

THỰC TẬP CNTT 5: TRIỂN KHAI ỨNG DỤNG AIoT

ĐỀ TÀI: XÂY DỰNG HỆ THỐNG GIÁM SÁT VÀ PHÂN TÍCH HÀNH ĐỘNG TẾ NGÃ

Giảng viên hướng dẫn: ThS. Lê Trung Hiếu, GV. Nguyễn Văn Nhân.
Trình bày: Nhóm 3_CNTT 16-04.
Thành viên: Phạm Thị Hồng Ngọc, Nguyễn Đức Thường, Nguyễn Đào Nguyên Giáp, Nguyễn Hải Phong.

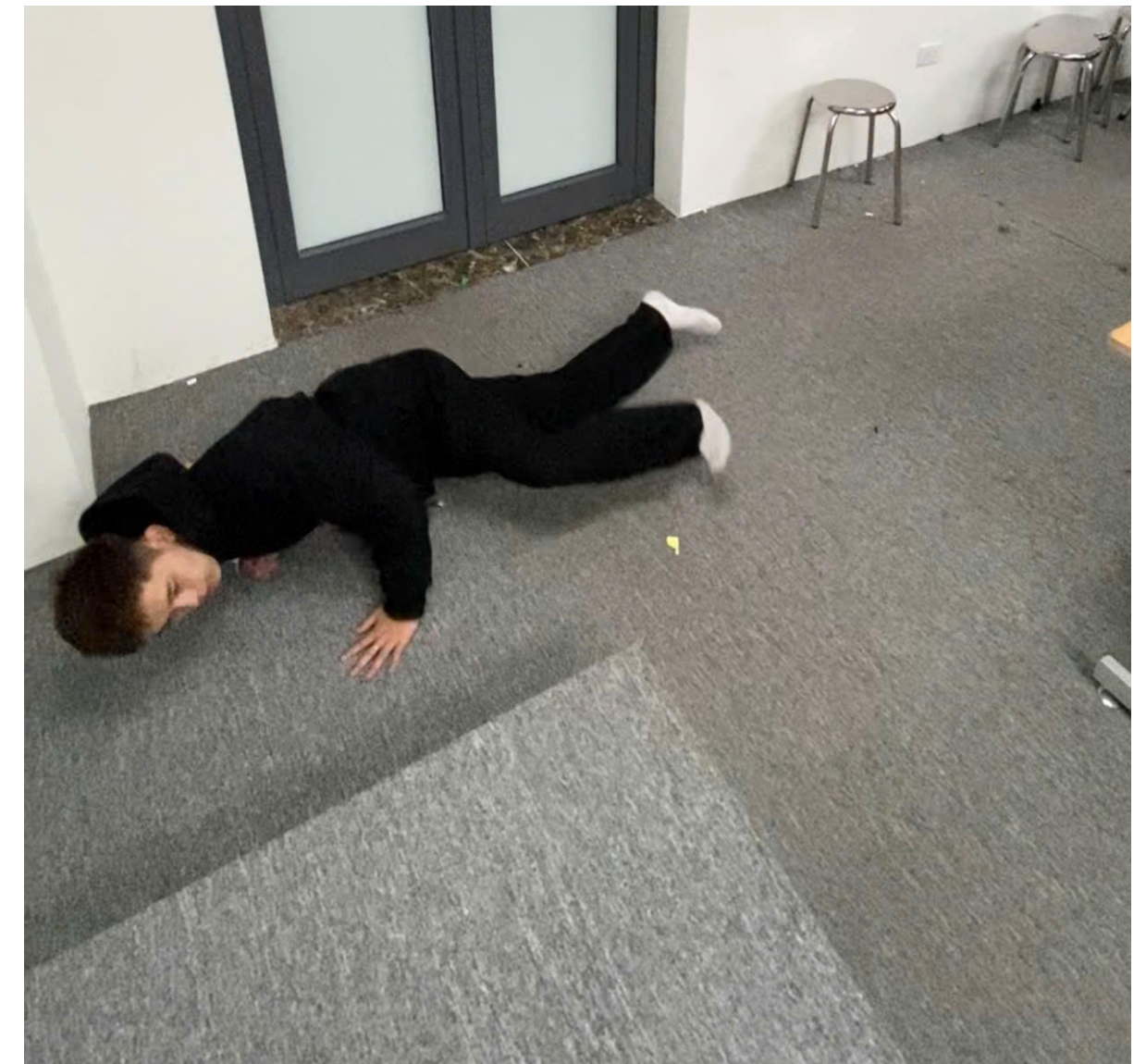
MỤC LỤC

1. Đặt vấn đề.
2. Mục tiêu và đề xuất.
3. Nghiên cứu liên quan.
4. Phương pháp của nhóm.
5. Sơ đồ hệ thống.
6. Các thiết bị.
7. Cách thức triển khai đề tài.
8. Thực nghiệm đánh giá kết quả.
9. Ưu điểm, hạn chế.
10. Kết luận và phương hướng phát triển.



1. ĐẶT VẤN ĐỀ

- **Té ngã** ở người có thể **gây chấn thương** nghiêm trọng, thậm chí tử vong.
- Việc **phát hiện** và **phân tích** té ngã giúp **cảnh báo** sớm, **hỗ trợ y tế** và **ngăn ngừa** rủi ro.
- Cần **nghiên cứu** các **phương pháp** **hiệu quả**, **chính xác** và **dễ triển khai** để **bảo vệ sức khỏe** con người.

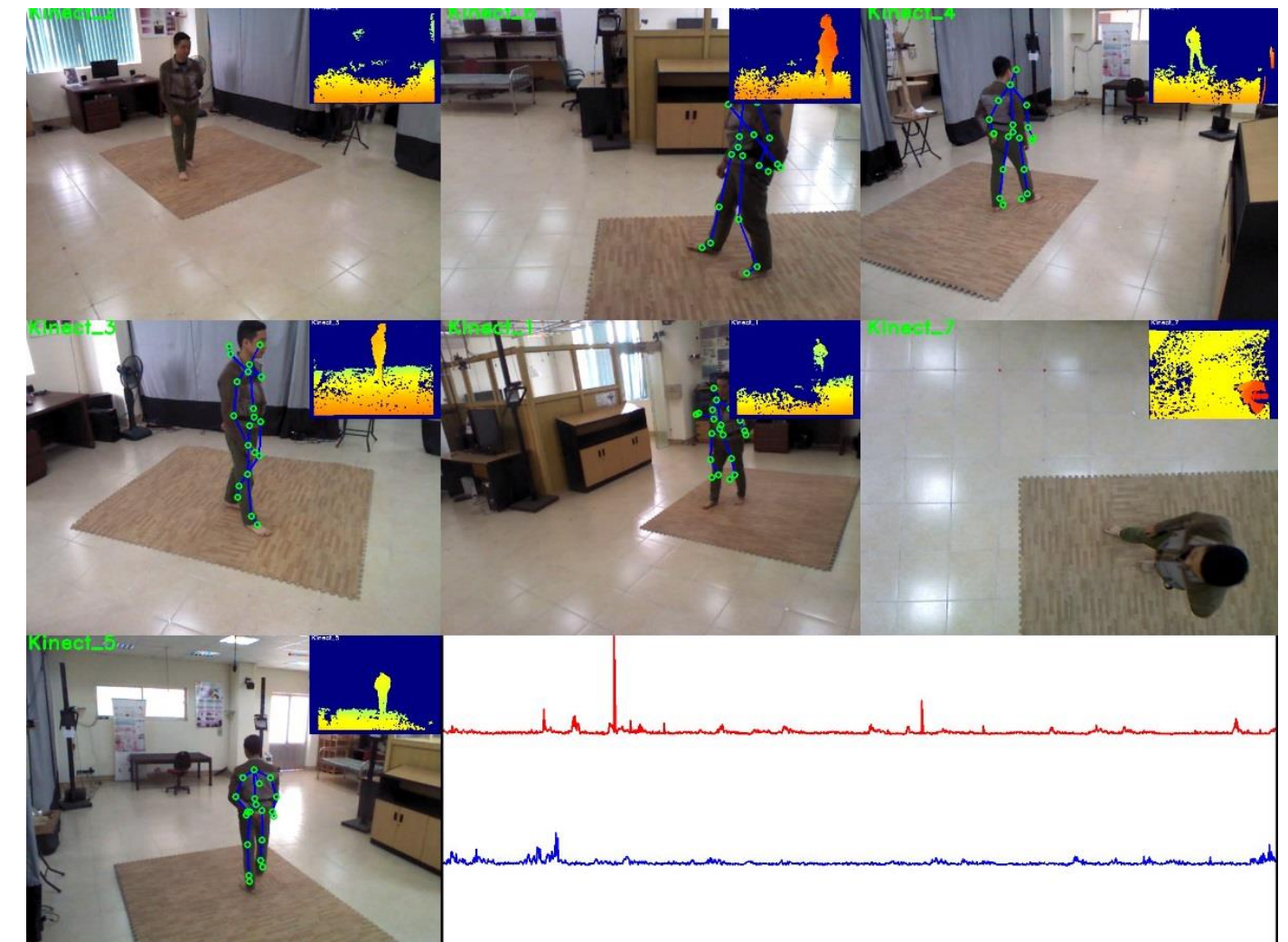


2. MỤC TIÊU

- Phát hiện **chính xác** – Xác định té ngã qua dữ liệu cảm biến.
- Cảnh báo **nhANH** – Gửi thông báo vào tin nhắn đến người thân.
- **Tiết kiệm** năng lượng – Hoạt động lâu dài, ít sạc.
- Chi phí **thấp** – Thiết bị nhỏ gọn, dễ triển khai.
- **Phân tích** dữ liệu – Ghi lại để cải thiện hệ thống.

3. NGHIÊN CỨU LIÊN QUAN

- **Continuous Multimodal Multi-view Dataset of Human Fall (CMD FALL)**
- Dự án xây dựng một Tập dữ liệu Đa phương thức Đa góc nhìn liên tục về hành động ngã của con người và phát triển các thuật toán nhận diện ngã của con người.
- Nguồn: <https://www.mica.edu.vn/perso/Tran-Thi-Thanh-Hai/CMD FALL.html>



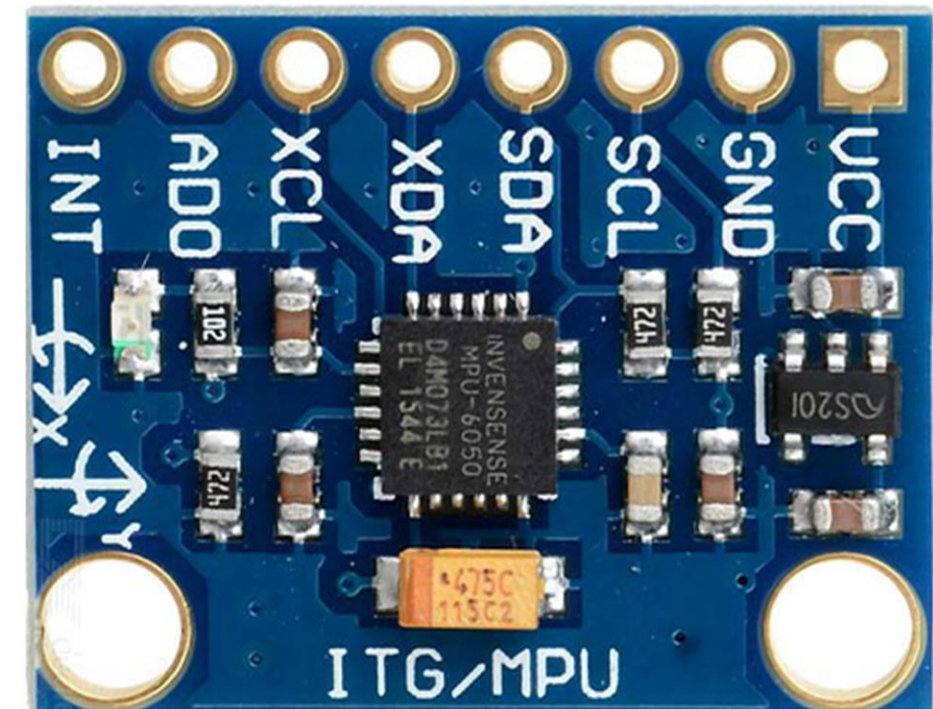
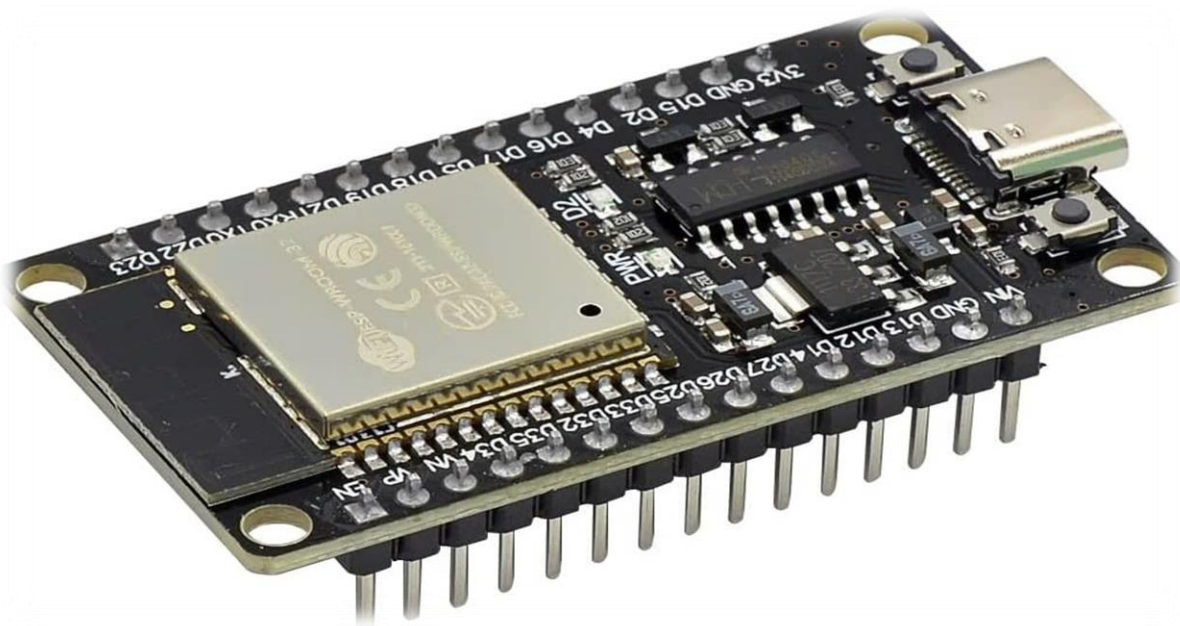
3. NGHIÊN CỨU LIÊN QUAN

- **Real-Time Fall Detection**
- Hệ thống phát hiện ngã đa người dựa trên IoT sử dụng **YOLO** và **ESP32-CAM**. Dự án này tích hợp công nghệ phát hiện đối tượng tiên tiến với ESP32-CAM để phát hiện ngã theo thời gian thực.
- Nguồn: <https://github.com/tonlongthuat/Real-Time-Fall-Detection>

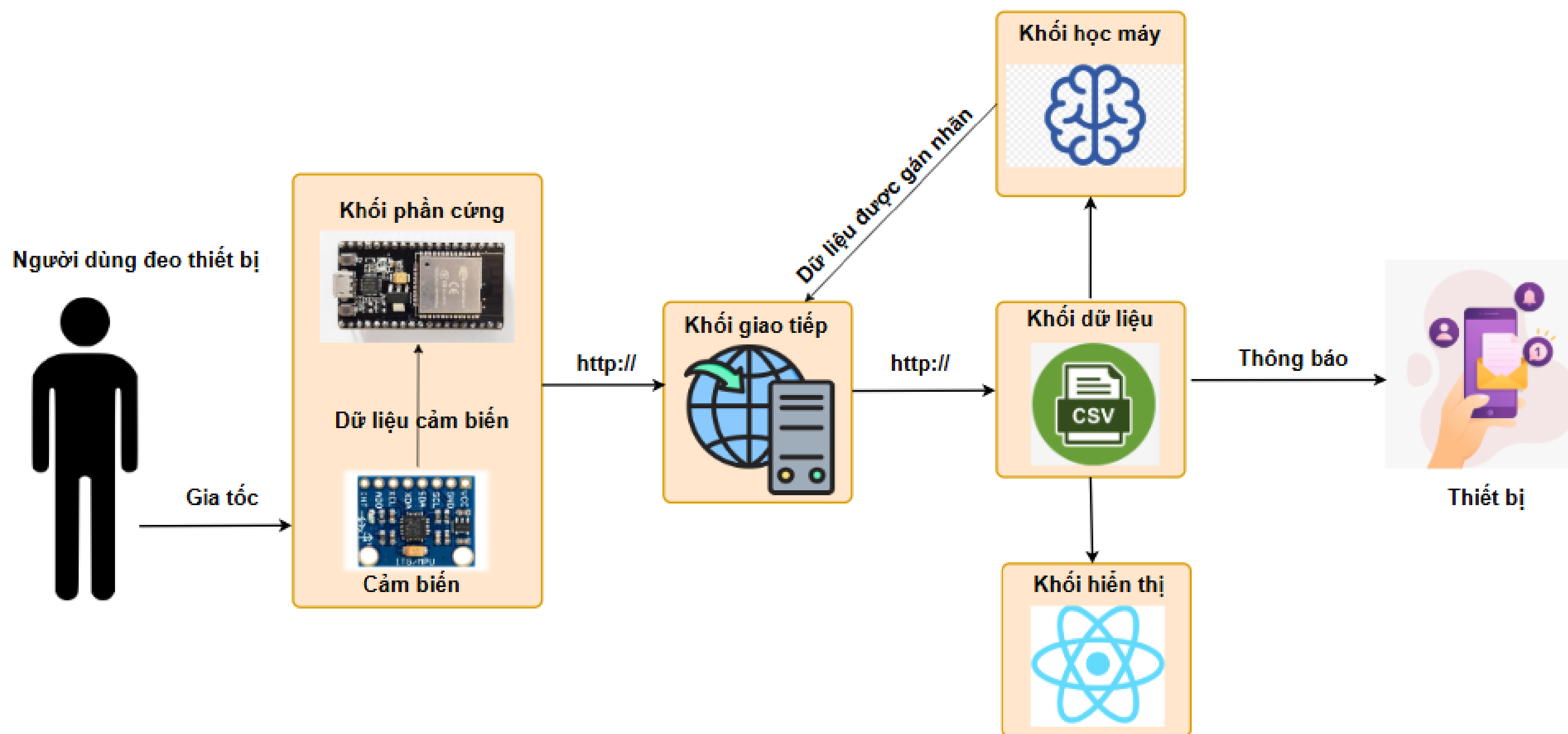


4. PHƯƠNG PHÁP CỦA NHÓM

Sử dụng **ESP32** và cảm biến **MPU6050** để phát hiện té ngã dựa trên **gia tốc**.
Khi té ngã xảy ra, thiết bị sẽ thu, gửi dữ liệu vào hệ thống để phân tích dữ liệu.
Gửi tin nhắn thông báo.

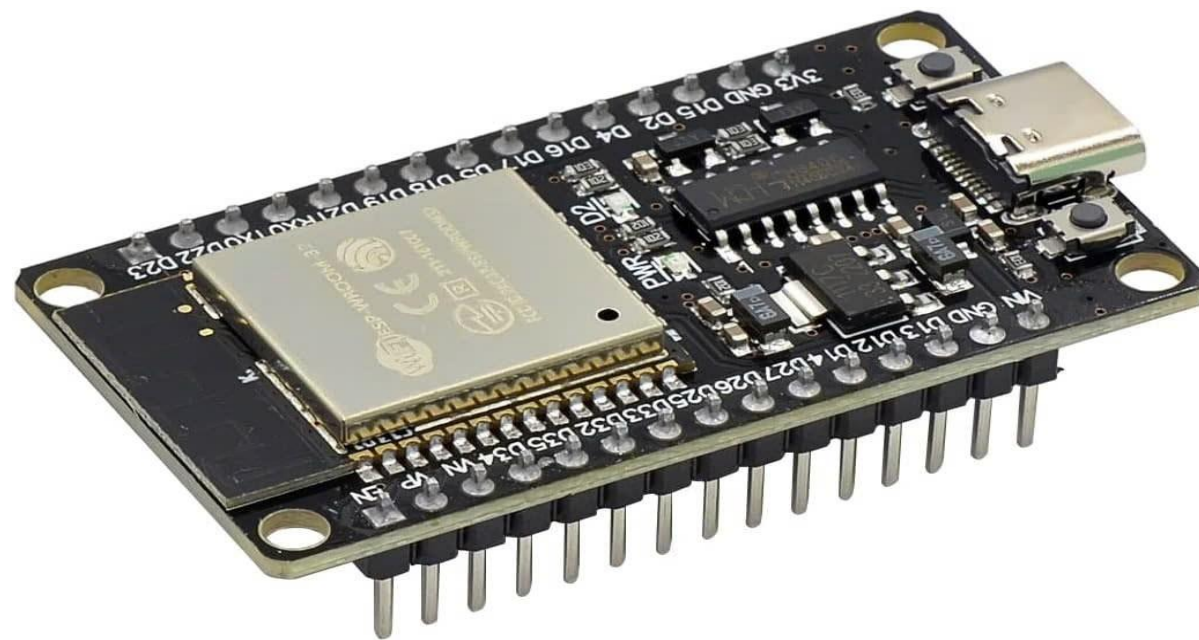


5. SƠ ĐỒ HỆ THỐNG



6. THIẾT BỊ SỬ DỤNG

- Các thiết bị sử dụng:

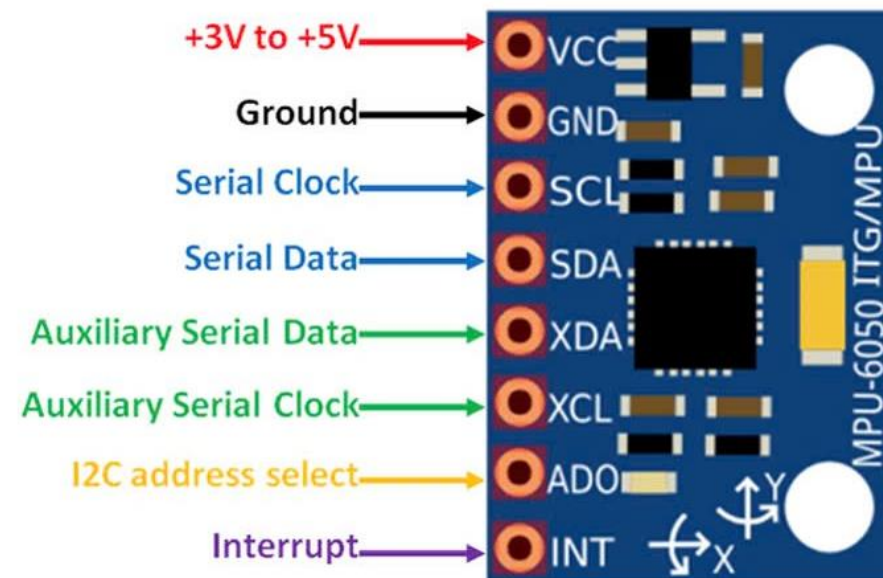


ESP32-WROOM

- Xử lý dữ liệu: Nhận dữ liệu từ cảm biến MPU6050, phát hiện té ngã.
- Kết nối không dây: Gửi dữ liệu qua Wi-Fi, hỗ trợ cảnh báo.
- Tiết kiệm năng lượng: Hỗ trợ chế độ tiết kiệm điện, phù hợp ứng dụng di động.

6. THIẾT BỊ SỬ DỤNG

- Các thiết bị sử dụng:



- Gia tốc kế 3 trục: Đo gia tốc X, Y, Z để xác định chuyển động, tư thế.
- Con quay hồi chuyển: Đo tốc độ quay, xác định góc nghiêng. DMP: Xử lý dữ liệu cảm biến, giảm tải cho ESP32.

Cảm biến gia tốc MPU6050

7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 1:
Thu dữ liệu.

Bước 2:
Tiền xử lý
dữ liệu.

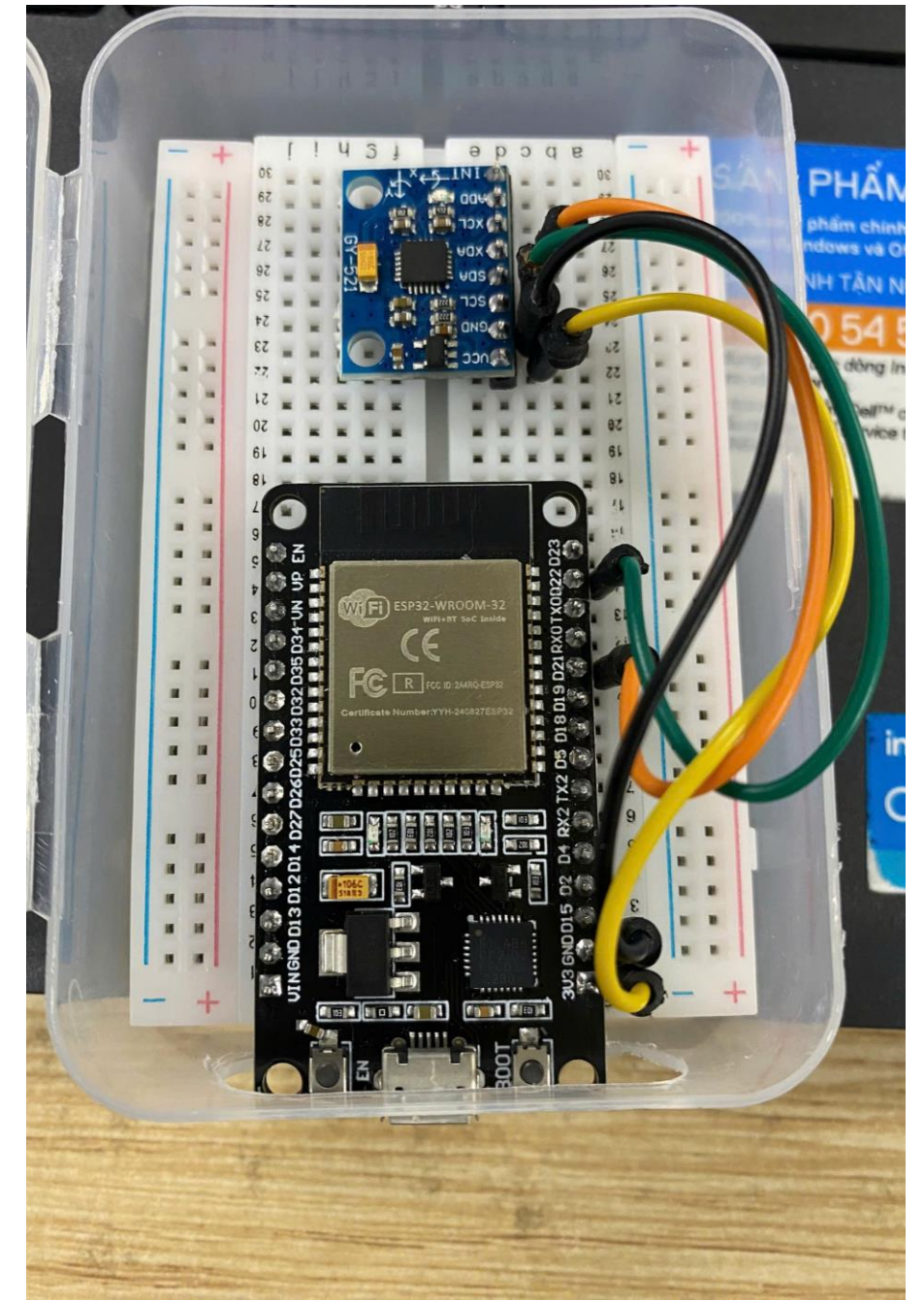
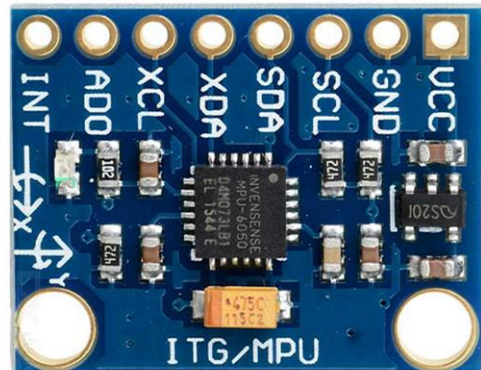
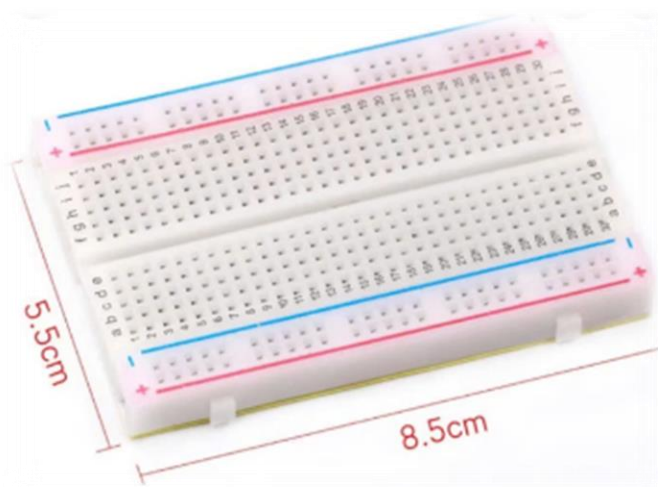
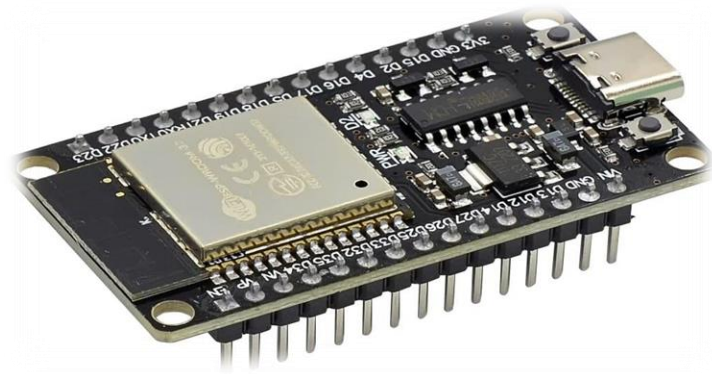
Bước 3:
Trực quan
hóa dữ liệu.

Bước 4:
Huấn luyện
mô hình.

Bước 5:
Kiểm tra mô
hình huấn
luyện.

7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

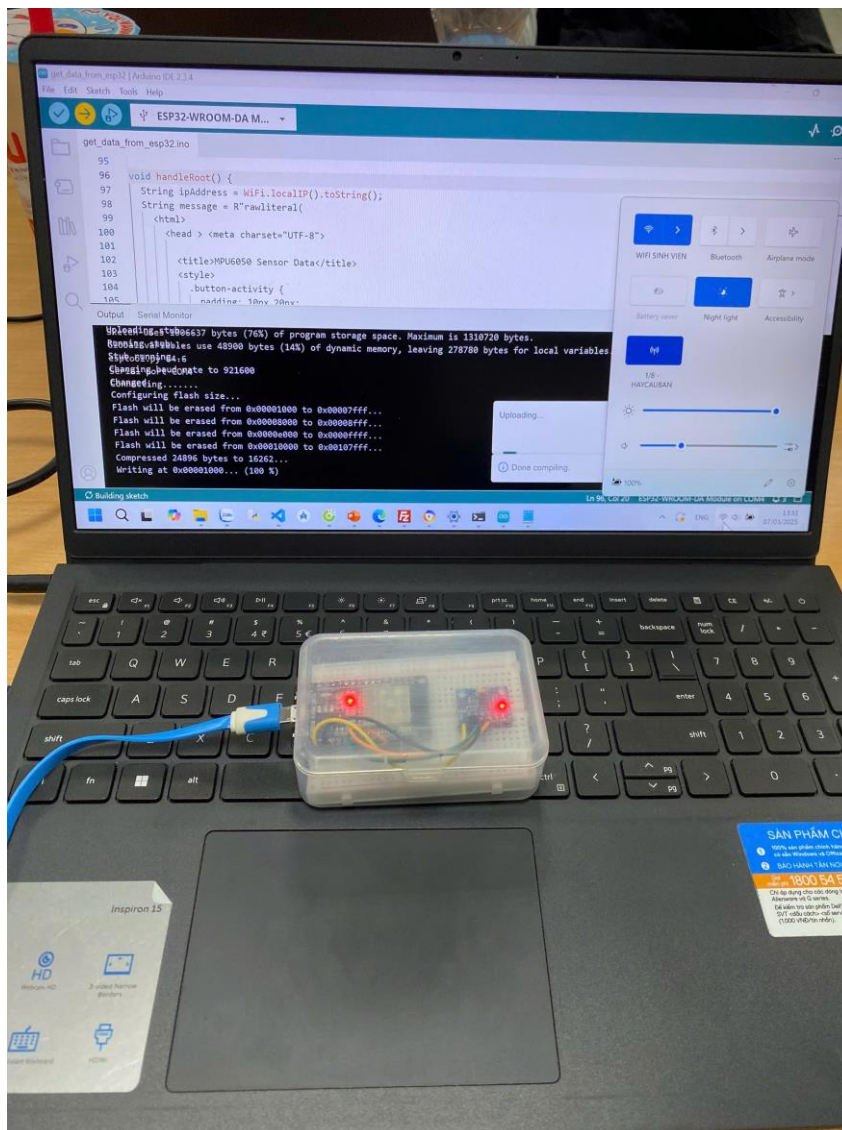
Bước 1: Thu dữ liệu.



Thiết bị thu dữ liệu

7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 1: Thu dữ liệu.



Nạp code Arduino

[Tải dữ liệu CSV](#)



MPU6050 Sensor Data

Fall_forward

Fall_backward

Fall_left

Fall_right

Standing

Hoạt động hiện tại: Chưa xác định

Dữ liệu cảm biến trực tiếp

Thời gian: 13247

Hoạt động: Chưa xác định

AccelX: 6.12 m/s²

AccelY: 0.05 m/s²

AccelZ: -8.65 m/s²

GyroX: -0.03 °/s

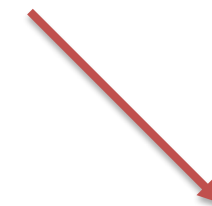
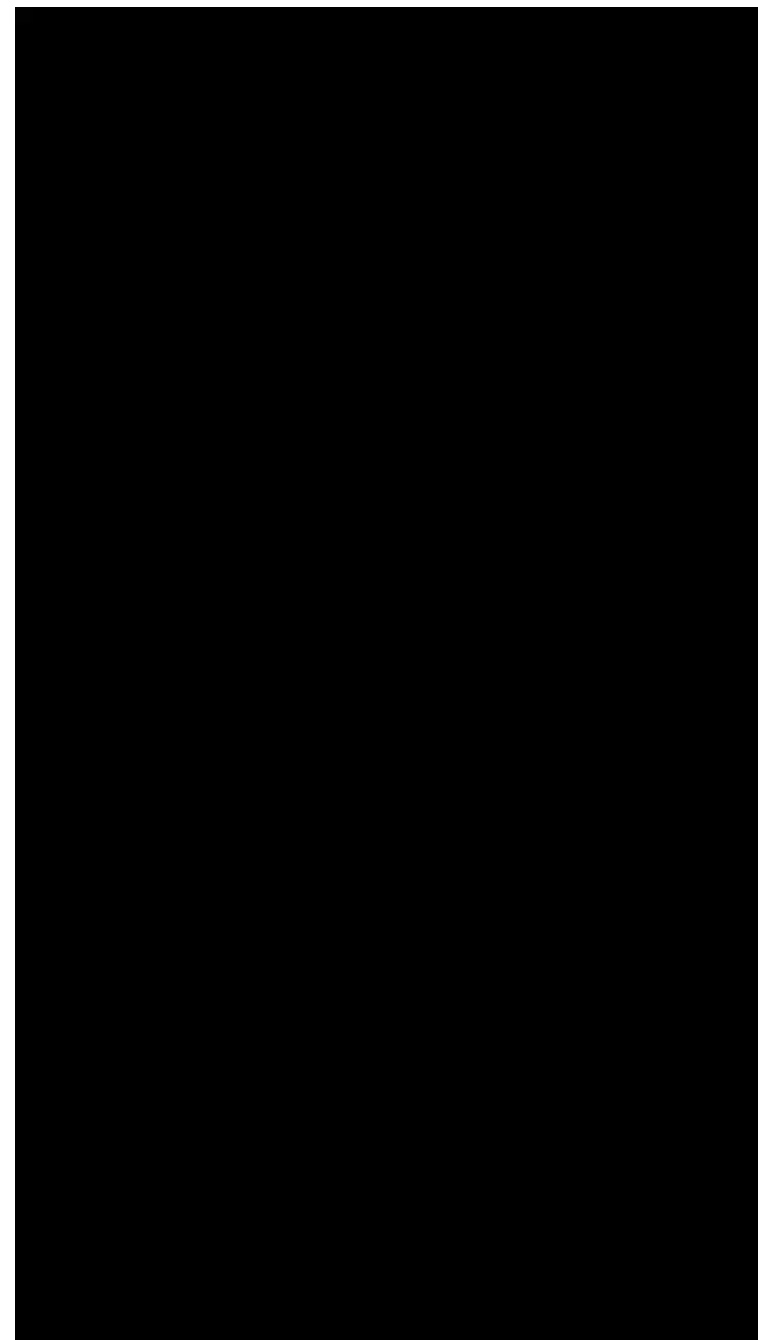
GyroY: -0.02 °/s

GyroZ: -0.04 °/s

Trang Web

7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 1: Thu dữ liệu.



[Tải dữ liệu CSV](#)

MPU6050 Sensor Data

Fall_forward

Fall_backward

Fall_left

Fall_right

Standing

Hoạt động hiện tại: Chưa xác định

Dữ liệu cảm biến trực tiếp

Thời gian: 13247

Hoạt động: Chưa xác định

AccelX: 6.12 m/s²

AccelY: 0.05 m/s²

AccelZ: -8.65 m/s²

GyroX: -0.03 °/s

GyroY: -0.02 °/s

GyroZ: -0.04 °/s

7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 1: Thu dữ liệu.

Time	AccelX	AccelY	AccelZ	GyroX	GyroY	GyroZ	ActivityLabel
379991	9.59	-1.7	2.84	0.03	-0.03	-0.06	fall_forward
380010	9.39	-1.61	3	0.04	-0.02	-0.06	fall_forward
380030	9.47	-1.63	3.14	0.01	-0.04	-0.05	fall_forward
380050	9.53	-1.64	2.97	-0.02	-0.03	-0.06	fall_forward
380070	9.5	-1.65	2.91	-0.01	-0.02	-0.06	fall_forward
380090	9.51	-1.67	2.87	0.01	-0.03	-0.06	fall_forward
380110	9.59	-1.65	3.07	0	-0.04	-0.06	fall_forward
380130	9.65	-1.73	2.87	0.01	-0.02	-0.05	fall_forward
380150	9.51	-1.66	2.98	0.03	0	-0.05	fall_forward
380170	9.54	-1.6	3.12	0.01	0.01	-0.05	fall_forward
380190	9.5	-1.7	2.96	0	-0.01	-0.05	fall_forward
380210	9.55	-1.64	2.9	0.02	-0.04	-0.04	fall_forward
380230	9.47	-1.59	2.95	0.03	-0.03	-0.04	fall_forward
380250	9.53	-1.62	3.05	0.02	-0.03	-0.04	fall_forward
380270	9.46	-1.58	3.13	0.01	-0.03	-0.04	fall_forward
380290	9.52	-1.56	3.06	0	-0.04	-0.03	fall_forward
380310	9.57	-1.56	3.04	0.01	-0.04	-0.04	fall_forward
380330	9.61	-1.58	2.94	0.01	-0.04	-0.03	fall_forward
380350	9.49	-1.68	2.94	0	-0.03	-0.03	fall_forward
380370	9.49	-1.58	3.02	0.01	-0.02	-0.03	fall_forward
380390	9.5	-1.63	2.98	0.02	-0.04	-0.03	fall_forward
380410	9.54	-1.57	3.04	0.01	-0.03	-0.03	fall_forward

Dữ liệu có 5 loại nhãn hành động:

- fall_forward (ngã về phía trước): 6.855 mẫu
- fall_left (ngã sang trái): 2.542 mẫu
- fall_backward (ngã ngửa): 1.228 mẫu
- fall_right (ngã sang phải): 915 mẫu
- standing (đứng yên): 660 mẫu

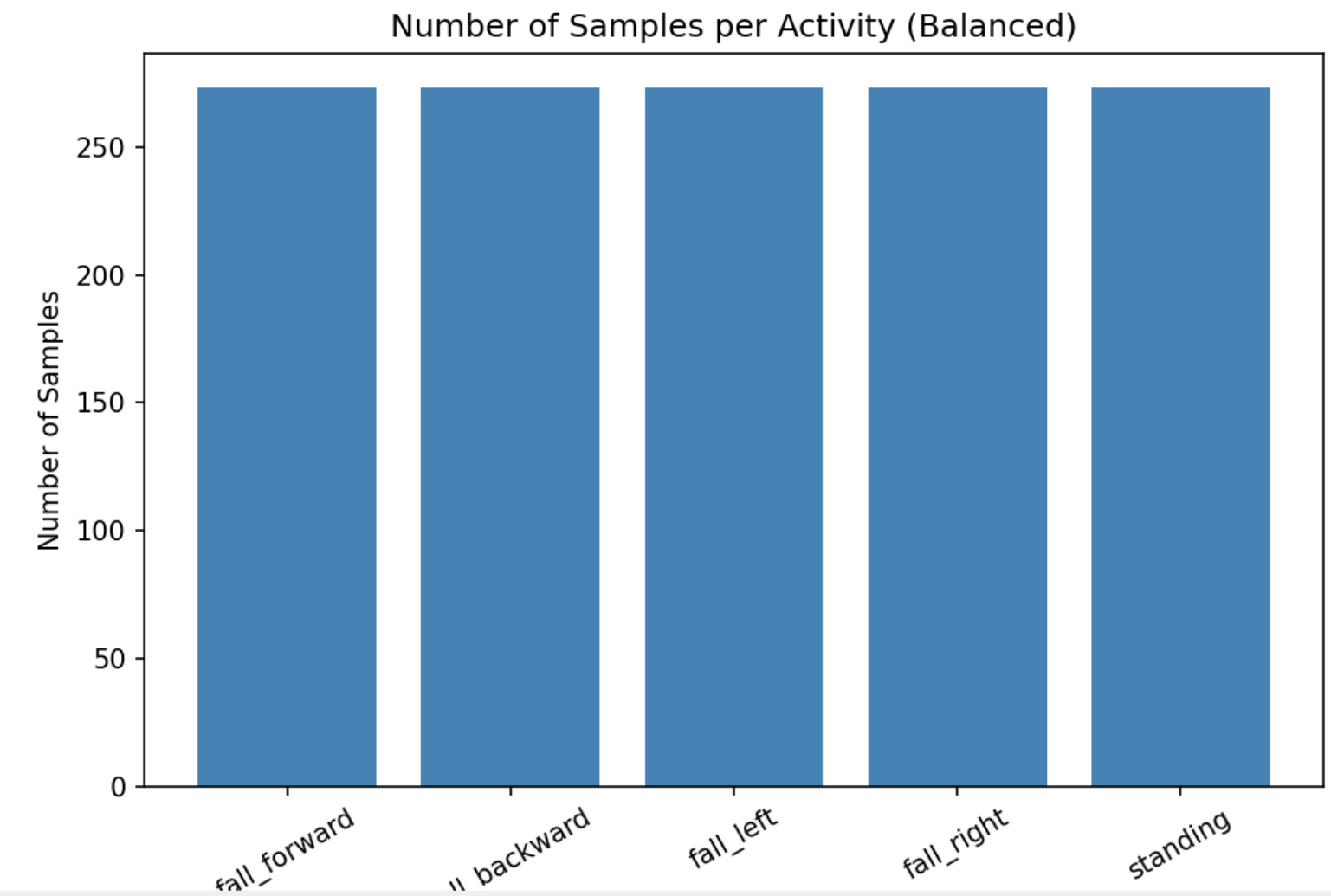
7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 2: Tiền xử lý dữ liệu.

```
37 X = np.array(X_windows) # Shape: (num_windows, 50, 6)
38 y = np.array(y_labels) # Shape: (num_windows,)
39
40 # 6. Xử lý mất cân bằng dữ liệu (SMOTE)
41 smote = SMOTE(sampling_strategy="auto", random_state=42)
42 X_flat = X.reshape(X.shape[0], -1) # Reshape để fit_resample
43 X_resampled, y_resampled = smote.fit_resample(X_flat, y)
44 X_resampled = X_resampled.reshape(-1, 50, 6) # Đưa về dạng ban đầu
45
46 # 7. Lưu dữ liệu đã cân bằng
47 save_path = "D:/KY2/AIoT/BTL/"
```

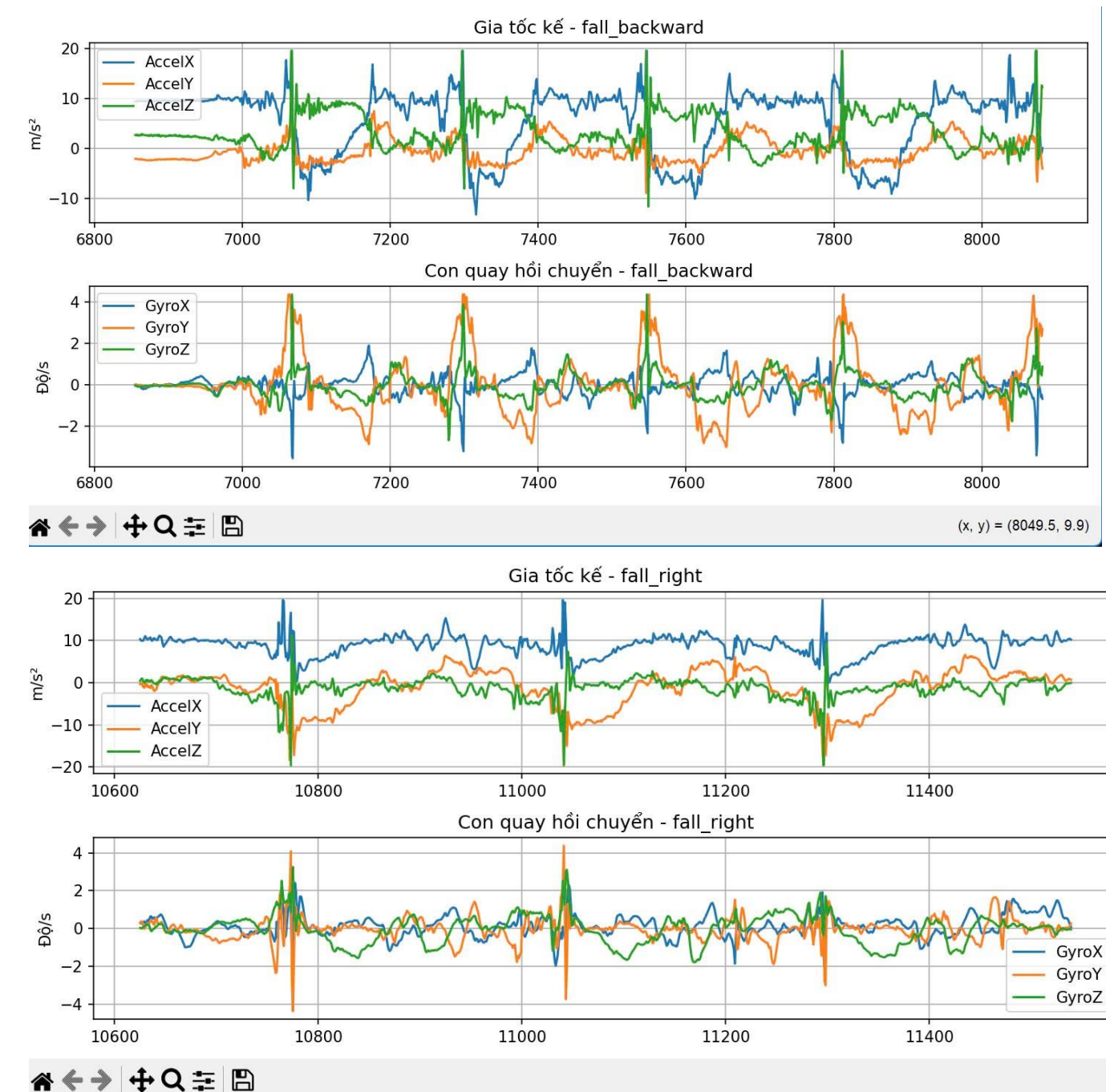
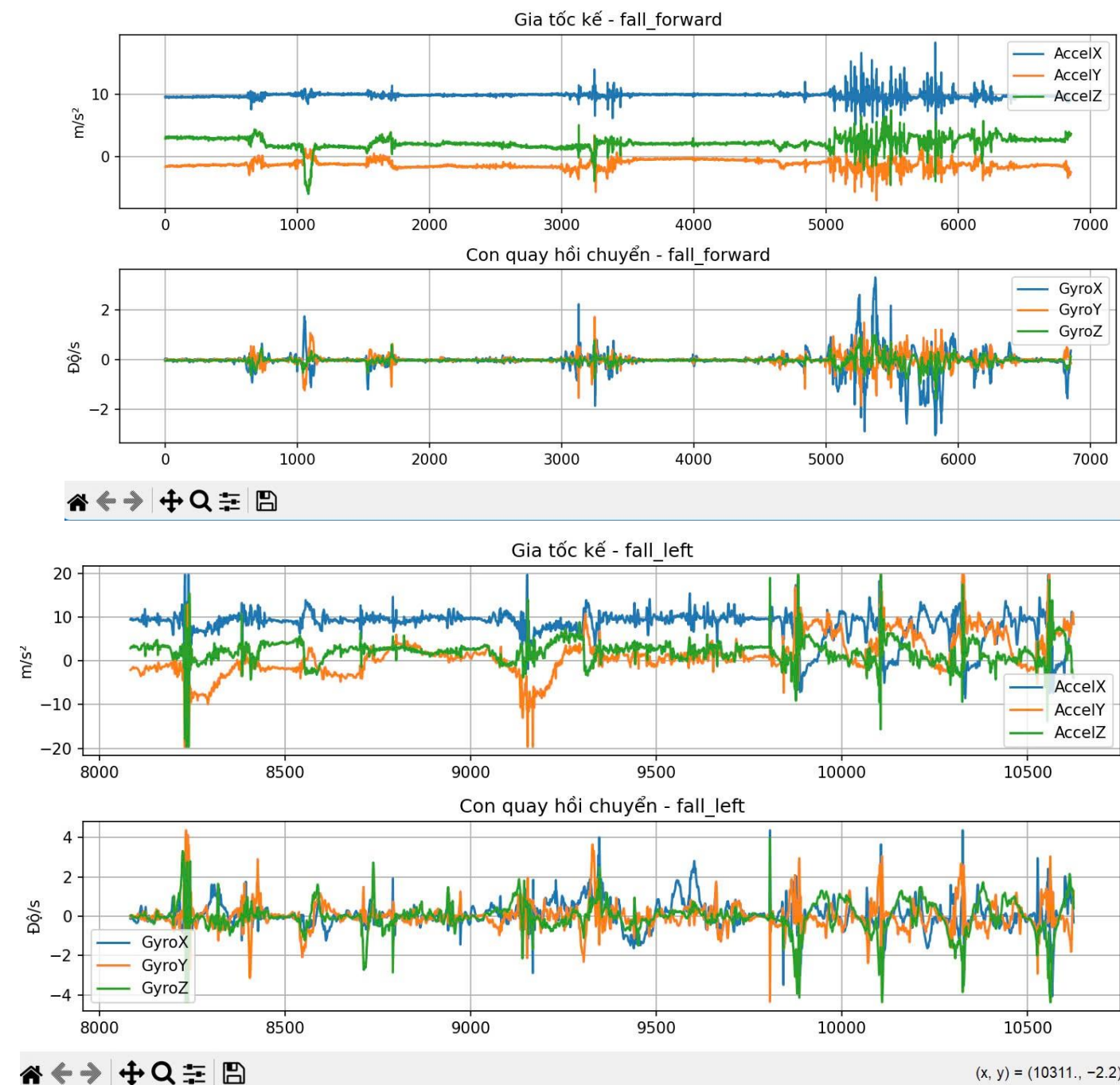
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE PORTS python - BTL +

```
(Ngoc) PS D:\KY2\AIoT\BTL> python .\client1.py
Số lượng của số sau khi cân bằng: 1365
Shape của X: (1365, 50, 6)
Số lượng của số sau khi cân bằng: 1365
Shape của X: (1365, 50, 6)
Shape của X: (1365, 50, 6)
Shape của y: (1365,)
```



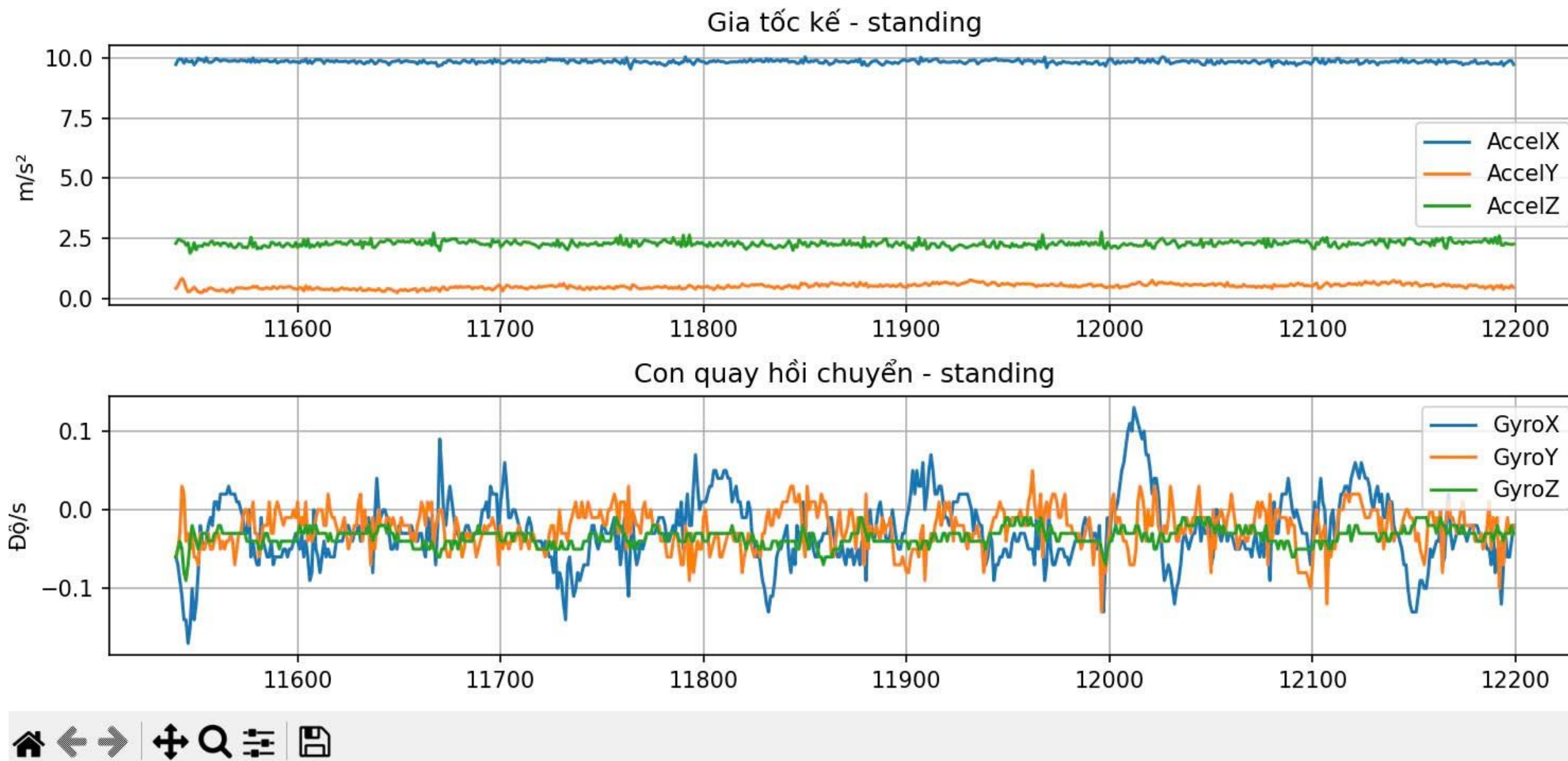
7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 3: Trực quan hóa dữ liệu.



7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 3: Trực quan hóa dữ liệu.



7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

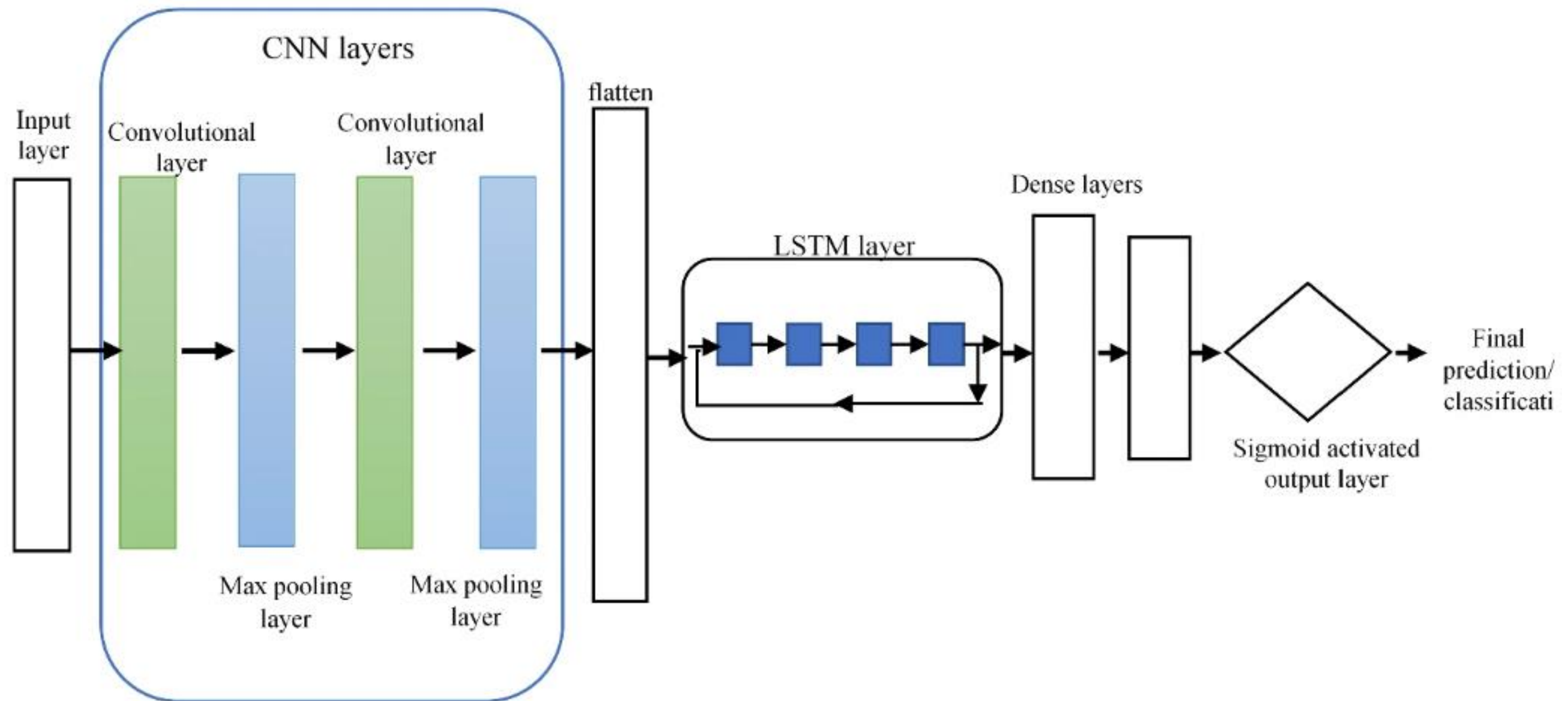
Bước 4: Huấn luyện mô hình.

Convolutional Neural Network + Long Short-Term Memory.

- CNN (Convolutional Neural Network): Mạng nơ-ron tích chập, thường được dùng để trích xuất đặc trưng từ dữ liệu có cấu trúc không gian, chẳng hạn như hình ảnh hoặc chuỗi thời gian.
- LSTM (Long Short-Term Memory): Một dạng đặc biệt của mạng nơ-ron hồi quy (RNN) có khả năng ghi nhớ thông tin trong thời gian dài, giúp xử lý dữ liệu chuỗi như tín hiệu cảm biến hoặc văn bản.

7. CÁCH THỨC TRIỂN KHAI ĐỀ

Bước 4: Huấn luyện mô hình.



Sơ đồ mô hình huấn luyện CNN + LSTM

7. CÁCH THỨC TRIỂN KHAI ĐỀ

Bước 4: Huấn luyện mô hình.

CNN + LSTM

- **Training accuracy:** Đạt 100% ở epoch cuối
→ Mô hình học rất tốt trên tập huấn luyện.
- **Validation accuracy:** Khoảng **92.36%**
→ Khá cao nhưng thấp hơn huấn luyện, có thể có hiện tượng overfitting.
- **Hàm mất mát (loss):**
 - **Training loss:** Giảm mạnh về gần 0.
 - **Validation loss:** Không giảm đều, có lúc tăng, có thể do mô hình chưa tối ưu hoặc dữ liệu chưa đủ đa dạng.

```
8/3/8/3 _____ 11s 12ms/step - accuracy: 1.0000 - loss: 3.0145e-04 - val_accuracy: 0.9498 - val_loss: 0.3686
Epoch 32/50
873/873 _____ 21s 13ms/step - accuracy: 0.9897 - loss: 0.0580 - val_accuracy: 0.9315 - val_loss: 0.2964
Epoch 33/50
873/873 _____ 20s 12ms/step - accuracy: 0.9937 - loss: 0.0197 - val_accuracy: 0.9361 - val_loss: 0.2979
Epoch 34/50
873/873 _____ 20s 12ms/step - accuracy: 0.9977 - loss: 0.0164 - val_accuracy: 0.9406 - val_loss: 0.3026
Epoch 35/50
873/873 _____ 21s 13ms/step - accuracy: 0.9948 - loss: 0.0176 - val_accuracy: 0.9269 - val_loss: 0.3390
Epoch 36/50
873/873 _____ 20s 12ms/step - accuracy: 0.9993 - loss: 0.0055 - val_accuracy: 0.9132 - val_loss: 0.5180
Epoch 37/50
873/873 _____ 20s 12ms/step - accuracy: 0.9959 - loss: 0.0143 - val_accuracy: 0.9361 - val_loss: 0.5085
Epoch 38/50
873/873 _____ 21s 13ms/step - accuracy: 0.9917 - loss: 0.0598 - val_accuracy: 0.9361 - val_loss: 0.3896
Epoch 39/50
873/873 _____ 20s 13ms/step - accuracy: 0.9968 - loss: 0.0201 - val_accuracy: 0.9361 - val_loss: 0.4745
Epoch 40/50
873/873 _____ 10s 12ms/step - accuracy: 0.9824 - loss: 0.0671 - val_accuracy: 0.9406 - val_loss: 0.3289
Epoch 41/50
873/873 _____ 19s 11ms/step - accuracy: 0.9985 - loss: 0.0027 - val_accuracy: 0.9361 - val_loss: 0.6697
Epoch 42/50
873/873 _____ 11s 12ms/step - accuracy: 0.9876 - loss: 0.0651 - val_accuracy: 0.9498 - val_loss: 0.2429
Epoch 43/50
873/873 _____ 11s 13ms/step - accuracy: 0.9958 - loss: 0.0100 - val_accuracy: 0.9543 - val_loss: 0.3060
Epoch 44/50
873/873 _____ 11s 12ms/step - accuracy: 0.9830 - loss: 0.0644 - val_accuracy: 0.9224 - val_loss: 0.4007
Epoch 45/50
873/873 _____ 11s 12ms/step - accuracy: 0.9916 - loss: 0.0236 - val_accuracy: 0.9498 - val_loss: 0.2872
Epoch 46/50
873/873 _____ 10s 12ms/step - accuracy: 0.9976 - loss: 0.0092 - val_accuracy: 0.9361 - val_loss: 0.3903
Epoch 47/50
873/873 _____ 11s 12ms/step - accuracy: 0.9789 - loss: 0.0683 - val_accuracy: 0.9589 - val_loss: 0.2331
Epoch 48/50
873/873 _____ 21s 13ms/step - accuracy: 0.9953 - loss: 0.0286 - val_accuracy: 0.9452 - val_loss: 0.2425
Epoch 49/50
873/873 _____ 20s 12ms/step - accuracy: 1.0000 - loss: 7.3069e-04 - val_accuracy: 0.9498 - val_loss: 0.2795
Epoch 50/50
873/873 _____ 19s 11ms/step - accuracy: 1.0000 - loss: 1.4065e-04 - val_accuracy: 0.9498 - val_loss: 0.2905
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file fr
Inference Time: 2.884757 s/mẫu
✅ Đã lưu mô hình vào fall_detection_model.h5

Validation Loss: 0.3532
Validation Accuracy: 0.9236
```

7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 4: Huấn luyện mô hình.

Đánh giá mô hình:

- Dữ liệu giữa các lớp khá cân bằng, hoặc mô hình phân loại đều tốt cho mọi lớp.
- > Mô hình duy trì chất lượng dự đoán rất ổn định trên tất cả các lớp.

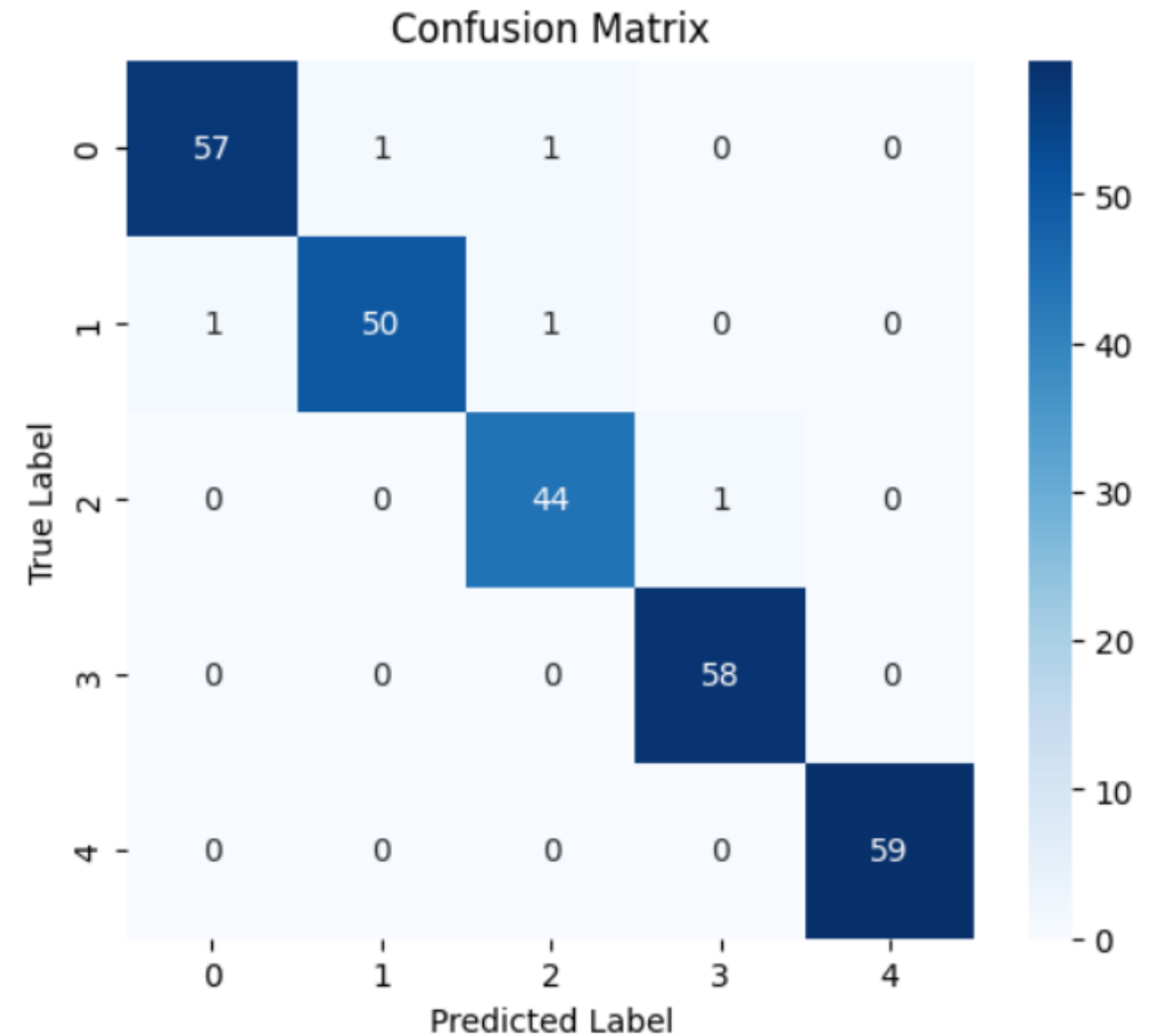
Action class	Precision	Recall	F1-score	Support
0.0 (Lean forward)	96.6%	96.6%	96.6%	59 mẫu
1.0 (Lean back)	98.0%	94.2%	96.1%	52 mẫu
2.0 (Lean to the left)	95.7%	97.8%	96.7%	45 mẫu
3.0 (Lean to the right)	98.3%	100%	99.1%	58 mẫu
4.0 (Stand normally)	100%	100%	100%	59 mẫu

7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 4: Huấn luyện mô hình.

Nhận xét:

- Hai lớp 3 và 4 có dự đoán hoàn toàn chính xác (không có mẫu nào bị nhầm).
- Lớp 0, 1 và 2 có một vài trường hợp dự đoán nhầm, nhưng số lượng rất ít.
- Nhìn chung, đường chéo chính có giá trị rất cao, thể hiện mô hình có độ chính xác cao trong việc phân loại.
- Màu sắc đậm ở đường chéo (theo thang màu bên phải) cho thấy số mẫu đúng chiếm ưu thế vượt trội so với các mẫu sai.

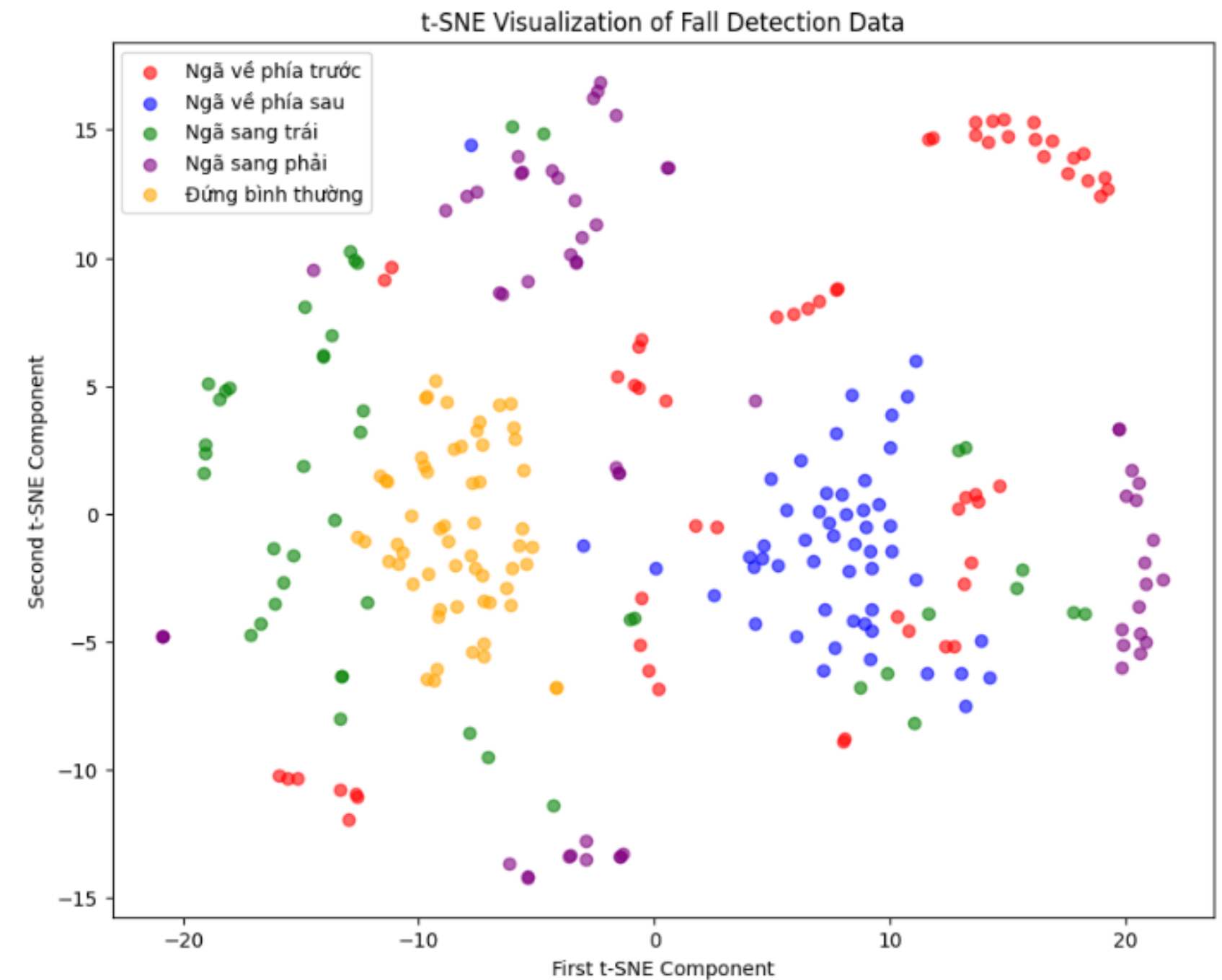


7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 4: Huấn luyện mô hình.

Nhận xét:

- Các lớp hành động được phân tách khá tốt
- Lớp "Đứng bình thường" có phạm vi rộng



7. CÁCH THỨC TRIỂN KHAI ĐỀ TÀI

Bước 5: Kiểm tra mô hình đã huấn luyện.

Xây dựng một ứng dụng web sử dụng Flask để nhận dữ liệu cảm biến từ ESP32 + MPU6050

Xây dựng code Arduino để thu dữ liệu cảm biến

Chạy Flask Server trên máy tính

Nạp code vào ESP32 và chạy

8. THỰC NHIỆM ĐÁNH GIÁ KẾT QUẢ

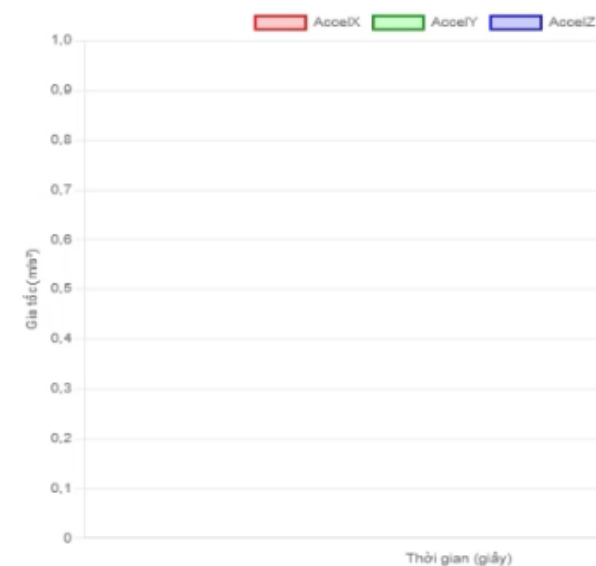
PHÁT HIỆN HÀNH ĐỘNG TẾ NGÃ

HÀNH ĐỘNG HIỆN TẠI:

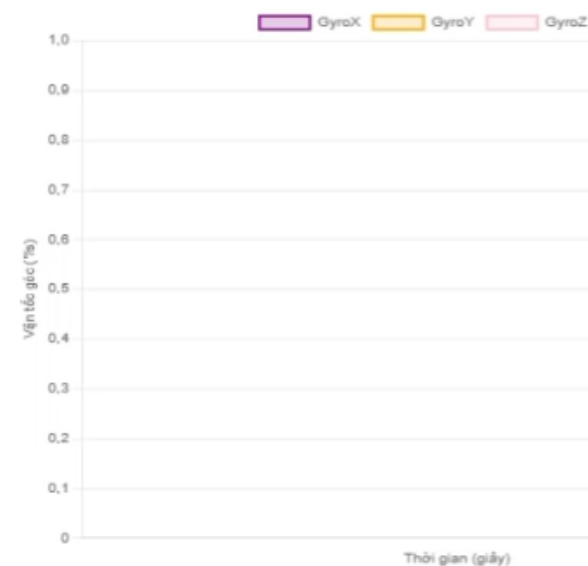
ĐỨNG BÌNH THƯỜNG

Biểu đồ dữ liệu cảm biến

Gia tốc kế



Con quay hồi chuyển



Thông báo



9. ƯU ĐIỂM, HẠN CHẾ

Ưu điểm

- Chi phí thấp, tiết kiệm năng lượng.
- Gửi cảnh báo kịp thời qua tin nhắn.
- Dễ triển khai, có thể mở rộng.
- Lưu trữ và phân tích dữ liệu.

Hạn chế

- Độ chính xác bị ảnh hưởng bởi nhiễu và tư thế té ngã phức tạp.
- Cần kết nối mạng ổn định, có thể bị trễ khi xử lý dữ liệu.
- Hạn chế phần cứng, cần bảo mật dữ liệu khi truyền tải.

10. KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

Kết luận

- Hệ thống AloT để phát hiện và giám sát ngã có độ chính xác cao
- Phản ứng nhanh và hoạt động ổn định
- Giúp theo dõi sức khỏe và cải thiện chất lượng cuộc sống cho người cao tuổi thông qua các ứng dụng học sâu và tối ưu hóa trên các thiết bị nhúng.

10. KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

Phương hướng phát triển

- Mở rộng dữ liệu để tăng độ chính xác
- Tích hợp AI tiên tiến để nhận diện ngữ cảnh và giảm thiểu cảnh báo sai
- Áp dụng nó vào các thiết bị đeo thông minh để theo dõi sức khỏe toàn diện.



Thank You