ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Đề tài: Phát hiện và khoanh vùng đối tượng có hành vi lạ trong chung cư bằng camera CCTV

Người hướng dẫn: PGS.TS Lê Hoàng Thái

Nhóm sinh viên thực hiện:

- Đỗ Vương Phúc – 19127242
- Võ Hoàng Bảo Duy – 19127027
- Lê Minh Sĩ – 19127064

Thành phố Hồ Chí Minh, 2022

MỤC LỤC

MỤC LỤC HÌNH

1 Thông tin nhóm và bảng phân chia công việc

1.1 Thông tin nhóm

- Sinh viên 1: Đỗ Vương Phúc
  - MSSV: 19127242
  - Địa chỉ email: phuc16102001@gmail.com
  - Điện thoại liên lạc: (+84) 707 953 475
- Sinh viên 2: Võ Hoàng Bảo Duy
  - MSSV: 19127027
  - Địa chỉ email: v.hbaoduy@gmail.com
  - Điện thoại liên lạc: (+84) 776 562 199
- Sinh viên 3: Lê Minh Sĩ
  - MSSV: 19127064
  - Địa chỉ email: hungtiensi.lms@gmail.com
  - Điện thoại liên lạc: (+84) 842 429 138
- Giảng viên hướng dẫn: PGS. TS. Lê Hoàng Thái
  - Cơ quan công tác: Khoa Công nghệ Thông tin ĐH Khoa học Tự nhiên
  - Địa chỉ email: lhthai@hcmus.edu.vn

1.2 Bảng phân chia công việc

Mã số SV	Họ và tên	Công việc	Mức hoàn thành
19127242	Đỗ Vương Phúc	- Lên ý tưởng - Viết phần mở đầu - Thiết kế demo - Tìm hiểu và viết về MediaPipe - Kiểm tra và định dạng documents	100%
19127027	Võ Hoàng Bảo Duy	- Viết tổng quan bài toán - Tìm hiểu LSTM và RNN - Viết đề xuất mô hình - Viết phần nhận diện và truy vết	100%
19127067	Lê Minh Sĩ	- Tìm hiểu và viết về CNN - Viết hướng phát triển - Tìm kiếm database - Thiết kế slide	100%

2 Danh mục các từ viết tắt

Từ viết tắt	Từ gốc / Tiếng Anh
CCTV	Closed-Circuit Television and Video
AI	Artificial Intelligent
LSTM	Long-short Term Memory
CNN	Convolution Neural Network
RNN	Recurrent Neural Network
CV	Computer Vision
DL	Deep Learning
ANN	Artificial Neural Network
NLP	Natural Language Processing
YOLO	You Look Only Once
MOT	Multiple Object Tracking
SORT	Simple Online Realtime Object Tracking

3 Lời mở đầu

3.1 Lý do chọn đề tài

Trong thời đại công nghệ thông tin và kỹ thuật phát triển, khối lượng dữ liệu và lượng thông tin đến từ nhiều nguồn không ngừng tăng lên, đặc biệt một trong số đó là số lượng hình ảnh, video,... ngày càng lớn. Do đó, việc phân loại và phát hiện các vấn đề từ các hình ảnh, video thu thập được là nhu cầu hết sức cần thiết để phục vụ cho công tác nghiên cứu phát triển các ứng dụng hỗ trợ con người giải quyết các vấn đề khó khăn trong cuộc sống hàng ngày.

Những năm gần đây, với sự phát triển vượt bậc của lĩnh vực Thị giác máy tính (Computer Vision). Các hệ thống xử lý ảnh lớn trên thế giới như Facebook, Google… đã đưa những sản phẩm của mình vào áp dụng thực tế như: xe tự hành, nhận diện khuôn mặt người dùng,…

Ở nhiều nước, các mô hình sử dụng CCTV để theo dõi đã đạt được nhiều kết quả tích cực. Như các dự án sử dụng camera để nhận biết sớm và ngăn chặn việc tự tử[3], cụ thể là ở sông Hàn ở Hàn Quốc[4].

Các phương pháp hiện tại đã đạt được các thành công cho các nhất định như nhận diện hành vi bất thường trong nhà thông minh[5], trong đám đông[6]. Tuy nhiên, vẫn chưa có các ứng dụng cho các chung cư, tòa nhà doanh nghiệp.

Hiện nay, ở hầu hết chung cư đều đã được trang bị hệ thống camera giám sát, chúng cung cấp các dữ liệu hình ảnh ở mọi lúc, mọi nơi. Tuy nhiên, trong khi số lượng camera được triển khai dường như phát triển với tốc độ đáng kinh ngạc cũng như với chất lượng hình ảnh được cải thiện,

và mục đích chính là để theo dõi trực tiếp hoặc ghi dữ liệu. Với một số lượng lớn camera như vậy, cần một đội ngũ nhân nhiên bảo vệ theo dõi sát sao, liên tục để đảm bảo được tình hình an ninh của khu vực cũng như là đạt được hiệu quả nhất.

Vì vậy, với một số cải tiến trong công nghệ cho phép các hệ thống camera phát hiện và cảnh báo với đội ngũ giám sát khi phát hiện một số hoạt động đáng ngờ. Với sự phát triển của trí tuệ nhân tạo (AI), học máy, học sâu; các hoạt động đáng ngờ, bất thường được tự động hiểu và phát hiện bởi việc phân tích hành vi của người trên các hành động cụ thể của họ, khu vực xung quanh hoặc sự kiện mà họ đang tham gia.

3.2 Mục đích nghiên cứu

Ngày nay, mỗi chung cư/tòa nhà trong các thành phố lớn đều trang bị hệ thống camera an ninh khắp các lối đi. Vì vậy, việc quan sát các camera an ninh của đội ngũ giám sát đôi khi xảy ra sai sót, chỉ với 1 giây không quan sát những hoạt động đáng nghi ngờ (trộm, cắp,…) có thể làm ảnh hưởng đến an ninh của chung cư. Do đó, để hỗ trợ đội ngũ giám sát và đảm bảo camera ở chung cư luôn được theo dõi 24/7 thì cần có sự can thiệp của công nghệ. Cụ thể, trong trường hợp này cần có mô hình trích xuất các đặc trưng của video để có thể phát hiện, khoanh vùng chính xác những đối tượng có hành vi bất thường trong thời gian thực (real-time) là một điều hết sức quan trọng.

Với sự phát triển của các mô hình học sâu (Deep learning), cùng với đó là các vấn đề về phần cứng (CPU, GPU,…) không trở thành mối e ngại trong việc huấn luyện mô hình học sâu. Do đó việc lựa chọn mô hình học sâu trong trường hợp này là hướng tiếp cận có thể giải quyết được bài toán đặt ra.

3.3 Mục tiêu nghiên cứu

Như đã phân tích trước đó, việc tự động phát hiện và khoanh vùng các hành vi bất thường từ camera là nhu cầu cấp thiết trong việc đảm bảo an ninh, xác định những rủi ro sớm nhất để có thể ngăn chặn chúng. Với mục đích nghiên cứu để giải quyết các vấn đề ở trên, trong việc nghiên cứu này chúng tôi đặt ra mục tiêu nghiên cứu sau.

Mục tiêu của đề tài nghiên cứu là phát hiện và khoanh vùng các đối tượng có hành vi lạ theo thời gian thực thông qua đoạn video trích xuất từ camera.

Để thực hiện được mục tiêu như trên, ta cần tìm hiểu và nghiên cứu những vấn đề về những mô hình học sâu (Deep Learning) cụ thể là: CNN trích xuất đặc trưng của video, từ đó theo vết các đối tượng, ước lượng dáng người (Pose Estimation) và kết hợp LSTM để phân loại hành vi, từ đó có thể giải quyết được bài toán đã đặt ra.

3.4 Đối tượng nghiên cứu và phạm vi nghiên cứu

3.4.1 Đối tượng nghiên cứu

Để có thể hoàn thành mục tiêu nghiên cứu, đầu tiên ta phải nghiên cứu về những vấn đề liên quan đến xử lý video (định dạng, các chuẩn loại phim) để có những bước triển khai thích hợp. Kế đến, ta cần tìm hiểu các nghiên cứu liên quan về việc phát hiện đối tượng (object detection), theo vết đối tượng (object tracking) và ước lượng dáng người của đối tượng (pose estimation) trong video.

Bên cạnh đó, nghiên cứu còn khảo sát, phân tích và đánh giá ưu nhược điểm của các phương pháp phát hiện, truy vết và dự đoán tư thể của đối tượng, kết hợp với các mô hình tính toán của các phương pháp có sẵn để xây dựng một mô hình mới phù hợp với mục tiêu của bài toán theo phát hiện và khoanh vùng các đối tượng có hành vi lạ.

3.4.2 Phạm vi nghiên cứu

Đề tài được thực hiện chủ yếu trong việc phát hiện, theo dõi và dự đoán tư thế của đối tượng từ đó khoanh vùng các đối tượng có hành vi lạ thông qua camera được lắp đặt tại các tòa nhà, chung cư. Tập dữ liệu để thực hiện là các video chuẩn dành cho các nghiên cứu trong lĩnh vực liên quan và dữ liệu thu thập trong thực tế từ camera ghi hình được đặt tại một số tòa nhà, chung cư tại Việt Nam.

Mô hình được đề xuất sẽ có kỳ vọng phù hợp với bài toán phát hiện và khoanh vùng các đối tượng có hành vi lạ, tập dữ liệu để huấn luyện mô hình này có kích thước vừa đủ để huấn luyện mô hình. Ngoài ra, mô hình còn có thể chạy trên những được hệ thống phần cứng không quá mạnh trong quá trình thực hiện phân tích các đoạn video từ camera.

3.5 Phương pháp nghiên cứu

Vận dụng nhiều phương pháp khác nhau: khảo sát, thực nghiệm, phân tích và mô hình hóa. Cụ thể:

- Khảo sát, phân tích đánh giá các phương pháp đã có. Tìm hiểu các bài báo liên quan đến vấn đề nghiên cứu, đánh giá ưu và khuyết điểm từ đó chọn phương pháp phù hợp để giải yêu cầu đặt ra.
- Xây dựng mô hình trên dữ liệu thực tế (nếu có thể) để nâng cao và đưa ra mô hình có tính khả thi và kết quả tốt nhất.

Phương pháp tìm hiểu lý thuyết: Tìm hiểu mô hình học sâu: CNN, LSTM (RNN) và Pose Estimation của MediaPipe.

Phương pháp thực nghiệm trên dữ liệu mẫu: nghiên cứu và sử dụng các mô hình CNN, LSTM và RNN để xây dựng, cài đặt và huấn luyện mô hình đã đề xuất. Từ mô hình đề xuất đã được huấn luyện, đánh giá kết quả, hiệu năng của mô hình trên tập dữ liệu độc lập với dữ liệu huấn luyện.

3.6 Đóng góp đề tài

- Cấu trúc mạng CNN áp dụng cho bài toán phát hiện (detecting) và truy vết (tracking) cho bài toán được nêu ra.
- Mô hình MediaPipe được kết hợp từ BlazePose và GHUM
- Cấu trúc LSTM được áp dụng cho việc ước lượng dáng người từ đó phân lớp các hành vi bất thường.
- Ở bài cáo báo này, chúng tôi sẽ trình bày một mô hình kết hợp giữa CNN và LSTM để có thể giải quyết bài toán đặt ra.
- Thu thập và xây dựng hệ thống dữ liệu để phục vụ cho quá trình huấn luyện và đánh giá mô hình đề xuất.

3.7 Nội dung báo cáo

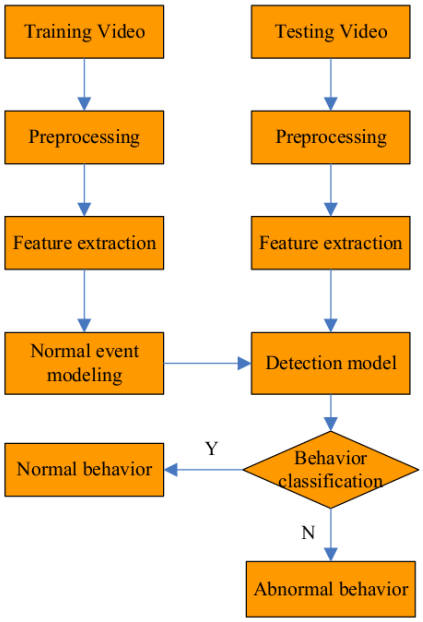
Nội dung của báo cáo gồm các phần chính:

- Tổng quan về bài toán phát hiện hành vi lạ và bài toán khoanh vùng đối tượng: Trong phần này, báo cáo tập trung vào việc giới thiệu, khái quát vấn đề việc phát hiện, truy vết các đối tượng và việc ước lượng dáng người để phân loại hành vi lạ. Bên cạnh đó, trình bày những bài báo, nghiên cứu liên quan đến vấn đề mà chúng tôi đang nghiên cứu.
- Cơ sở lý thuyết: Ở phần này, chúng tôi tập trung nói về cơ sở lý thuyết khi thực hiện đề tài, cụ thể là các vấn đề liên quan đến mô hình CNN, RNN, LSTM, BlazePose và GHUM.
- Phương pháp đánh giá mô hình đã đề xuất.
- Xây dựng mô hình học sâu phát hiện và khoanh vùng đối tượng có hành vi lạ trong chung cư: Trong phần này, chúng tôi nêu lên mô hình mà chúng tôi đã nghiên cứu và đề xuất
- Thực nghiệm và đánh giá kết quả: Chúng tôi tập trung vào thực hiện phân tích tập dữ liệu để huấn luyện mô hình. Dựa vào đó mà đánh giá độ chính xác của mô hình đã thực hiện.
- Kết luận và hướng phát triển.

4 Tổng quan về bài toán phát hiện hành vi lạ và truy vết khoanh vùng đối tượng

4.1 Bài toán phát hiện hành vi lạ

Bài toán phát hiện hành vi lạ đã được nghiên cứu từ rất lâu[1], mục đích của bài toán là xác định đối tượng cụ thể ở đây người có trong ảnh hoặc các khung hình của video có hành vi bất thường hay không. Để giải quyết bài toán này, chúng ta cần phải thu thập dữ liệu, xử lý các dữ liệu, rút trích các đặc trưng và đào tạo model để thực hiện việc xác định xem mỗi khung hình của video có xảy ra hành vi nào được coi là bất thường.



Hình 1. Mô hình cơ bản của phát hiện hành vi bất thường

Có thể coi bài toán phát hiện hành vi bất thường là một bài toán phân lớp.

Phân lớp (Classification): là một quá trình xử lý nhằm sắp xếp, gán nhãn (label) của đối tượng vào một lớp (class) nào đó. Các mẫu dữ liệu hoặc đối tượng được phân lớp dựa vào thuộc tính hoặc đặc trưng của chúng. Cụ thể ở bài toán đề cập, chúng ta cần sử dụng đặc trưng của từng frame video để xem xét các đối tượng trong video thuộc lớp nào: *normal behavior* hoặc *abnormal behavior*.

Với sự phát triển của công nghệ thông tin, kỹ thuật số và bùng nổ dữ liệu như hiện nay, việc phân lớp trở thành nhu cầu không thể thiếu trong nhiều lĩnh vực, bài toán chúng ta đang xem xét cũng không ngoại lệ. Hiện nay, có rất nhiều mô hình phân lớp khác nhau, do đó việc lựa chọn mô hình phân lớp để mang lại hiệu quả và độ chính xác là việc hết sức quan trọng.

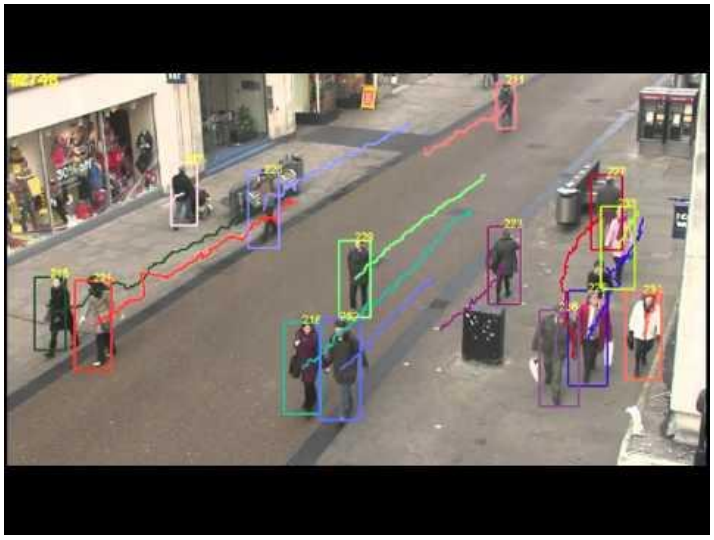
Nhìn chung, đa số các hành vi ở con người đều có thể xác định bằng dáng người. Do đó, chúng tôi chúng tôi sử dụng việc ước lượng dáng (***pose estimation***) để giúp bài toán phân lớp chạy tốt hơn[2].

Pose estimation: là vấn đề chung trong lĩnh vực Thị giác Máy tính (CV). Mục đích của nó là xác định vị trí và hướng của con người hay đối tượng. Thông thường, việc này là dự đoán các vị trí của các điểm trọng yếu (key points) như tay, đầu, chân,… trong trường hợp ước tính tư thế của người (*Human Pose Estimation*).

Hình 2. Ước lượng dáng người (Pose estimation)

4.2 Bài toán truy vết và khoanh vùng đối tượng

Bài toán truy vết đối tượng (*Object tracking*) cũng là một trong những bài toán kinh điển của Thị giác máy tính (CV). Ở đây, bài toán phải lập ra tập hợp các đối tượng đã được phát hiện (*Object detection*), từ đó gán định danh (*ID*) cho mỗi đối tượng đã phát hiện được và cuối cùng là hiện truy vết các đối tượng khi chúng di chuyển trên các khung hình (frame) tiếp theo trong video. Truy vết người (*People tracking*) trong video là một trường hợp cụ thể của bài toán truy vết đối tượng.



Hình 3. Truy vết đối tượng (Object tracking)

Dựa vào việc truy vết đối tượng, từ đó mới thực hiện khoanh vùng các đối tượng. Với sự hiệu quả của các mô hình học sâu như CNN thì việc sử dụng chúng trong bài toán truy vết là việc hết sức khả thi[7].

4.3 Tình hình nghiên cứu

Những năm gần đây, bài toán phát hiện hành vi bất thường được các nhóm nghiên cứu sôi nổi bằng việc sử dụng các phương pháp khác nhau. Dưới đây là một số phương pháp mà chúng tôi tìm kiếm được.

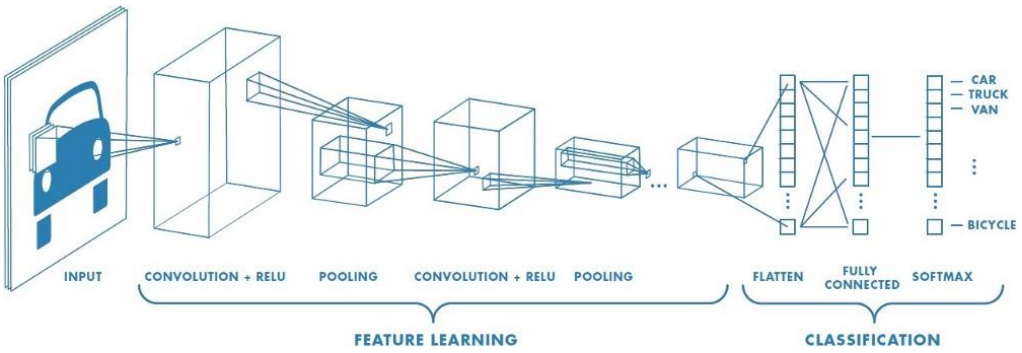
Ở bài báo này[8], nhóm tác giả đề xuất việc nhận dạng hoạt động và những hành vi bất thường sử dụng các mô hình RNN như: LSTM, GRU… Tại việc nhận dạng hoạt động được xem xét như việc gán nhãn cho một trình tự (sequence labeling), khi đó các hành vi bất thường được gắn cờ (flag) dựa trên độ lệch của chúng so với những hành vi bình thường.

Nhóm tác giả[9] kết hợp giữa việc truy vết các đối tượng và rút trích đặc trưng sử dụng mô hình CNN và việc dùng mô hình LSTM để đưa ra phân lớp các hành vi mang lại hiệu quả tương đối đáng kể trong bài toán này.

5 Cơ sở lý thuyết

5.1 Convolution Neural Network (CNN)

Trong các mô hình học sâu, có thể coi CNN hay còn gọi là mạng nơ-ron tích chập là một trong những mô hình đặc trưng trong DL. Trong những năm gần đây, mô hình CNN được sử dụng nhiều trong lĩnh vực CV, được xây dựng với nhiệm vụ để nhận dạng và phân loại ảnh. Trong đó, xác định các đối tượng (object detection) là một trong những lĩnh vực áp dụng rộng rãi.



Hình 4. Mô hình cơ bản của CNN trong nhiệm vụ phân lớp ảnh

Trong mô hình này, thay vì chỉ sử dụng các lớp Fully Connected như trong ANN, mô hình CNN còn sử dụng các lớp tích chập (Convolution layer) và Pooling layer trước khi sử dụng Fully Connected để phân lớp hay dự đoán.

5.1.1 Lớp tích chập – Convolution Layer.

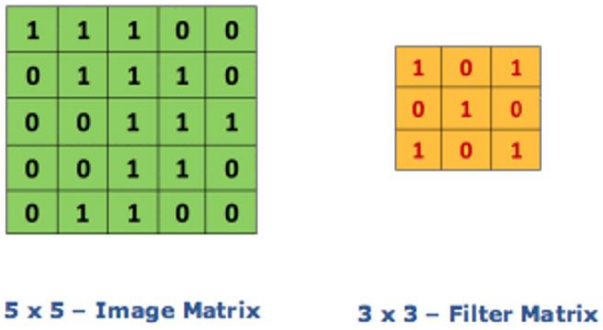
Lớp tích tập là đầu tiên để trích xuất đặc trưng từ các ảnh đầu vào trong mô hình CNN. Việc sử dụng các lớp tích chập này nhằm duy trì mối quan hệ giữa các điểm ảnh (pixel) bằng sử dụng kernel (được khởi tạo) và trượt chúng trên các điểm ảnh.

Tích chập là một phép toán thực hiện với hai hàm số f và g, kết quả sẽ cho ra một hàm số - là kết quả của phép tích chập $f°g$. Trong lĩnh vực xử lý ảnh, tích chập được sử dụng rộng rãi và hết sức quan trọng trong việc rút trích các đặc trưng của ảnh.

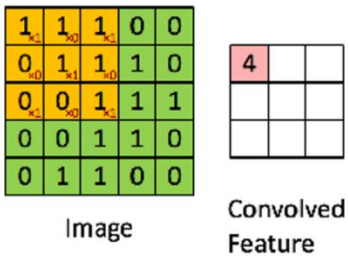
Công thức tích chập giữa hàm ảnh $f(x,y)$ và bộ lọc (filter) $k(x,y)$ (có kích thước $m \times n$) được biểu diễn như sau:

$$f(x,y)\cdot k(x,y) = \sum_{u=\frac{-m}{2}}^{\frac{m}{2}} \sum_{v=\frac{-n}{2}}^{\frac{n}{2}} k(u,v)\, f(x-u, y-u)$$

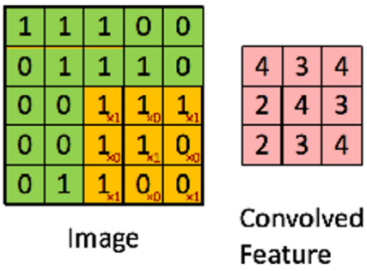
Ví dụ: Với một đầu vào có kích thước 5x5, và filter được khởi tạo có kích thước 3x3. Quá trình thực hiện tích chập (convole) được mô tả như sau.



Hình 5. Tích chập giữa ảnh đầu vào và bộ lọc



Hình 6. Mô phỏng quá trình bộ lọc trượt trên ảnh (1)



Hình 7. Mô phỏng quá trình bộ lọc trượt trên ảnh (2)

Bộ lọc (filter): là một phần quan trọng của lớp lớp tích chập. Đây là một ma trận được khởi tạo với kích thước *m x n*, và thước có kích thước rất nhỏ so với kích thước của ảnh (thường được khởi tạo với kích thước 5x5 hoặc 3x3) nhằm mang lại kết quả tốt nhất.

Stride: trong quá trình trượt bộ lọc trên ảnh đầu vào, bộ lọc sẽ lần lượt dịch chuyển theo chiều ngang hoặc dọc một giá trị, giá trị này được gọi là **Stride** (bước trượt). Ở mỗi lần dịch chuyển (trượt từ trái qua phải, từ trên xuống dưới), sẽ thực hiện tính toán kết quả cho điểm ảnh đang xét bằng công thức tích tập được mô tả như trên.

Padding: đôi khi trong việc thực hiện tích chập, bộ lọc filter không phù hợp với ảnh đầu vào. Ta có thể thêm ***padding*** (đường viền) vào 4 đường biên của ảnh trước khi thực hiện phép tích chập để đảm bảo kích thước đầu ra là không đổi.

Feature map: là kết quả sau khi thực hiện tích chập giữa ảnh đầu vào và bộ lọc khi quét qua hết. Kích thước của kết quả đầu ra được tính theo công thức sau:

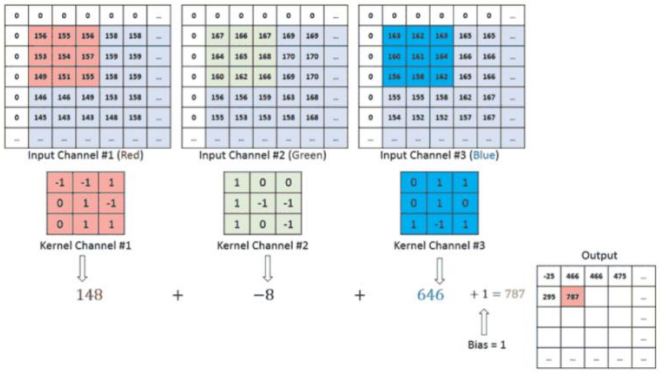
$$W_1 = \frac{W_0 + 2P - F}{S} + 1, H_1 = \frac{H_0 + 2P - F}{S} + 1$$

Với

- W_0, H_0 là kích thước của ảnh đầu vào.
- W_1, H_1 là kích thước của ảnh đầu vào.
- P là kích thước của padding.

- F là kích thước của bộ lọc.
- S là bước trượt (stride)

Việc sử dụng ảnh đầu vào với kích thước $W_0 \times H_0 \times C_0$ (với C_0 là số channel của ảnh), thì quá trình tích chập được thực hiện lần lượt trên từng channel và sau đó lấy tổng giữa các channel. Ta có thể thêm bias vào kết quả sau khi thực hiện tích chập.

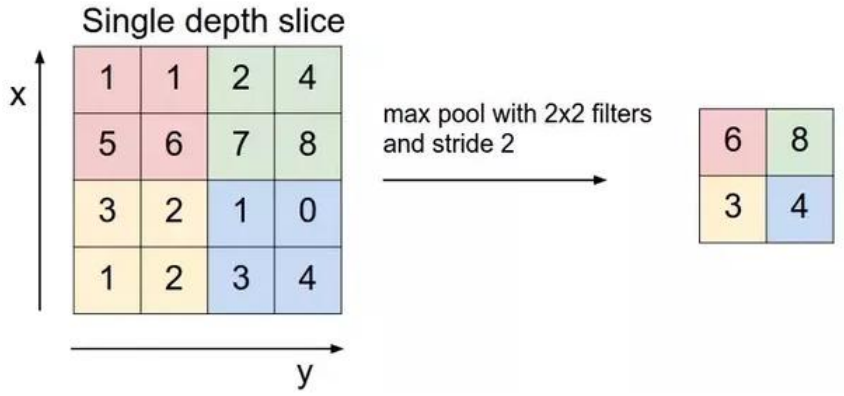


Hình 8. Quá trình thực hiện tích chập với ảnh nhiều channel và áp dụng bias

5.1.2 Pooling layer.

Trong mô hình CNN, thông thường giữa các lớp tích tập với nhau, người ta thường chèn vào một lớp **Pooling** để giảm bớt số lượng tham số (parameter) lại nếu như ảnh đầu vào quá lớn. Việc áp dụng pooling layer giúp giảm kích thước của không gian mẫu, tuy nhiên vẫn giữ được những nét đặc trưng cơ bản của ảnh đầu vào. Ở quá trình này, cũng sử dụng cửa sổ (window size) trượt trên ảnh, pooling có nhiều loại khác nhau như:

- Max pooling: với mỗi lần trượt window, chọn ra pixel lớn nhất.
- Average pooling: chọn ra trung bình giữa các pixel ứng với kích thước của window.
- Sum pooling: tổng tất cả các pixel trong window.



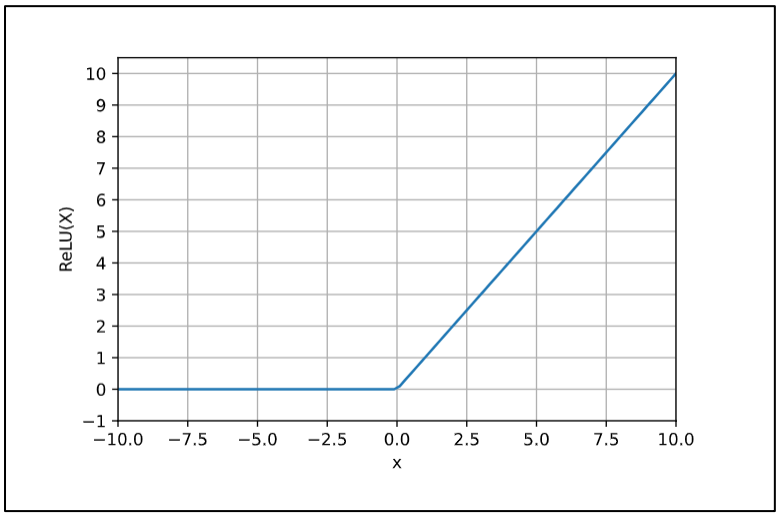
Hình 9. Quá trình áp lớp Max Pooling sau khi thực hiện tích chập

5.1.3 Hàm kích hoạt – Activation function.

Thông thường sau khi thực hiện tích chập, feature map được tính ra sẽ được áp dụng hàm kích hoạt lên tất cả các giá trị của feature map. Một số hàm kích hoạt thường được sử dụng như: **ReLU (Rectified linear unit), Tanh, Sigmoid.**

- **ReLU**: là hàm số được mô tả theo công thức:

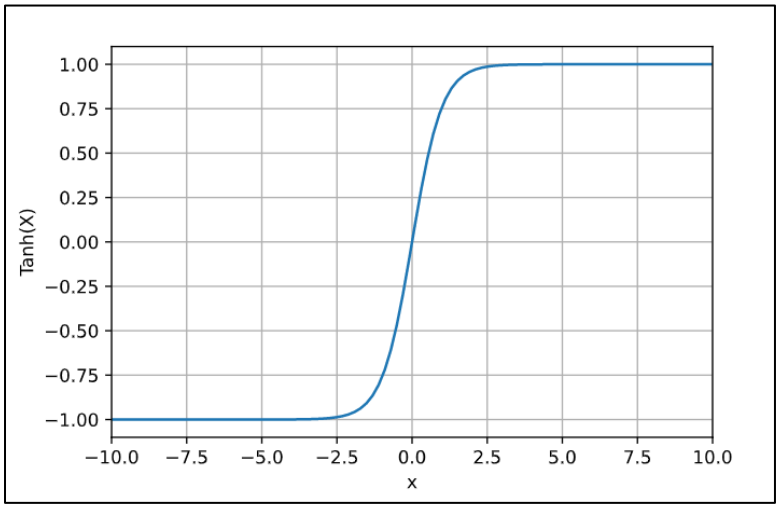
$$f(x) = \max(0, x)$$



Hình 10. Hàm kích hoạt - ReLU

- **Tanh:** là hàm kích hoạt phi tuyến tính được định nghĩa theo công thức:

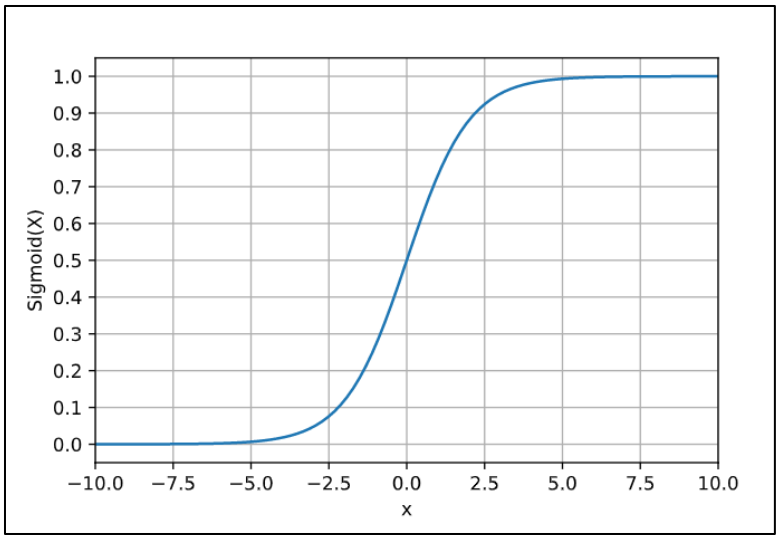
$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



Hình 11. Hàm kích hoạt - Tanh

- **Sigmoid:** là hàm kích hoạt phi tuyến tính.

$$\mathrm{f}(x) = \frac{1}{1 + e^{-x}}$$



Hình 12. Hàm kích hoạt - Sigmoid

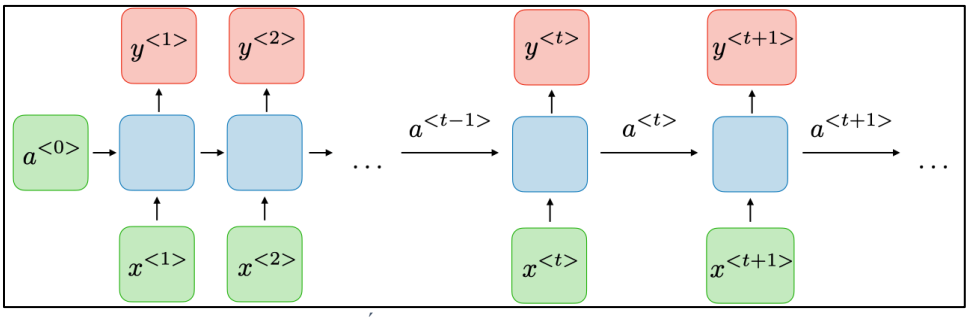
5.1.4 Fully connected layer.

Sau khi thực hiên rút trích đặc trưng của ảnh qua các lớp tích chập và pooling, kết đến để thực hiện quá trình dự đoán kết quả, ta sử dụng lớp *Fully connected* thêm vào sau một mạng CNN. Lớp này cũng giống như các lớp trong mô hình mạng ANN.

5.2 Recurrent Neural Network (RNN)

Mô hình RNN cũng là một dạng của mô hình ANN, mô hình thường được sử dụng trong việc dữ liệu mang ý nghĩa trình tự (hay con gọi là **sequential data**) như: giọng nói (speech), text, âm thanh…, tức là nếu thay đổi trình tự của dữ liệu thì bài toán sẽ nhận được một kết quả khác.

RNN giúp chúng ta lưu lại những trạng thái hoặc thông tin của đầu vào trước đó để tạo ra một trình tự học. Dưới đây là kiến trúc thường gặp của mô hình RNN:



Hình 13. Kiến trúc thường gặp của mô hình RNN

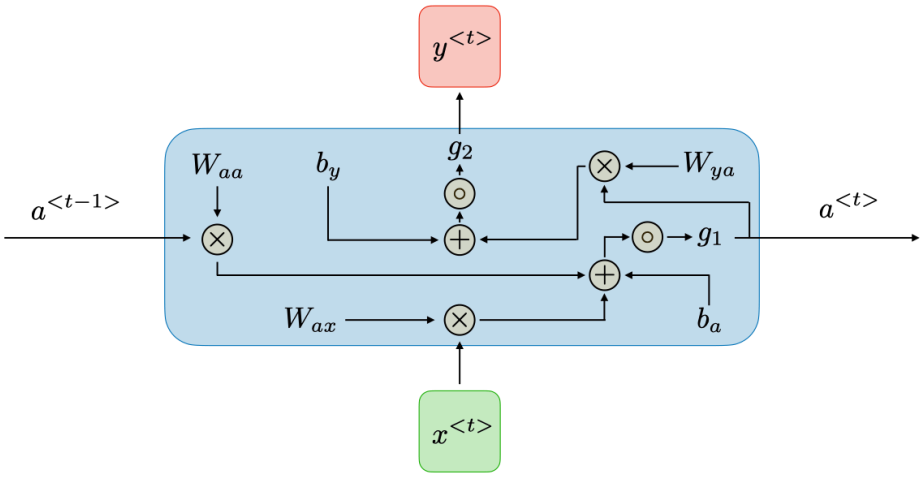
Ở mỗi bước t, ta có giá trị đã được áp dụng hàm kích hoạt $a^{<t>}$ và giá trị đầu ra là $y^{<t>}$. Ta có thể biểu diễn như sau:

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Với :

- $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ là các hệ số được chia sẻ tạm thời
- g_1, g_2 là các hàm kích hoạt

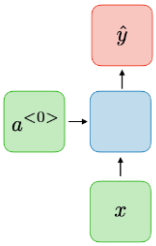


Hình 14. Mô hình tính toán của RNN

Mô hình RNN được sử dụng trong nhiều lĩnh vực của xử lý ngôn tự nhiên (NLP) và nhận diện giọng nói. Với mỗi việc áp dụng mô hình RNN vào từng mục đích cụ thể, RNN được chia thành các loại sau:

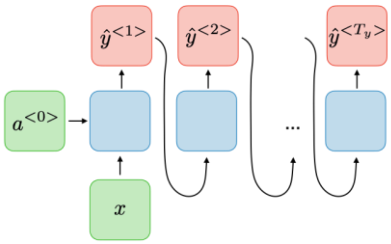
Gọi T_x là số dữ liệu đầu vào, T_y là số giá trị cần trả ra.

- **One-to-one:** mạng neural truyền thống. $T_x = T_y = 1$.



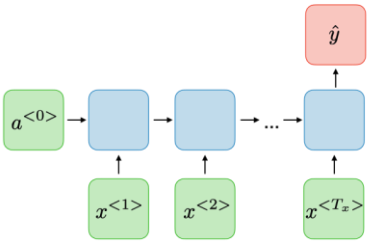
Hình 15. Mô hình One-to-one

- **One-to-many:** thường được sử dụng trong Music generation. $T_x = 1, T_y > 1$.



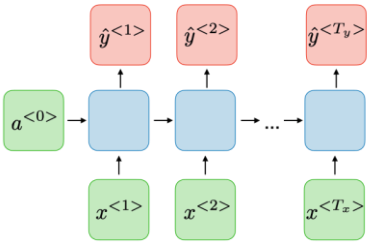
Hình 16. Mô hình One-to-many

- **Many-to-one:** sử dụng trong các bài toàn phân loại cảm xúc (Sentiment classification). $T_x > 1, T_y = 1$.

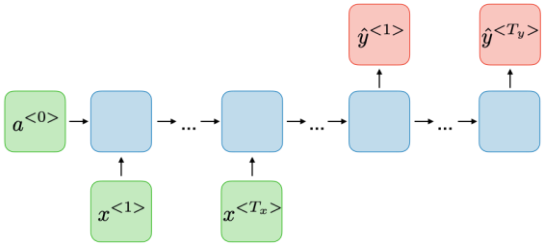


Hình 17. Mô hình Many-to-one

- **Many-to-many**: thường được sử dụng trong lĩnh vực dịch máy (Machine Translation) nếu $T_x \neq T_y$, hoặc sử dụng trong nhận dạng tên của thực thể (Name entity Recognition) nếu $T_x = T_y$.



Hình 18. Mô hình Many-to-many (1)



Hình 19. Mô hình Many-to-many (2)

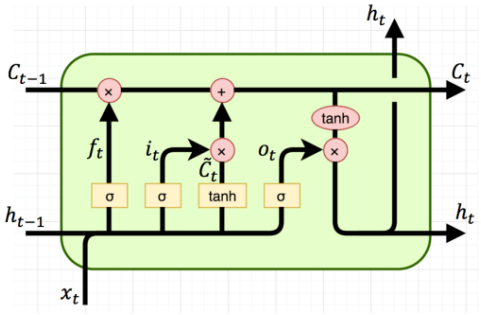
Tuy nhiên, ở mô hình RNN nói trên lại xảy ra 2 vấn đề lớn trong quá trình huấn luyện:

- **Exploding gradients**: trong quá trình lan truyền ngược (Backpropagation) có thể dẫn đến tình trạng đạo hàm trên từng tham số rất lớn dẫn đến việc cập nhật tham số sai sót, việc này cải thiện bằng cách sử dụng kỹ thuật cắt giảm giá trị của đạo hàm (truncating gradients).
- **Vanishing gradients**: xảy ra các giá trị đạo hàm quá nhỏ (xấp xỉ gần bằng 0), nên việc cập nhật các tham số trong quá trình huấn luyện trở nên vô nghĩa. Có thể giải quyết vấn đề này qua việc sử dụng LSTM.

5.3 Long Short-Term Memory (LSTM)

Để giải quyết vấn đề mà mô hình RNN gặp phải, LSTM được lên ý tưởng bởi Sepp Hochreiter và Juergen Schmidhuber để hạn chế vấn đề **Vanishing gradients**. LSTM là một mô hình mở rộng của RNN, về cơ bản là mở rộng bộ nhớ hơn so với mô hình RNN thông thường.

LSTM cho phép RNN ghi nhớ các đầu vào trước đó trong khoảng thời gian dài, bởi vì LSTM chứa các thông tin trên bộ nhớ giống như bộ nhớ của máy tính. LSTM có thể đọc, ghi và xóa các thông tin từ bộ nhớ. Ở LSTM, nó bao gồm các cổng (**gates**) để điều chỉnh luồng thông tin qua các unit một cách tốt hơn. Cụ thể:



Hình 20. Mô hình cơ bản của LSTM

Ta có thể biểu diễn công thức của LSTM như sau:

$$
\begin{aligned}
f_t &= \sigma_g(W_t x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_0 x_t + U_0 h_{t-1} + b_0) \\
\tilde{c}_t &= \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

Với:

- σ_g là hàm kích hoạt sigmoid
- σ_c là hàm kích hoạt tanh
- σ_h là hàm kích hoạt tanh hoặc $\sigma_h(x) = x$
- f_i là cổng forget
- i_t là cổng input
- o_t là cổng output
- $\tilde{c}_t$ là cổng input của cell
- c_t là trạng thái cell
- h_t là vector của trạng thái ẩn (hidden state)

Nhìn vào cấu trúc một unit của LSTM, có thể thấy các cổng đầu ra sử dụng hàm kích hoạt sigmoid (giá trị của hàm này nằm trong khoảng 0 đến 1) nên vấn đề ***vanishing gradients*** đã được giải quyết, giúp quá trình huấn luyện mô hình nhanh chóng hội tụ và mang lại độ chính xác cao hơn so với mô hình truyền thống RNN.

6 Xây dựng mô hình phát hiện, khoanh vùng đối tượng có hành vi lạ trong chung cư

Trong phần này, chúng tôi sẽ trình bày về các mô hình sử dụng trong từng nhiệm vụ riêng biệt từ đó thực hiện kết hợp giữa các mô hình để giải quyết vấn đề đã đặt ra trước đó. Bao gồm:

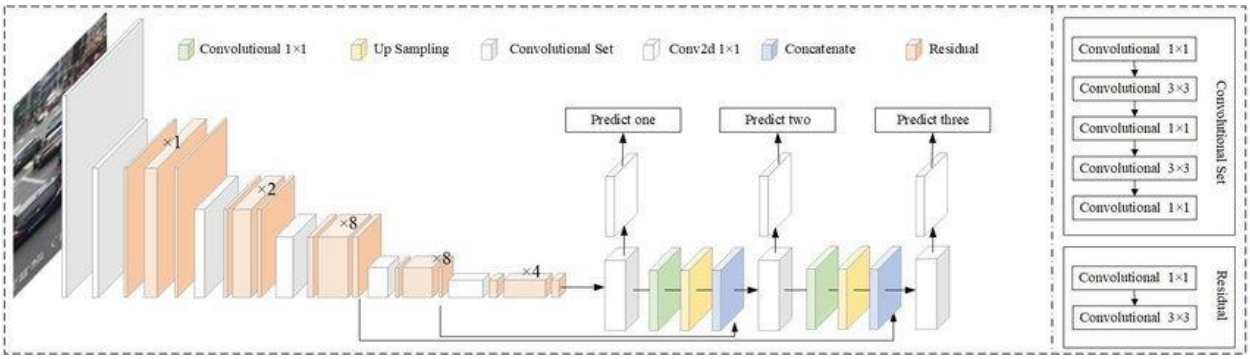
- Kỹ thuật phát hiện đối tượng
- Truy vết những đối tượng đã phát hiện được
- Trích xuất các dáng người của đối tượng đang truy vết
- Mô hình học sâu kết hợp để giải quyết vấn đề đặt ra

6.1 Phát hiện và truy vết đối tượng

Phát hiện đối tượng – Object detection

Trong nhiệm vụ phát hiện và truy vết đối tượng (cụ thể là người) trong từng khung hình của video, chúng tôi sử dụng mô hình học sâu CNN để sử dụng rút trích đặc trưng và đưa ra những dự đoán về các đối tượng xuất hiện trong khung hình.

Để đáp ứng được việc phát hiện đối tượng trong thời gian thực (real-time), chúng tôi sử dụng mô hình YOLO. Đây là một mô hình CNN với nhiệm vụ phát hiện và phân loại đối tượng trong thời gian thực, so với các mạng nhân tạo khác YOLO mang lại hiệu năng cao hơn về cả tốc độ và độ chính mang lại.



Hình 21. Mô hình YOLOv3

Đầu vào của mô hình: lần lượt các khung hình của video.

Đầu ra: lần lượt chứa các bounding box và các đối tượng trong bouding box. Mỗi bounding box gồm có 5 phần (x, y, w, h, prediction) với (x, y) là tọa độ của bounding box, (w, h) lần lượt là chiều rộng và cao của bounding box, prediction là score của đối tượng được dự đoán.

Với mô hình đã giải quyết được phần đầu là bài toán đặt ra, là phát hiện các đối tượng có trong ảnh/video, là tiền đề để thực hiện bước truy vết và khoanh vùng đối tượng. Ngoài ra, YOLO được phân ra nhiều pre-trained model để lựa chọn. Từ model nặng nhất (YOLOv5l) đến các model nhỏ hơn như YOLOv5n. Dĩ nhiên, đây là một sự đánh đổi, nếu muốn độ chính xác cao thì thời gian xử lý phải lâu. Đây cũng là một vấn đề cần được cân nhắc bằng cách thực nghiệm.

Truy vết đối tượng – Object tracking

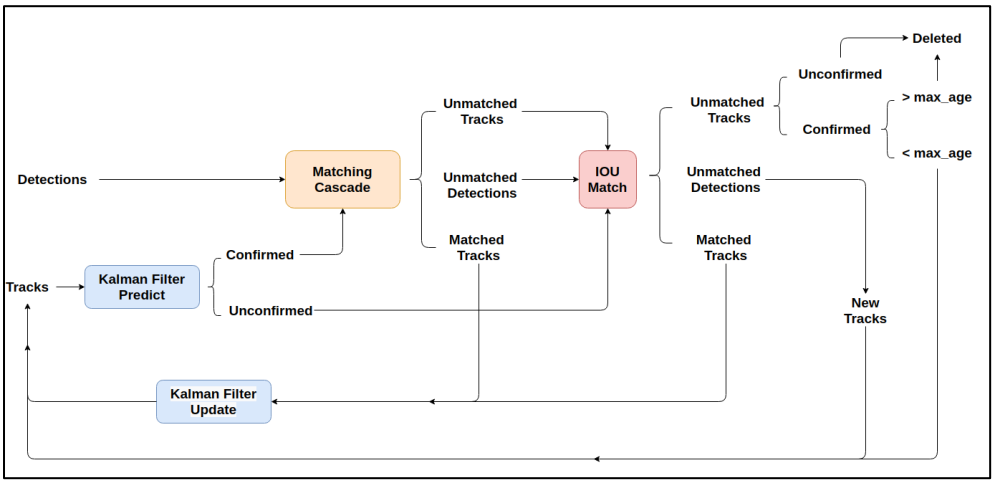
Với nhiệm vụ là truy vết đối tượng trong các khung hình của video, chúng tôi sử dụng mô hình Deep SORT[10] để giúp việc liên kết các đối tượng sau khi đã biến mất một thời gian được hiệu quả hơn.

Deep SORT là một mô hình được sử dụng trong nhiệm vụ MOT. Mô hình được chúng tôi thực hiện tuần tự các bước sau:

- Sử dụng YOLO (đã nêu ở phần trên) để phát hiện các đối tượng trong khung hình hiện tại.
- Sau đó, sử dụng Kalman Filter để dự đoán các trạng thái truy vết mới dựa trên các truy vết đã thực hiện trong quá khứ. Các trạng thái này lúc khởi tạo sẽ gán giá trị mang tính thăm dò, nếu giá trị vẫn đảm bảo duy trì trong 3 khung hình tiếp theo, trạng thái sẽ

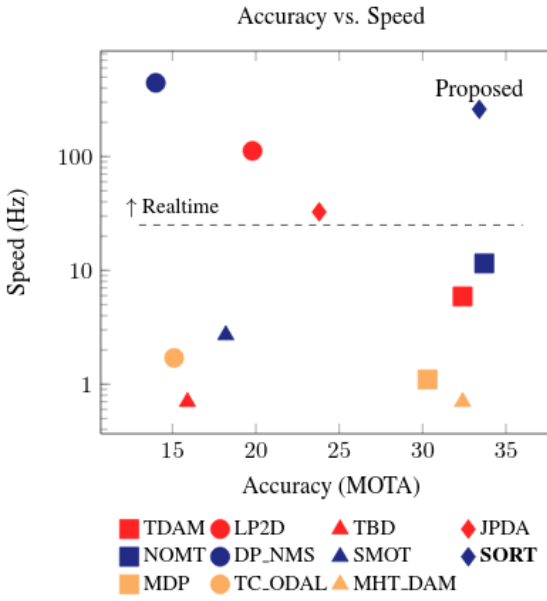
chuyển từ trạng thái thăm dò dang trạng thái xác nhận và sẽ cố gắng duy trì và theo dõi trong 30 khung hình tiếp theo.

- Dựa vào những truy vết đã xác nhận, đưa chúng vào chiến lược đối sánh phân tầng (matching cascade) nhằm liên kết với các phát hiện, dựa trên độ đo về khoảng cách và đặc trưng.
- Xử lý, phân lỗi các phát hiện và truy vết.
- Sử dụng Kalman Filter để hiệu chỉnh lại giá trị truy vết.



Hình 22. Luồng xử lý của deep SORT

Hơn nữa, nhóm đã lựa chọn Deep SORT vì tính real-time và độ chính xác cao hơn các giải thuật tracking. Dưới đây là biểu đồ so sánh một số giải thuật tracking:

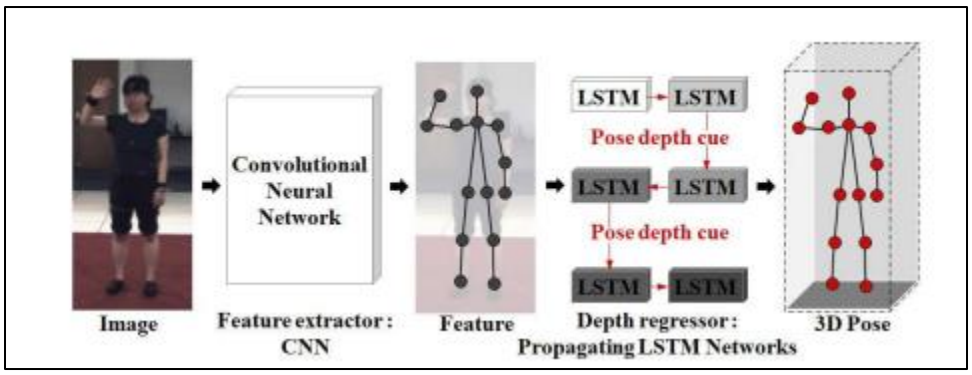


Hình 23. So sánh các giải thuật Tracking

6.2 Ước lượng dáng – Pose estimation

Để thực hiện việc phân lớp các hành vi lạ (hay bất thường) trong các video một cách chính xác, chúng tôi sử dụng việc ước lượng dáng trước khi thực hiện phân lớp để xem xét các dáng mà chúng tôi dự đoán, có phải là hành vi bất thường hay không.

Ví dụ: nếu khung hình trước dáng người trong video thực hiện hành vi đi đứng một cách bình thường, nhờ vào việc truy vết đối tượng này mà ở khung hình kế tiếp phát hiện đối tượng này đang thực hiện hành vi chạy, một cách bất chợt thì trong trường hợp này có thể coi ở khung hình này xuất hiện hành vi lạ và thực hiện cảnh báo nguy hiểm.

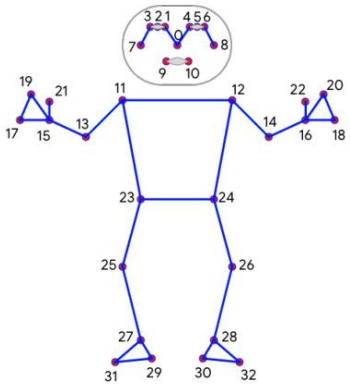


Hình 24. Quy trình sử dụng LSTM để ước lượng dáng người[11]

Tương tự như các mô-đun khác, Pose Estimation cũng cần có đặc tính xử lý nhanh và có khả năng chạy real-time. Để đạt được điều đó, mô hình MediaPipe do đội ngũ Google phát triển là lựa chọn phù hợp cho chúng tôi. MediaPipe được chia ra làm hai giai đoạn bao gồm:

- Ước lượng dáng 2D bằng BlazePose[12]
- Sau đó, ước lượng cho 3D bằng GHUM[13]

BlazePose là một mô hình dùng để ước lượng dáng người 2D. Điểm đặc biệt của BlazePose so với các mô hình khác là ở chỗ tốc độ xử lý. Ở đây, với ảnh input đầu vào, mô hình sẽ cho ra 33 key points của các vị trí trên cơ thể người.



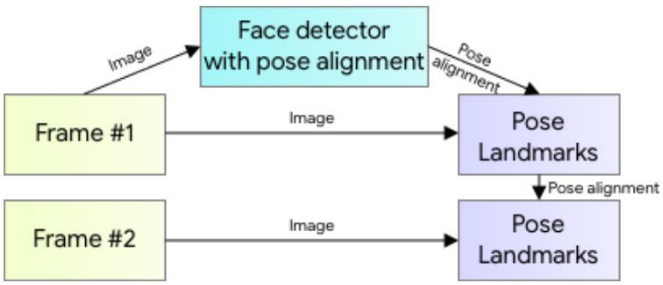
Hình 25. 33 Keypoints được cung cấp bởi BlazePose

Để có thể thực hiện được việc pose estimation một cách nhanh chóng, nhóm tác giả đã chia quá trình này ra thành hai phần chính:

- Nhận diện khuôn mặt (face detector)

- Truy vết dáng người (pose tracking)

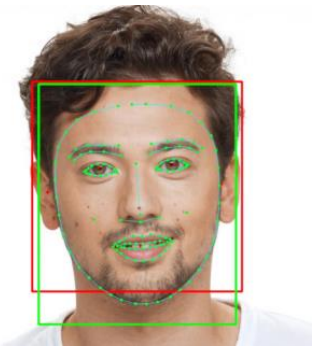
Bởi vì bài toán cần giải quyết của chúng ta là ước tính dáng người cho một chuỗi các video hoặc hình trực tiếp từ camera, nên dáng người giữa hai frame sẽ không thay đổi quá nhiều. Dựa vào đặc điểm đó, thay vì phải nhận diện lại từ đầu, nhóm tác giả chỉ cần căn chỉnh lại dáng sau khi lần nhận diện đầu tiên.



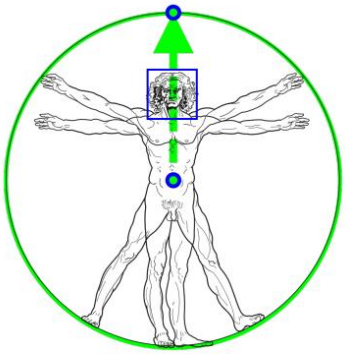
Hình 26. Quy trình Pose Estimation của BlazePose

Rất nhiều mô hình nhận diện đối tượng (object detection) ngày nay sử dụng Non-maximum suppression (NMS) ở bước cuối để loại bỏ các bounding box chồng nhau. Tuy nhiên, việc này còn có nhiều nhược điểm trong các ngữ cảnh khác nhau như là bắt tay, ôm nhau,… Do đó, nhóm tác giả chú trọng vào nhận dạng ra một bộ phận trên cơ thể người để ước tính những vị trí còn lại. Ở đây, nhóm đã kết luận rằng mặt người là vị trí dễ nhận ra nhất cho mạng neural network, vì chúng có các đặc trưng mang tính tương phản cao và ít biến thể khác nhau về mặt ngoại hình.

Để phát hiện được khuôn mặt với thời gian real-time, đội ngũ của Google sử dụng một mô hình khác cũng chính do họ thiết kế, đó là BlazeFace. Ngoài ra, mô hình này cũng đưa ra một số tham số khác để có thể căn chỉnh như là điểm giữa hai hông người, kích thước vòng tròn bao quanh người.



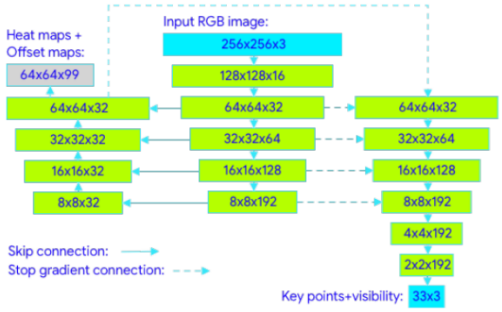
Hình 27. Nhận diện khuôn mặt BlazeFace



Hình 28. Căn chỉnh Vitruman man qua bộ nhận dạng khuôn mặt

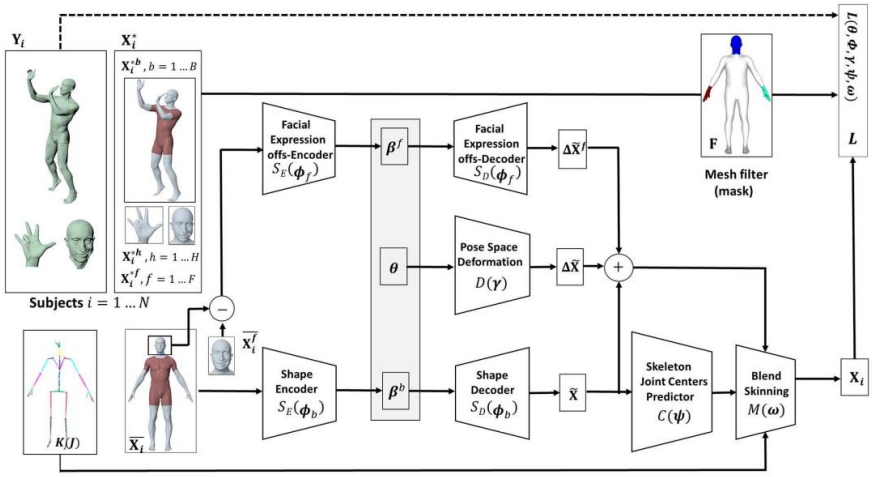
Cuối cùng, kiến trúc mô hình của BlazePose được sử dụng hai phần bao gồm heat maps, offset maps (bên trái); và một mạng hồi quy (bên phải). Trong quá trình huấn luyện, nhóm tác giả sử dụng mô hình trái và giữa trước. Sau đó, phần regression mới được train bằng cách chia sẻ các đặc trưng của mạng bên trái, tuy nhiên không thực hiện back-propagation cho mạng bên trái.

Trong quá trình chạy inference, đầu ra của mô hình heat map sẽ được loại bỏ, chỉ sử dụng đầu ra của mạng regression.



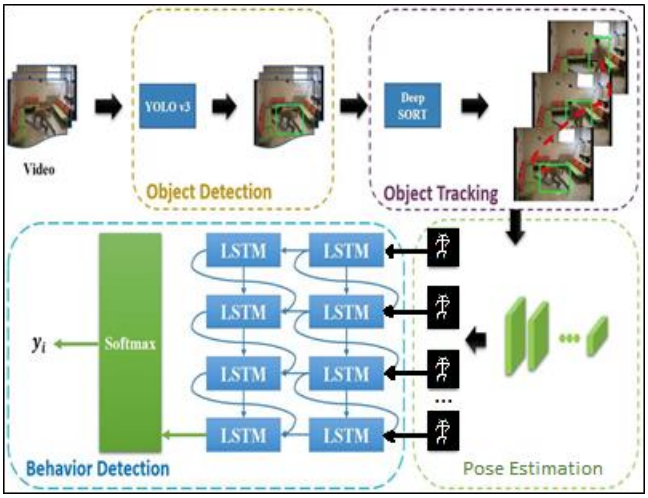
Hình 29. Mô hình của BlazePose

Sau đó, mô hình GHUM được áp dụng trên kết quả đầu ra của BlazePose. Đây là một mô hình rất phức tạp vì nó không chỉ sử dụng các thông tin đơn giản, mà còn dùng các mô hình người đã được scan dưới 3D để tạo thành các không gian mẫu.



Hình 30. Mô hình GHUM

6.3 Kiến trúc mô hình đề xuất



Hình 31. Mô hình phát hiện hành vi lạ do nhóm đề xuất

Kiến trúc mô hình được đề xuất gồm các giai đoạn:

- Phát hiện và truy vết đối tượng.

- Với mỗi đối tượng được phát hiện và truy vết thực hiện ước lượng dáng.
- Phân loại hành vi từ dáng được ước lượng.
- Khoanh vùng các đối tượng có hành vi lạ.

Để hiểu rõ hơn về mô hình, chúng tôi sẽ đi từ bài toán cơ bản nhất là nhận biết cho một đối tượng (single object). Với bài toán này, chúng ta dễ dàng sử dụng Pose Estimation để nhận biết dáng người và dùng LSTM để phân lớp. Tuy nhiên, với bài toán khó hơn – nhiều đối tượng (multiple objects), việc Pose Estimation sẽ khó khăn hơn vì lúc đó chúng ta không biết có bao nhiêu đối tượng. Để đảm nhiệm việc này, chúng tôi đề xuất sử dụng object detection, cụ thể là YOLO. Sau khi biết được vị trí của các đối tượng, vấn đề kế tiếp là cần phải biết được dáng người nào là của ai trong mỗi khung hình. Việc này sẽ được giải quyết bằng cách đánh định danh (ID) cho mỗi người trong khung ảnh, đây là bài toán Object tracking.

7 Thực nghiệm và đánh giá kết quả

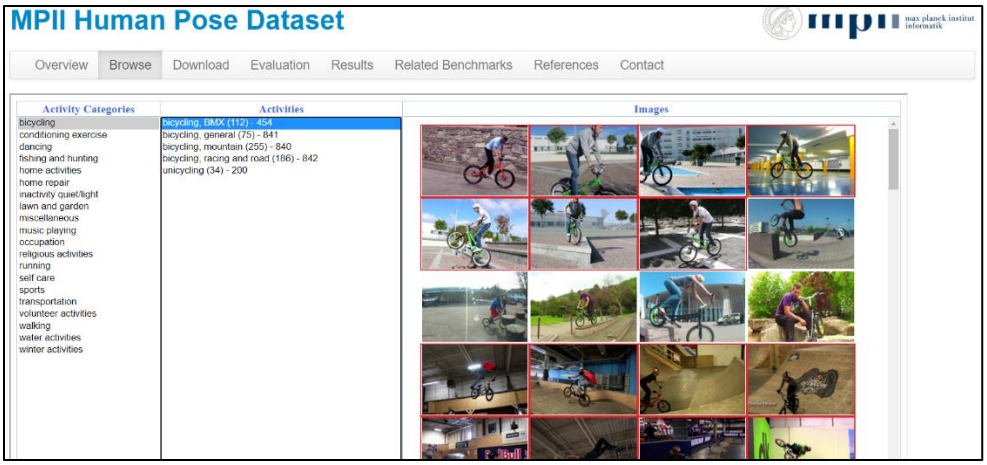
7.1 Cơ sở dữ liệu

Về bài toán Tracking: Chúng ta có thể sử dụng dữ liệu từ một workshop nổi tiếng trong mảng Computer Vision – ICCV 2021 MMPTRACK (iccv2021-mmp.github.io). Bộ dữ liệu này bao gồm các video khoảng 5 tiếng dùng để train và 1.5 tiếng dùng để validate. Các đối tượng trong video đều được kí hiệu bounding box xung quanh và đánh định danh sẵn. Hơn nữa, các video được ghi nhận từ camera ở các góc khác nhau. Các đối tượng tham gia vào xây dựng video này được trải rộng ở các độ tuổi, giới tính và chủng tộc khác nhau.

Hình 32. Dataset MMPTRACK

Về bài toán nhận diện Pose và phân loại hành vi: MPII Human Pose là một cơ sở dữ liệu state of the art để đánh giá mô hình dáng người cũng như là phân loại hành vi. Bộ dữ liệu có hơn 25.000 ảnh và hơn 40.000 người được đánh các khớp nối trên cơ thể. Những hình ảnh này được thu thập mỗi ngày với hơn 410 hành vi khác nhau.

Chúng ta có thể vào trang chủ các dataset để xem một số mẫu. Các hình có viền đỏ là các hình dùng cho việc train, còn lại sẽ được dùng cho việc test.



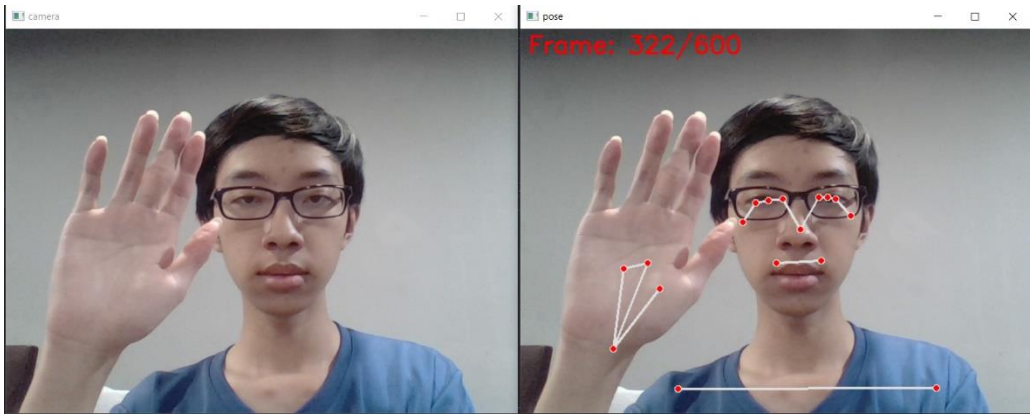
Hình 33. Dataset MPII Human Pose

7.2 Thực nghiệm

Bởi vì giới hạn về mặt thời gian, nhóm sinh viên đã tự xây dựng một thí nghiệm nhỏ cho bài toán phân loại hành vi bằng Pose Estimation. Đây chỉ mới là hai pha cuối cùng trong mô hình được đề xuất. Do đó, thí nghiệm chỉ có thể hoạt động trên một đối tượng. Khi đưa ra thực tế, cả 4 pha sẽ giúp cho mô hình hoạt động được trên nhiều đối tượng.

Đối với bài toán Pose Estimation, nhóm chọn thư viện MediaPipe được cung cấp bởi Google. Đây là một thư viện được cài đặt với mô hình BlazePose và GHUM đã được đề cập bên trên. Lý do thư viện này được nhóm chọn là nó có tốc độ xử lý rất nhanh để có thể chạy được trong thời gian thực.

Đầu tiên, chúng tôi thiết kế một chương trình giúp chúng ta sinh ra bộ dữ liệu để train và validation mô hình. Ở chương trình này, chúng ta sẽ ghi nhận 600 frames ảnh liên tục cho từng hành vi và lưu lại các landmarks của Pose Estimation thành một file csv. Ví dụ bên dưới chúng ta đang sinh data cho class “Swing hand” tại frame thứ 322.



Hình 34. Quá trình huấn luyện

Ở đây, nhóm thử nghiệm trên 4 lớp khác nhau bao gồm:

- Bình thường (Idle)
- Vẫy tay (Swing_hand)
- Che mắt (Eye_cover)

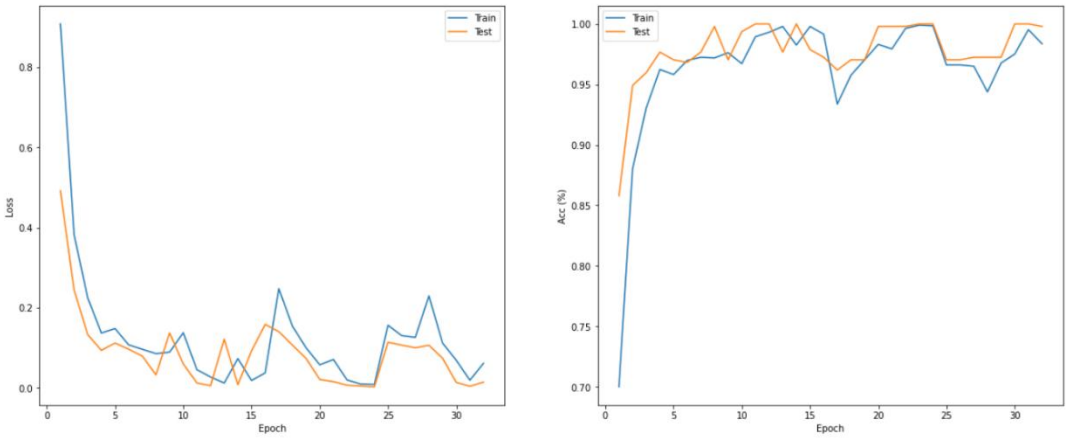
- Tay để hình chữ W (W_hand)

Sau đó, nhóm huấn luyện mô hình LSTM bằng cách trích 10 frame liên tục nhau làm 1 data point. Mô hình LSTM nhóm thử nghiệm là một mô hình đơn giản chứa 4 layers LSTM có 50 units. Ngoài ra, để chống hiện tượng quá khớp (overfitting), các lớp Dropout cũng được bổ sung vào sau từng lớp LSTM. Cuối cùng, mô hình kết thúc bằng một fully connected layer có số unit tương ứng với số class. Hiển nhiên, để đầu ra là một phân phối xác suất (distributed probability) thì hàm kích hoạt sẽ là softmax.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 10, 50)	36600
dropout (Dropout)	(None, 10, 50)	0
lstm_1 (LSTM)	(None, 10, 50)	20200
dropout_1 (Dropout)	(None, 10, 50)	0
lstm_2 (LSTM)	(None, 10, 50)	20200
dropout_2 (Dropout)	(None, 10, 50)	0
lstm_3 (LSTM)	(None, 50)	20200
dropout_3 (Dropout)	(None, 50)	0
dense (Dense)	(None, 4)	204

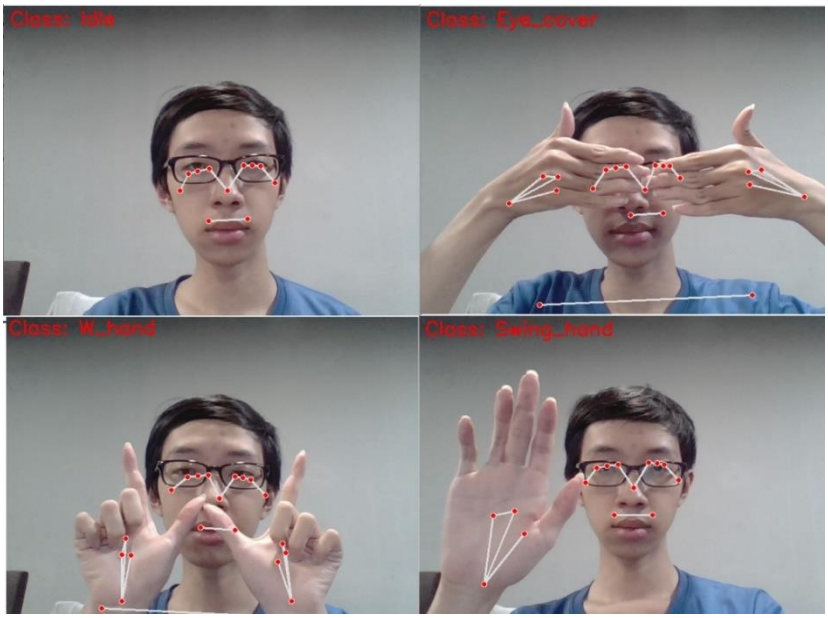
Hình 35. Mô-đun phân lớp bằng LSTM

Mô hình trên được nhóm huấn luyện với 32 epochs và chia dữ liệu thành 64 batchs để tối ưu bằng phương pháp Adam. Và kết quả cho thấy rằng mô hình này học rất tốt, chỉ khoảng với 5 epochs đầu thì mô hình đã đạt được hiệu quả rất cao.



Hình 36. Kết quả huấn luyện

Cuối cùng, nhóm thực hiện chạy inference trên mô hình vừa train:



Hình 37. Chạy inference real-time

7.3 Đánh giá

Nhìn chung, tốc độ lẫn kết quả của mô hình khá tốt. Với mô hình cho single object, chúng ta hoàn toàn có thể chạy với thời gian thực. Đồng thời, việc áp dụng mô hình YOLO vào cũng sẽ không ảnh hưởng đến thời gian vì YOLO vốn được sinh ra để chạy trên các ứng dụng thời gian thực.

Mặt khác, một số vấn đề tồn đọng cần được phân tích và giải quyết như là việc sử dụng dáng người sẽ có thể hạn chế một số hành vi. Điển hình là hành vi giơ hai ngón tay và vẫy tay đều nhận là một vì các điểm landmark không thay đổi. Để khắc phục nhược điểm này, nhóm sẽ đề xuất một giải pháp cải tiến hơn ở mục sau.

8 Kết luận và hướng phát triển

8.1 Kết luận

Bài báo cáo nghiên cứu về các mạng học sâu như CNN, RNN và LSTM để dùng cho các bài toán phát hiện và khoanh vùng các đối tượng có hành vi lạ ở các camera CCTV.

Việc sử dụng kết hợp các mạng CNN và LSTM để làm các nhiệm vụ phát hiện, truy vết và ước lượng dáng người đều giúp việc phân loại các hành vi mang lại hiệu quả cao. Qua đó, có thể thấy tính khả thi khi sử dụng mô hình đề xuất trong vấn đề đã đặt ra.

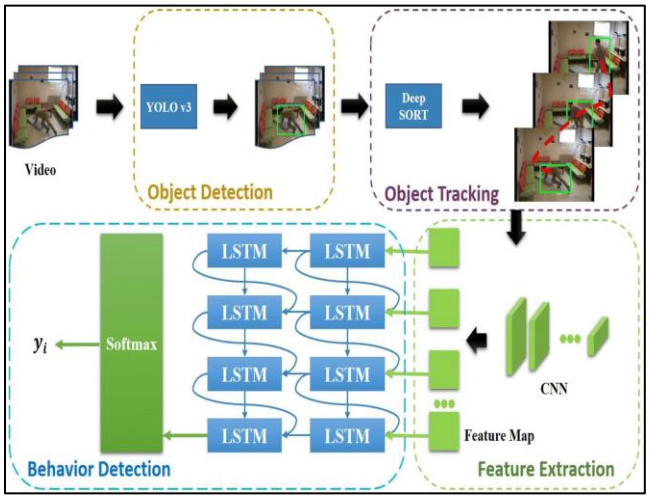
8.2 Hướng phát triển

Học viên sẽ tiếp tục nghiên cứu và phát triển mô hình đề xuất để có cái thiện độ chính xác và thời gian thực thi, phân tích nội dung.

Cài đặt trên các nền tảng khác nhau và tiến hành kiểm thử trên môi trường thực tế tại các camera CCTV để có thể đánh giá mô hình đề xuất một cách khách quan.

Phát triển bài toán trên nhiều tập dữ liệu khác nhau, trên nhiều môi trường khác nhau. Ngoài phân loại các hành vi bất thường tại các chung cư, tòa nhà,… có thể đa dạng hóa các môi trường bằng cách thực hiện tại những nơi có lắp đặt camera để tăng cường độ bảo mật, năng cao hiệu suất và giảm nguồn lực cho các đội giám sát an ninh tại các khu vực.

Ngoài ra, thay vì chỉ sử dụng dáng người bằng Pose Estimation, nhóm sinh viên cũng sẽ thử nghiệm một số mô hình CNN như MobileNet để rút trích đặc trưng ảnh real-time, từ đó cải tiến mô hình.



Hình 38. Mô hình đề xuất trong hướng phát triển

9 Tài liệu tham khảo

[1] Oluwatoyin P. Popoola, and Kejun Wang. Video-Based Abnormal Human Behavior Recognition.

[2] T. Wang, Q. Li, Y. Liu , Y. Zhou. Abnormal human body behavior recognition using pose estimation

[3] Sandersan Onie et al., The Use of Closed-Circuit Television and Video in Suicide Prevention: Narrative Review and Future Directions, 7th May 2021

[4] Julak Lee et al., Application of sensor network system to prevent suicide from the bridge, Nov 2016

[5] G. Spathoulas et al., Detection of abnormal behavior in smart-home environments, IEEE, 21st Nov 2019

[6] Cem Direkoglu et al., Abnormal crowd behavior detection using novel optical flow-based features, IEEE, 23rd Oct 2017

[7] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, Francisco Herrera. Deep learning in video multi-object tracking: A survey

[8] Damla Arifoglu, Abdelhamid Bouchachia. Activity Recognition and Abnormal Behaviour Detection with Recurrent Neural Networks.

[9] Chuan-Wang Chang, Chuan-Yu Chang & You-Ying Lin. A hybrid CNN and LSTM-based deep learning model for abnormal behavior detection.

[10] Nicolai Wojke, Alex Bewley, Dietrich Paulus. Deep SORT - Simple Online and Realtime Tracking with a Deep Association Metric (2017)

[11] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating LSTM: 3D Pose Estimation based on Joint Interdependency.

[12] Valentin Bazarevsky et al., BlazePose: On-device Real-time Body Pose tracking, 17 June

2020
[13] Hongyi Xu et al., GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models, CVPR 2020