh.

4. ***Đặt vị trí camera:***

- Đặt vị trí camera sao cho nó có thể nhìn thấy toàn bộ cảnh và các đối tượng trong đó. Thường thì camera sẽ được đặt cách đối tượng một khoảng cách nhất định để đảm bảo mọi thứ hiển thị rõ ràng.

5. ***Thêm ánh sáng:***

- Để các đối tượng hiển thị rõ ràng và chân thực, thêm ánh sáng vào cảnh. Có nhiều loại ánh sáng khác nhau như ánh sáng môi trường, ánh sáng điểm và ánh sáng hướng. Chọn loại phù hợp và thêm nó vào cảnh.

6. ***Tạo hàm render:***

- Tạo một hàm để cập nhật và render cảnh liên tục, giúp tạo ra các hiệu ứng động. Hàm này thường được gọi là `'animate'` hoặc `'render'`, và nó sẽ sử dụng `'requestAnimationFrame'` để tạo vòng lặp render mượt mà.

7. ***Tải và hiển thị tệp OBJ (tùy chọn):***

- Nếu muốn tải và hiển thị mô hình 3D từ tệp OBJ, bao gồm thêm thư viện `'OBJLoader'` của Three.js.
- Sử dụng `'OBJLoader'` để tải tệp OBJ và thêm mô hình vào cảnh. Có thể đặt các thuộc tính như vị trí, tỷ lệ và quay của mô hình để hiển thị đúng cách.

8. ***Chạy ứng dụng:***

- Cuối cùng, chạy hàm `'animate'` hoặc `'render'` để bắt đầu vòng lặp render và hiển thị cảnh 3D trên trang web.

1.2 **PHÁT BIỂU BÀI TOÁN**

Mục tiêu: Xây dựng ứng dụng điều khiển đối tượng 3D bằng cử chỉ tay.

Phạm vi nghiên cứu:

- Thu thập dữ liệu về các đối tượng 3D dưới dạng file OBJ.
- Xây dựng tập dữ liệu hành động cử chỉ tay cho các hành động điều khiển đối tượng 3D và đào tạo model AI để nhận diện cử chỉ tay.
- Triển khai hệ thống tương tác giữa Model nhận diện hành động và WebGL Threejs bằng công nghệ Socket kết hợp với Database Mysql.

Các vấn đề đặt ra:

- Các mẫu đối tượng 3D có sẵn: Cách thu thập và hiển thị các mẫu 3D có sẵn.

- Tương tác với người dùng: Thiết kế giao diện người dùng thân thiện và dễ sử dụng, cung cấp tính năng lựa chọn đối tượng và các nút nhấn để điều chỉnh đối tượng.
- Triển khai hệ thống: Lựa chọn nền tảng phù hợp và quy trình triển khai ứng dụng WebGL.

1.3 KẾT CHƯƠng

Chương này đã cung cấp một cái nhìn tổng quan về lý thuyết và công nghệ liên quan đến đề tài, bao gồm lý thuyết 3D, WebGL, ThreeJs, Socket, AI Model.

Hướng phát triển: Để mở rộng đề tài, có thể nghiên cứu về tính năng mở rộng của hệ thống WebGL như: Xây dựng thế giới ảo, tích hợp thêm các bảng chọn để điều khiển tham số, hiển thị một tập đối tượng cùng một lúc, xây dựng tập đối tượng có liên quan.

Chương 2. PHÂN TÍCH THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

2.1 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1.1 Kiến trúc hệ thống

Module	Thành phần	Công nghệ sử dụng
Nhận diện hành động	Camera/Webcam, OpenCV, AI/ML Model	Python, OpenCV
Cơ sở dữ liệu	MySQL, bảng hành động	MySQL
Giao tiếp WebSocket	WebSocket server/client	Python, JavaScript
Xử lý và điều khiển	JavaScript, mô hình 3D	JavaScript, WebGL, Three.js
Giao diện người dùng	Web UI	HTML, CSS, JavaScript

Bảng 2.1 Kiến trúc hệ thống

Luồng Hoạt Động

1. Người dùng thực hiện các hành động trước camera.
2. Module nhận diện hành động nhận diện và lưu các hành động vào cơ sở dữ liệu MySQL.
3. Module giao tiếp WebSocket truyền dữ liệu hành động mà máy chủ đọc được từ database tới client.
4. Module xử lý và điều khiển nhận dữ liệu hành động và điều khiển mô hình 3D dựa trên hành động đó.
5. Giao diện người dùng hiển thị mô hình 3D và phản hồi theo các hành động của người dùng.

2.1.2 Mô tả các module trong hệ thống

1. Module nhận diện hành động:

- Mô tả: Nhận diện các hành động từ người dùng qua webcam.
- Chi tiết:
 - Camera/Webcam: Thu thập hình ảnh của người dùng.
 - OpenCV/Python: Sử dụng thư viện OpenCV để xử lý hình ảnh và nhận diện hành động.

- Mô hình AI/ML: Sử dụng mô hình học máy để nhận diện các hành động của người dùng từ hình ảnh.

2. Module Cơ sở dữ liệu:

- Mô tả: Lưu trữ và quản lý các hành động được nhận diện.
- Chi tiết:
 - MySQL: Hệ quản trị cơ sở dữ liệu để lưu trữ các hành động được nhận diện.
 - Bảng hành động: Bảng trong cơ sở dữ liệu để lưu các hành động với các trường gồm ID, tên hành động

3. Module Giao tiếp WebSocket:

- Mô tả: Truyền dữ liệu hành động từ máy chủ Python tới client JavaScript.
- Chi tiết:
 - WebSocket: Sử dụng giao thức WebSocket để truyền thông tin giữa máy chủ và client một cách hiệu quả và thời gian thực.

4. Module Xử lý và Điều khiển:

- Mô tả: Nhận dữ liệu từ máy chủ và điều khiển mô hình 3D.
- Chi tiết:
 - JavaScript: Xử lý hành động nhận được và điều khiển mô hình 3D.
 - Mô hình 3D: Mô hình 3D được điều khiển bởi các hành động nhận diện từ người dùng.

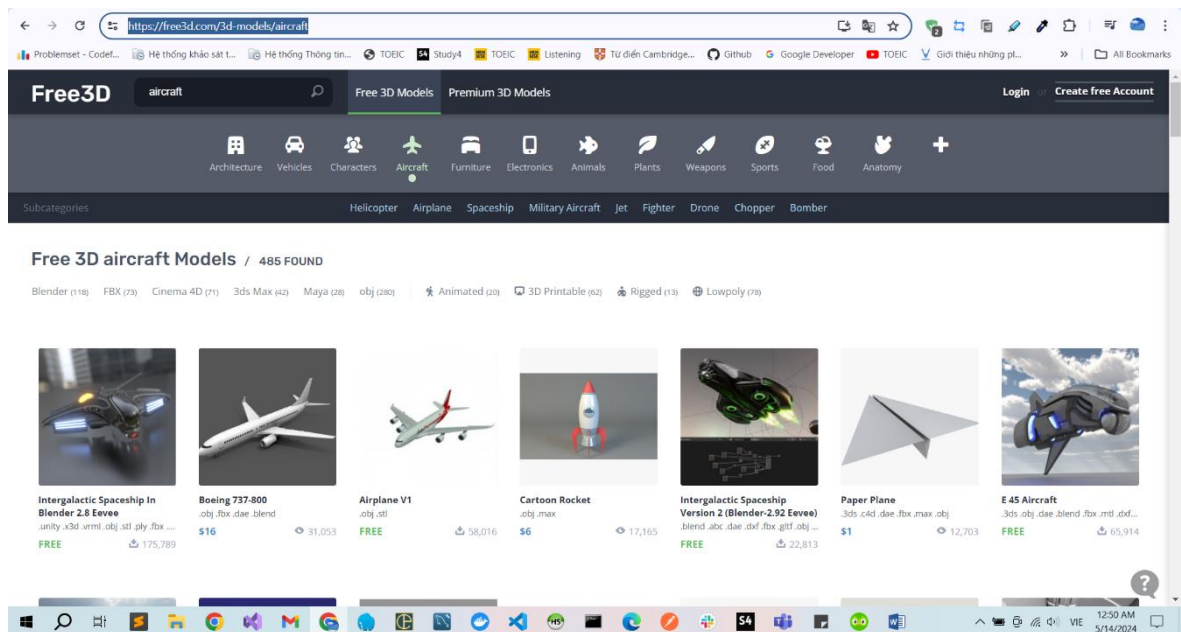
5. Module Giao diện người dùng (UI):

- Mô tả: Hiển thị và tương tác với người dùng.
- Chi tiết:
 - Web UI: Giao diện người dùng trên trình duyệt để hiển thị mô hình 3D và nhận đầu vào từ người dùng.

2.2 PHÂN TÍCH DỮ LIỆU

2.2.1 Thu thập các mẫu đối tượng 3D có sẵn

- Có nhiều nguồn để thu thập và một trong số chúng là trang web: <https://free3d.com/3d-models/aircraft>, ta có thể vào đây để tải về các file OBJ của các đối tượng có sẵn miễn phí hoặc trả phí.



Hình 2.1 Thu thập các mẫu đối tượng 3D có sẵn

2.2.2 Xây dựng tập dữ liệu cử chỉ tay

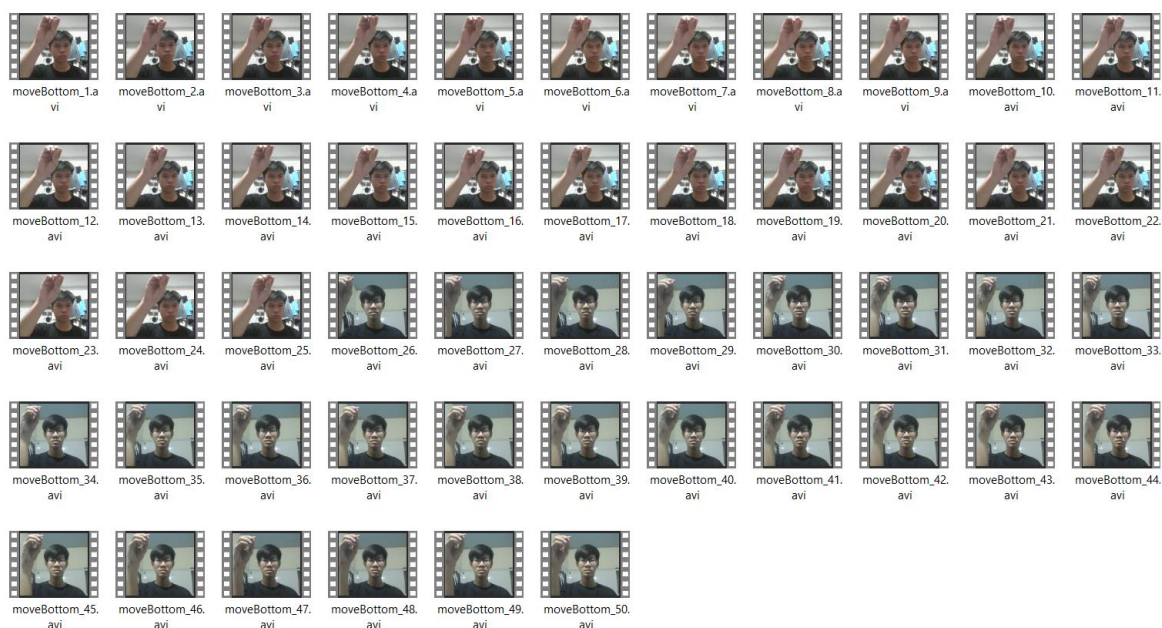
Các thử nghiệm để nhận dạng ngôn ngữ ký hiệu động được thực hiện bằng cách sử dụng tập dữ liệu DSL14 gồm 14 ký hiệu từ 2 người quay khác nhau (chứa 700 video). Tập dữ liệu video được quay bằng camera laptop thông thường.

Mỗi ký hiệu được quay 50 video được chia ngẫu nhiên để đào tạo và thử nghiệm. Tất cả video trong DSL14 được ghi lại trong môi trường trong nhà với ánh sáng thông thường và máy ảnh di động trung bình (720p). Mỗi video được ghi ở 30 khung hình mỗi giây (FPS) và có cùng số khung hình và thời lượng (1s).

Tập dữ liệu thu được gồm 14 hành động tương ứng với 14 lệnh điều khiển vật thể:

1. **Move Left:** Hành động tương tự với việc nắm một vật thể ngang ngực và bắt đầu kéo nó từ phải sang trái bằng tay phải.
2. **Move Right:** Hành động tương tự với việc nắm một vật thể ngang ngực và bắt đầu kéo nó từ trái sang phải bằng tay phải.
3. **Move Top:** Hành động tương tự với việc nắm một vật thể trước mặt và bắt đầu kéo nó từ dưới lên trên bằng tay phải.
4. **Move Bottom:** Hành động tương tự với việc nắm một vật thể trước mặt và bắt đầu kéo nó từ trên xuống dưới bằng tay phải.
5. **Zoom In:** Hành động tương tự với việc 2 bàn tay đang nắm một vật thể đối diện trước ngực và kéo căng ra 2 bên.

6. **Zoom Out:** Hành động tương tự với việc 2 lòng bàn tay mở ra, đối diện nhau, đặt song song với cơ thể, vuông góc với mặt đất sau đó ép chặt lại đối diện trước ngược.
7. **Rotate XC:** Duỗi thẳng tay phải ra song song với mặt đất và vuông góc với thân người, sau đó hất tay lên phía trên giống như đang dùng tay để lật một vật.
8. **Rotate XN:** Duỗi thẳng tay ra và hợp mới mặt đất một góc khoảng 135 độ, sau đó gập tay xuống như đang nhấn một vật xuống.
9. **Rotate YC:** Sử dụng tay phải, đặt cánh tay phải song song với thân người, lòng bàn tay mở ra và đưa cánh tay từ phải sang trái.
10. **Rotate YN:** Sử dụng tay phải, đặt cánh tay phải song song với thân người, lòng bàn tay mở ra và đưa cánh tay từ trái sang phải
11. **Rotate ZC:** Sử dụng tay phải, lòng bàn tay mở ra, đưa tay phải từ góc dưới bên phải lên góc trên bên trái.
12. **Rotate ZN:** Sử dụng tay phải, lòng bàn tay mở ra, đưa tay phải từ góc trên bên trái xuống góc dưới bên phải.
13. **Move Far:** Sử dụng tay phải, để lòng bàn tay mở ra và song song mới thân người, sau đó đẩy lòng bàn tay tới trước như đang đẩy một vật thể ra xa.
14. **Move Near:** Đưa bàn tay tới phía trước, chụm lại và kéo bàn tay lui trước ngực như đang kéo một vật thể tới gần hơn phía trước ngực.



Hình 2.2 Xây dựng tập dữ liệu cử chỉ tay

Chia tập dữ liệu: Nhóm thực hiện chia tập train:test theo tỉ lệ 80%:20% và thực hiện huấn luyện trên tập train đồng thời thống kê các kết quả trên tập test.

2.3 PHÂN TÍCH CHỨC NĂNG

2.3.1 Chức năng lựa chọn mô hình

Người dùng có thể lựa chọn nhiều loại mô hình nào để tăng tính đa dạng cho ứng dụng.

2.3.2 Chức năng hướng dẫn sử dụng

Chức năng hướng dẫn người dùng sử dụng hệ thống bằng cách cho người dùng xem các video mẫu và mô tả hành động điều khiển.

2.3.3 Chức năng điều khiển đối tượng bằng nút nhấn

Người dùng có thể điều khiển đối tượng bằng các nút nhấn, tổng cộng gồm có 14 nút nhấn tương ứng với 14 cách điều khiển đối tượng, gồm có:

1. **Move Left:** Di chuyển đối tượng sang trái trên trục X.
2. **Move Right:** Di chuyển đối tượng sang phải trên trục X.
3. **Move Bottom:** Di chuyển đối tượng xuống dưới trên trục Y.
4. **Move Top:** Di chuyển đối tượng lên trên trục Y.
5. **Zoom In:** Phóng to đối tượng, làm cho nó trở nên lớn hơn.
6. **Zoom Out:** Thu nhỏ đối tượng, làm cho nó trở nên nhỏ hơn.
7. **Rotate XN:** Quay đối tượng ngược chiều kim đồng hồ quanh trục X.
8. **Rotate YN:** Quay đối tượng ngược chiều kim đồng hồ quanh trục Y.
9. **Rotate ZN:** Quay đối tượng ngược chiều kim đồng hồ quanh trục Z.
10. **Rotate XC:** Quay đối tượng theo chiều kim đồng hồ quanh trục X.
11. **Rotate YC:** Quay đối tượng theo chiều kim đồng hồ quanh trục Y.
12. **Rotate ZC:** Quay đối tượng theo chiều kim đồng hồ quanh trục Z.
13. **Move Far:** Di chuyển đối tượng xa hơn.
14. **Move Near:** Di chuyển đối tượng gần hơn.

Các hoạt động này cung cấp sự linh hoạt và kiểm soát chi tiết đối với việc điều chỉnh vị trí, phóng to, xoay và di chuyển đối tượng 3D trong không gian.

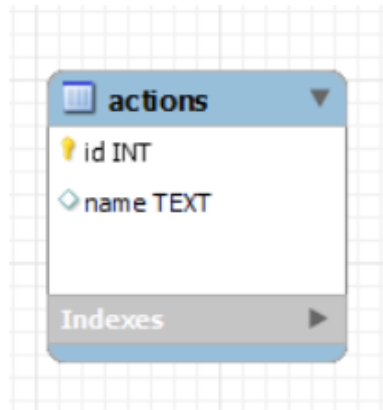
2.3.4 Chức năng điều khiển đối tượng bằng nhận diện cử chỉ tay

Thông qua cửa sổ được hiển thị hình ảnh quay bằng camera và các khớp tay được vẽ lên, hành động được tự động nhận diện và chuyển thành các câu lệnh để điều khiển vật thể

2.4 THIẾT KẾ CƠ SỞ DỮ LIỆU

Trong phần này, chúng ta sẽ mô tả cấu trúc của cơ sở dữ liệu được sử dụng trong hệ thống WebGL. Các yếu tố cơ bản bao gồm:

Bảng Dữ Liệu: Mô tả bảng dữ liệu được sử dụng để lưu trữ thông tin về hành động nhận diện được.

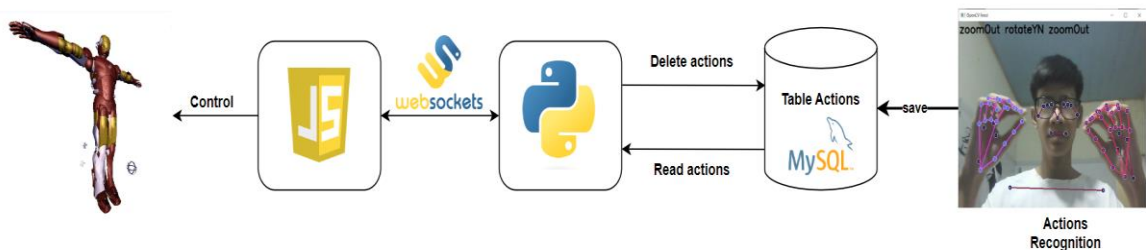


Hình 2.3 Diagram của cơ sở dữ liệu hệ thống

2.5 XÂY DỰNG HỆ THỐNG

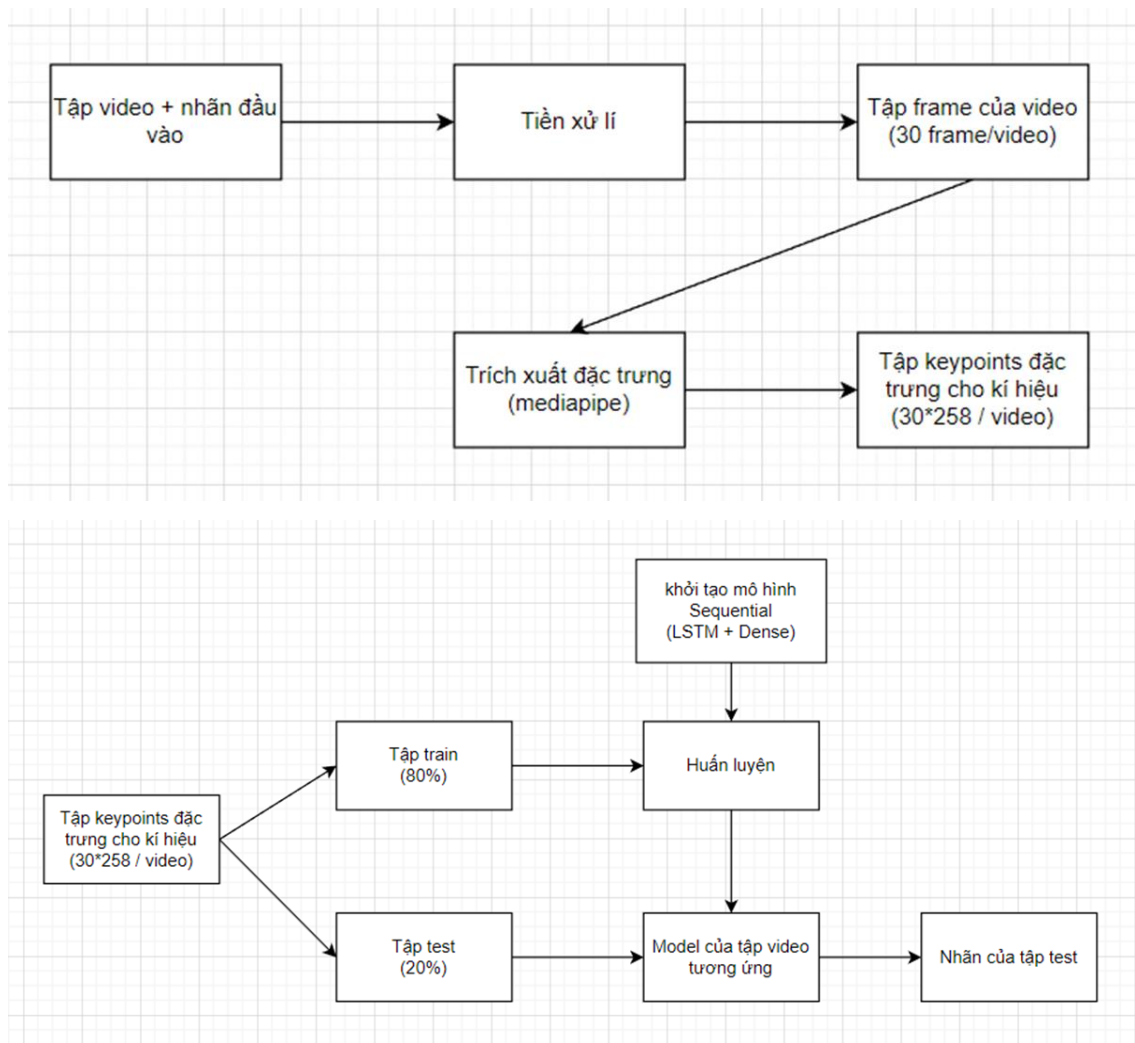
2.5.1 Tổng quan hệ thống

- Đầu tiên file recog.py được bật lên để nhận diện hành động sau đó sẽ lưu hành động nhận diện được và table actions trong database.
- Cùng với đó file server.py sẽ đọc và xóa liên tục các hành động từ table actions, sau đó sử dụng socket để gửi về cho client js.
- Client js nhận được hành động sẽ sử dụng hành động đó để điều khiển đối tượng 3D.



Hình 2.4 Tổng quan hệ thống

2.5.2 Xây dựng và huấn luyện model nhận diện hành động



Hình 2.5 Pipeline đề xuất đối với hệ thống nhận diện DSL

Phần này lý giải các phương pháp trích xuất đặc trưng các bộ phận trên cơ thể người thể hiện hành động và chi tiết triển khai mô hình Sequential cho nhận diện luồng hành động.

a. Tiền xử lí

- Điều chỉnh số frame của video về 30 frame/video: Đồng nhất số frame cho quá trình huấn luyện
- Cắt bỏ đoạn đầu cuối video không thuộc ngôn ngữ kí hiệu: Tập trung vào các hành động của ngôn ngữ kí hiệu
- Giảm số chiều ảnh hưởng của các keypoints không quan trọng: Tập trung vào các keypoints quan trọng, đặc trưng cho hành động

b. Trích xuất đặc trưng các bộ phận trên cơ thể người

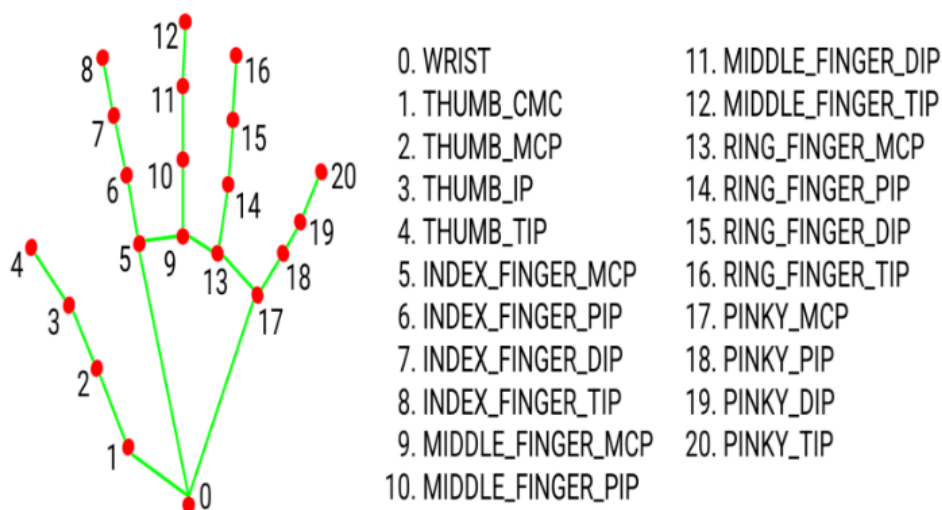
Ngôn ngữ ký hiệu dựa trên việc sử dụng tay và ước tính tư thế, tuy nhiên, DSL đối mặt với nhiều khó khăn do sự chuyển động liên tục. Những khó khăn này bao gồm việc xác định vị trí của tay, hình dạng và hướng đi. MediaPipe được sử dụng như một giải pháp cho những vấn đề này. Nhóm nghiên cứu đã sử dụng mediapipe để bắt bộ khung trên cơ thể và liên kết chúng thành các keypoints. Các keypoints này là các điểm đặc trưng trên cơ thể, được sử dụng để phân tích và nhận diện các hành động. Hành động được định nghĩa như sự chuyển tiếp giữa các keypoints trong quá trình thực hiện hành động đó.

Mediapipe trích xuất các điểm chính cho ba chiều X, Y, Z của cả hai tay và ước tính tư thế cho mỗi khung như được thể hiện trong Hình 1.

Kỹ thuật ước tính tư thế được sử dụng để dự đoán và theo dõi vị trí của tay liên quan đến cơ thể. Kết quả đầu ra của khung MediaPipe là một danh sách các điểm chính cho tay và ước tính tư thế.

Đối với mỗi tay, MediaPipe trích xuất 21 điểm chính [29] như được thể hiện trong Hình 2. Các điểm chính được tính toán trong không gian ba chiều: X, Y và Z cho cả hai tay. Do đó, số điểm chính được trích xuất của hai tay được tính toán như sau:

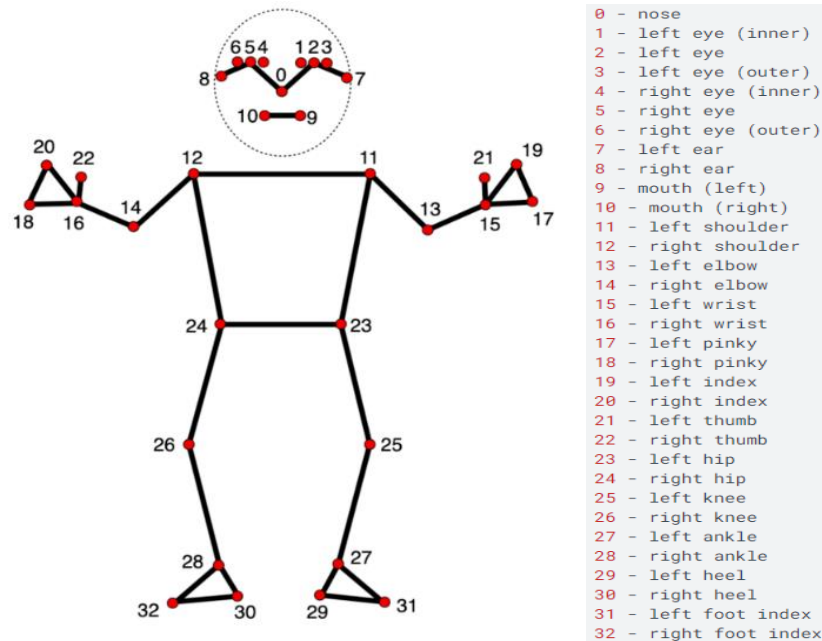
- keypoints in hand \times Three dimensions \times No. of hands = $(21 \times 3 \times 2) = 126$ keypoints.



Hình 2.6 Thứ tự và nhãn cho các điểm chính có trong tay của MediaPipe

Để ước tính tư thế, MediaPipe trích xuất 33 điểm chính [29] như được thể hiện trong Hình 3. Chúng được tính toán trong không gian ba chiều: X, Y và Z cộng với khả năng nhìn thấy. Khả năng nhìn thấy là một giá trị chỉ ra liệu điểm đó có thể nhìn thấy được hay bị che bởi một phần thân thể khác trên một khung. Do đó, số điểm chính được trích xuất từ ước tính tư thế được tính toán như sau:

- keypoints in pose \times (Three dimensions + Visibility) = $(33 \times (3 + 1)) = 132$ keypoints.



Hình 2.7 Thứ tự và nhãn cho các điểm chính có trong pose

Các điểm dưới thân sẽ nằm trong khoảng 25-32. Do đó khi thực hiện giảm chiều của keypoints các điểm này sẽ bị loại bỏ:

- keypoints in pose without legs \times (Three dimensions + Visibility) = $(25 \times (3 + 1)) = 100$ keypoints.

Bao gồm các điểm chân, tổng số điểm chính cho mỗi khung được tính như sau:

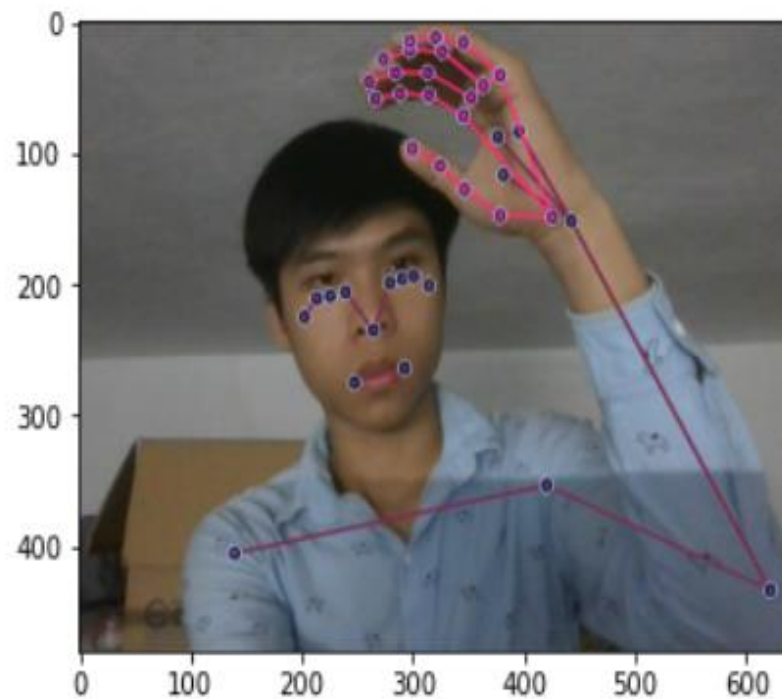
- keypoints in hands + keypoints in pose = $(126 + 132) = 258$ keypoints.

Nếu không bao gồm các điểm chân, tổng số điểm chính cho mỗi khung được tính như sau:

- keypoints in hands + keypoints in pose without legs = $(126 + 100) = 226$ keypoints.

Thao tác này được lặp lại trong toàn bộ video để trích xuất các điểm chính cho mỗi khung. Vị trí của tay, cơ thể và khuôn mặt được phát hiện với xác định hình dạng và hướng đi của chúng trong tất cả các video của DSL14-Dataset.

Như vậy sau khi thực hiện trích xuất đặc trưng của cơ thể bằng mediapipe sẽ thu được tọa độ các điểm trên không gian 3 chiều. Các dữ liệu thu thập được mà nhóm nghiên cứu chọn để thực hiện đặc trưng cho hành động của cơ thể người là tay trái, tay phải, khung xương. Xét riêng về ngôn ngữ ký hiệu thì chỉ cần có đủ 3 thành phần trên thì có thể nhận diện được toàn bộ các từ vựng của ngôn ngữ này.



Hình 2.8 Trích xuất đặc trưng các bộ phận trên cơ thể (tay, khung xương)

c. Nhận diện hành động với mô hình RNN:

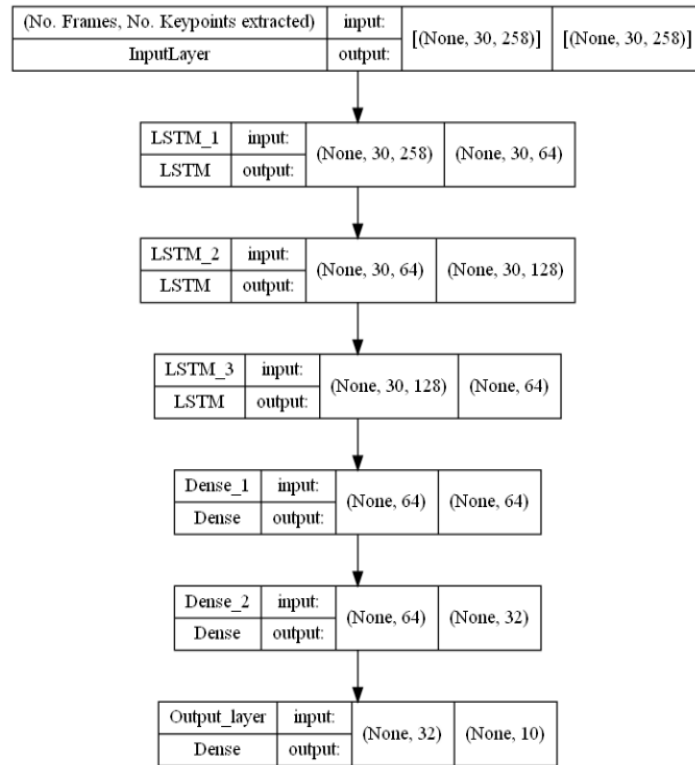
Mạng nơ-ron tuần hoàn (RNNs) là một loại mạng nơ-ron nhân tạo (ANN) sử dụng dữ liệu chuỗi thời gian và tuần tự. RNNs được gọi là tuần hoàn vì chúng thực hiện cùng một chức năng cho mỗi phần tử trong chuỗi, với sự phụ thuộc tính toán của các trạng thái trước đó. Đặc điểm chính của RNN là mạng có các kết nối phản hồi [33]. Một cách khác để nghĩ về RNN là chúng chứa một bộ nhớ lưu trữ thông tin về tính toán các trạng thái trước đó. Công việc của chúng tôi đối mới hai mô hình liên quan đến RNN: GRU và LSTM [bài báo gốc].

GRU tương tự như LSTM với một cổng quên; tuy nhiên, nó có độ phức tạp thấp hơn và ít tham số hơn. LSTM được tạo ra để giải quyết vấn đề gradient biến mất có thể xảy ra khi cố gắng huấn luyện RNN truyền thống.

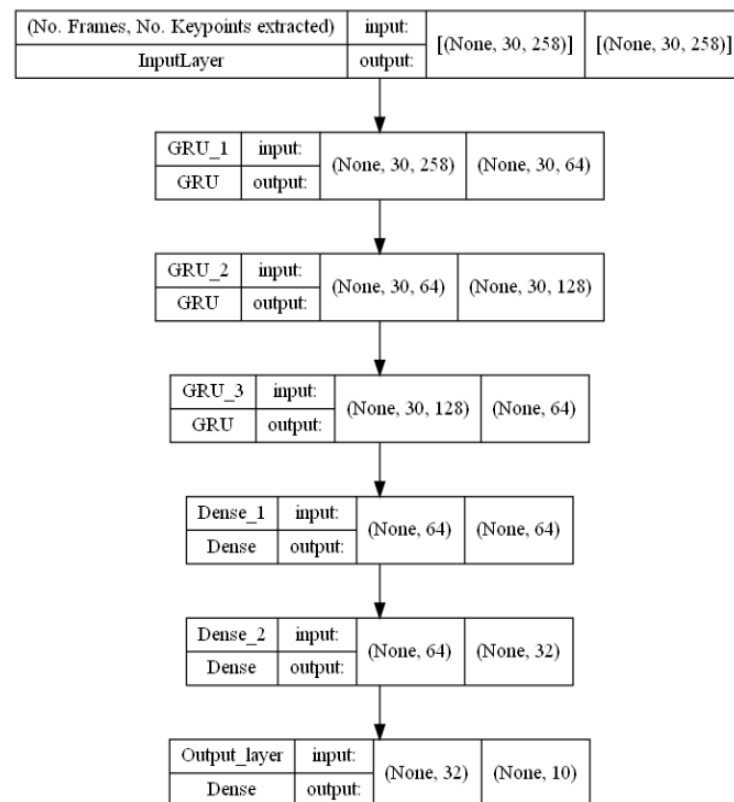
Đầu ra của các mô hình này dựa trên chuỗi đầu vào, cải thiện khả năng phát hiện chuyển động của DSL. Các hình sau minh họa tóm tắt và cấu trúc cho mỗi mô hình trong công việc này.

Cấu trúc của các mô hình được thể hiện trong Hình 5-7. Ba lớp đầu tiên thuộc về mô hình RNN trong khi ba lớp cuối cùng là các lớp dày đặc. Sau đó, các lớp được biên dịch bằng cách lựa chọn giá trị tối ưu nhất của tham số tối ưu hóa [4] như được thể hiện trong Bảng 1. Khi sử dụng mỗi mô hình, giá trị của các tham số của lớp có

thể được điều chỉnh bằng cách chọn bất kỳ giá trị nào từ Bảng 1 để chuẩn bị cho giai đoạn huấn luyện.



Hình 2.9 Cấu trúc mô hình LSTM



Hình 2.10 Cấu trúc mô hình GRU

Các đầu vào cho các mô hình là độ dài chuỗi và tổng số điểm chính. Độ dài chuỗi là số khung hình có trong mỗi đoạn. Tổng số điểm chính là 258 nếu bao gồm phần chân hoặc 226 nếu không bao gồm phần chân.

Parameters	Value
RNN Model	GRU, LSTM
Number of Nodes	Between (64,256)
Activation	'Relu' or 'Softmax'
Optimizer	'Adam'

Bảng 2.2 Mô hình đề xuất

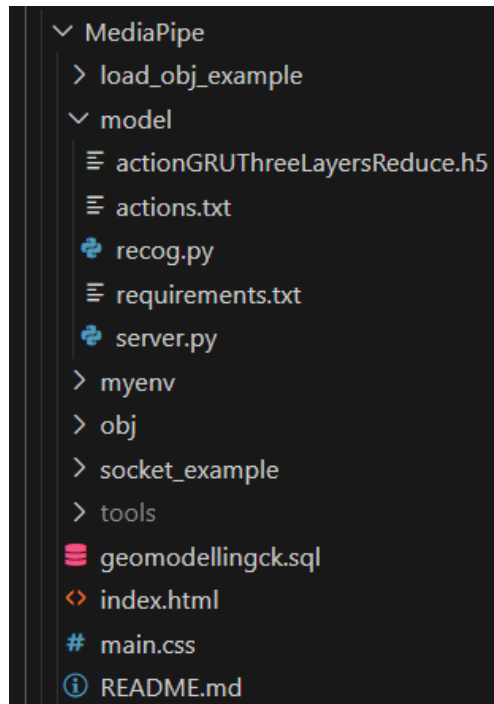
Tham khảo kết quả trên bài báo gốc của các mô hình được đề xuất trên có thể thấy được kết quả trên tập huấn luyện và kiểm thử của mô hình GRU là khả quan hơn so với mô hình LSTM. Do đó chúng tôi dự đoán độ phức tạp của model nhận diện ngôn ngữ ký hiệu có thể được giảm lại. Theo đó, chúng tôi thực hiện giảm số lớp RNN từ 3 xuống còn 2 lớp và thực hiện các thử nghiệm.

2.5.3 Tổ chức chương trình

a. Tổ chức thư mục

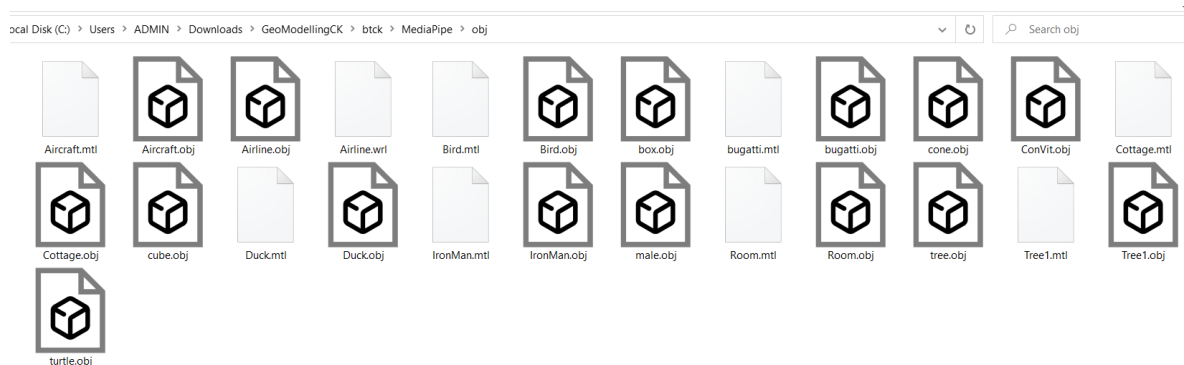
Cấu trúc thư mục của dự án được tổ chức một cách cẩn thận để dễ dàng quản lý và phát triển. Các thư mục chính có thể bao gồm:

- **MediaPipe:** Thư mục chứa mã nguồn của toàn bộ dự án, bao gồm code python và code html js chính để xây dựng giao diện.
- **Folder model:** Gồm file recog.py để chứa code nhận dạng hành động và file Server.py để thực hiện chức năng socket gửi hành động nhận dạng về cho client.
- **File index.html:** Chứa code để xây dựng giao diện, trong đó có chứa code js thực hiện các chức năng load đối tượng, điều khiển đối tượng và nhận dữ liệu từ socket trả về.





Hình 2.11. Tổ chức mã nguồn của hệ thống

b. Các tập tin OBJ liên quan đến đối tượng 3D



Hình 2.12. Các tập tin OBJ liên quan đến đối tượng 3D

c. Các tập tin liên quan đến database và diagram

 ckgeomodeling.mwb	5/14/2024 1:15 AM	MySQL Workbench...	6 KB
 geomodellingck.sql	5/12/2024 5:16 PM	SQL Source File	2 KB

Hình 2.13. Các tập tin liên quan đến database và diagram

2.6 KẾT CHƯƠng

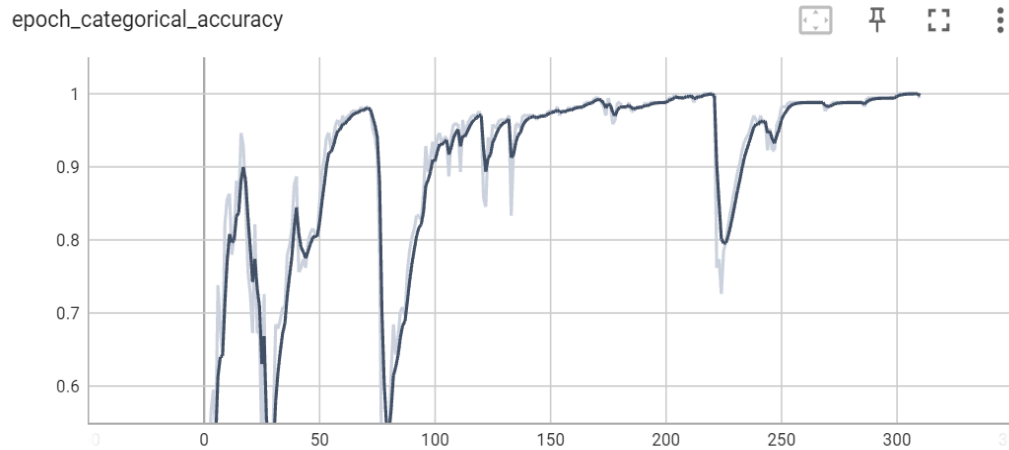
Chương này đã trình bày chi tiết về phân tích chức năng, thiết kế cơ sở dữ liệu và tổ chức chương trình của hệ thống WebGL về hệ thống điều khiển đối tượng bằng nhận diện cử chỉ tay. Các nội dung đã được trình bày nhằm mục đích giúp hiểu rõ hơn về cách triển khai dự án và quản lý mã nguồn.

Chương 3. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

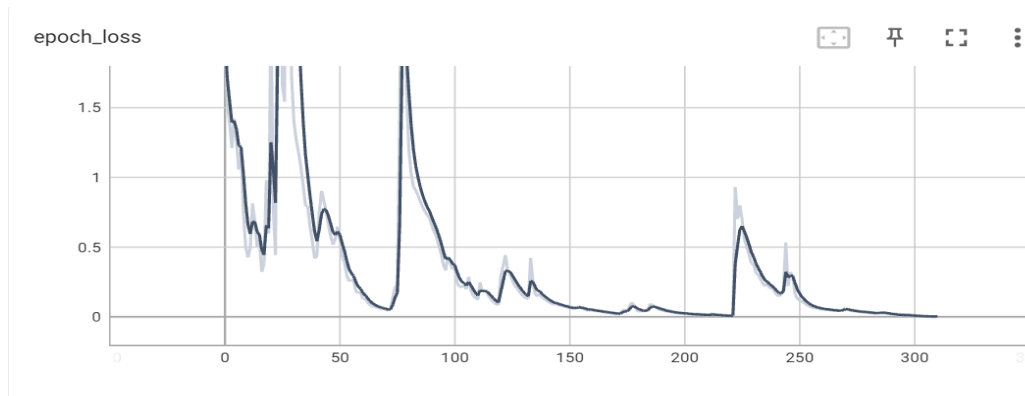
3.1 KẾT QUẢ TRÊN TẬP KIỂM THỬ

3.1.1 Kết quả huấn luyện

Hầu hết các thử nghiệm trên tập train của các mô hình đều cho thấy hình dạng loss và accuracy là như nhau, các đồ thị sau minh họa quá trình huấn luyện.



Hình 3.1 Đồ thị accuracy khi huấn luyện trên tập train



Hình 3.2 Đồ thị loss function khi huấn luyện trên tập train

3.1.2 Kết quả kiểm thử

Bảng thống kê kết quả chạy trên các mô hình:

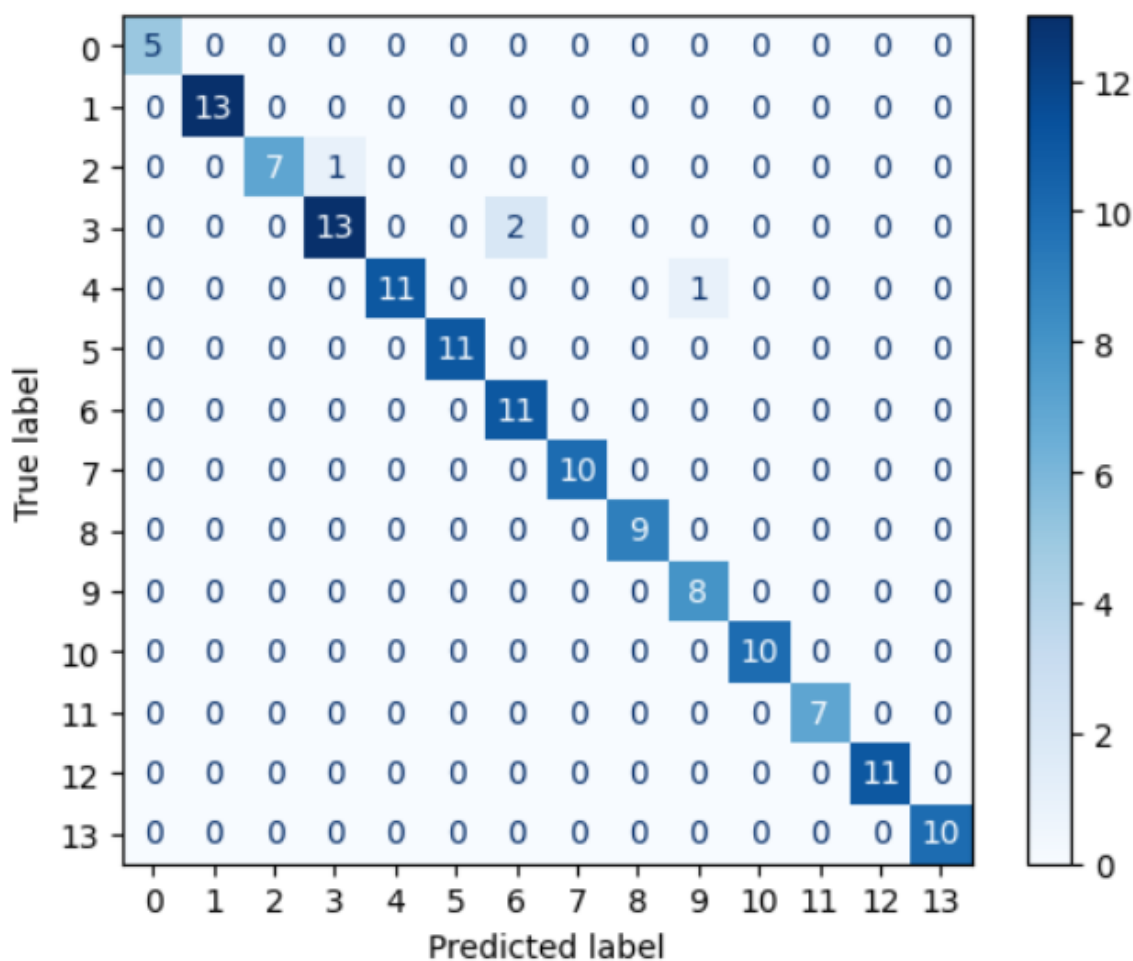
		Không tiền xử lí	Tiền xử lí
GRU	3 lớp	98.67%	95.20%
	2 lớp	98.13%	97.07%
LSTM	3 lớp	96.53%	93.07%
	2 lớp	98.40%	96.27%

Bảng 3.1 Bảng thống kê kết quả chạy trên các mô hình

GRU		LSTM	
3 lớp	2 lớp	3 lớp	2 lớp
0.2124	0.1838	0.2914	0.2712

Bảng 3.2 Bảng thống kê thời gian dự đoán trên tập kiểm thử (140 video)

Ma trận nhầm lẫn trên 14 loại như sau:



Hình 3.3 Ma trận nhầm lẫn cho 14 loại cử chỉ tay

3.1.3 Bàn luận

Kết quả đạt được tốt nhất là 98.67% đối với mô hình GRU (kiến trúc gồm 3 lớp GRU) và không thực hiện tiền xử lý dữ liệu.

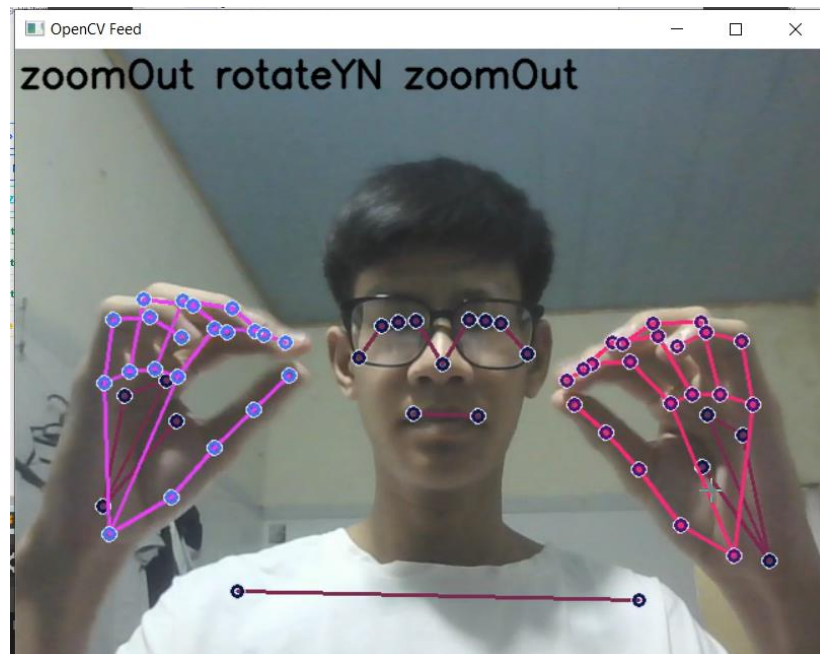
So sánh việc thực hiện giảm số lượng lớp của kiến trúc LSTM và kiến trúc GRU:

- Đối với GRU thì hiệu suất giảm nhẹ từ 98.67% xuống còn 98.13%.
- Đối với LSTM thì hiệu suất lại tăng mạnh từ 96.53% lên 98.40%

Việc giảm số lớp đi chính là việc giảm độ phức tạp của kiến trúc được sử dụng. Với kết quả thu được, LSTM có kiến trúc phức tạp hơn GRU khi giảm đi số lớp thì hiệu suất cũng tăng mạnh. Có thể trong bài báo gốc dự đoán sai về độ phức tạp của ngôn ngữ ký hiệu. Nhiều kết quả sử dụng SVM để nhận diện ngôn ngữ ký hiệu đạt hiệu suất rất tốt là mô hình có độ phức tạp thấp hơn so với nhận diện bằng luồng hành động. Vì vậy, trong các nghiên cứu tiếp theo có thể thực hiện đánh giá về độ phức tạp cần thiết để nhận diện một ngôn ngữ kí hiệu.

3.2 KẾT QUẢ THỰC NGHIỆM

3.2.1 Nhận diện hành động



Hình 3.4 Nhận diện hành động

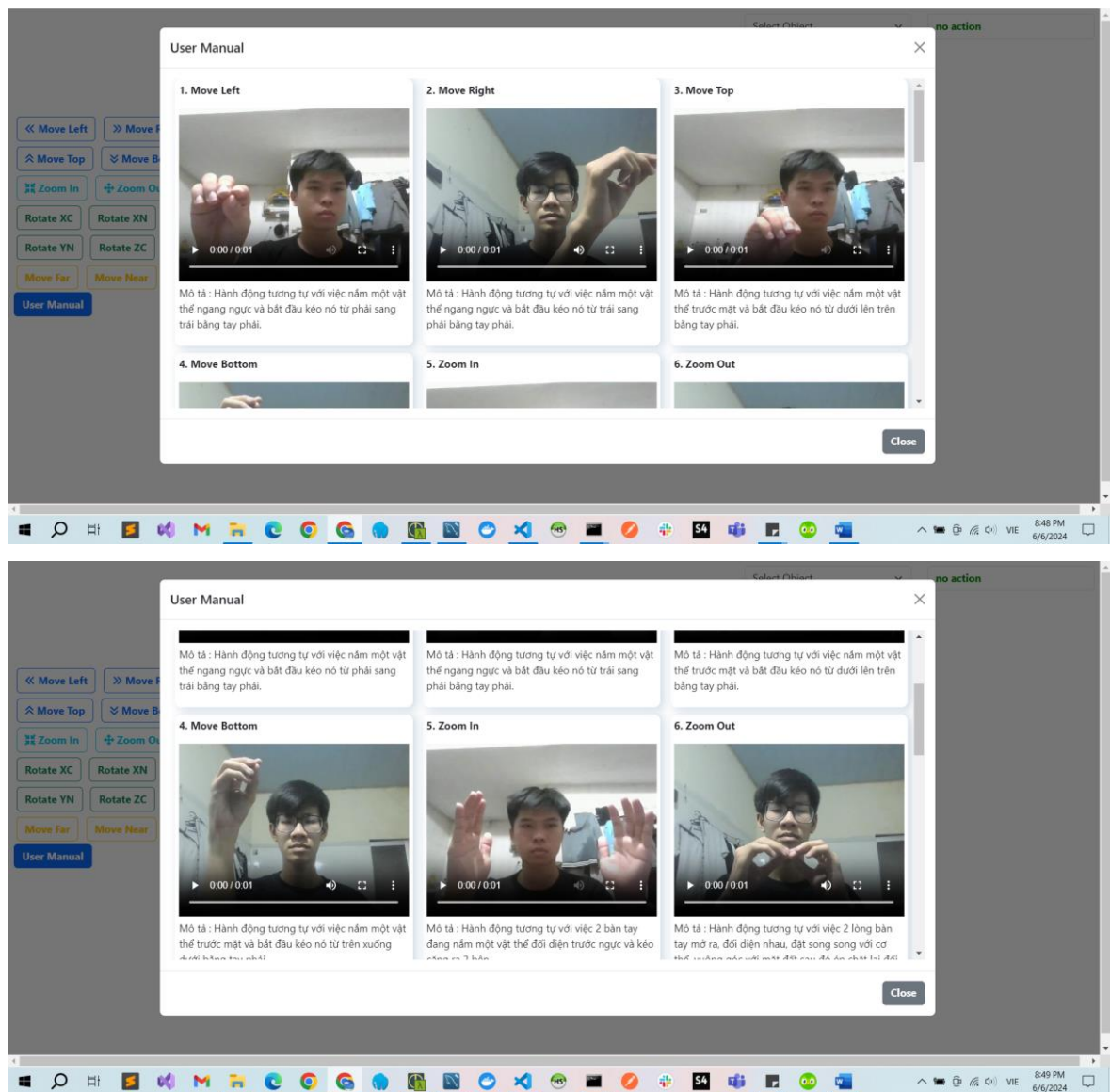
3.2.2 Dữ liệu hành động được lưu vào database

geomodellingck.actions: 11 rows total (approximately)

id		name
100		zoomOut
101		zoomOut
102		zoomOut
103		moveLeft
104		moveLeft
105		zoomOut
106		rotateYN
107		zoomOut
108		zoomOut
109		zoomOut
110		zoomOut

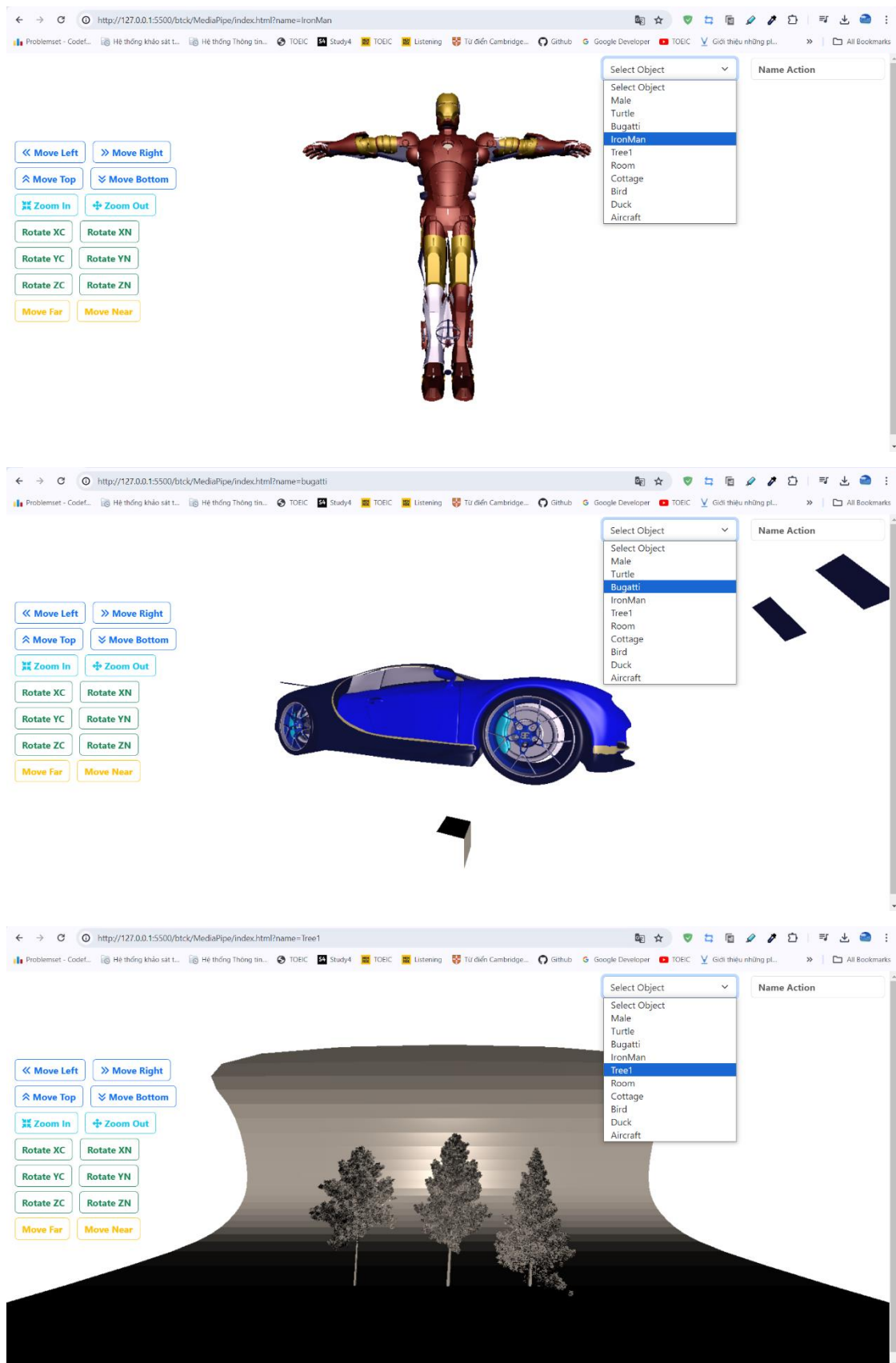
Hình 3.5 Dữ liệu hành động được lưu vào database

3.2.3 Chức năng hướng dẫn sử dụng



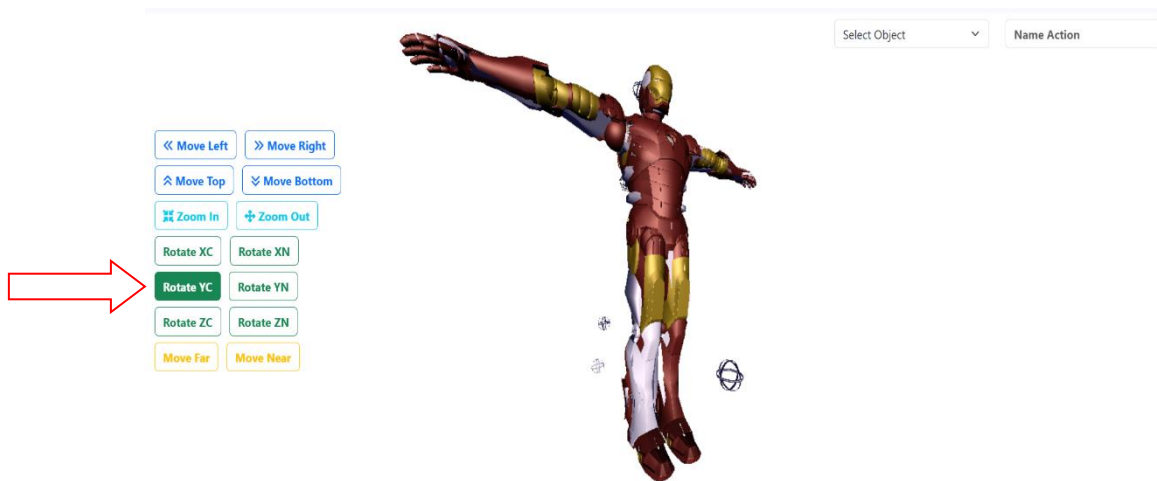
Hình 3.6 Chức năng hướng dẫn sử dụng

3.2.4 Chức năng lựa chọn đối tượng



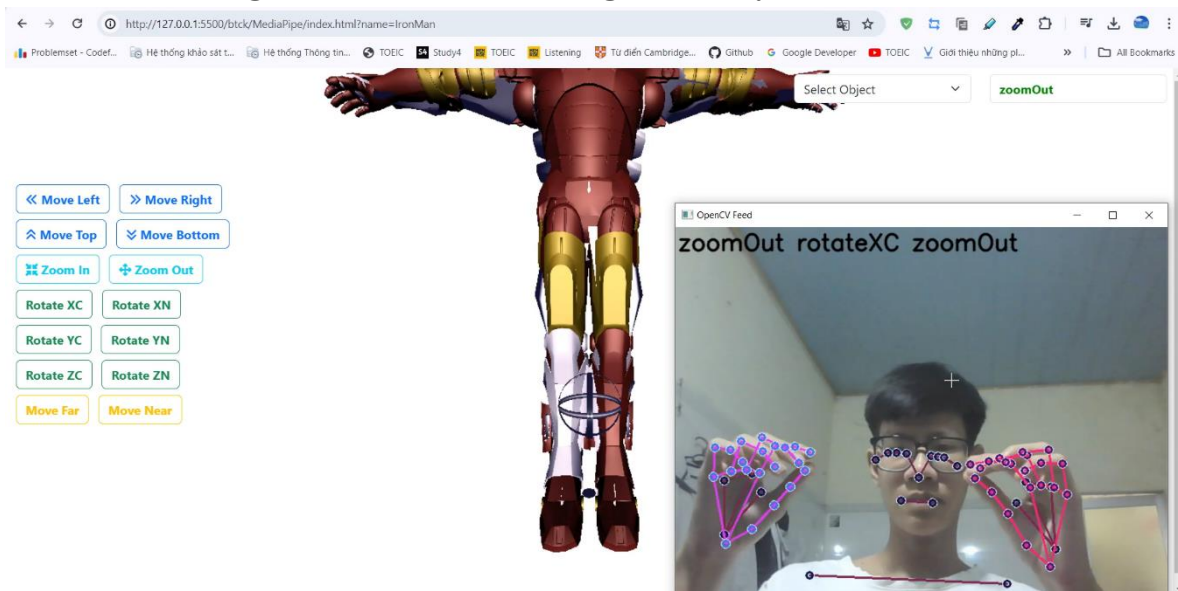
Hình 3.7 Chức năng lựa chọn đối tượng

3.2.5 Chức năng điều khiển đối tượng bằng nút nhấn



Hình 3.8 Chức năng điều khiển vật thể bằng nút nhấn

3.2.6 Chức năng điều khiển vật thể bằng cử chỉ tay



Hình 3.9 Chức năng điều khiển vật thể bằng cử chỉ tay

3.3 NHẬN XÉT ĐÁNH GIÁ KẾT QUẢ

3.3.1 Chức năng socket

- Có thể sau khi nhận diện hành động xong thì gửi về cho client js luôn cũng được hoặc để rõ ràng thì tác ra 2 luồng. Một luồng cho nhận diện hành động và 1 luồng cho socket gửi hành động.
- Khi 2 luồng cùng mở 1 file.txt thì sẽ gây ra lỗi. Chính vì thế cần dùng đến database để lưu dữ liệu hành động thay vì lưu vào file.

3.3.2 Load màu các đối tượng 3D

- Nhiều đối tượng vẫn chưa có màu sắc, cần khai thác các file OBJ của các đối tượng mà có màu sắc để cho nó đa dạng hơn. Nghiên cứu cách tự tô màu cho các đối tượng 3D.

3.4 KẾT CHƯƠng

Chương này trình bày những kết luận và hướng phát triển tiếp theo dựa trên những kết quả thực nghiệm và nhận xét đánh giá. Các điểm nổi bật và hạn chế của hệ thống sẽ được tổng kết và đề xuất các giải pháp cải thiện. Đồng thời, chương cũng có thể đề cập đến sự quan trọng của dự án, ứng dụng thực tiễn và tiềm năng phát triển trong tương lai.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

A. KẾT QUẢ ĐẠT ĐƯỢC

Trong quá trình tìm hiểu, nghiên cứu cơ sở lý thuyết và triển khai ứng dụng công nghệ, đồ án đã đạt được những kết quả sau:

Về mặt lý thuyết, đồ án đã đạt được sự hiểu biết sâu sắc về các khái niệm trong lý thuyết mô hình 3D, WebGL, Threejs, Socket và AI. Đồ án đã phân tích và trình bày một cách rõ ràng về cách thức hoạt động của các công nghệ này.

Về mặt thực tiễn ứng dụng, đồ án đã triển khai thành công một hệ thống WebGL để nhận diện hành động cử chỉ tay để điều khiển đối tượng. Hệ thống này cho phép người dùng thao tác điều khiển và lựa chọn đối tượng một cách dễ dàng và nhanh chóng.

Kết quả đóng góp của đồ án được thể hiện như sau:

- Huấn luyện và xây dựng được model AI nhận diện hành động cử chỉ tay, góp phần vào chức năng điều khiển vật thể 3D
- Xây dựng được một giao diện người dùng thân thiện và dễ sử dụng, giúp người dùng thao tác dễ dàng hơn trong việc điều khiển và lựa chọn các đối tượng 3D.
- Đề xuất phương pháp sử dụng socket để truyền dữ liệu trong thời gian thực từ server về client.

Tuy nhiên, vẫn còn tồn tại các vấn đề như sau:

- Vấn đề về tốc độ tải các đối tượng 3D có kích thước lớn chưa nhanh
- Cải thiện độ chính xác của model AI nhận diện hành động
- Đề xuất các tính năng mở rộng và cải thiện trải nghiệm người dùng để nâng cao giá trị của hệ thống.

B. KIẾN NGHỊ VÀ HƯỚNG PHÁT TRIỂN

Một số hướng nghiên cứu và phát triển của đề tài như sau:

- Tối ưu hóa thuật toán nhận dạng, điều chỉnh lại mô hình và thuật toán để có thể nhận diện các hành động một cách chính xác và phức tạp.
- Tối ưu giao diện web thân thiện hơn với người dùng, tăng tính mượt mà khi điều khiển vật thể.
- Xây dựng ứng dụng mobile app để có thể điều khiển vật thể trên điện thoại.
- Ứng dụng vào giáo dục phổ thông liên quan đến các bài học về các khối 3D, ngoài ra có thể ứng dụng vào y tế và công nghiệp.
- Phát triển thêm về bảo mật và quyền riêng tư cho người dùng.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Huu Loc Nguyen (2022), "Giáo trình mô hình hóa hình học", Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh.

[2] PGS.TS Nguyen Tan Khoi (2024), "Slide WebGL", Đại học Bách Khoa Đà Nẵng.

Tiếng Anh

[1] Parisi, T. (2014). "Learning Three.js: The JavaScript 3D Library for WebGL". O'Reilly Media.

[2] Williams, J. (2018). "WebGL Insights". CRC Press.

[3] Cabello, R. (2017). "Learning Three.js: A JavaScript 3D Library for WebGL". Packt Publishing.

[4] Dirksen, J. (2016). "Learning Three.js: The JavaScript 3D Library for WebGL". Packt Publishing.

[5] Dirksen, J. (2018). "Discover Three.js: An Interactive Journey Through 3D Programming with WebGL". Apress.

Internet

[1] Trang web chính thức của Three.js: <https://threejs.org/>

[2] Trang web của WebGL: <https://www.khronos.org/webgl/>

[3] Trang web chính thức của Socket.io: <https://socket.io/>

[4] Trang web của TensorFlow.js <https://www.tensorflow.org/js>

[5] Trang web chính thức của OpenAI: <https://openai.com/>

PHỤ LỤC

1. Nội dung mã nguồn của hệ thống

<https://github.com/NguyenVanManh-AI/GeoModellingCK>