Nhận dạng hoạt động con người

1st Đinh Nguyễn Gia Bảo - 22127027
Khoa Công nghệ Thông tin
Trường Đại học Khoa học Tự nhiên, ĐHQG-TPHCM
Thành phố Hồ Chí Minh, Việt Nam

0358180131

2nd Hoàng Bảo Khanh - 22127183
Khoa Công nghệ Thông tin
Trường Đại học Khoa học Tự nhiên, ĐHQG-TPHCM
Thành phố Hồ Chí Minh, Việt Nam

0932164473

***Tóm tắt nội dung*—Nhận dạng hoạt động của con người (HAR) là một trong những đề tài phổ biến, thu hút nhiều sự quan tâm từ các nhà khoa học nhờ vào các ứng dụng liên quan đến chăm sóc sức khỏe, theo dõi thể chất và tương tác giữa người và máy. Nghiên cứu này đề xuất một hệ thống HAR để phân loại các hoạt động của con người với độ chính xác cao. Mô hình đề xuất sẽ trải qua các bước tiền xử lý dữ liệu, trích xuất đặc trưng và tối ưu hóa để cải thiện hiệu suất. Kết quả thực nghiệm sẽ cho biết độ chính xác của mô hình trên các tập dữ liệu chuẩn, vượt trội so với các phương pháp truyền thống. Những phát hiện này nhằm nhấn mạnh tiềm năng của mô hình nhận dạng hoạt động con người trong các ứng dụng thực tế.**

I. Giới thiệu

Human Activity Recognition (HAR) là một đề tài nghiên cứu quan trọng trong khoa học máy tính và trí tuệ nhân tạo, có tính ứng dụng cao trong thực tiễn khi kết hợp với nhiều mô hình như học máy, học sâu nhằm phục vụ mục đích nhận diện và phân loại các hoạt động của con người. Bên cạnh đó, với sự phát triển mạnh mẽ của hệ thống Internet of Things (IOT) thì các hệ thống thông minh trong nhà như chống trộm, giám sát an ninh, phát hiện hành vi bất thường tại ngân hàng hay sân bay đang bùng nổ mạnh mẽ hơn bao giờ hết. Điều này cho thấy vai trò và tiềm năng của đề tài này là vô cùng to lớn.

A. Ý nghĩa về khoa học

Việc nghiên cứu và phát triển các phương pháp nhận diện hoạt động của con người không chỉ đóng góp nhiều vào sự hiểu biết về các hành vi, thói quen sinh hoạt của một người, mà còn trở thành nền tảng, từ đó tạo ra các nghiên cứu sâu hơn về tâm lý học, khoa học hành vi và tương tác giữa người và máy. Các nghiên cứu trong HAR còn có các đóng góp trong học máy, xử lý tín hiệu, nhận dạng mẫu và tự động hóa. Những nghiên cứu trong lĩnh vực này không chỉ cải thiện mô hình AI, mà còn tạo ra những đột phá trong việc hiểu và mô hình hóa hành vi con người, mở ra nhiều nghiên cứu liên ngành.

B. Ý nghĩa về ứng dụng

Về mặt ứng dụng, HAR được sử dụng rộng rãi trong các sản phẩm như hệ thống thông minh, giúp phát hiện, theo dõi các thành viên trong gia đình khi ở xa, cũng như phát hiện các hành động bất thường và cảnh báo kịp thời. Ngoài ra, đề tài này còn được ứng dụng trong lĩnh vực y tế để giúp các bác sĩ, y tá theo dõi các hành động của bệnh nhân nhằm có được sự đánh giá trong quá trình điều trị bệnh, cũng như phát hiện các triệu chứng bất thường. Hoặc trong lĩnh vực tài chính ngân hàng, các sản phẩm HAR còn được cài đặt trong các thiết bị theo dõi như camera, nhằm kiểm soát, phát hiện và cảnh báo những hành động có ý đồ xấu gây nguy hại đến sự an toàn của người dân trong khu vực.

C. Phát biểu bài toán

Đầu vào - đầu ra: Hệ thống nhận dạng hoạt động (Human Activity Recognition) HAR với đầu vào là đoạn video chứa đối tượng và bối cảnh. Sau đó, chương trình sử dụng mô hình đã được huấn luyện dựa trên các nhãn (label) mà phân lớp cho video. Kết quả là ta thu được video kèm nhãn label.

Hình 1. Hình ảnh ví dụ về nhận dạng hoạt động con người

Dữ liệu: Dữ liệu mà chúng em dùng cho bài toán lần này là các bộ dữ liệu chuẩn liên quan đến hoạt động của con người. Cụ thể các bộ dữ liệu là UCF11 [1], UCF50 [11], UCF101 [10], HMDB101.

Môi trường và thư viện: Đồ án môn học này dự kiến sẽ được trình bày bằng Jupyter Notebook, sử dụng ngôn ngữ lập trình Python 3.10 trở lên. Công cụ phát triển chương trình chính là Visual Studio Code (VS Code), một IDE phổ biến với các tính năng mạnh mẽ, hỗ trợ tốt cho Python và Jupyter Notebook.

Còn về thư viện sử dụng, chúng em dự định sẽ xây dựng mô hình có thể bằng TensorFlow hoặc Pytorch, ngoài ra còn sử dụng numpy trong việc tính toán, pandas cho thống kê dữ liệu, scikit-learn, matplotlib và seaborn trong trực quan hóa dữ

liệu và hiệu suất của mô hình. Hệ thống nhận dạng dự kiến sẽ bao gồm các công đoạn chính như sau:

1) Tải và đọc dữ liệu UCF-11
2) Tiền xử lý dữ liệu video: chuyển đổi video thành khung, thay đổi kích thước hoặc chuẩn hóa dữ liệu,...
3) Trích xuất đặc trưng từ video: trong bước này chúng em dự kiến sẽ sử dụng mô hình CNN và LSTM, còn về lý do lựa chọn giải pháp trên thì sẽ được trình bày sau bên dưới.
4) Thiết kế, xây dựng mô hình **CNN-LSTM**
5) Huấn luyện mô hình
6) Thực hiện dự đoán và đánh giá hiệu suất của mô hình.

II. Các công trình nghiên cứu liên quan

A. Phương pháp truyền thống

Trong những năm qua, nhiều nghiên cứu đã đề xuất các phương pháp dựa trên đặc trưng thủ công và mạng nơ-ron học sâu cho bài toán nhận dạng hoạt động. Các phương pháp trước đây chủ yếu dựa vào các đặc trưng thủ công để nhận diện một số hoạt động đơn giản. Các hệ thống này trích xuất các đặc trưng cấp thấp từ dữ liệu video, qua đó đưa vào các bộ phân loại như máy vector hỗ trợ (SVM), cây quyết định và KNN để nhận dạng.

Lấy ví dụ, Yilmaz và Shah [2] đã nghiên cứu các thuộc tính hình học của thể tích không-thời gian (STV) để tạo ra sự biểu diễn bằng cách phát họa. Họ xếp chồng các đường viền có thể theo trục thời gian, thu thập các thông tin về hướng, tốc độ và hình dạng của STV để nhận dạng hoạt động. Vào năm 2007, Gorelick et al. [3] đã biểu diễn hoạt động con người dưới dạng cấu trúc ba chiều được hình thành từ bóng của cơ thể trong STV. Họ áp dụng phương trình Poisson để phân tích các dạng hình ảnh 2D của hoạt động và rút trích các đặc trưng không-thời gian STF. Tuy nhiên, vì sử dụng tập dữ liệu phi thực tế, phương pháp này gặp vấn đề khi một số hành động khác nhau có thể tạo ra hình dạng 2D tương tự trong STV, qua đó làm giảm khả năng phân biệt giữa các hoạt động.

Ngoài ra, phương pháp wavelet rời rạc cũng là một phương pháp thủ công. Đây là hướng tiếp cận phổ biến trong HAR, khi sử dụng mô hình quỹ đạo dày đặc (Dense Trajectories) cho phép theo dõi chuyển động của các điểm đặc trưng trong không gian - thời gian. Tuy nhiên, phương pháp gặp hạn chế trong việc phân biệt các chuyển động chính và thứ cấp ở các dải tần số khác nhau.

Để khắc phục vấn đề trên, nghiên cứu của Chaolong Zhang [4] đã đề xuất tích hợp biến đổi wavelet rời rạc (Discrete Wavelet Transform - DWT) vào quỹ đạo trên. Điều đó giúp mô hình có thể phân tách các video thành các thành phần có tần số và hướng chuyển động khác nhau. Tuy nhiên về mặt nhược điểm, việc áp dụng DWT đòi hỏi thêm tài nguyên tính toán, đồng thời yêu cầu tinh chỉnh tham số phù hợp với mức phân giải của wavelet. Bên cạnh đó, đối với những video chứa dữ liệu nhiễu cao, thì phương pháp DWT chỉ cải thiện nhiễu một phần.

B. Nhận dạng hoạt động của con người bằng Multi-Head CNN và LSTM

Tuy nhiên, các phương pháp truyền thống như SVM, KNN và Random Forest đã được áp dụng rộng rãi nhưng gặp nhiều hạn chế trong việc xử lý dữ liệu chuỗi thời gian phức tạp. Để cải thiện độ chính xác và khả năng tổng quát hóa của mô hình, Waqar Ahmad [5] đã đề xuất một phương pháp *Multi-Head CNN-LSTM*, trong đó ba mạng CNN hoạt động song song để trích xuất đặc trưng từ dữ liệu cảm biến gia tốc (accelerometer) và con quay hồi chuyển (gyroscope). Đầu ra của các CNN này sau đó được hợp nhất và đưa vào một lớp LSTM để mô hình hóa mối quan hệ thời gian, giúp nhận diện chính xác các hoạt động như đi bộ, chạy, ngồi và nằm.

Cấu trúc của mô hình Multi-Head CNN-LSTM bao gồm ba CNN hoạt động độc lập với cùng một cấu hình để xử lý ba loại dữ liệu cảm biến khác nhau: tổng gia tốc, gia tốc cơ thể và dữ liệu con quay hồi chuyển. Các CNN này có bốn lớp tích chập (1D-CNN) với kích thước bộ lọc 3×1, xen kẽ với các lớp Max-Pooling để giảm chiều dữ liệu. Sau khi hợp nhất đầu ra của ba CNN, dữ liệu được đưa vào lớp LSTM với 128 đơn vị nhớ, sau đó đi qua một lớp Dense với 1000 neuron và cuối cùng là lớp Softmax để phân loại hoạt động. Nhóm tác giả sử dụng thuật toán Adam với learning rate 0.001, huấn luyện mô hình trong 17 epochs trên tập dữ liệu UCI HAR.

Kết quả thực nghiệm cho thấy phương pháp *Multi-Head CNN-LSTM đạt độ chính xác 95.76%*, cao hơn so với mô hình *Single CNN-LSTM (94.1%)* và vượt trội hơn các phương pháp máy học truyền thống như *SVM (84%)* và *KNN (90%)*. Việc chia dữ liệu thành ba luồng song song giúp tối ưu hóa khả năng trích xuất đặc trưng, giúp mô hình học sâu hơn và chính xác hơn trong nhận diện hoạt động con người. Kết quả này cũng chứng minh rằng thay vì sử dụng một mạng CNN duy nhất, việc sử dụng nhiều CNN song song có thể tăng khả năng nắm bắt đặc trưng quan trọng trong dữ liệu cảm biến.

C. MultiWave: Kiến trúc sâu đa độ phân giải thông qua phân tách wavelet để dự đoán chuỗi thời gian đa biến

MultiWave, được đề xuất bởi Deznabi và Fiterau [6], là một kiến trúc học sâu sử dụng biến đổi wavelet rời rạc (DWT) để phân rã tín hiệu chuỗi thời gian thành các thành phần có tần số khác nhau, giúp tận dụng cả thông tin miền thời gian và miền tần số trong phân tích dữ liệu đa biến. Phương pháp này khắc phục hạn chế của các mô hình học sâu truyền thống như LSTM, CNN và Transformer khi xử lý dữ liệu có tần số lấy mẫu không đồng nhất. MultiWave chia tín hiệu thành các dải tần số khác nhau và xử lý chúng bằng các thành phần mô hình riêng biệt, trong đó một cơ chế gating được sử dụng để hợp nhất các đặc trưng quan trọng nhằm tối ưu hóa hiệu suất.

Kết quả thực nghiệm cho thấy MultiWave đạt hiệu suất cao trong nhiều tác vụ, bao gồm nhận diện căng thẳng và cảm xúc từ thiết bị đeo thông minh, dự báo tỷ lệ tử vong do COVID-19 từ mẫu máu bệnh nhân và nhận diện hoạt động

con người (HAR) từ dữ liệu cảm biến. Trong bài toán HAR, MultiWave được áp dụng trên tập dữ liệu MHEALTH và giúp tăng độ chính xác của mô hình CNN-LSTM lên 96.48%, cao hơn 5.6% so với mô hình gốc. Điều này chứng tỏ rằng việc khai thác thông tin miền tần số có thể cải thiện đáng kể hiệu suất nhận diện hoạt động, đặc biệt khi xử lý dữ liệu cảm biến có tần số lấy mẫu khác nhau. Ngoài ra, trong dự báo tỷ lệ tử vong do COVID-19, MultiWave cũng giúp cải thiện 5% AUC so với mô hình deep learning tốt nhất trước đó, cho thấy khả năng xác định chính xác các thành phần tần số quan trọng liên quan đến nguy cơ tử vong, chẳng hạn như mức CRP, D-D dimer và LDH.

Những kết quả này nhấn mạnh tiềm năng của MultiWave trong việc tăng cường hiệu suất của các mô hình học sâu bằng cách tích hợp thông tin miền tần số, mở ra hướng đi mới cho các ứng dụng dự báo chuỗi thời gian đa biến trong y tế, giám sát và nhận diện hoạt động.

D. Nhận dạng hoạt động con người bằng mô hình CNN-LSTM-GRU

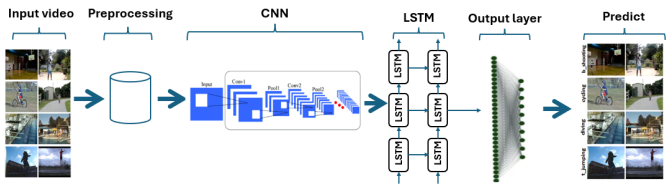
Trong bối cảnh phát triển của phương pháp này, Pandey [7] đã đề xuất một mô hình lai kết hợp Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU), nhằm khai thác đồng thời đặc trưng không gian và mối quan hệ thời gian trong dữ liệu cảm biến. Mô hình sử dụng CNN để trích xuất đặc trưng không gian từ dữ liệu đầu vào, trong khi LSTM học các mối quan hệ theo chuỗi thời gian, giúp mô hình nhận diện chính xác các hành vi có tính tuần hoàn hoặc biến đổi theo thời gian. Bên cạnh đó, GRU được tích hợp để tối ưu hóa quá trình ghi nhớ, giúp giảm độ phức tạp tính toán so với LSTM truyền thống nhưng vẫn giữ lại khả năng xử lý thông tin dài hạn. Nghiên cứu được thực hiện trên hai bộ dữ liệu chuẩn là iSPL và UCI HAR, với các hoạt động như đi bộ, chạy, đứng yên và ngồi.

Kết quả thực nghiệm cho thấy mô hình CNN-LSTM-GRU đạt *99% độ chính xác trên iSPL* và *92% trên UCI HAR*, cao hơn so với các phương pháp truyền thống dựa trên SVM và các kiến trúc học sâu khác như CNN-LSTM thuần túy. Những kết quả này nhấn mạnh hiệu quả của việc kết hợp CNN, LSTM và GRU, không chỉ giúp tăng độ chính xác mà còn cải thiện hiệu suất tính toán, tạo tiền đề cho việc ứng dụng mô hình vào các hệ thống nhúng và thiết bị di động nhằm nhận diện hoạt động theo thời gian thực.

III. Phương pháp và mô hình đề xuất

A. Thiết kế mô hình

Trong các nghiên cứu liên quan, phương pháp phổ biến được sử dụng nhiều là xây dựng mô hình mạng nơ-ron CNN-LSTM, với mô hình CNN sử dụng cho việc trích xuất đặc trưng và LSTM để thực hiện huấn luyện mô hình. Phương pháp này không chỉ phổ biến với các bài khảo sát hiện nay mà còn là một phương pháp trả về độ chính xác khá cao, lên tới 99% [8]. Trong phần đề xuất lần này, chúng em lấy ý tưởng chính



Hình 2. Mô hình luồng xử lý của hệ thống

từ mô hình CNN - Bidirectional LSTM từ tác giả Amin Ullah [9]. Cụ thể thì phương pháp chúng em thực hiện sẽ bao gồm những công đoạn sau:

1) Chương trình đọc dữ liệu từ bộ dữ liệu, từ đó chuyển đổi các video trên về các chuỗi khung hình (frames).
2) Thực hiện bước tiền xử lý dữ liệu nhằm cải thiện chất lượng các mẫu đầu vào cho mô hình. Các bước tiền xử lý có thể áp dụng các phương pháp làm mịn ảnh như Gaussian, điều chỉnh độ sáng hoặc tương phản sao cho phù hợp. Ngoài ra các phương pháp tiền xử lý có để được áp dụng như tăng cường dữ liệu (Data Augmentation), có thể thực hiện các phép biến đổi như xoay ảnh, lật ngang dọc để tăng độ đa dạng dữ liệu.
3) Dữ liệu sau khi được xử lý sẽ được đưa vào mô hình CNN để thực hiện trích xuất đặc trưng. Cấu trúc CNN thông thường bao gồm các tầng tích chập (Convolutional layers) và tầng gộp (Pooling layers).
4) CNN chỉ xử lý từng frame riêng lẻ. Do đó để xây dựng được mối quan hệ giữa các frame trong thời gian, mô hình còn sử dụng LSTM. Đây là phương pháp giúp nhận diện chuỗi thời gian từ đặc trưng CNN, giúp lưu trữ thông tin của các khung hình trước đó để hiểu hoạt động đang diễn ra.
5) LSTM trả về một vector đầu ra, sau đó đi qua lớp full connected để tạo ra xác suất của mỗi lớp hoạt động. Từ đó chương trình sử dụng hàm Softmax để dự đoán lớp hoạt động có xác suất cao nhất.
6) Cuối cùng, sau khi nhận được dự đoán từ mô hình, chương trình sẽ gán nhãn hoạt động vào video, đồng thời đánh giá hiệu suất hoạt động của mô hình bằng các thông số đánh giá như (Accuracy, Precision, Recall, F1-score,...).

B. Trích xuất đặc trưng

Trong bài toán nhận diện hành động từ video, việc trích xuất đặc trưng (features extraction) là một công đoạn thiết yếu nhằm chuyển đổi dữ liệu video thô thành dạng biểu diễn có ý nghĩa để mô hình học sâu có thể xử lý. Mỗi video bao gồm một chuỗi các khung hình theo thời gian, trong đó mỗi khung hình chứa đựng thông tin không gian (spatial) quan trọng cho việc nhận diện hành vi. Tuy nhiên, xử lý trực tiếp các video thô vừa tốn tài nguyên tính toán, vừa khó khăn trong việc khai thác tri thức. Do đó, việc sử dụng mô hình học sâu đã được huấn luyện để trích xuất đặc trưng không gian từ khung hình giúp tối ưu hóa hiệu quả học và giảm độ phức tạp mô hình.

1) Phát biểu bài toán: Cho một tập dữ liệu video được ký hiệu là:

$$\mathcal{D} = \{(V_i, y_i) \mid i = 1, 2, \ldots, N\}$$

trong đó:

- V_i là video thứ i,
- $y_i \in \mathcal{Y} = \{1, 2, \ldots, C\}$ là nhãn hành động tương ứng,
- N là số lượng video trong tập dữ liệu,
- C là tổng số lớp hành động.

Mỗi video V_i được lưu trữ như một chuỗi các khung hình liên tục:

$$V_i = \left[f_i^{(1)}, f_i^{(2)}, \ldots, f_i^{(T_i)}\right], \quad f_i^{(t)} \in \mathbb{R}^{H \times W \times 3}$$

Mục tiêu của công đoạn này là trích xuất một chuỗi đặc trưng có độ dài cố định từ mỗi video:

$$\mathbf{X}_i = \left[\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \ldots, \mathbf{x}_i^{(T)}\right] \in \mathbb{R}^{T \times D}$$

trong đó:

- T là số khung hình cố định được chọn từ mỗi video,
- D là chiều của vector đặc trưng không gian (ví dụ: $D = 2048$),
- $\mathbf{x}_i^{(t)} = \phi(f_i^{(t)}) \in \mathbb{R}^D$ là đặc trưng không gian trích xuất từ khung hình $f_i^{(t)}$ thông qua hàm ánh xạ ϕ.

Bài toán đặt ra là xây dựng thành công một pipeline trích xuất đặc trưng:

$$(V_i, y_i) \xrightarrow{\text{Features Extraction}} (\mathbf{X}_i, y_i)$$

nhằm ánh xạ video đầu vào thành chuỗi đặc trưng ngữ nghĩa, phục vụ cho mô hình tuần tự phía sau như Bi-LSTM.

2) Phương pháp và hướng tiếp cận: Trong nghiên cứu này, chúng tôi lựa chọn phương pháp **fine-tuning mô hình ResNet50** được huấn luyện sẵn (pretraining) trên tập dataset **ImageNet** để trích xuất đặc trưng không gian từ từng khung hình. ResNet50 là một mô hình học sâu mạnh mẽ với kiến trúc residual giúp khắc phục hiện tượng mất gradient khi huấn luyện mạng sâu. Thay vì sử dụng toàn bộ mô hình, chúng tôi loại bỏ lớp phân loại cuối cùng (Fully Connected layer) và chỉ giữ lại phần convolutional làm bộ trích xuất đặc trưng. Điều này cho phép mỗi ảnh đầu vào được mã hóa thành một vector 2048 chiều chứa thông tin hình ảnh ngữ nghĩa nhưng không gắn với nhãn cụ thể nào.

Về mặt toán học, mô hình ResNet50 có thể được biểu diễn như một hàm ánh xạ:

$$\phi_{\text{ResNet}} : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^D$$

trong đó:

- $H \times W \times 3$ là kích thước của khung hình ảnh đầu vào (ví dụ: $224 \times 224 \times 3$),
- $D = 2048$ là số chiều của vector đặc trưng đầu ra.

Mô hình ResNet50 bao gồm chuỗi các khối residual $\mathcal{F}_l$ với $l = 1, \ldots, L$, và có thể được biểu diễn như sau:

$$\phi_{\text{ResNet}}(\mathbf{I}) = \mathbf{z}_L = \mathcal{F}_L(\mathcal{F}_{L-1}(\ldots \mathcal{F}_1(\mathbf{I}) \ldots))$$

Trong đó $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ là ảnh đầu vào, và $\mathbf{z}_L \in \mathbb{R}^{2048}$ là vector đặc trưng sau khi đã qua toàn bộ các khối residual và một phép pooling toàn cục.

Cụ thể, mỗi khối residual thực hiện phép biến đổi có dạng:

$$\mathbf{z}_{l+1} = \sigma\left(\mathcal{F}_l(\mathbf{z}_l) + \mathbf{z}_l\right)$$

với:

- $\mathcal{F}_l(\cdot)$ là tổ hợp của các lớp convolution, batch normalization, và hàm kích hoạt ReLU,
- σ là hàm phi tuyến (thường là ReLU),
- $\mathbf{z}_l$ là đầu vào của tầng residual thứ l,
- $\mathbf{z}_{l+1}$ là đầu ra của tầng thứ l.

Sau khi đi qua toàn bộ các tầng L, đầu ra sẽ được xử lý bằng một phép *Global Average Pooling*:

$$\mathbf{z}_{\text{GAP}} = \frac{1}{H' \cdot W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} \mathbf{f}_{i,j}$$

trong đó:

- $\mathbf{f}_{i,j} \in \mathbb{R}^D$ là vector đặc trưng tại vị trí (i, j) trong feature map cuối cùng,
- $H' \times W'$ là kích thước spatial của feature map cuối cùng.

Kết quả là một vector $\mathbf{z}_{\text{GAP}} \in \mathbb{R}^D$, chính là đặc trưng không gian của ảnh đầu vào.

Khi áp dụng cho chuỗi video $V_i = \{f_i^{(1)}, f_i^{(2)}, \ldots, f_i^{(T)}\}$, ta thu được chuỗi đặc trưng tương ứng:

$$\mathbf{X}_i = \left[\phi_{\text{ResNet}}(f_i^{(1)}), \phi_{\text{ResNet}}(f_i^{(2)}), \ldots, \phi_{\text{ResNet}}(f_i^{(T)})\right] \in \mathbb{R}^{T \times D}$$

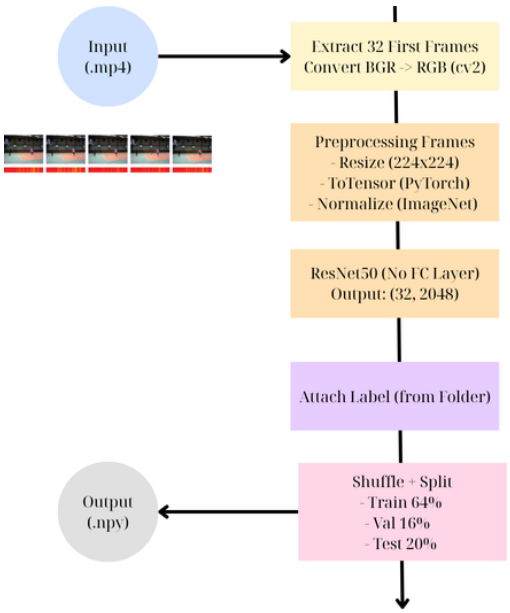
Chuỗi đặc trưng $\mathbf{X}_i$ này là biểu diễn không gian-thời gian của video V_i, giữ nguyên thứ tự thời gian giữa các khung hình, và sẽ được sử dụng làm đầu vào cho mô hình học tuần tự như Bi-LSTM ở giai đoạn tiếp theo.

3) Cách triển khai và thiết kế tham số: Toàn bộ pipeline được triển khai theo các bước sau:

- **Tiền xử lý ảnh:** Mỗi khung hình được resize về kích thước chuẩn 224×224, chuyển đổi từ định dạng OpenCV (BGR) sang RGB, và chuẩn hóa theo giá trị trung bình và độ lệch chuẩn của tập ImageNet. Việc chuẩn hóa này giúp đảm bảo ổn định đầu ra của mô hình trích đặc trưng.
- **Chọn số lượng khung hình T:** Chúng tôi chọn $T = 32$ khung hình đầu tiên trong mỗi video. Số lượng này đảm bảo mô hình có đủ thông tin theo thời gian để học động lực học của hành động, đồng thời giữ mức tài nguyên tính toán ở mức hợp lý.
- **Bộ trích đặc trưng:** Mỗi khung hình sau khi tiền xử lý được đưa qua ResNet50 (bỏ FC layer), tạo ra tensor đầu ra có kích thước $(2048, 1, 1)$, sau đó được flatten thành vector 2048 chiều.
- **Tích hợp đặc trưng theo thời gian:** Các vector đặc trưng từ 32 khung hình được nối lại thành tensor có dạng $(32, 2048)$, đại diện cho toàn bộ video trong không gian đặc trưng.

- **Gắn nhãn và lưu trữ:** Các đặc trưng cùng với nhãn hành động được lưu lại theo định dạng `.npy` để phục vụ cho bước huấn luyện mô hình tuần tự như LSTM. Dữ liệu cũng được chia thành ba tập: huấn luyện (64%), kiểm thử (20%), và xác thực (16%) để đảm bảo tính tổng quát của mô hình.

Thông qua quá trình này, toàn bộ video đầu vào đã được rút gọn thành các chuỗi vector đặc trưng giàu ngữ nghĩa, phù hợp để đưa vào mô hình học sâu mà không cần xử lý trực tiếp trên ảnh thô. Đây là bước trung gian quan trọng giúp nâng cao hiệu quả huấn luyện và độ chính xác trong bài toán nhận diện hành động từ video.



Hình 3. Hình minh họa pipeline trích xuất đặc trưng: video được cắt 32 khung hình đầu tiên, xử lý tiền xử lý chuẩn hóa, trích xuất đặc trưng bằng ResNet50 (không có FC layer), sau đó ghép lại thành tensor, gắn nhãn và chia tập dữ liệu để phục vụ huấn luyện mô hình phân loại hành động.

C. Recurrent Neural Network

Video là một dạng dữ liệu tuần tự, trong đó các chuyển động trong nội dung hình ảnh được thể hiện qua nhiều khung hình. Do đó, RNNs là một phương pháp hợp lý để phân tích các mẫu tuần tự ẩn trong cả dữ liệu tuần tự theo thời gian và dữ liệu tuần tự theo không gian. Tuy nhiên, RNNs lại có một nhược điểm là mô hình có thể diễn giải các chuỗi tuần tự, nhưng lại quên các đầu vào trước đó của chuỗi trong trường hợp chuỗi dài. Vấn đề trên được gọi là Vanishing Gradient. Và để giải quyết vấn đề trên, một biến thể đặc biệt của RNN là LSTM đã được phát triển.

LSTM có khả năng học được các quan hệ dài hạn nhờ cấu trúc đặc biệt của nó, bao gồm ba cổng: cổng vào (input gate), cổng ra (output gate) và cổng quên (forget gate) giúp kiểm soát cách mô hình tiếp nhận và duy trì thông tin từ chuỗi dữ liệu. Các cổng này được điều chỉnh thông qua một đơn vị sigmoid, có nhiệm vụ học trong quá trình huấn luyện để quyết định khi nào nên giữ lại hoặc loại bỏ thông tin. Dưới đây là các phương trình [9] mô tả chi tiết cách hoạt động của một đơn vị LSTM. Trong đó x_t là đầu vào tại thời điểm t (trong trường hợp này, là đoạn dữ liệu C). Cổng quên f_t có nhiệm vụ xóa đi thông tin không cần thiết ra khỏi ô nhớ và duy trì thông tin các khung hình trước đó nếu cần thiết. Cổng đầu ra o_t lưu giữ thông tin để chuẩn bị cho bước tiếp theo, trong khi đó, g là đơn vị hồi quy với hàm kích hoạt $tanh$, được tính toán dựa trên đầu vào hiện tại và trạng thái trước đó s_{t-1}. Trạng thái ẩn của mỗi bước trong RNN được tính bằng cách sử dụng hàm kích hoạt $tanh$ và ô nhớ c_t. Do quá trình nhận diện hoạt động con người không yêu cầu đầu ra trung gian từ mô hình LSTM, nên mô hình sẽ sử dụng bộ phân loại softmax để đưa ra quyết định cuối cùng dựa trên trạng thái cuối cùng của mạng RNN.

$$i_t = ((x_t + s_{t-1})W^i + b_i)$$
$$f_t = ((x_t + s_{t-1})W^f + b_f)$$
$$o_t = ((x_t + s_{t-1})W^o + b_o)$$
$$g = tanh((x_t + s_{t-1}W^g + b_g)$$
$$c_t = c_{t-1}.f_t + g.i_t$$
$$s_t = tanh(c_t).o_t$$
$$final\_state = softmax(Vs_t)$$

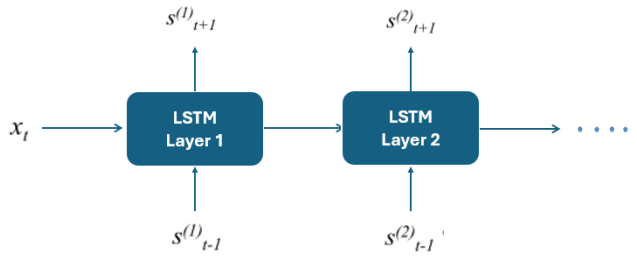
Việc huấn luyện chuỗi dữ liệu tuần tự lớn và phức tạp (cụ thể như dữ liệu từ video) không được xác định hiệu quả bởi một lớp LSTM đơn lẻ. Do đó, trong phần đề xuất, chúng tôi sẽ sử dụng mô hình LSTM đa lớp (Multi layers LSTM) bằng cách xếp chồng nhiều lớp LSTM với nhau để mô hình học các quan hệ dài hạn trong dữ liệu video.

D. Multi layers LSTM

Hiệu suất của mạng nơ-ron học sâu có thể được cải thiện bằng cách tăng số lượng lớp trong mô hình. Chiến lược tương tự cũng được áp dụng cho RNN bằng cách xếp chồng hai tầng LSTM vào mạng. Qua đó, việc bổ sung tầng mới giúp RNN có thể nắm bắt được thông tin trình tự ở cấp độ cao hơn. Trong mô hình tiêu chuẩn RNN, dữ liệu chỉ được đưa vào một tầng duy nhất để kích hoạt và xử lý trước khi tạo đầu ra. Tuy nhiên đối với các bài toán chuỗi thời gian, việc xử lý dữ liệu qua nhiều tầng là cần thiết. Khi xếp chồng các tầng LSTM, mỗi tầng trong RNN hoạt động theo một hệ thống phân cấp, trong đó trạng thái ẩn của tầng trước sẽ được làm đầu vào cho tầng tiếp theo.

Hình 4 minh họa một mô hình LSTM nhiều tầng. Tầng 1 nhận dữ liệu đầu vào x_t, trong khi tầng 2 nhận đầu vào từ trạng thái trước đó của chính nó $s_t^{(2)}$ tại thời điểm $t-1$ và trạng thái đầu ra $s_t^{(1)}$ của tầng 1 tại thời điểm hiện tại. Việc tính toán của mỗi lớp LSTM vẫn tuân theo các phương trình đã đề cập ở trên, nhưng thông tin về tầng đã được thêm vào chỉ số trên của các biến i_t, f_t, o_t, c_t, s_t. Phương trình dưới đây [9] mô tả cách tính trạng thái của một tầng trong mạng.

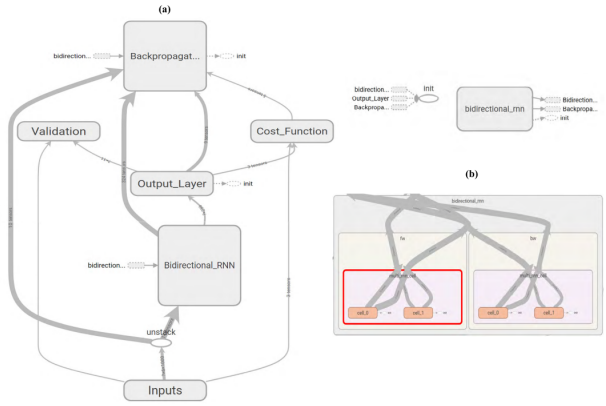
$$s_t^l = tanh(c_t^l).o_t^l$$



Hình 4. Kiến trúc hai lớp LSTM

E. Bidirectional LSTM

Trong LSTM hai chiều (Bidirectional LSTM), đầu ra tại thời điểm t không chỉ phụ thuộc vào các khung hình trước trong chuỗi mà còn chịu ảnh hưởng từ các khung hình tiếp theo. Bidirectional RNN có cấu trúc khá đơn giản, gồm hai mạng RNN được xếp chồng lên nhau, trong đó một mạng xử lý theo chiều tiến, mạng còn lại xử lý theo chiều lùi. Sau đó, đầu ra kết hợp được tính toán dựa trên trạng thái ẩn của cả hai mạng RNN.



Hình 5. Cấu trúc bên ngoài và bên trong của mô hình mạng DB-LSTM [9]

Hình 4a thể hiện cấu trúc bên ngoài của giai đoạn huấn luyện, trong đó dữ liệu đầu vào được truyền vào RNN hai chiều, và trạng thái ẩn của các bước tiến và lùi được kết hợp tại tầng đầu ra. Sau khi tính toán đầu ra, dữ liệu xác thực và hàm mất mát được sử dụng để điều chỉnh trọng số và hệ số điều chỉnh (biases) thông qua lan truyền ngược. Để xác thực mô hình, 20% dữ liệu được tách ra làm tập kiểm tra và hàm cross entropy được sử dụng để tính toán lỗi. Quá trình tối ưu hóa ngẫu nhiên với tốc độ học 0.001 được áp dụng để giảm thiểu chi phí.
Hình 4b thể hiện cấu trúc bên trong của RNN hai chiều, trong đó "fw" đại diện cho hướng tiến và "bw" đại diện cho hướng lùi. Cả hai hướng này đều bao gồm hai lớp LSTM, giúp mô hình trở thành một LSTM hai chiều sâu. Phương pháp đề xuất vượt trội hơn các phương pháp hiện đại khác nhờ vào cơ chế tính toán đầu ra. Đầu ra của một khung hình tại thời điểm t được xác định không chỉ từ khung hình trước đó tại thời điểm $t-1$ mà còn từ khung hình kế tiếp tại thời điểm $t+1$, vì các tầng trong mô hình thực hiện xử lý theo cả hai hướng.

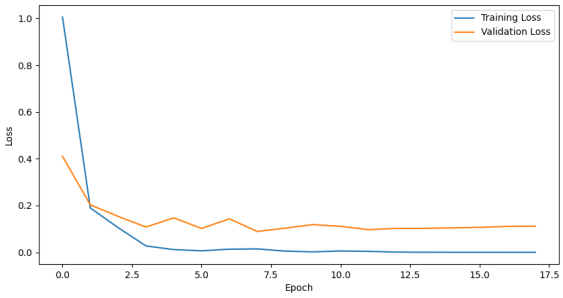
F. Cải tiến mô hình

IV. Kết quả thực nghiệm

Trong phần này, mô hình sẽ trải qua quá trình thực nghiệm từ các bộ dữ liệu như UCF11, UCF50, UCF101, HMDB101. Bộ dữ liệu khi được đọc vào sẽ được phân chia thành các tập huấn luyện, tập xác thực và tập kiểm thử. Trước tiên, chương trình xáo dữ liệu, rồi phân chia thành tập huấn luyện và tập kiểm thử trước theo tỷ lệ 80% và 20%. Từ tập huấn luyện, chương trình sẽ phân chia tiếp để tạo tập xác thực với tỷ lệ 20%. Dữ liệu được đọc vào là những video, sau đó chương trình sẽ chuyển đổi các video trên thành các frame với số lượng frame mặc định là 32. Trong quá trình trích xuất đặc trưng và huấn luyện mô hình, chương trình còn hỗ trợ chạy trên cả CPU và GPU. Tốc độ học mặc định là 0.001, kích thước lớp ẩn là 256 và số lượng epochs là 100.

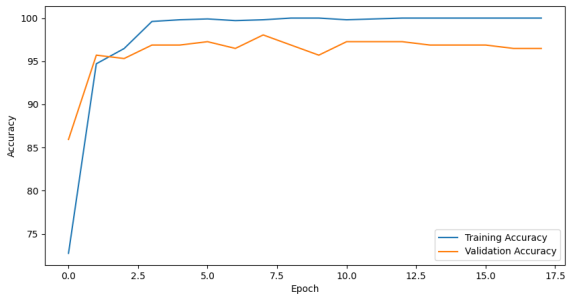
A. UCF11 Dataset

Bộ dữ liệu UCF-11 (hay còn được gọi là UCF Youtube Action Dataset) là một tập dữ liệu chứa các video clip từ YouTube. Đây là bộ dataset với số lượng khá nhỏ so với các bộ còn lại tuy nhiên đây là bộ dữ liệu thích hợp để thực nghiệm các mô hình mới được thiết kế. Bộ dữ liệu được thu thập 11 môn thể thao khác nhau (bao gồm volleyball, basketball, golf, horse riding, biking/cycling, tennis, diving, football, swinging, jumping, and walking with a dog). Với mỗi bộ môn sẽ bao gồm 25 thư mục khác nhau và mỗi thư mục sẽ chứa ít nhất 4 video. Các video clips sẽ có những đặc trưng giống nhau, ví dụ như cùng một diễn viên, cùng một phong cảnh và cùng thực hiện một môn thể thao.
Về thực nghiệm trên bộ dữ liệu này, kết quả cho thấy mô hình đạt tỷ lệ huấn luyện và dự đoán rất cao luôn trên 96%.



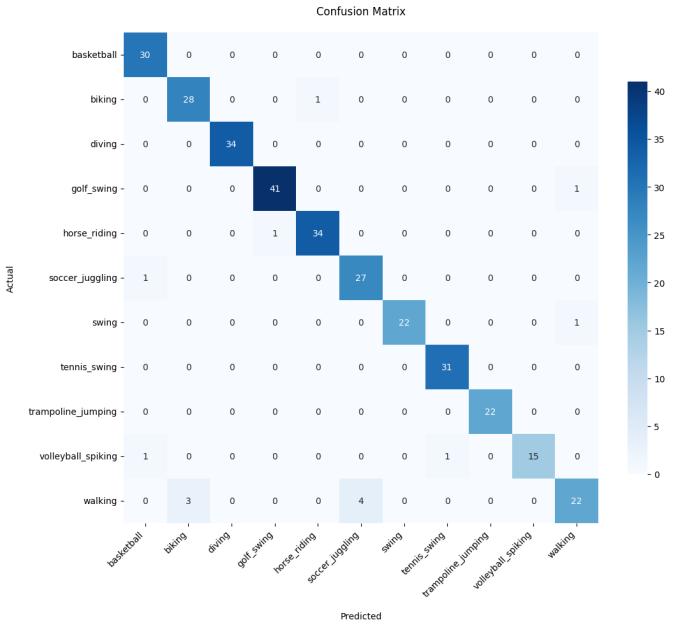
Hình 6. Biểu đồ lỗi trong quá trình huấn luyện tập UCF11

Từ hai hình ảnh 6 và 7, ta nhận xét được rằng độ lỗi trong quá trình học được khá thấp với xấp xỉ 0.1 và độ chính xác của mô hình đối với tập huấn luyện đạt 100% và tập xác thực là hơn 96%. Số lượng epoch khi huấn luyện là không cao, do mô hình học khá nhanh và chương trình còn cài đặt thêm cơ



Hình 7. Biểu đồ về độ chính xác giữa tập huấn luyện và tập xác thực khi huấn luyện bộ UCF11

chế Early Stopping, giúp mô hình dừng học sớm nếu độ lỗi không cải thiện hơn so với lúc trước 10 lần.



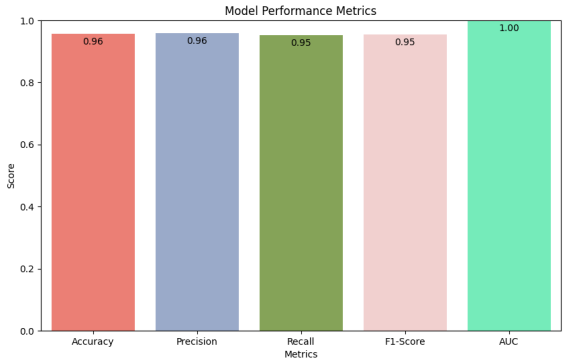
Hình 8. Ma trận tương quan của bộ UCF11

Ma trận hình 8 trên cho thấy sự tương quan giữa tập Ground Truth và tập dự đoán. Ta thấy rằng tỷ lệ dự đoán của mô hình rất cao, chỉ có vài chỗ dự đoán sai sót. Từ hình 9, các thông số đánh giá về độ chính xác, precision, recall, f1 và auc đều rất cao và cân bằng nhau. Trong lần thử nghiệm bất kỳ này thì độ chính xác của mô hình đạt được là 96%. Một số lần thử nghiệm khác thì con số có thể dao động từ 95% hoặc thậm chí có thể đạt đến 98, 99% tùy thuộc vào sự ngẫu nhiên khi chia bộ dữ liệu ban đầu.

TÀI LIỆU

[1] University of Central Florida, "UCF Youtube Action Data Set", [Online]. Available: https://www.crcv.ucf.edu/data/UCF_YouTube_Action.php

[2] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2005, pp. 984–989.



Hình 9. Biểu đồ thể hiện thông số đánh giá của mô hình

[3] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[4] Chanlong Zhang, Yuanping Xu, Zhijie Xu, Jian Huang, Jun Ju. 2021. Hybrid handcrafted and learned feature framework for human action recognition. 52:12771–12787.

[5] Waqar Ahmad, Misbah Kazmi, Hazrat Ali, "Human Activity Recognition using Multi-Head CNN followed by LSTM", 21 Feb 2020.

[6] Iman Deznabi, Madalina Fiterau, "MultiWave: Multiresolution Deep Architectures through Wavelet Decomposition for Multivariate Time Series Prediction", 16 Jun 2023

[7] Garima Pandey1, Abhishek Kumar Karn2, Manish Jha, "Human Activity Recognition Using CNN-LSTM-GRU Model", April 2024

[8] Erdal Genc, Mustafa Eren Yildirim, Yucel Batu Salman, "Human activity recognition with fine-tuned CNN-LSTM", February 2024

[9] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, Sung Wook Baik. "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features", February 14 2018.

[10] University of Central Florida, "UCF101 - Action Recognition Data Set", [Online]. Available: https://www.crcv.ucf.edu/data/UCF101.php

[11] University of Central Florida, "UCF50 - Action Recognition Data Set", [Online]. Available: https://www.crcv.ucf.edu/data/UCF50.php