**TRƯỜNG ĐẠI HỌC HÀNG HẢI VIỆT NAM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**-----\*\*\*-----**

Logo

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**HỌC PHẦN “THỊ GIÁC MÀY TÍNH”**

**Đề tài: *Xây dựng hệ thống nhận dạng cử chỉ cơ thể con người***

**GVHD:** T.S Hồ Thị Hương Thơm

**Sinh viên thực hiện:** Bùi Hoàng Cường - 82435

Phạm Duy Phong - 84391

*Hải Phòng, tháng 11 năm 2022*

MỤC LỤC

[**MỞ ĐẦU** 2](#_Toc120057335)

[**CHƯƠNG 1: MÔ TẢ ĐỀ TÀI** 3](#_Toc120057336)

[1.1. Lý do chọn đề tài 3](#_Toc120057337)

[1.2. Mục đích nghiên cứu 3](#_Toc120057338)

[1.3. Nội dung nghiên cứu 3](#_Toc120057339)

[1.4. Nhiệm vụ nghiên cứu 3](#_Toc120057340)

[1.5. Đối tượng và phạm vi nghiên cứu 4](#_Toc120057341)

[1.6. Phương pháp nghiên cứu 4](#_Toc120057342)

[**CHƯƠNG 2: TỔNG QUAN VỀ NHẬN DẠNG CỬ CHỈ, HÀNH ĐỘNG** 6](#_Toc120057343)

[2.1. Mô hình khung xương cơ thể người 6](#_Toc120057344)

[2.2.Phân loại cử chỉ, hành động 6](#_Toc120057345)

[2.3. Tiến trình nhận dạng 6](#_Toc120057346)

[**CHƯƠNG 3: CƠ SỞ LÝ THUYẾT** 10](#_Toc120057347)

[3.1.Tổng quan về MediaPipe 10](#_Toc120057348)

[3.2 : Tổng quan về RNN - LSTM 11](#_Toc120057349)

[**CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ** 21](#_Toc120057350)

**MỞ ĐẦU**

Xử lý ảnh (DIP - Digital Image Processing) là một lĩnh vực mang tính khoa học và công nghệ. Nó là một ngành khoa học mới mẻ so với nhiều ngành khoa học khác nhưng tốc độ phát triển của nó rất nhanh, kích thích các trung tâm nghiên cứu, ứng dụng, đặc biệt là máy tính chuyên dụng riêng cho nó. Đầu tiên phải kể đến xử lý tín hiệu số với các khái niệm về tích chập, các biến đổi Fourier, biến đổi Laplace, các bộ lọc hữu hạn… Thứ hai, các công cụ toán như đại số tuyến tính, xác suất thống kê. Một số kiến thức cần thiết như trí tuệ nhân tạo, mạng noron nhân tạo cũng được đề cập trong quá trình phân tích và nhận dạng ảnh. Đặc biệt nhận dạng đối tượng (Object Recognition) đang được ứng dụng rất nhiều. Cụ thể là các bài toán là nhận dạng dấu vân tay, nhận dạng khuôn mặt, nhận dạng chữ viết…

Nhận dạng hành động của con người là một miền ứng dụng quan trọng trong thị giác máy tính. Mục đích chính của nó là mô tả chính xác các hành động của con người và tương tác của họ từ một chuỗi dữ liệu chưa từng thấy trước đây mà các cảm biến thu được

**CHƯƠNG 1: MÔ TẢ ĐỀ TÀI**

* 1. **Lý do chọn đề tài**Với sự gia tăng nhu cầu an ninh và giám sát, đặc biệt là ở những nơi đông đúc như các khu vực sân bay, trung tâm mua sắm thì việc nhận biết được hành vi của con người là cực kỳ quan trọng. Chính vì vậy chúng e chọn đề tài này để học hỏi them kiến thức về lĩnh vực thị giác, cũng như phát triển phần mềm nhận biết hành vi con người

**1.2. Mục đích nghiên cứu**

- Nắm rõ kiến thức về RNN, LSTM

- Xây dựng giải pháp nhận dạng cử chỉ, hành độngcủa con người

**1.3. Nội dung nghiên cứu**

- Phân tích, trích chọn đặc trưng và nhận dạng cư chỉ, hành động

- Đưa ra giải pháp thích hợp đáp ứng mục đích đề tài.

- Viết chương trình để kiểm tra tính hiệu quả của giải pháp.

**1.4. Nhiệm vụ nghiên cứu**

Luận văn tập trung nghiên cứu các vấn đề sau:

- Phân tích, trích chọn đặc trưng và nhận dạng cử chỉ.

- Đưa ra giải pháp thích hợp đáp ứng mục đích yêu cầu đề bài

- Viết chương trình để kiểm tra tính hiệu quả

**1.5. Đối tượng và phạm vi nghiên cứu**

a, Đối tượng nghiên cứu

* Các cử chỉ con người.
* Các kỹ thuật trích chọn đặc trưng cử chỉ , hành động
* Các kỹ thuật nhận dạng cử chỉ , hành động

b, Phạm vi nghiên cứu

* Nghiên cứu kỹ thuật xử lý ảnh và nhận dạng cử chỉ ,hành động
* Nghiên cứu các kỹ thuật trích chọn đặc trưng

**1.6. Phương pháp nghiên cứu**

a, Phương pháp lý thuyết

Tập trung nghiên cứu một số lý thuyết:

- Các tài liệu về cử chỉ, nhận dạng cử chỉ, hành độngtrong lĩnh vực xử lý ảnh và thị giác máy tính

- Cơ sở lý thuyết về các thuật toán trích chọn đặc trưng và nhận dạng cử chỉ tay từ camera.

b, Phương pháp thực nghiệm

Nghiên cứu một số vấn đề thực nghiệm như sau:

- Cài đặt chương trình ứng dụng dựa trên giải pháp đề xuất

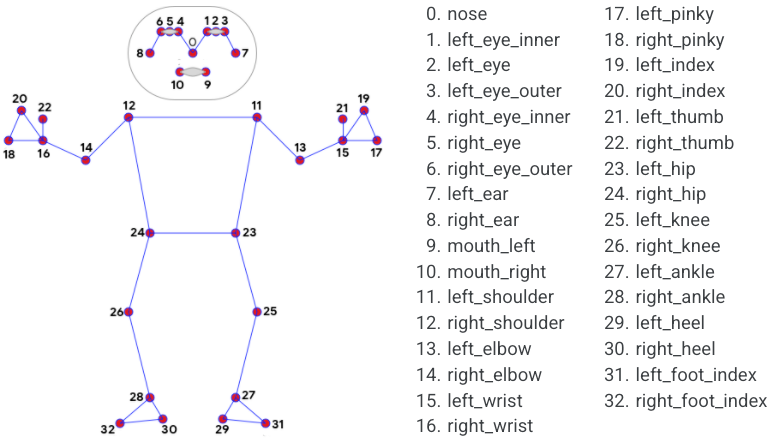
- Triển khai thực nghiệm

- Kiểm thử hiệu quả của chương trình với các cử chỉ tay khác nhau

**CHƯƠNG 2: TỔNG QUAN VỀ NHẬN DẠNG CỬ CHỈ, HÀNH ĐỘNG**

**2.1. Mô hình khung xương con người**

- Mô hình khung xương con người có cấu trúc như sau:



Có tổng cộng 33 bộ phân trên cơ thể người mà chúng ta cần xác định

- Nhờ vào việc thay đổi vị trí của các địa điểm, chúng ta có thể dự đoán được hành động của con người thông qua việc thay đổi trên.

**2.2.Phân loại cử chỉ , hành động**

- Mỗi cử chỉ là một cấu hình bàn tay đặc biệt đại diện bởi một hình ảnh duy nhất. Có nhiều cách để phân loại cử chỉ, thông thường cử chỉ được phân loại dựa trên mối quan hệ thời gian. Khi đó, cử chỉ tay được phân thành 2 loại: cử chỉ tĩnh và cử chỉ động

* Cử chỉ tĩnh: là cử chỉ mà trong đó vịt trí khung xương không thay đổi theo thời gian ( độc lập với thời gian). Cử chỉ tĩnh chủ yếu dựa vào hình dạng và góc uốn của cơ thể. Cử chỉ tay tĩnh là cử chỉ tay diễn đạt một trạng thái duy nhất của cơ thể
* 
* Cử chỉ động: là cử chỉ trong đó vị trí tay thay đổi liên tục theo thời gian. Trong thực tế, nhiều cử chỉ được tạo ra bởi cách chúng ta di chuyển bàn như hướng đi, quỹ đạo hay tốc độ thực hiện. Như vậy có thể thấy cử chỉ tay động chủ yếu dựa vào quỹ đạo, hướng đi, thêm vào đó là hình dáng và góc uốn của khung xương
* 

**2.3. Tiến trình nhận dạng cử chỉ ,hành động**

**2.3.1. Sơ đồ khối của hệ thống nhận dạng cử chỉ ,hành động**

**Diagram

Description automatically generated**

*Hình 1: Sơ đồ khối của hệ thống nhận dạng cử chỉ tay*

**2.3.2. Dữ liệu vào**

Dữ liệu vào có thể là một hình ảnh, hoặc một chuỗi các hình ảnh được chụp bởi một máy tính duy nhất hướng về phía bàn tay người thực hiện các cử chỉ. Tuy nhiên, cũng tồn tại các hệ thống sử dụng hai hoặc nhiều máy ảnh để thu được nhiều thông tin về các tư thế.

**2.3.3. Tiền xử lý**

Mục đích cơ bản của giai đoạn này là tối ưu hình ảnh thu được từ giai đoạn trước để chuẩn bị trích xuất các đặc trưng trong giai đoạn tiếp theo. Tiền xử lý ảnh là bắt buộc để nâng cao chất lượng ảnh đồng thời cải thiện độ chính xác của hệ thống được đề xuất. Dưới đây là mội số kỹ thuật tiền xử lý thường được áp dụng trong giai đoạn này:

a, Chuyển ảnh RGB sang ảnh nhị phân

Ảnh nhị phân là ảnh mà giá trị của các điểm ảnh chỉ được biểu diễn bởi một trong hai giá trị 0 (màu đen) hoặc 1 (màu trắng). Chẳng hạn những điểm ảnh trong vùng tay có màu trắng và điểm ảnh nền có màu đen.

b, Lọc nhiễu

Trong ảnh nhị phân, có thể có một số giá trị 0 trong cùng tay và một số giá trị 1 trong khu vực nền. Khi đó, nền có những số 1 được biết đến như nhiễu xung quanh và cử chỉ tay có mội vài số 0 được biết đến như nhiễu cử chỉ. Những nhiễu này cần phải được loại bỏ.

c, Phân đoạn ảnh

Mục tiêu của phân đoạn ảnh là làm nổi bật hoặc tách hẳn đối tượng cần quan tâm ra từ ảnh ban đầu, làm đơn giản hoắ và thay đổi cách biểu diễn để dễ dàng phân tích hơn. Một số kỹ thuật phân đoạn ảnh phổ biến như:

* Phân đoạn ảnh dự trên ngưỡng
* Phân đoạn ảnh dựa trên biên ( cạnh)
* Phân đoạn ảnh dựa vào màu da

**2.3.4. Trích chọn đặc trưng**

Trích đoạn đặc trưng là chiết xuất thông tin từ dữ liệu thô mà nó phù hợp nhất cho mục đích nhận dạng để giảm thiểu các biến thể trong một lớp và tối đa hóa các biến thể giữa các lớp khác nhau.

Trong giai đoạn trích chọn đặc trưng, mỗi đối tượng được đại diện bởi một vecto đặc trưng. Mục tiêu chủ yếu của trích chọn đặc trưng là trích xuất một tập hợn các đặc trưng, trong đó có thể tối đa hóa tỷ lệ nhận dạng với sự nhầm lẫn nhỏ nhất. Các đặc điểm của đối tượng được trích chọn tùy theo mục đích nhận dạng trong quá trình xử lý ảnh. Có thể nêu ra một số đặc điểm của ảnh sau đây: Đặc điểm không gian, đặc điểm biến đổi, đặc điểm biên và đường biên…

**2.3.5. Nhận dạng**

Giai đoạn nhận dạng thực hiện việc phân tích dữ liệu đầu vào của người dùng để kiểm tra trong dữ liệu đầu vào đó có chứa cử chỉ đang xét hay không. Dữ liệu đầu vào cũng qua các quá trình tiền xử lý, trích chọn đặc trưng vào các lớp đã được định nghĩa trong giai đoạn huấn luyện trước đó. Sau đó cử chỉ được phân đoạn nhờ vào việc kết hợp các bộ nhận dạng cử chỉ. Mỗi bộ nhận dạng cử chỉ có chức năng nhận dạng một cử chỉ ứng với các đặc trưng được huấn luyện.

**CHƯƠNG 3: CƠ SỞ LÝ THUYẾT**

**3.1.Tổng quan về MediaPipe**

AI (Trí tuệ nhân tạo) đang dần "xâm chiếm" hầu như tất cả các lĩnh vực trong cuộc sống. Với tính chất tự động hóa cao, có thể thực hiện các công việc khó và có độ chính xác ngày càng được cải thiện, AI đang dần trở thành một công cụ khó có thể thay thế được. Tuy nhiên, các mô hình AI hiện tại được đánh giá là "nặng, yêu cầu phần cứng cao" khiến cho việc áp dụng vào trong các dự án thực tế trở nên khá khó khăn, nhất là với các thiết bị mobile hoặc edge devices - xu thế hiện tại. Vì vậy, Google đã đưa ra một giải pháp, chính xác hơn là một bộ công cụ, cung cấp các công cụ cho các bài toán AI/ML đã được tối ưu để chạy trên nhiều nền tảng khác nhau, với tên gọi là [MediaPipe](https://github.com/google/mediapipe" \t "_blank).

Về tổng quan, [MediaPipe](https://github.com/google/mediapipe" \t "_blank) là tập hợp của một loạt các giải pháp Machine Learning đa nền tảng, có thể can thiệp được và cực kỳ lightweight. Một số ưu điểm có thể kể tới của giải pháp này bao gồm:

* Cung cấp một giải pháp inference nhanh chóng: Google khẳng định rằng bộ công cụ này có thể chạy ổn định trên hầu hết các cấu hình phần cứng thông dụng.
* Dễ dàng cài đặt và triển khai: Việc cài đặt cực kỳ dễ dàng và tiện lợi, có thể triển khai trên nhiều nền tảng khác nhau như Mobile (Android/iOS), Desktop/Cloud, Web và IoT devices.
* Mã nguồn mở và miễn phí: Toàn bộ source code được công khai trên [MediaPipe](https://github.com/google/mediapipe" \t "_blank), người dùng hoàn toàn có thể sử dụng và tùy chỉnh trực tiếp để phù hợp với bài toán của mình.

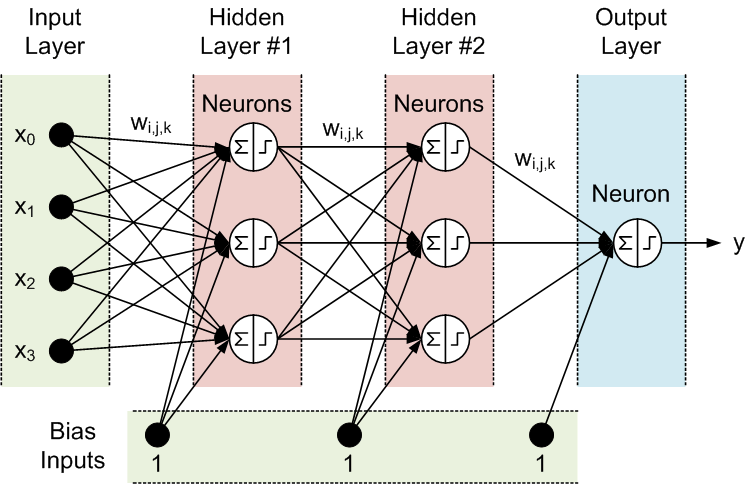
3.1.1 Giới thiệu về Human Pose Estimation

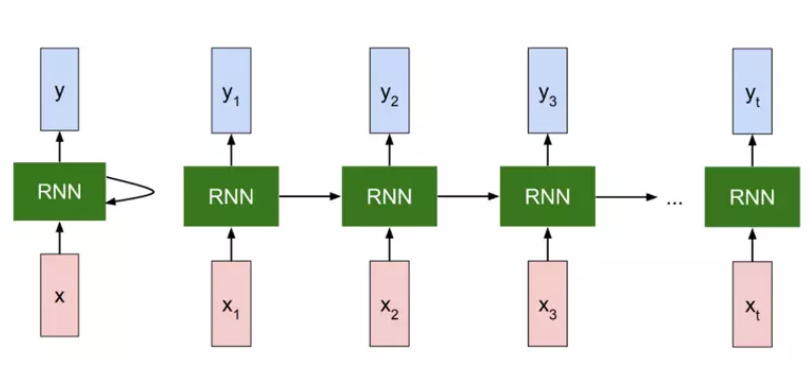
Mở rộng từ bài toán Hands Detection, Human Pose Estimation cung cấp một mô hình skeleton 3D cho cả cơ thể, với các khớp quan trọng được định nghĩa sẵn và được nối với nhau để tạo thành khung của người. Chiến thuật được đặt ra cho bài toán này tương tự như bài Hands Detection và Face Mesh. BlazeFace, một lần nữa, được sử dụng làm tư tưởng chính cho thuật toán xử lý bài này.



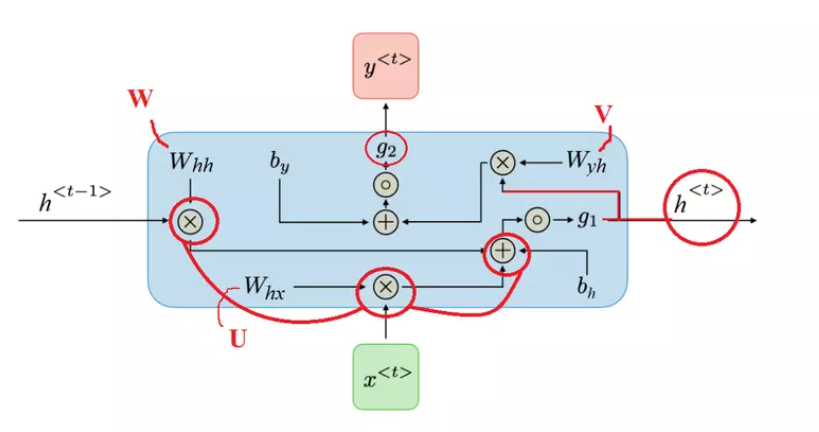
Ngoài ra, còn rất nhiều giải pháp nữa được Google cung cấp trong bộ cung cấp này, bao gồm như các bài toán Segmentation (Selfie, Hair, ...), Object Detection, Motion Tracking, 3D Object Detection, ... Tuy nhiên, mình sẽ không liệt kê hết trong bài viết này do quá dài. Còn bây giờ, mình sẽ giới thiệu một ứng dụng đơn giản để áp dụng bộ công cụ này vào thực tế.

**3.2 : Giới thiệu về RNN**

Neural Network bao gồm 3 phần chính là Input layer, Hidden layer và Output layer, ta có thể thấy là đầu vào và đầu ra của mạng neuron này là độc lập với nhau. Như vậy mô hình này không phù hợp với những bài toán dạng chuỗi như mô tả, hoàn thành câu, ... vì những dự đoán tiếp theo như từ tiếp theo phụ thuộc vào vị trí của nó trong câu và những từ đằng trước nó.  
Và như vậy RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại thông tin từ từ những bước tính toán xử lý trước để dựa vào nó có thể đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại

Mỗi block RNN sẽ lấy thông tin từ các block trước và input hiện tại.  
  
Các x ở đây đại diện cho dữ liệu đầu vào lần lượt (được chia theo time step).

*xt*​ đại diện cho time step thứ t, và  *yt*​ là output của một step. Ví dụ,  *x*2​ sẽ là vector đại diện của từ thứ 2 trong câu văn bản.

Hình ảnh dưới đây cho thấy rõ hơn điều gì thực sự xảy ra trong một step.  


### **Hidden state**  *ht*​ (trong một số tài liệu tường ký hiện  *st*​). Đây chính là bộ nhớ của mạng.  *ht*​ là tổng hợp thông tin của hidden state trước ( *ht*−1​) cộng với input tại time step t (*xt*​). Activation function ở đây $g\_1$chủ yếu là tanh hoặc ReLu. Hạn chế của RNN là gì?

* **Phải thực hiện tuần tự:** Không tận dụng được khả năng tính toán song song của máy tính (GPU/TPU).
* **Vanishing gradient** (Đạo hàm bị triệt tiêu)

Vì hàm kích hoạt (tanh hay sigmoid) của ta sẽ cho kết quả đầu ra nằm trong đoạn [-1, 1][−1,1] (với sigmoid là [0, 1][0,1]) nên đạo hàm của nó sẽ bị đóng trong khoảng [0, 1][0,1] (với sigmoid là [0, 0.25][0,0.25]).

Ở trên, chúng ta đã dùng chain rule để tính đạo hàm. Có một vấn đề ở đây là, hàm tanh lẫn sigmoid đều có đạo hàm bằng 0 tại 2 đầu. Mà khi đạo hàm bằng 0 thì nút mạng tương ứng tại đó sẽ bị bão hòa. Lúc đó các nút phía trước cũng sẽ bị bão hoà theo. Nên với các giá trị nhỏ trong ma trận, khi ta thực hiện phép nhân ma trận sẽ đạo hàm tương ứng sẽ xảy ra Vanishing gradient, tức đạo hàm bị triệt tiêu chỉ sau vài bước nhân. Như vậy, các bước ở xa sẽ không còn tác dụng với nút hiện tại nữa, làm cho RNN không thể học được các phụ thuộc xa. Vấn đề này không chỉ xảy ra với mạng RNN mà ngay cả mạng neural truyền thống với nhiều lớp cũng có hiện tượng này.

Với cách nhìn như trên, ngoài Vanishing gradient, ta còn gặp phải **Exploding Gradient** (bùng nổ đạo hàm). Tùy thuộc vào hàm kích hoạt và tham số của mạng, vấn đề này xảy ra khi các giá trị của ma trận là lớn (lớn hơn 1). Tuy nhiên, người ta thường nói về vấn đề Vanishing nhiều hơn là Exploding, vì 2 lý do sau.

Thứ nhất, bùng nổ đạo hàm có thể theo dõi được vì khi đạo hàm bị bùng nổ thì ta sẽ thu được kết quả là một giá trị phi số NaN làm cho chương trình của ta bị dừng hoạt động.

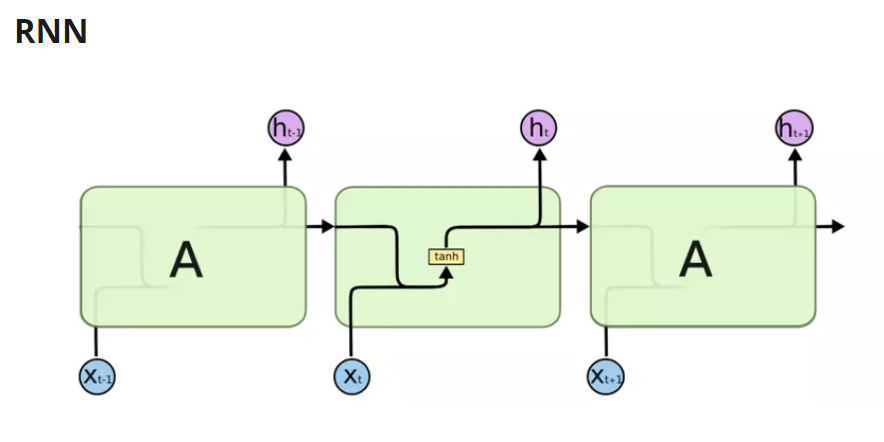
Thứ hai, bùng nổ đạo hàm có thể ngăn chặn được khi ta đặt một ngưỡng giá trị trên (tham khảo kỹ thuật **Gradient Clipping**). Còn rất khó để theo dõi sự mất mát đạo hàm cũng như tìm cách xử lí nó.

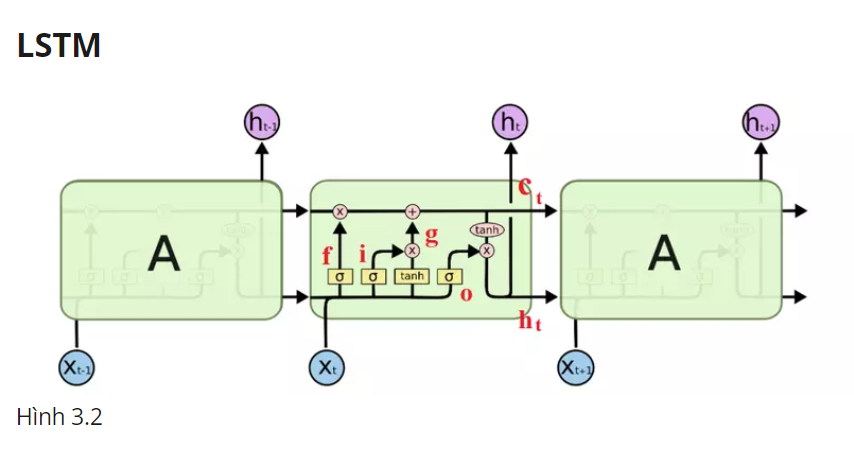
Để xử lý **Vanishing Gradient**, có 2 cách phổ biến:

* Cách thứ nhất, thay vì sử dụng activation function là tanh và sigmoid, ta thay bằng ReLu (hoặc các biến thể như Leaky ReLu). Đạo hàm của ReLu hoặc là 0 hoặc là 1, nên ta có thể kiểm soát phần nào vấn đề mất mát đạo hàm.
* Cách thứ hai, ta thấy RNN thuần không hề có thiết kế nào để lọc đi những thông tin không cần thiết. Ta cần thiết kế một kiến trúc có thể nhớ dài hạn hơn, đó là LSTM và GRU.

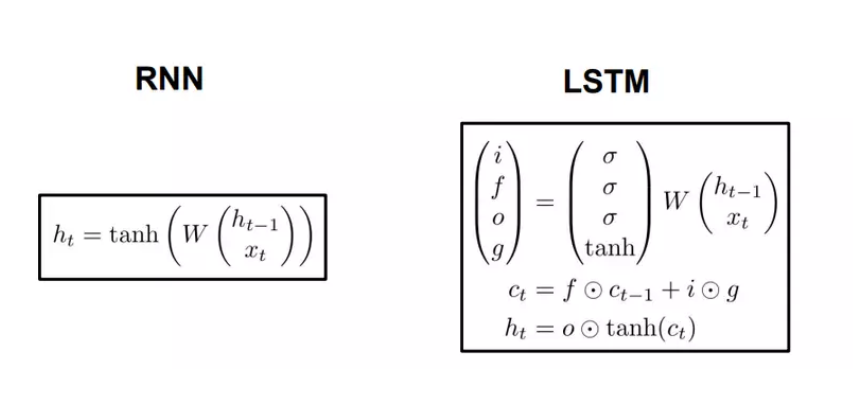
**3.2 LSTM (Long Short-Term Memory)**

So sánh giữa RNN và LSTM





Về cơ bản, ý tưởng không khác nhau là mấy. Chúng ta chỉ thêm một số tính toán ở đây. Tất cả được tóm tắt trong hình sau.



### Tổng kết

Nhìn một lượt qua kiến trúc LSTM, ta có thể tóm tắt:

* Thứ nhất, LSTM có long-term memory. Tuy nhiên,  ​ khá giống với RNN truyền thống, tức có short-term memory. Nhìn chung, LSTM giải quyết phần nào vanishing gradient so với RNN, nhưng chỉ một phần.
* Với lượng tính toán như trên, RNN đã chậm, LSTM nay còn chậm hơn.

Tuy vậy, với những cải tiến so với RNN thuần, LSTM đã và đang được sử dụng phổ biến. Trên thực tế, cách cài đặt LSTM cũng rất đa dạng và linh hoạt theo bài toán, tuy nhiên vẫn dựa trên LSTM chuẩn như trên.

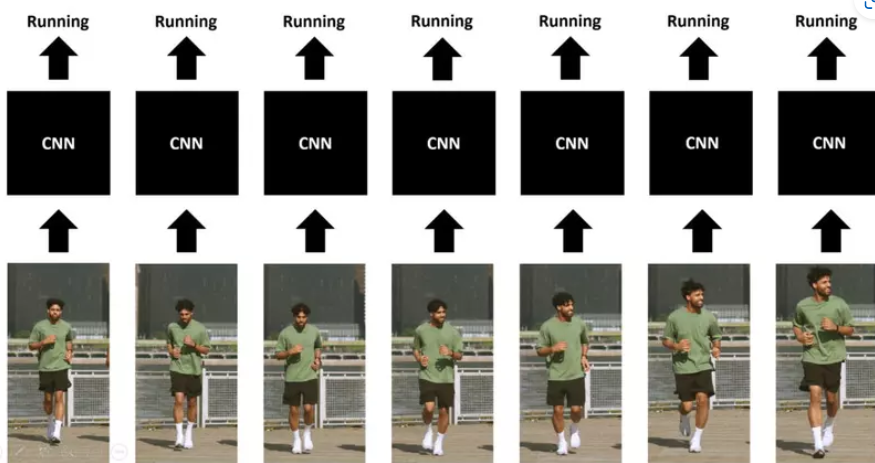
**CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ**

**4.1 Xây dựng bài toán**

# Các phương pháp phân loại

## Single-Frame CNN

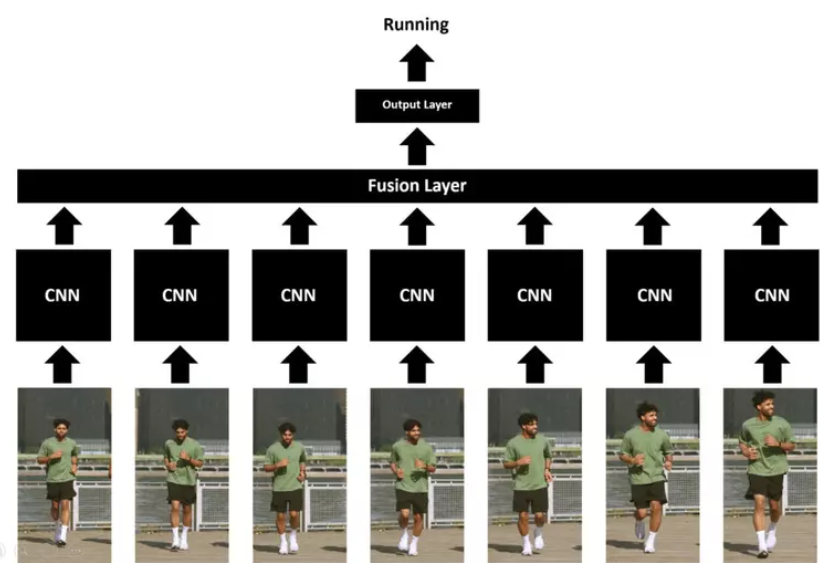
Đây là phương pháp Image Classification, dùng model CNN predict từng frame. Rồi lấy trung bình cộng các xác suất độc lập của từng frame để ra xác suất cuối cùng.



## Late Fusion

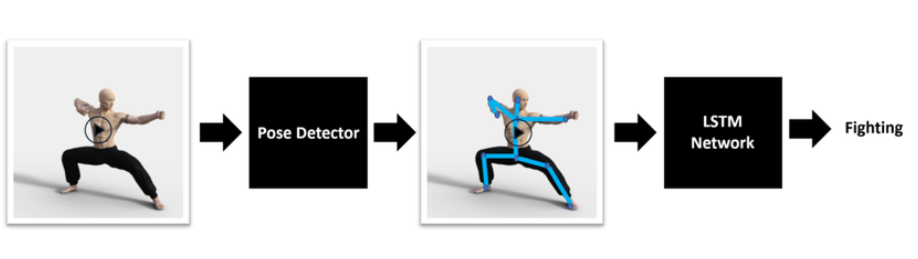
Cũng là dùng CNN predict từng frame nhưng output không còn là xác suất mà là features. Các features này sẽ được merge với nhau ở lớp cuối cùng. Lớp này gọi là lớp fusion, fusion thì có nhiều cách nhưng ở đây tác giả đề xuất dùng max pooling, average pooling hoặc flatten.

Cách này giúp mô hình học thêm các thông tin về không gian, thời gian, chuyển động của người trong từng frame.

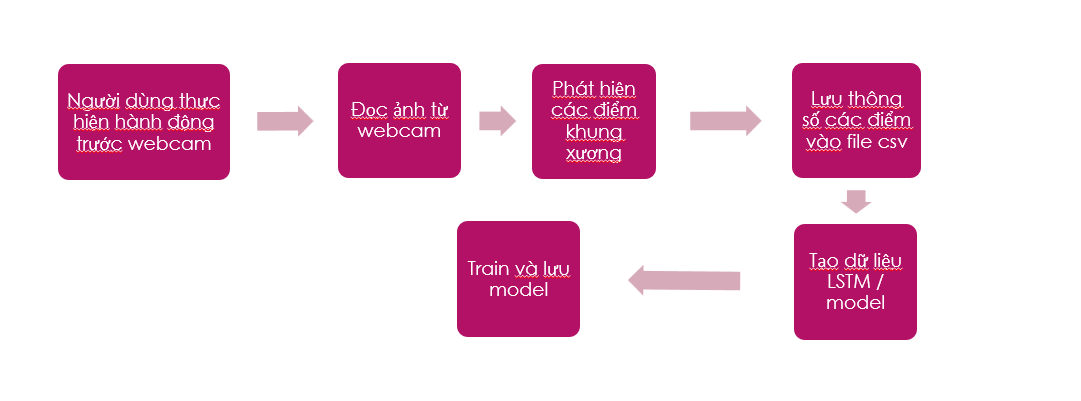


## Pose Detection and LSTM

Cách này là dùng keypoints a.k.a các điểm trên cơ thể người mỗi frame cho qua LSTM để xác định hành vi của người trong video. Trước đó thì phát hiện keypoints bằng mô hình hoặc framework (Openpose hoặc MediaPipe).



4.2 Lưu đồ thuật toán



4.3 Nhận diện hành vi

