



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



**BÁO CÁO
PBL5 - ĐỒ ÁN KỸ THUẬT MÁY TÍNH**

CHUYỂN ĐỔI NGÔN NGỮ KÝ HIỆU TAY THÀNH VĂN BẢN

Giảng viên đồng hướng dẫn: TS. Ninh Khánh Duy

STT NHÓM: 28 HỌ VÀ TÊN SINH VIÊN	LỚP HỌC PHẦN ĐỒ ÁN
Trương Bích Quỳnh	20.12A
Nguyễn Thị Thanh Hiền	20.12A
Nguyễn Phước Nhâm	20.12A

ĐÀ NẴNG, 06/2023

TÓM TẮT ĐỒ ÁN

Đồ án nhận diện ngôn ngữ ký hiệu tập trung vào việc giải quyết vấn đề nhận dạng hành động riêng lẻ trong ngôn ngữ ký hiệu. Phương pháp được áp dụng bao gồm việc sử dụng module esp32cam để ghi hình ảnh từ camera, sau đó gửi lên server để xử lý. Trên server, chúng em sử dụng mô hình LSTM kết hợp với thư viện media pipe để nhận dạng hành động từ các hình ảnh đã gửi lên. Kết quả đạt được là hệ thống có khả năng nhận dạng những hành động riêng lẻ trong ngôn ngữ kí hiệu. Nhờ sử dụng mô hình LSTM và media pipe, hệ thống có khả năng phân loại và nhận diện các hành động đơn lẻ như "Hello", "Bye", "love" và nhiều hành động khác. Điều này mang lại tiềm năng ứng dụng rộng rãi trong việc tương tác và giao tiếp với người dùng thông qua ngôn ngữ kí hiệu, đồng thời đảm bảo tính chính xác và đáng tin cậy của quá trình nhận dạng.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá theo 3 mức (Đã hoàn thành/Chưa hoàn thành/Không triển khai)
Nguyễn Phước Nhâm	<ul style="list-style-type: none"> -Tìm hiểu và chuẩn bị phần cứng. -Trích xuất đặc trưng. -Xây dựng mô hình nhận dạng. -Xây dựng model nhận dạng. -Viết báo cáo. 	<ul style="list-style-type: none"> -Hoàn thành -Hoàn thành -Hoàn thành -Hoàn thành -Hoàn thành
Nguyễn Thị Thanh Hiền	<ul style="list-style-type: none"> -Thu thập dữ liệu -Web FE -Sử dụng ESP cam gửi video lên server -Viết báo cáo 	<ul style="list-style-type: none"> -Hoàn thành -Hoàn thành -Hoàn thành -Hoàn thành
Trương Bích Quỳnh	<ul style="list-style-type: none"> -Thu thập dữ liệu -Triển khai API vào hệ thống -Sử dụng ESP cam gửi video lên server -Viết báo cáo 	<ul style="list-style-type: none"> -Hoàn thành -Hoàn thành -Hoàn thành -Hoàn thành

Bảng 1. Bảng phân công nhiệm vụ

MỤC LỤC

1. Giới thiệu.....	6
1.1. Thực trạng.....	6
1.2. Các vấn đề cần giải quyết.....	6
1.3. Đề xuất giải pháp.....	6
2. Giải pháp.....	7
2.1. Giải pháp về phần cứng và truyền thông.....	7
2.1.1. Giải pháp phần cứng.....	7
2.1.2. Giải pháp truyền thông.....	7
2.2. Giải pháp mô hình.....	9
2.2.1 Model ConvLSTM (Model tham khảo).....	9
2.2.2 Model LSTM - Long Short-Term Memory (Model đề xuất).....	10
2.2.3 Mediapipe.....	13
3. Kết quả.....	14
3.1. Dữ liệu.....	14
3.1.1. Thu thập dữ liệu lần 1.....	14
3.1.2. Thu thập dữ liệu lần 2.....	14
3.1.3. Làm giàu dữ liệu.....	16
3.2. Trích xuất đặc trưng.....	17
3.2.1. Thử nghiệm lần 1.....	17
3.2.2. Thử nghiệm lần 2.....	18
3.3 Kết quả huấn luyện.....	19
3.3.1. Huấn luyện lần 1 sử dụng model ConvLSTM.....	20
3.3.2. Huấn luyện lần 2 sử dụng model LSTM.....	22
3.3.3. Huấn luyện lần 3 sử dụng model LSTM.....	25
3.4 Giao diện người dùng.....	28
4. Kết luận.....	28
4.1 Đánh giá.....	28
4.2 Hướng phát triển.....	28
5. Danh mục tài liệu tham khảo.....	29

MỤC LỤC HÌNH ẢNH

Hình 2.1. Sơ đồ truyền thông.....	8
Hình 2.2. Sơ đồ xử lý video và trả về kết quả.....	8
Hình 2.3. Chuyển đổi hình ảnh 2D thành 3D tensor.....	9
Hình 2.4. Cấu trúc bên trong của ConvLSTM.....	9
Hình 2.5. Cấu trúc mạng nơ-ron tái phát LSTM.....	10
Hình 2.6. Giải thích model.....	12
Hình 2.7. Các mốc mediapipe trên bàn tay.....	13
Hình 3.1. Số lượng video mỗi hành động lần 1.....	14
Hình 3.2. Số lượng video mỗi hành động lần 2.....	15
Hình 3.3. Làm giàu dữ liệu.....	16
Hình 3.4. Số lượng video sau khi làm giàu dữ liệu.....	17
Hình 3.5. Video trích xuất lần 1.....	18
Hình 3.6. Sơ đồ trích xuất đặc trưng.....	19
Hình 3.7. Phân chia dữ liệu.....	20
Hình 3.8. Độ chính xác của tập train và test huấn luyện lần 1.....	21
Hình 3.9. Mất mát train tập train và test huấn luyện lần 1.....	21
Hình 3.10. Độ chính xác của tập train và test huấn luyện lần 2.....	23
Hình 3.11. Mất mát train tập train và test huấn luyện lần 2.....	23
Hình 3.12. Ma trận nhầm lẫn trên tập test lần 1.....	24
Hình 3.14. Độ chính xác của tập train và test huấn luyện lần 3.....	26
Hình 3.15. Mất mát train tập train và test huấn luyện lần 3.....	26
Hình 3.16. Ma trận nhầm lẫn trên tập test của mô hình huấn luyện lần 3.....	27
Hình 3.18. Giao diện người dùng.....	28

MỤC LỤC BẢNG

Bảng 1. Bảng phân công nhiệm vụ.....	3
Bảng 2. Đề xuất giải pháp.....	7
Bảng 3. Danh sách các phần cứng sử dụng.....	7

1. Giới thiệu

1.1. Thực trạng

Ngôn ngữ ký hiệu tay đã trở thành một phương pháp giao tiếp quan trọng và phổ biến trong cộng đồng người khiếm thính. Tuy nhiên, hiện tại, các hệ thống nhận dạng ngôn ngữ ký hiệu tay vẫn chưa đáp ứng đầy đủ nhu cầu và tiện ích của người sử dụng. Vì vậy, nhóm chúng em đã đề xuất và phát triển một hệ thống nhận dạng ngôn ngữ ký hiệu tay sử dụng camera, nhằm giúp người khiếm thính giao tiếp một cách tự nhiên và thuận tiện hơn. Chúng em tập trung vào việc nhận dạng và chuyển đổi các cử chỉ ngôn ngữ ký hiệu tay thành văn bản, nhằm mang lại sự linh hoạt và tiện lợi cho người sử dụng.

1.2. Các vấn đề cần giải quyết

- Phần cứng để ghi video.
- Model để nhận dạng hình ảnh.
- Server xử lý video và trả về kết quả.
- Giao diện hiển thị video và kết quả.

1.3. Đề xuất giải pháp

Vấn đề	Giải pháp đề xuất
Phần cứng	- ESP32Cam
Dữ liệu	- Sử dụng esp32cam để thu thập và lưu vào thẻ nhớ. - Sử dụng camera của laptop
Model nhận dạng	- Sử dụng mô hình LSTM kết hợp với thư viện Media Pipe.
Server xử lý video và trả về kết quả	- Framework Flask
Giao diện hiển thị video và kết quả	- HTML, CSS, javascript


Bảng 2. Đề xuất giải pháp

2. Giải pháp

2.1. Giải pháp về phần cứng và truyền thông

2.1.1. Giải pháp phần cứng

Hệ thống sử dụng 1 ESP32CAM để ghi lại hình ảnh, sau đó stream lên giao diện web local. Trên giao diện có các nút bấm để bắt đầu ghi hình và gửi video lên server. Server xử lý video, dự đoán và trả kết quả về web, hiển thị kết quả ra web.

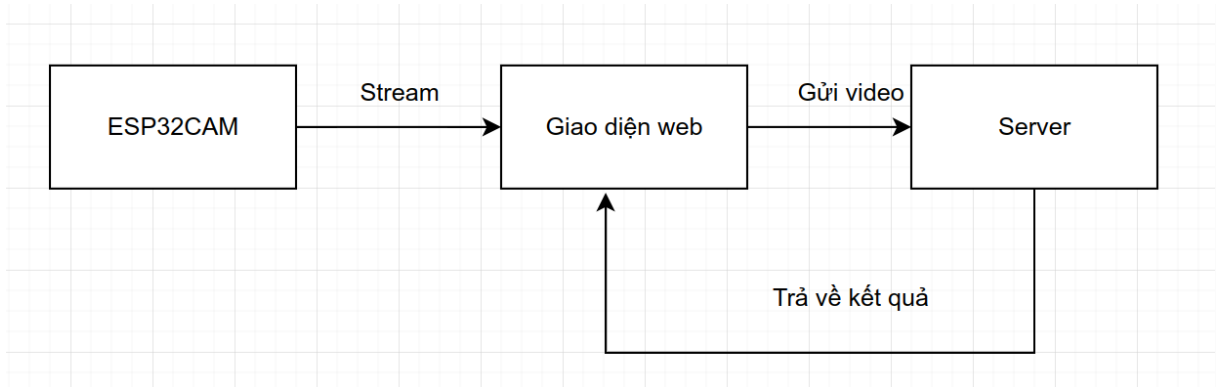
Tên linh kiện	Hình ảnh	Thông số
ESP32Cam		<ul style="list-style-type: none">-Giao diện hỗ trợ: UART, SPI, I2C, PWM- Wifi: 802.11 b/g/n/e/i- Cổng IO: 9- Định dạng đầu ra hình ảnh: JPEG (chỉ được hỗ trợ bởi OV2640), BMP

Bảng 3. Danh sách các phần cứng sử dụng

2.1.2. Giải pháp truyền thông

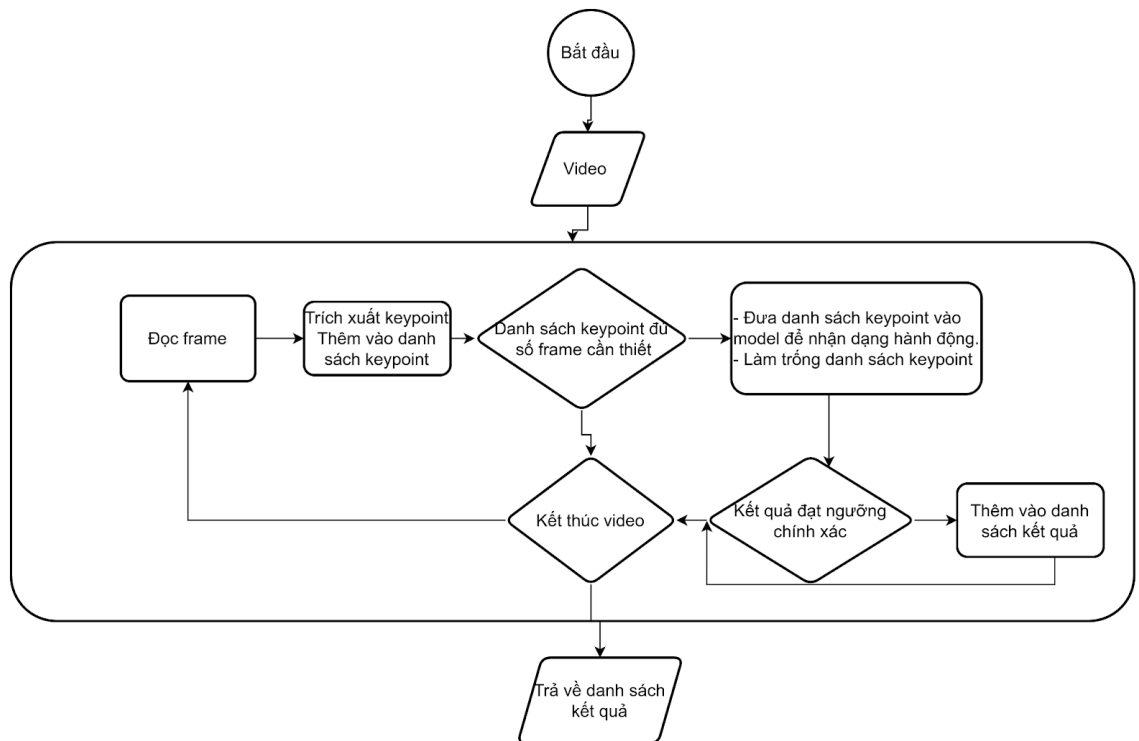
Web FE:

- Sử dụng thư viện DNS server để tạo một DNS server, cho phép client sử dụng tên miền là địa chỉ IP của WiFi để kết nối đến ESP32. Khi ESP32 kết nối đến WiFi, nó sẽ được cấp một địa chỉ IP, và đó sẽ là địa chỉ IP mà DNS server sẽ sử dụng để tạo tên miền, cho phép client sử dụng tên miền là địa chỉ ip của wifi để kết nối đến ESP32.
- Tạo trang HTML để gửi yêu cầu http đến server của ESP32.
- Khi người dùng click record/stop trên Web FE, sẽ gửi yêu cầu HTTP (POST request) đến ESP32, được xử lý bởi các hàm handle. Sau khi ghi xong, ESP32 sẽ gửi file video lên DNS server với URL là ip/stop. Trang Web FE sẽ lấy file video từ DNS server và gửi lên API, sau đó nhận kết quả về và hiển thị lên trang.



Hình 2.1. Sơ đồ truyền thông

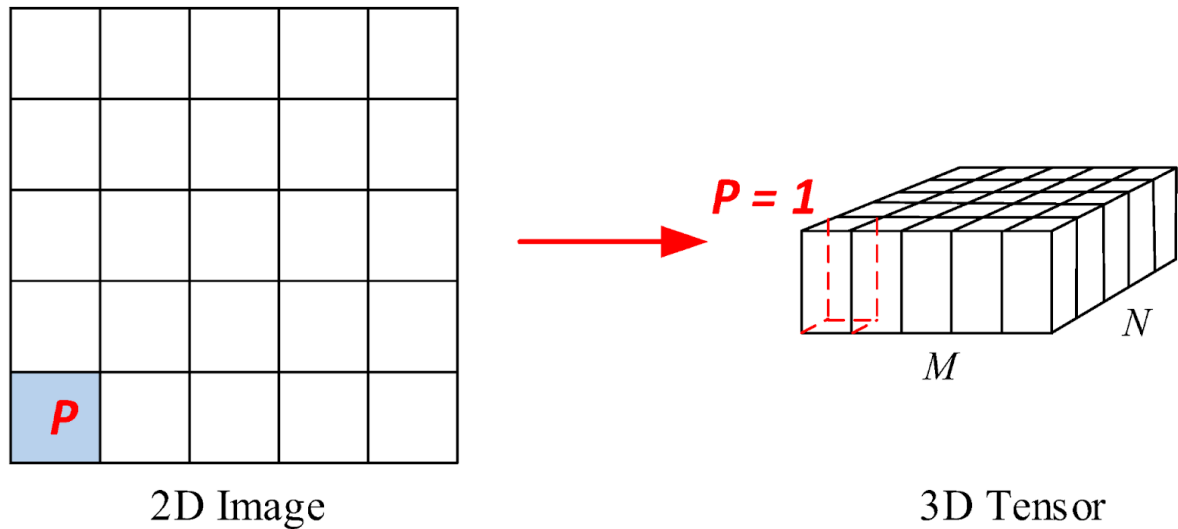
Server API nhận đầu vào là 1 video và kết quả là chuỗi các từ ngữ tương ứng.



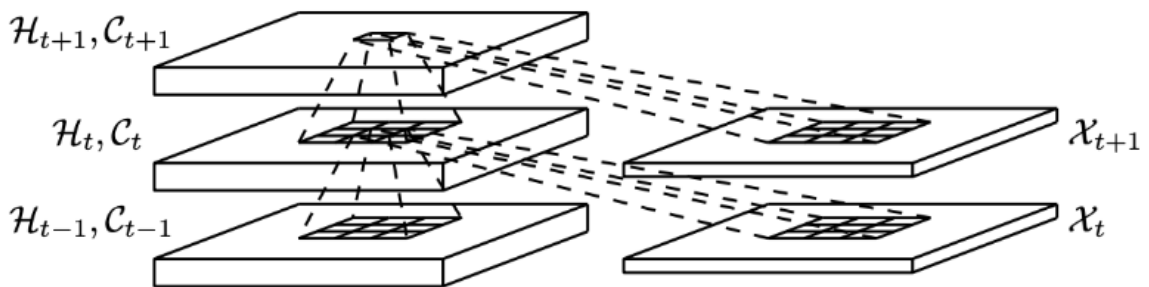
Hình 2.2. Sơ đồ xử lý video và trả về kết quả

2.2. Giải pháp mô hình

2.2.1 Model ConvLSTM (Model tham khảo)



Hình 2.3. Chuyển đổi hình ảnh 2D thành 3D tensor



Hình 2.4. Cấu trúc bên trong của ConvLSTM

- Mạng nơ-ron ConvLSTM (Convolutional LSTM) là một biến thể của mạng nơ-ron tái phát (LSTM) kết hợp với các lớp tích chập. Nó được sử dụng phổ biến trong các ứng dụng liên quan đến dữ liệu dạng chuỗi hoặc dữ liệu không gian, chẳng hạn như nhận dạng ngôn ngữ kí hiệu, xử lý video, dự báo thời tiết dựa trên chuỗi thời gian, v.v.
- ConvLSTM kết hợp cả khả năng học đặc trưng không gian từ các lớp tích chập và khả năng mô hình hóa thông tin dài hạn từ mạng nơ-ron tái phát LSTM. Nó giúp giải quyết vấn đề trong việc nhận dạng và dự báo trạng thái tiếp theo trong dữ liệu không gian hoặc dữ liệu dạng chuỗi.
- Dưới đây là cách ConvLSTM được áp dụng trong hệ thống nhận dạng ngôn ngữ kí hiệu:
 - + Chuẩn bị dữ liệu:

- Dữ liệu ngôn ngữ kí hiệu được chuyển đổi thành chuỗi các khung hình (frames) hoặc các khối hình ảnh.
- Mỗi frame hoặc khối hình ảnh được biểu diễn dưới dạng ma trận hình ảnh với các giá trị pixel.

+ Xây dựng mô hình ConvLSTM:

- Mô hình ConvLSTM bao gồm các lớp tích chập và lớp LSTM được xếp chồng lên nhau.
- Lớp tích chập được sử dụng để học đặc trưng không gian từ dữ liệu hình ảnh.
- Lớp LSTM giúp mô hình hóa thông tin dài hạn và ứng phó với mối quan hệ thời gian trong dữ liệu chuỗi hình ảnh.

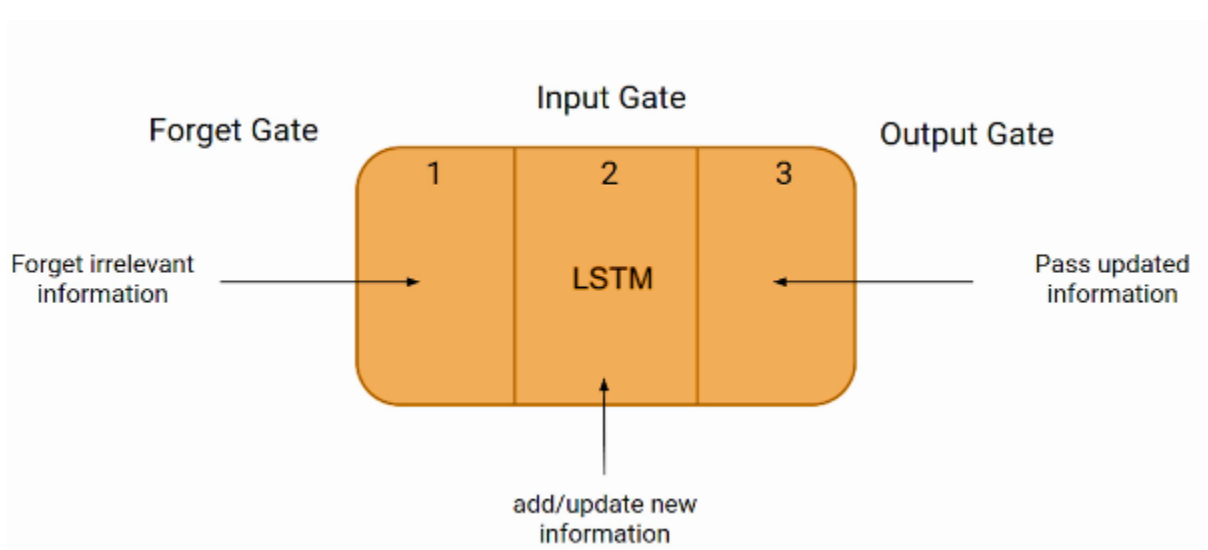
+ Đào tạo mô hình:

- Mô hình ConvLSTM được đào tạo bằng cách sử dụng các cặp dữ liệu đầu vào và đầu ra đã được gán nhãn.
- Quá trình đào tạo tối ưu hóa các trọng số của mô hình để dự đoán đầu ra chính xác cho mỗi khung hình hoặc khối hình ảnh.

+ Dự đoán và nhận dạng:

- Sau khi mô hình được đào tạo, nó có thể được sử dụng để dự đoán và nhận dạng các hành động, từ ngôn ngữ kí hiệu trong dữ liệu mới.
- Đầu vào mới được truyền qua mô hình ConvLSTM để dự đoán trạng thái tiếp theo hoặc các hành động tương ứng.

2.2.2 Model LSTM - Long Short-Term Memory (Model đề xuất)



Hình 2.5. Cấu trúc mạng nơ-ron tái phát LSTM

- Hệ thống nhận dạng ngôn ngữ kí hiệu là một ứng dụng trong lĩnh vực xử lý hình ảnh và công nghệ thị giác máy tính, nhằm nhận dạng và hiểu nghĩa của các biểu hiện ngôn ngữ ký hiệu dùng trong giao tiếp với người khiếm thính. Trong hệ thống này, Mạng

neuron tái phát (LSTM) có thể được áp dụng để mô hình hóa và phân loại các ngôn ngữ ký hiệu.

- LSTM là một biến thể của mạng neuron hồi quy (RNN), được thiết kế để giải quyết vấn đề gradient vanishing và gradient exploding trong quá trình huấn luyện mạng neuron. LSTM có khả năng lưu trữ và truyền thông tin trong thời gian dài, giúp nắm bắt các mối quan hệ thời gian trong dữ liệu chuỗi.

- Trong hệ thống nhận dạng ngôn ngữ ký hiệu, quá trình áp dụng LSTM có thể được mô tả như sau:

- + Tiền xử lý dữ liệu: Ảnh chứa ngôn ngữ ký hiệu được đưa vào hệ thống. Trước khi đưa vào mạng LSTM, ảnh có thể được chuyển đổi sang định dạng phù hợp và được chuẩn hóa.

- + Biểu diễn ảnh: Ảnh được đưa qua một mạng neuron tích chập (CNN - Convolutional Neural Network) để trích xuất các đặc trưng hình ảnh. CNN giúp tìm ra các đặc điểm quan trọng trong ảnh và tạo ra một biểu diễn gốc của ngôn ngữ ký hiệu.

- + Chuỗi hóa dữ liệu: Biểu diễn gốc của ngôn ngữ ký hiệu từ CNN được chia thành các chuỗi (sequence) các đặc trưng. Mỗi chuỗi đặc trưng tương ứng với một phần của ngôn ngữ ký hiệu.

- + Mạng LSTM: Các chuỗi đặc trưng được đưa vào mạng LSTM. Mạng LSTM sẽ học và mô hình hóa các mối quan hệ thời gian trong chuỗi đặc trưng. LSTM sử dụng các cơ chế cổng (gates) để điều chỉnh thông tin và quyết định xem thông tin nào cần được lưu trữ và thông tin nào cần được bỏ qua.

- + Đầu ra: Sau quá trình đi qua mạng LSTM, một đầu ra được tạo ra. Đầu ra có thể là một dự đoán về ngữ cảnh hoặc ý nghĩa của ngôn ngữ ký hiệu trong ảnh, hoặc có thể là một phân loại của ngôn ngữ ký hiệu dựa trên các lớp đầu ra đã được định nghĩa trước.

Giải thích model:

```
# sequence_length: số lượng frame trong 1 video train
# X.shape[-1]: số lượng các giá trị keypoint của trích xuất được từ 1 video.
model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', kernel_initializer=initializers.he_normal(), nput_shape=(sequence_length, X.shape[-1])))
model.add(LSTM(128, return_sequences=True, activation='relu', kernel_initializer= initializers.he_normal()))
model.add(LSTM(64, return_sequences=False, activation='relu', kernel_initializer= initializers.he_normal()))
model.add(Dense(64, activation='relu', kernel_initializer=initializers.he_normal()))
model.add(Dense(32, activation='relu', kernel_initializer=initializers.he_normal()))
```

```
model.add(Dense(actions.shape[0],activation='softmax',kernel_initialize
r= initializers.he_normal()))
```

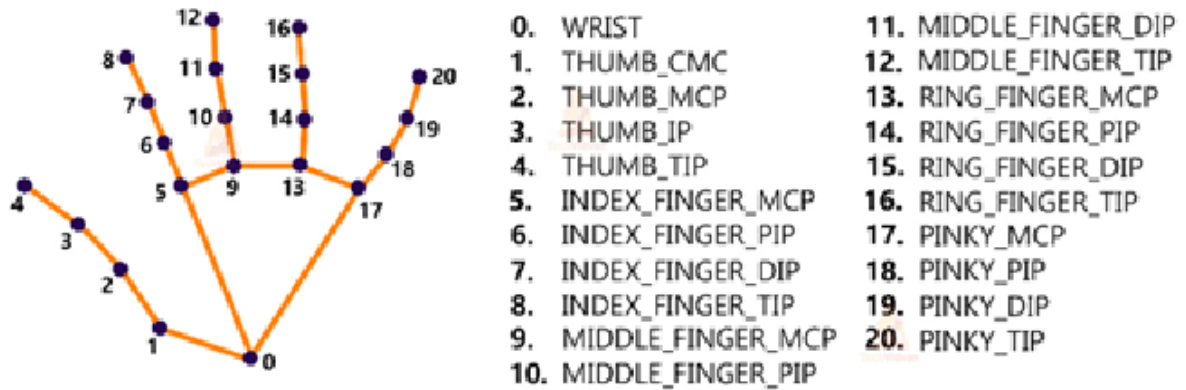
Model: "sequential_2"

Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 12, 64)	93440
lstm_4 (LSTM)	(None, 12, 128)	98816
lstm_5 (LSTM)	(None, 64)	49408
dense_2 (Dense)	(None, 64)	4160
dense_3 (Dense)	(None, 32)	2080
dense_4 (Dense)	(None, 14)	462
=====		
Total params: 248,366		
Trainable params: 248,366		
Non-trainable params: 0		

Hình 2.6. Giải thích model

- Model có input đầu vào là (12,300). Sử dụng 3 lớp LSTM và 3 lớp Dense. Số lượng các unit được điều chỉnh nhiều lần sau nhiều lần thử nghiệm và lựa chọn để đạt được kết quả huấn luyện đạt hiệu quả nhất.
- Quá trình huấn luyện sử dụng hàm tối ưu Adam với learning rate là 0.001. Learning được cài đặt giảm đi một nửa sau 10 epoch để đảm bảo quá trình huấn luyện hội tụ tốt hơn.
- Khởi tạo các giá trị mặc định cho các trọng số, để đảm bảo quá trình huấn luyện các giá trị của tham số không bị crash, không hợp lệ dẫn đến độ chính xác giảm.

2.2.3 Mediapipe



Hình 2.7. Các mốc mediapipe trên bàn tay

- Mediapipe là một framework mã nguồn mở của Google, được sử dụng rộng rãi trong việc xây dựng ứng dụng thị giác máy tính và xử lý dữ liệu đa phương tiện. Nó cung cấp các công cụ và thư viện để xây dựng các ứng dụng nhận dạng, theo dõi và phân tích dữ liệu hình ảnh và video.

- Khi áp dụng Mediapipe trong hệ thống nhận dạng ngôn ngữ kí hiệu, có thể sử dụng một số thành phần chính như sau:

+ Xử lý hình ảnh đầu vào:

- Mediapipe hỗ trợ xử lý hình ảnh và video từ nhiều nguồn đầu vào, bao gồm camera, video, hoặc các tệp hình ảnh.
- Hình ảnh đầu vào được truyền vào qua pipeline xử lý của Mediapipe để tiền xử lý và trích xuất thông tin quan trọng.

+ Mô hình nhận dạng ngôn ngữ kí hiệu:

- Trong Mediapipe, mô hình nhận dạng ngôn ngữ kí hiệu được xây dựng dựa trên mạng nơ-ron sâu (deep neural network).
- Mô hình được huấn luyện trước trên dữ liệu ngôn ngữ kí hiệu để nhận dạng các biểu hiện và cử chỉ tương ứng với ngôn ngữ kí hiệu.
- Xử lý dữ liệu và truyền thông tin:
- Mediapipe hỗ trợ xử lý dữ liệu và truyền thông tin giữa các thành phần của pipeline.
- Dữ liệu hình ảnh được truyền qua mô hình nhận dạng để dự đoán và nhận dạng ngôn ngữ kí hiệu.
- Thông tin về các biểu hiện và cử chỉ nhận dạng được truyền đến các thành phần khác để tiếp tục xử lý hoặc hiển thị kết quả.
- Hiển thị kết quả:

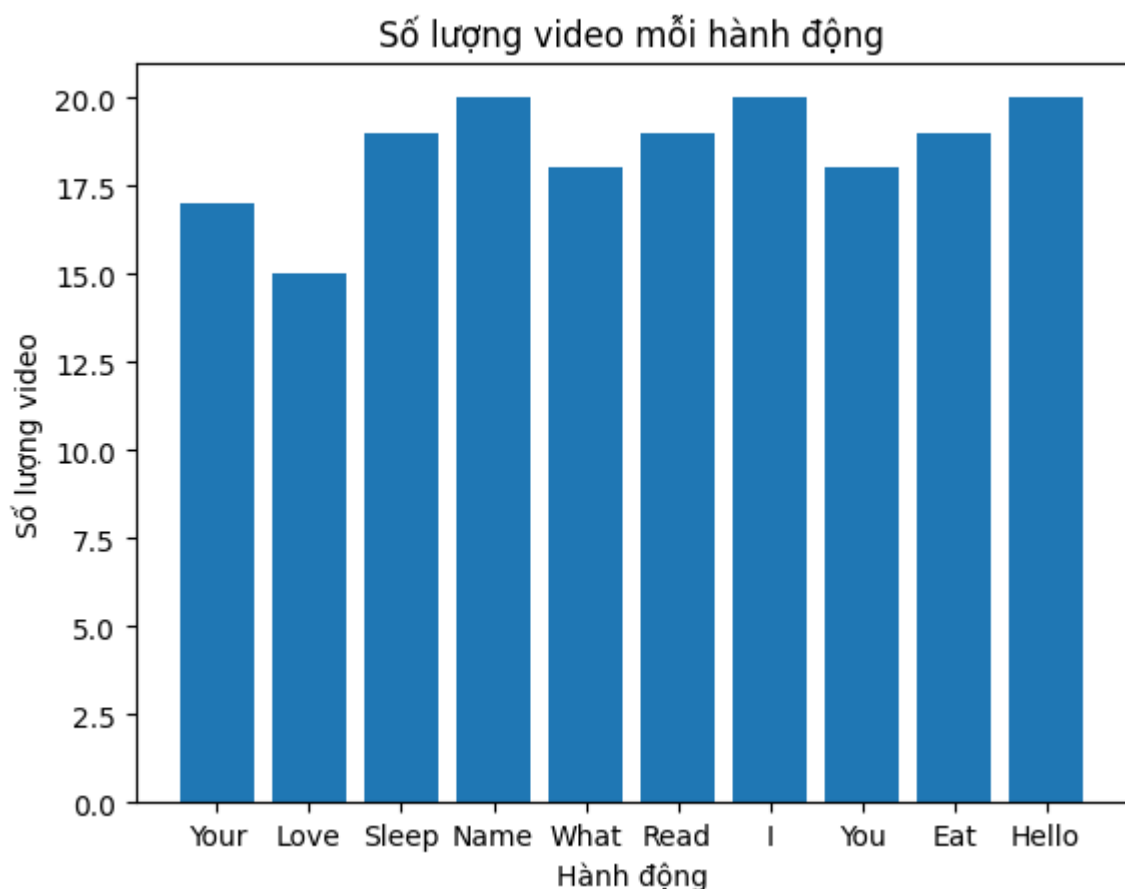
- Mediapipe cung cấp các công cụ và API để hiển thị kết quả nhận dạng ngôn ngữ ký hiệu.
- Kết quả có thể được hiển thị trực tiếp trên giao diện người dùng, hoặc có thể được gửi đến các thiết bị ngoại vi khác để thực hiện các hành động tương ứng.

3. Kết quả

3.1. Dữ liệu

3.1.1. Thu thập dữ liệu lần 1

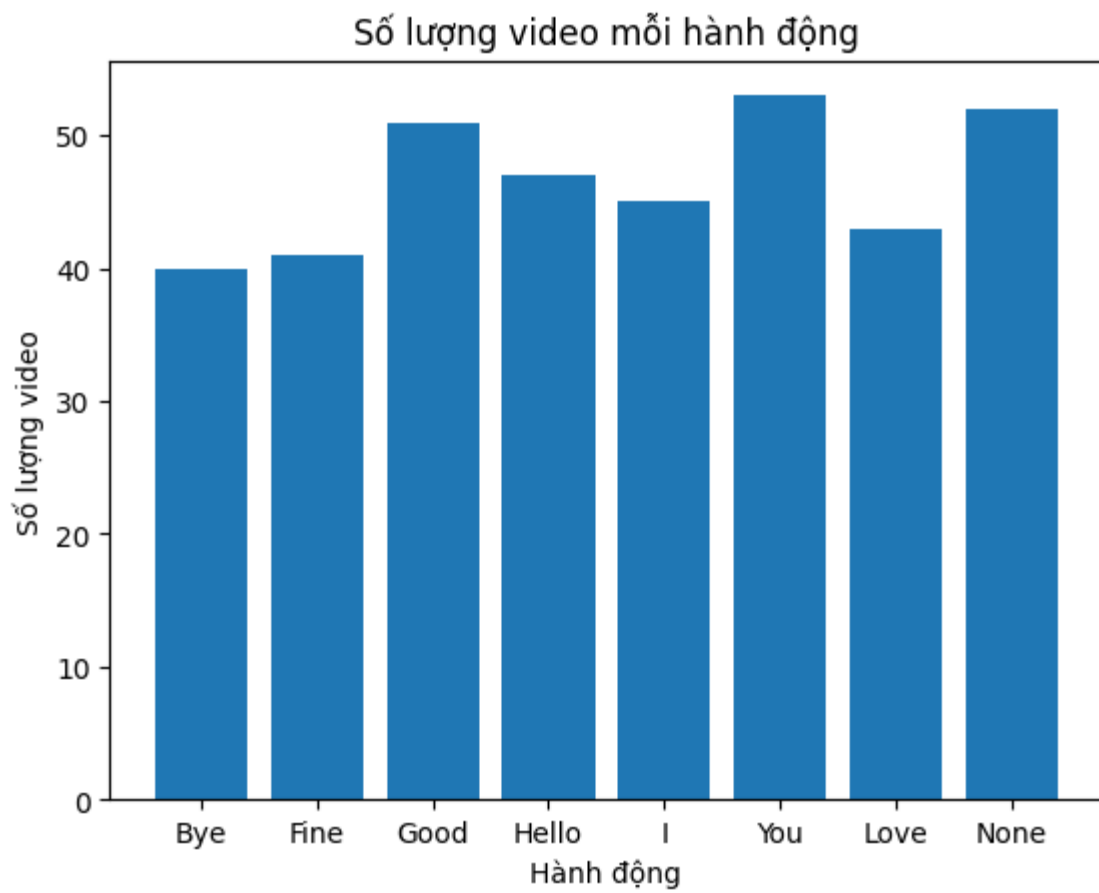
Gồm các hành động: 'Your', 'Love', 'Sleep', 'Name', 'What', 'Read', 'I', 'Bye', 'You', 'Eat', 'Hello'. Có 10 hành động, Mỗi hành động gồm 17 đến 20 video được quay bằng esp32 cam.



Hình 3.1. Số lượng video mỗi hành động lần 1

3.3.2. Thu thập dữ liệu lần 2

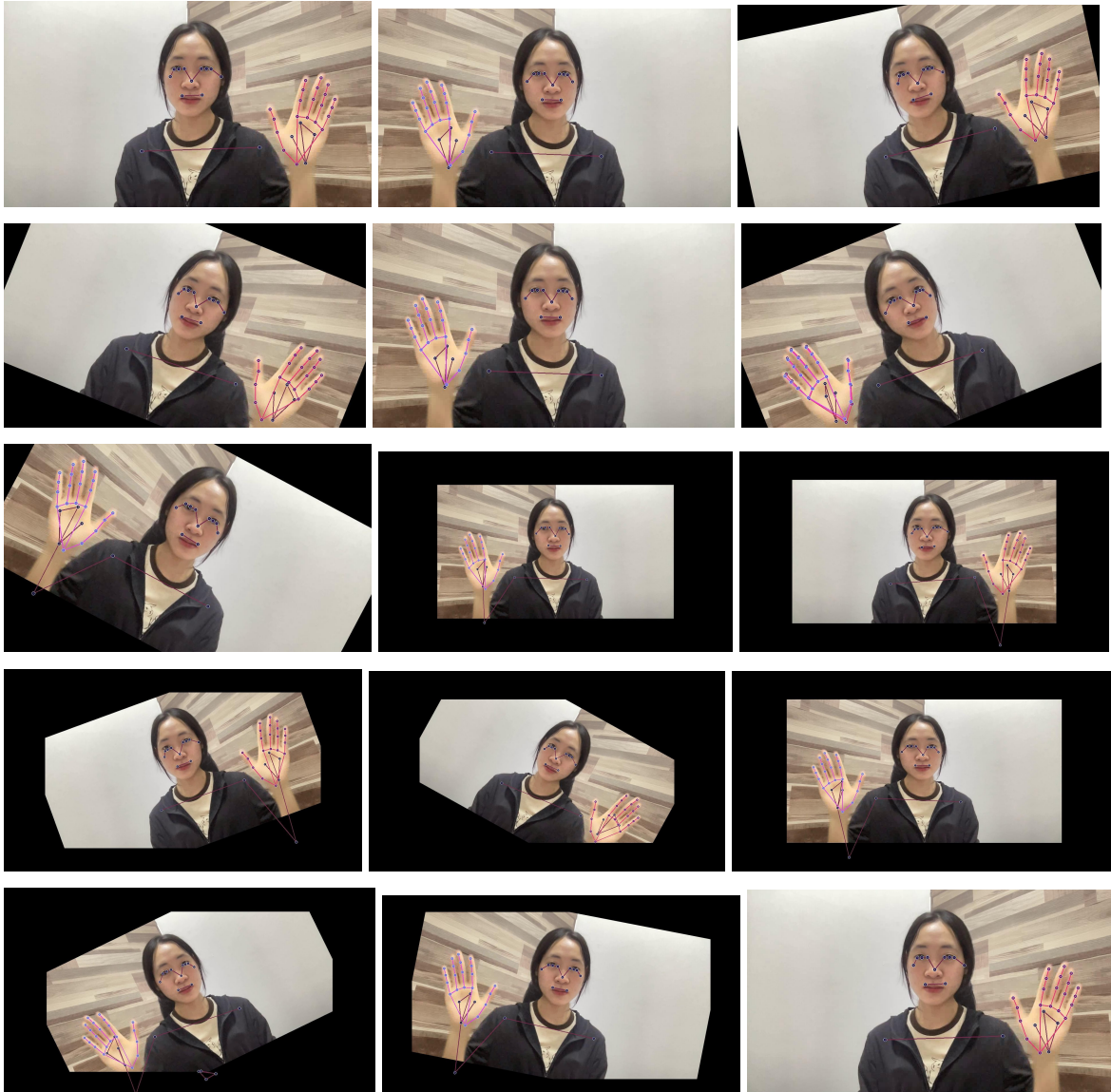
Gồm các hành động: 'Bye', 'Fine', 'Good', 'Hello', 'I', 'You', 'Love', 'None'. Không sử dụng những hành động dễ gây nhầm lẫn mơ hồ, thu thập thêm dữ liệu hành động mới. Video được quay ở nhiều góc độ và độ xa khác nhau. Có 8 hành động. Số lượng video mỗi loại từ 40 đến 50 video.



Hình 3.2. Số lượng video mỗi hành động lần 2

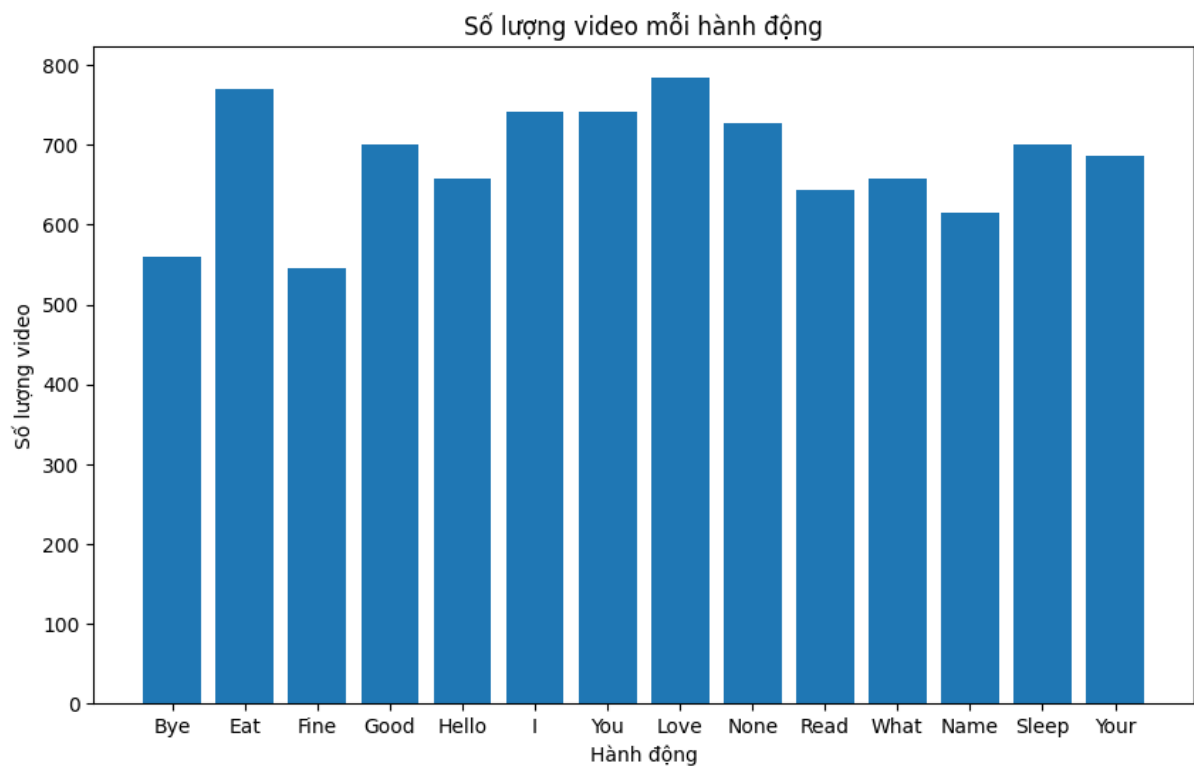
3.1.3. Làm giàu dữ liệu

Sử dụng tất cả các video đã thu thập, kết hợp với các phương pháp làm giàu dữ liệu hình ảnh như: flip, rotate, thêm padding. Từ mỗi video gốc ta có được 14 video mới.



Hình 3.3 Làm giàu dữ liệu

Sau khi làm giàu dữ liệu, có tổng cộng 14 hành động, mỗi hành động có từ 550 đến 750 video.

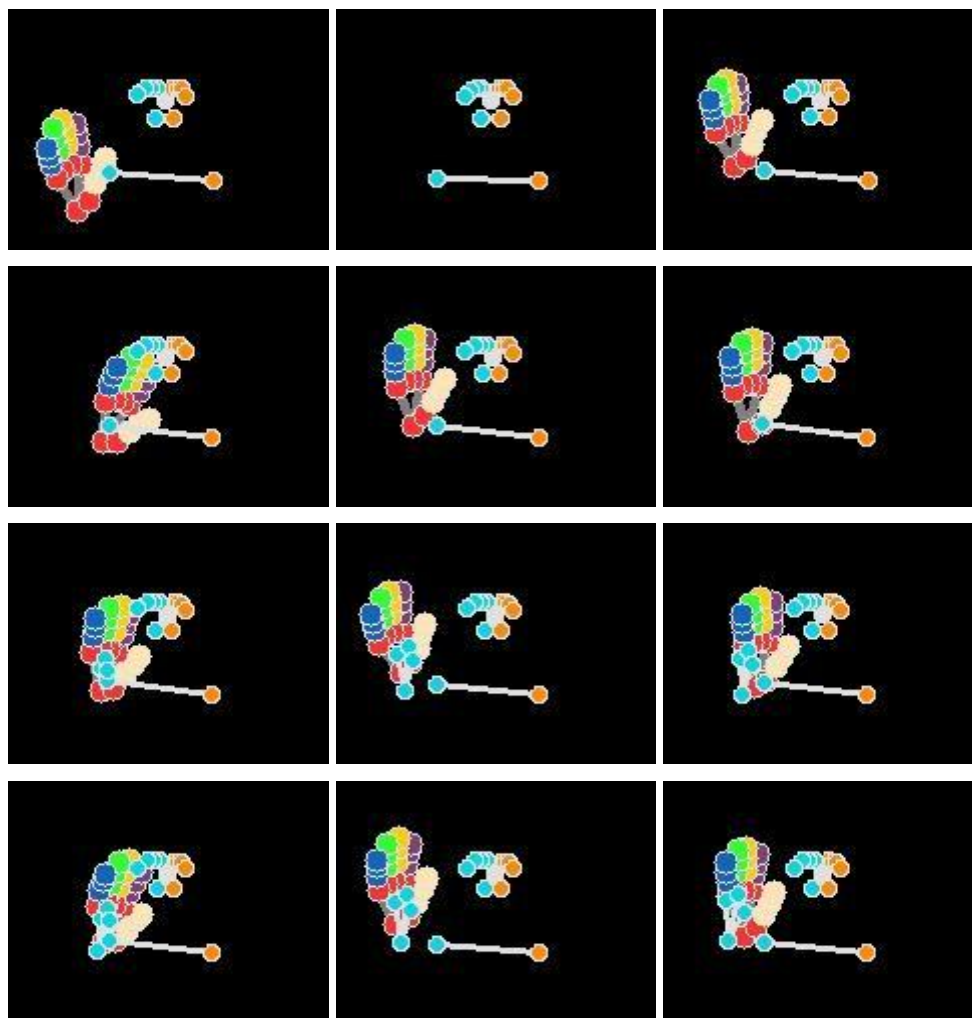


Hình 3.4. Số lượng video sau khi làm giàu dữ liệu

3.2. Trích xuất đặc trưng

3.2.1. Thử nghiệm lần 1

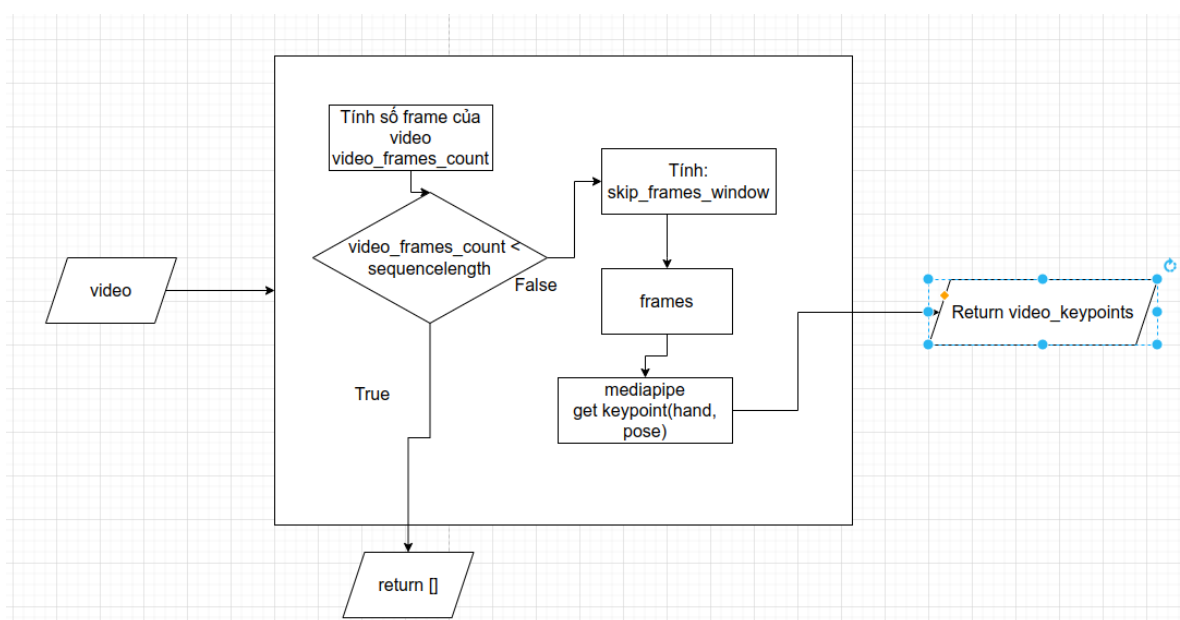
Sử dụng Mediapipe trích xuất các điểm keypoint của tay và form của người. Tạo ra video mới gồm các điểm point từ video gốc.



Hình 3.5. Video trích xuất lần 1

3.2.2. Thử nghiệm lần 2

Sử dụng thư viện Mediapipe trích xuất các keypoint, sau đó lưu các thông số keypoint đó vào ma trận, sử dụng để train.



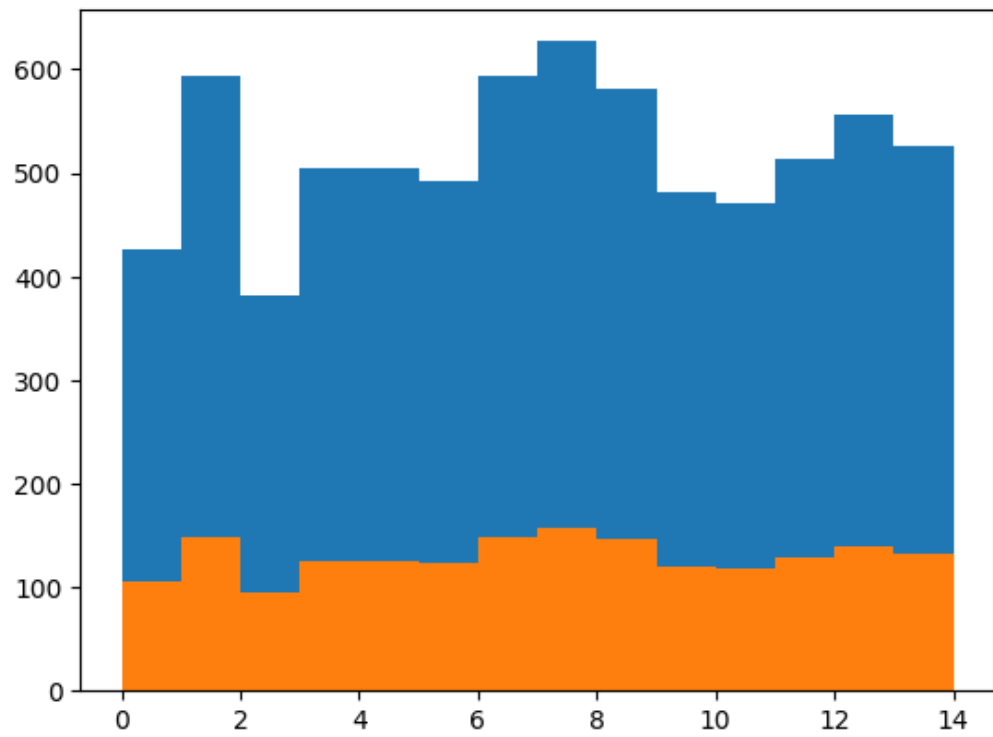
Hình 3.6. Sơ đồ trích xuất đặc trưng

- Mỗi keypoint: gồm các thông số [x, y, z, visibility] của hand và pose
- + x, y, z: Đây là ba giá trị tọa độ để định vị vị trí của đối tượng trong không gian ba chiều. Giá trị x đại diện cho hoành độ, y đại diện cho tung độ và z đại diện cho hoành độ.
- + visibility: Thuộc tính này đo lường mức độ mà đối tượng có thể được nhìn thấy hoặc phát hiện trong một hệ thống nhận dạng hoặc theo dõi. Giá trị của visibility có thể là một số thực trong khoảng từ 0 đến 1, trong đó 0 có nghĩa là không nhìn thấy hoặc không phát hiện được, và 1 có nghĩa là hoàn toàn nhìn thấy hoặc phát hiện được
- Tổng cộng 75 keypoint

Shape Input cho model: (12, 300)

3.3 Kết quả huấn luyện

Dữ liệu huấn luyện: Dùng StratifiedShuffleSplit của Sklearn, chia dữ liệu theo tỷ lệ train:val:test là 81:9:10.

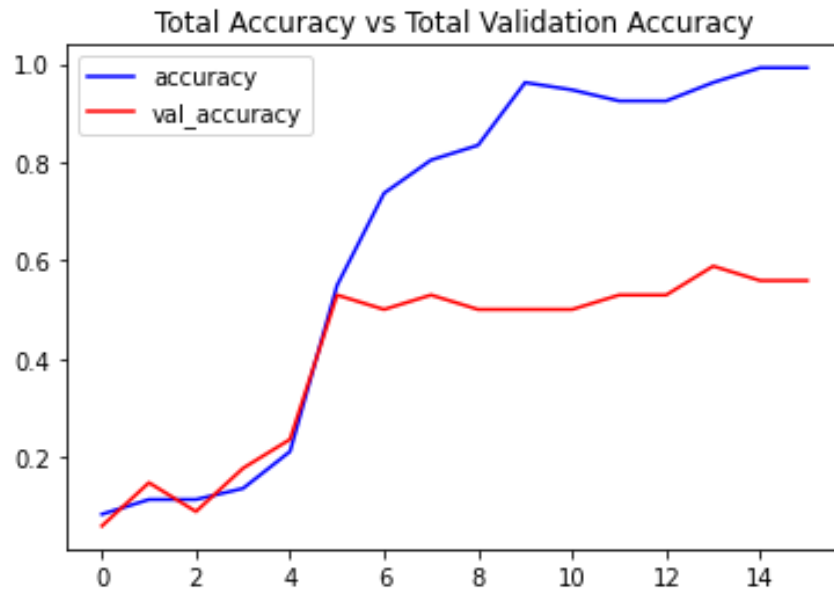


Hình 3.7. Phân chia dữ liệu

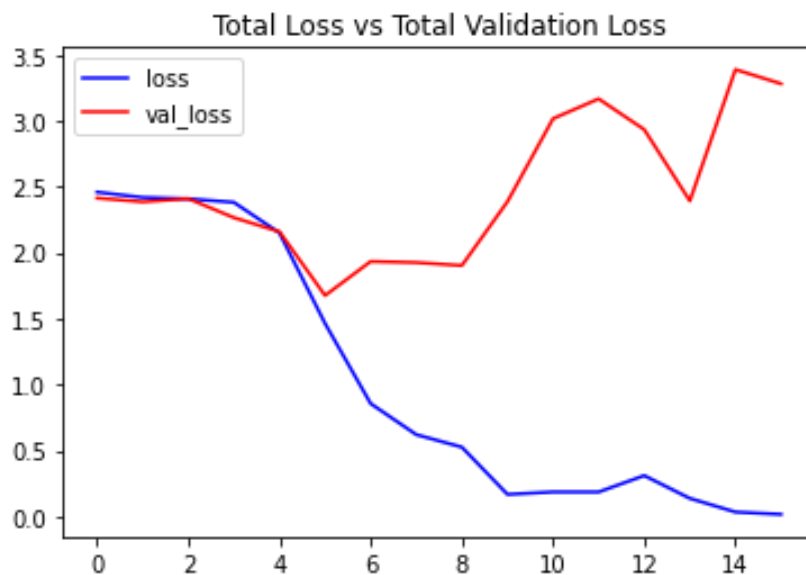
3.3.1. Huấn luyện lần 1 sử dụng model ConvLSTM

Huấn luyện lần này sử dụng các dữ liệu thu được lần 1 và cách trích xuất đặc trưng thứ nhất là sử dụng các video gồm các điểm keypoint. Sử dụng model ConvLSTM

Kết quả thu được:



Hình 3.8. Độ chính xác của tập train và test huấn luyện lần 1



Hình 3.9. Mất mát train tập train và test huấn luyện lần 1

Nhận xét:

Độ chính xác trên tập validation khoảng 60% nhưng khi đưa vào inference để kiểm tra thực tế, mô hình nhận dạng với độ chính xác rất thấp.

Nguyên nhân:

Với hình ảnh chỉ gồm các điểm keypoint, rất dễ bị nhầm lẫn giữa các hành động do không có thông tin về chiều sâu của vị trí cánh tay, ngón tay. Các hành động dễ thể hiện từ, "Tôi", "Bạn" rất dễ bị nhầm lẫn do vị trí tay khi trích xuất ra video chỉ có keypoint rất giống nhau.

Số lượng video cho mỗi hành động nhận dạng còn ít.

Giải pháp:

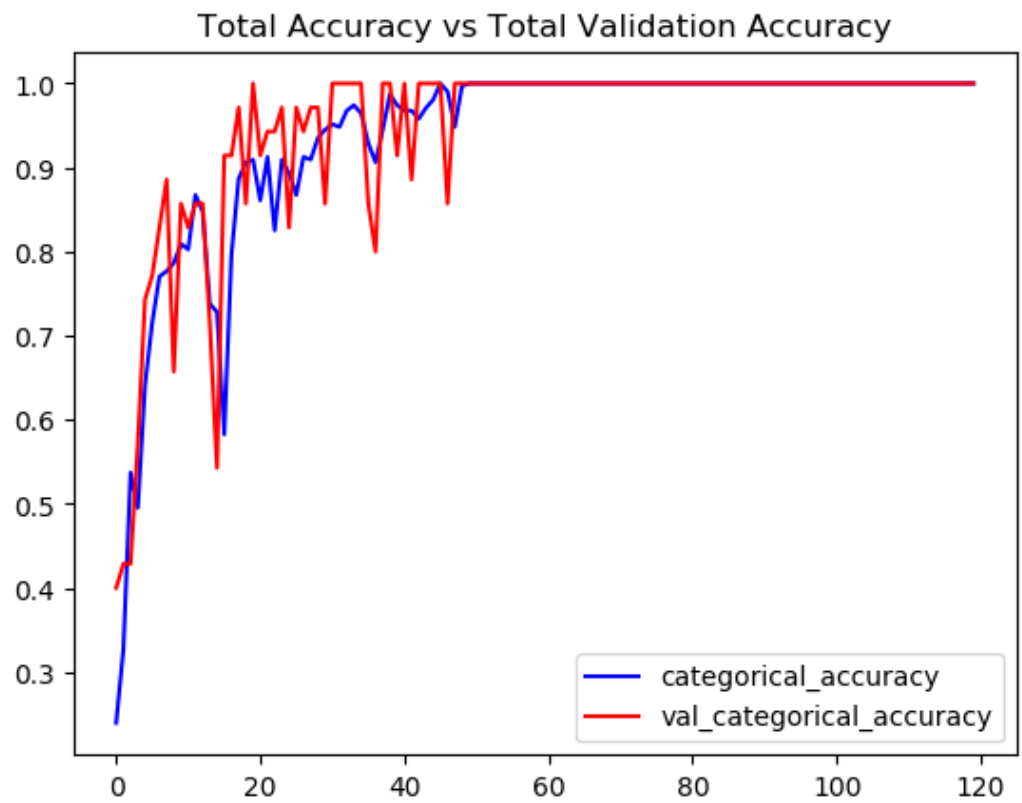
Chọn lại cách trích xuất đặc trưng khác (đã nêu ở mục 3.3.2).

Thu thập thêm dữ liệu mới (Kết quả ở mục 3.1.1)

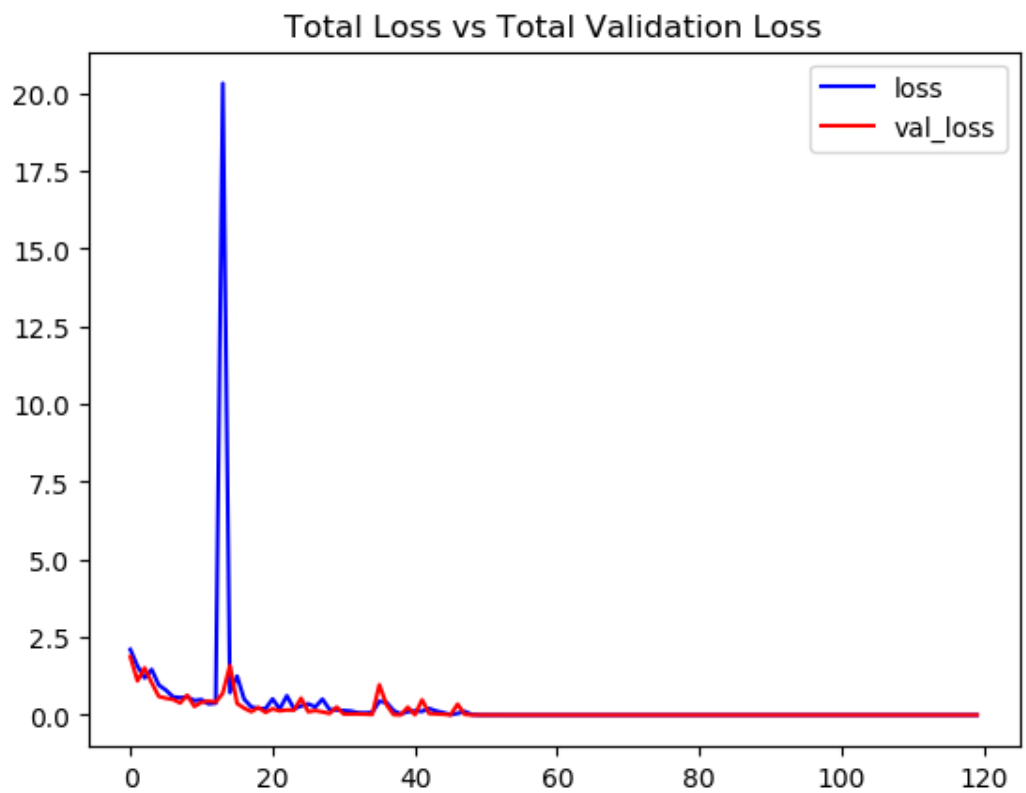
3.3.2. Huấn luyện lần 2 sử dụng model LSTM

Huấn luyện sử dụng dữ liệu thu thập được ở lần thứ 2 và các trích xuất đặc trưng lần 2 (mục 3.1.2).

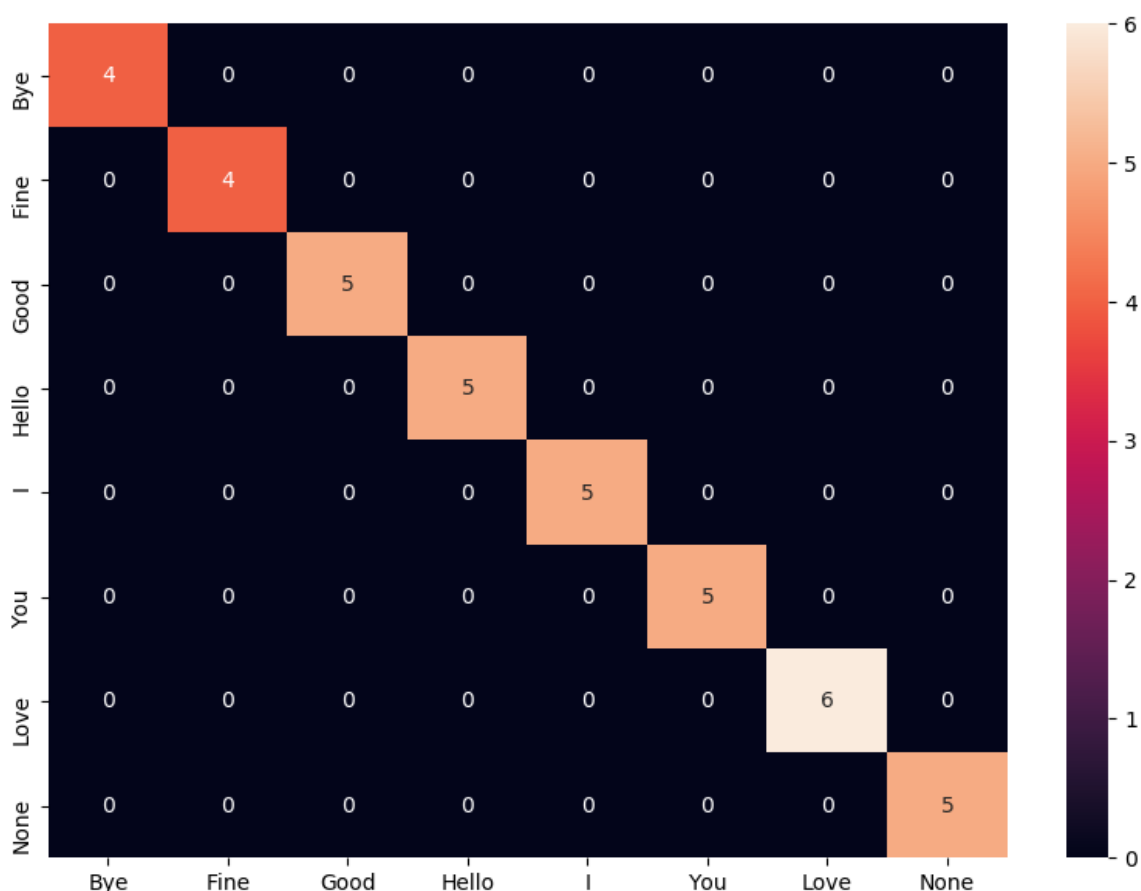
Kết quả huấn luyện thu được:



Hình 3.10. Độ chính xác của tập train và test huấn luyện lần 2



Hình 3.11. Mất mát train tập train và test huấn luyện lần 2



Hình 3.12. Ma trận nhầm lẫn trên tập test lần 1

Nhận xét:

Độ chính xác đạt xấp xỉ 100%. Tuy nhiên khi dự đoán cho chuỗi hành động, đưa ra những kết quả không mong muốn. Ví dụ chưa thực hiện hành động nào vẫn đưa ra dự đoán, làm cho kết quả trả về không được chính xác.

Nguyên nhân:

Dữ liệu không có sự mơ hồ giữa các hành động nên dự đoán chính xác với những hành động riêng lẻ nhưng với chuỗi hành động thì do lúc đọc đủ số frame cần thiết để dự đoán, nhưng vẫn chưa có hành động cụ thể để đưa ra dự đoán ngẫu nhiên, do gần như tất cả hành động vị trí tay ban đầu đều giống nhau.

Hạn chế:

Các hành động trong dữ liệu sử dụng 1 tay cố định để tránh nhầm lẫn dẫn đến lúc đưa vào thực tế nếu sử dụng tay khác dễ đưa ra dự đoán sai.

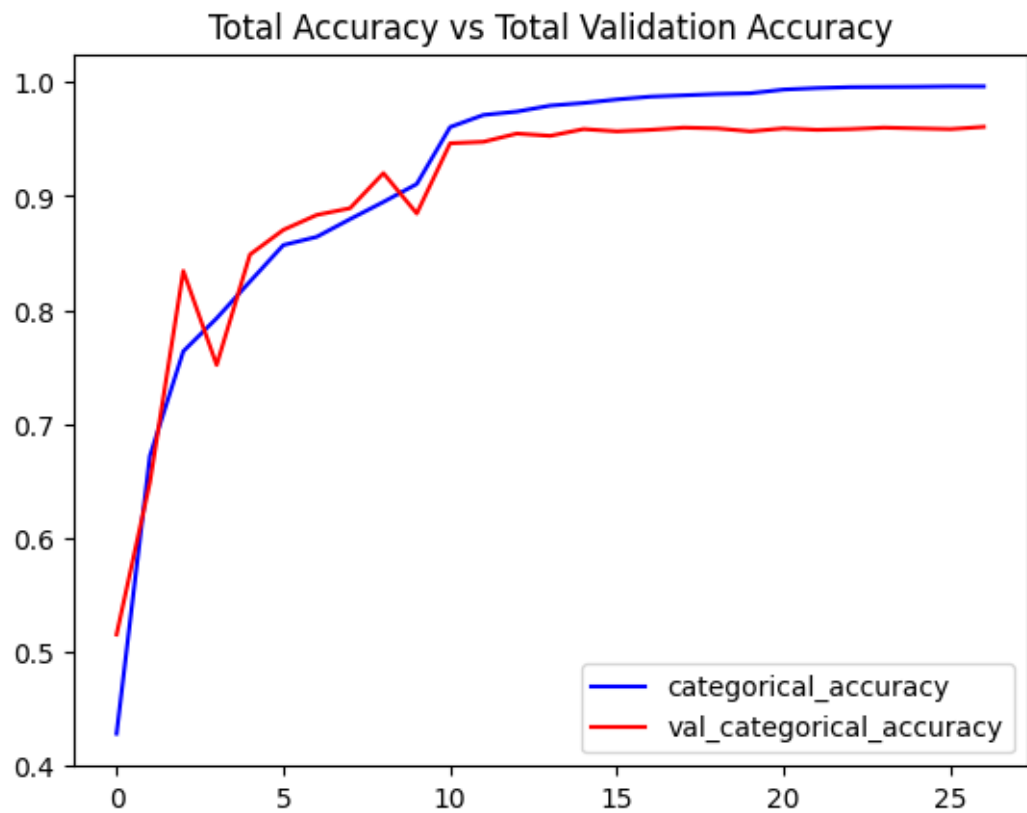
Độ chính xác đạt xấp xỉ 100%. Tuy nhiên khi dự đoán cho chuỗi hành động, đưa ra những kết quả không mong muốn. Ví dụ chưa thực hiện hành động nào vẫn đưa ra dự đoán, làm cho kết quả trả về không được chính xác.

Giải pháp xử lý:

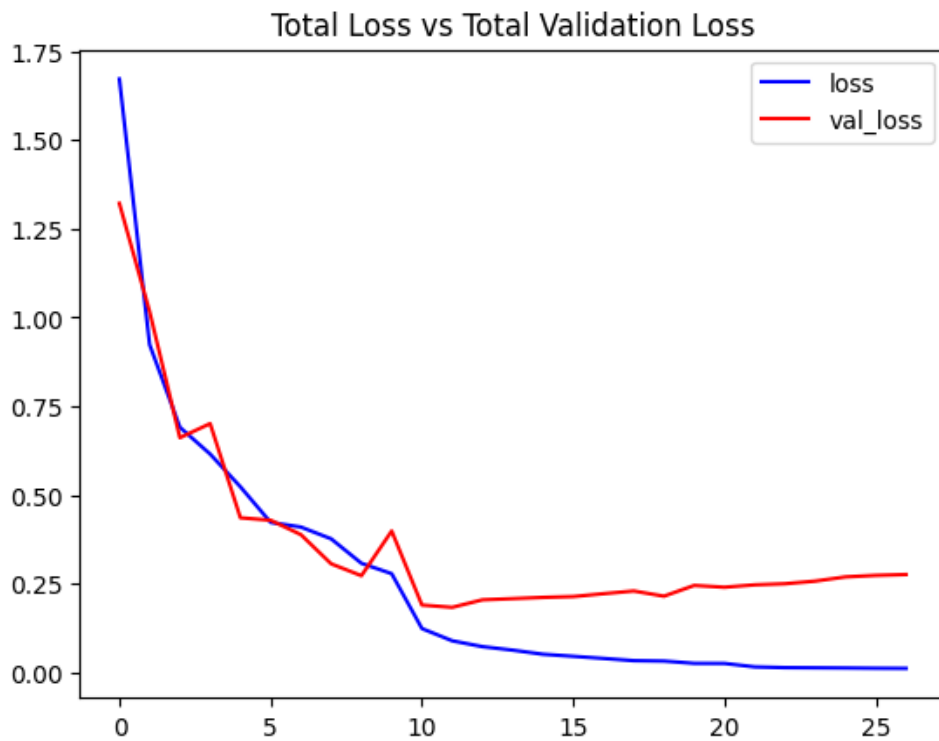
Làm giàu dữ liệu (Kết quả ở mục 3.1.3) để mô hình có thể phân loại được các hành động với các khoảng cách, tay khác nhau.

3.3.3. Huấn luyện lần 3 sử dụng model LSTM

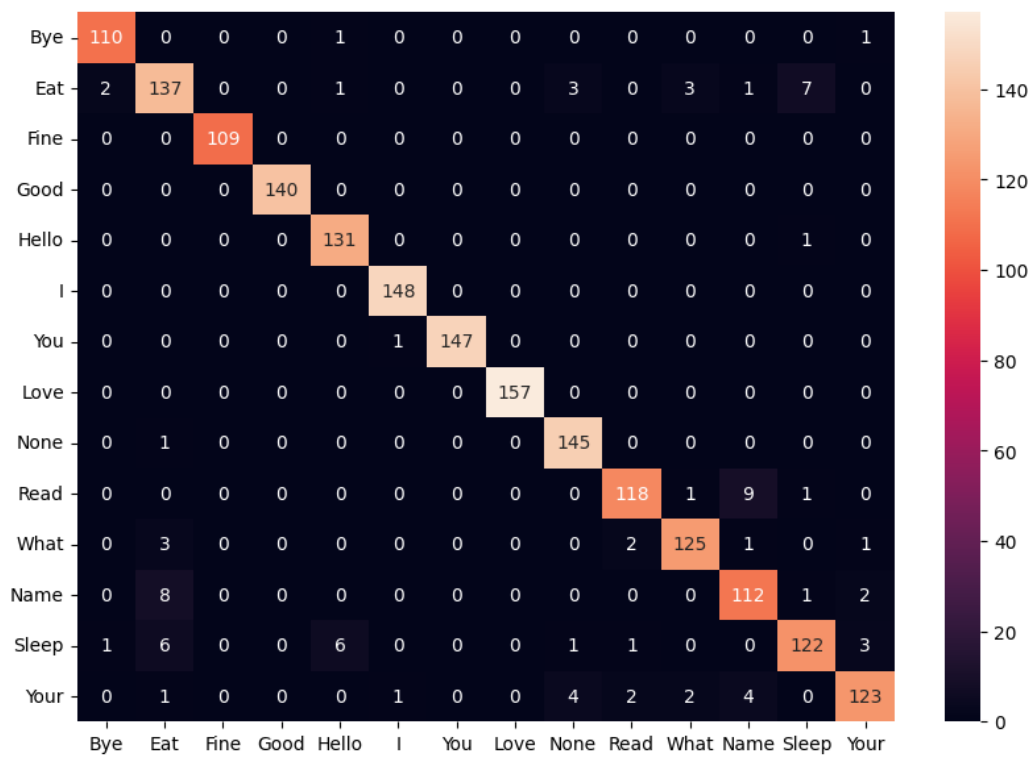
Dữ liệu sử dụng để huấn luyện lần này là dữ liệu đã được làm giàu, sử dụng các trích xuất đặc trưng giống với lần huấn luyện trước. Đây là kết quả huấn luyện:



Hình 3.14. Độ chính xác của tập train và test huấn luyện lần 3



Hình 3.15. Mất mát train tập train và test huấn luyện lần 3



Hình 3.16. Ma trận nhầm lẫn trên tập test của mô hình huấn luyện lần 3

Nhận xét:

Kết quả train và test đạt độ chính xác cao đối với những hành động riêng lẻ.

Có thể áp dụng để nhận dạng các hành động riêng lẻ.

3.4 Giao diện người dùng



Hình 3.18. Giao diện người dùng

4. Kết luận

4.1 Đánh giá

Đối với nhận dạng một hành động: Hệ thống nhận diện ngôn ngữ ký hiệu hoạt động tương đối hiệu quả và ổn định. Tuy nhiên độ chính xác khi nhận diện động còn chưa cao nếu gặp các điều kiện bất lợi như mặt không chính diện, thiếu sáng, ...

Đối với nhận dạng chuỗi hành động: Hệ thống nhận dạng chưa được chính xác, cần có sự cải tiến trong khả năng nhận dạng chuỗi hành động.

4.2 Hướng phát triển

Nhận diện chuỗi hành động: Hiện tại mô hình nhận dạng chuỗi hành động chưa được chính xác, chúng em cần nghiên cứu và cải thiện độ chính xác của mô hình để có thể nhận diện được chuỗi hành động liên tục với độ chính xác cao hơn.

Giao diện: Giao diện tương đối đơn giản, cần nâng cấp giao diện đẹp, thân thiện hơn với người dùng hơn.

5. Danh mục tài liệu tham khảo

- [1] Freecodecamp, "How to use transfer learning for sign language ",
<https://www.freecodecamp.org/news/asl-recognition-using-transfer-learning-918ba054c004>
- [2] developers google, "MediaPipe Solutions guide",
<https://developers.google.com/mediapipe/solutions/guide>
- [3] Taha Anwar (BleedAI.com, 2021), "Introduction to Video Classification and Human Activity Recognition",
<https://learnopencv.com/introduction-to-video-classification-and-human-activity-recognition/>
- [4] Keras, "How to use LSTM", https://keras.io/api/layers/recurrent_layers/lstm/