HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN QUỐC ĐẠT

KỸ THUẬT HỌC SÂU CHO BÀI TOÁN THEO VẾT ĐA ĐỐI TƯỢNG

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

TP.HỒ CHÍ MINH - 2021

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN QUỐC ĐẠT

KỸ THUẬT HỌC SÂU CHO BÀI TOÁN THEO VẾT ĐA ĐỐI TƯỢNG

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

PGS.TS LÊ HOÀNG THÁI

TP. HỒ CHÍ MINH - 2021

LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: "***Kỹ thuật học sâu cho bài toán theo vết đa đối tượng***" là công trình nghiên cứu của chính tôi.

Những kết quả nghiên cứu được trình bày trong luận văn là công trình của riêng của tôi dưới sự hướng dẫn của **PGS.TS Lê Hoàng Thái**.

Tôi cam đoan các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong luận văn này mà không được trích dẫn theo đúng quy định.

TP. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Trần Quốc Đạt

LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện luận văn, ngoài nỗ lực của bản thân, tôi đã nhận được sự hướng dẫn nhiệt tình quý báu của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới:

Tôi xin chân thành cảm ơn Ban Giám hiệu, quý Thầy Cô Khoa Đào tạo sau đại học của Học viện Công nghệ Bưu chính Viễn thông đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành luận văn.

Tôi cũng xin chân thành cảm ơn Thầy **PGS.TS Lê Hoàng Thái**, người thầy kính mến đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành luận văn.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn.

Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

DANH SÁCH HÌNH ẢNH

DANH SÁCH BẢNG

MỤC LỤC

I. MỞ ĐẦU

1. Lý do chọn đề tài

Trong những năm gần đây, việc phát hiện và tái xác định đối tượng đã có nhiều tiến bộ đáng kể. Hai kỹ thuật này là thành phần cốt lõi để hình thành hệ thống theo dõi đa đối tượng. Tuy nhiên, việc hoàn thành hai nhiệm vụ trong một mạng duy nhất để cải thiện tốc độ suy luận chưa được quan tâm nhiều. Các nỗ lực ban đầu cho việc hợp nhất hai nhiệm vụ trên cho kết quả thấp. Nguyên nhân chủ yếu: là do kỹ thuật tái nhận dạng chưa được huấn luyện phù hợp. Trong luận văn, chúng tôi tìm hiểu những lý do cơ bản đằng sau sự thất bại; tiến tới, đề nghị một phương pháp cơ bản đơn giản để giải quyết các vấn đề.

Mục tiêu của hệ thống đề xuất là: dự đoán đường đi của nhiều vật thể được chú ý trong các video. Nhiều ứng dụng của hệ thống đề nghị này sẽ rất hữu ích trong nhiều lĩnh vực thực tế khác nhau:

- Dự đoán hành động.
- Phân tích các video thể thao,
- Robot trợ giúp người già.
- Tương tác giữa người và máy tính….

2.Tổng quan về vấn đề nghiên cứu

Theo vết đa đối tượng (Multi-Object Tracking (MOT)) là một trong những bài toán kinh điển thuộc lĩnh vực thị giác máy tính.

Các phương pháp trước đây thường chia bài toán này thành hai model riêng biệt: model (1) Bộ dò tìm(detection): đầu tiên sẽ định vị và khoanh vùng vật thể cần chú ý tới bằng bounding box trong tập các ảnh, sau đó sang model (2), Bộ kết hợp (association) sẽ tạo ra các đặc trưng tái định danh (Re-identification (Re-ID)) cho mỗi bounding box và kết nối nó tới một trong những tuyến đường (tạo ra bởi vật thể) đã được xác định bởi các đặc trưng trước đó. Trong các năm gần đây, các kỹ thuật trên đã có những bước tiến đáng kinh ngạc về độ chính xác cũng như tốc độ. Tuy nhiên, khi kết hợp hai model thì lại không thể dùng ở các video có độ phân giải cao (30FPS), do tốc độ thực thi không đảm bảo, bởi vì các network đó không chia sẻ cùng một bộ đặc trưng (Tức là muốn dùng

được đặc trưng của (1)detection thì (2) Association phải qua một bước biến đổi nào đó - two-steps).

Với sự phát triển của học đa nhiệm (multi-task learning), phương pháp one-shot cho việc kết hợp (1) phát hiện vật thể và (2) học các đặc trưng Re-ID được chú ý đến nhiều hơn. Do phần lớn các đặc trưng có thể được chia sẻ giữa hai model nên phương pháp này có khả năng làm giảm thời kết hợp (inteference time) hai model. Tuy nhiên, độ chính xác (accuracy) của phương pháp one-shot hiện tại giảm đi rõ rệt, khi so sánh với phương pháp two-steps, dựa vào các thực nghiệm, thì rõ ràng việc kết hợp hai model này không thể thực hiện một cách đơn giản được, mà phải chú ý một cách cẩn thận.

Thay vì, sử dụng các trick trong máy học và học sâu để tăng độ chính xác thì chúng ta sẽ nghiên cứu lý do quan trọng cho thất bại này. Sau đây, sẽ là 3 nhân tố quan trọng nhất ảnh hưởng đến accuray:

- Anchors don’t fit Re-ID [6](tập đặc trưng của bộ dò tìm không khớp với tập đặc trưng tái định danh)
- Multi-Layer Feature Aggregation [8] (Tích hợp các đặc trưng qua nhiều lớp)
- Dimensionality of the RelD Features [2,9] (Kích thước của các đặc trưng Re-ID).

3. Mục đích nghiên cứu

Xây dựng một mô hình nhận dạng theo vết nhiều đối tượng (người) để tiến tới xa hơn có thể áp dụng mô hình cho một số lĩnh vực thực tế như: an ninh quốc phòng, giao thông vận tải,…

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Nhận dạng theo vết nhiều đối tượng (người) quan tâm trong video ở tốc độ 30 khung hình mỗi giây.

Phạm vi nghiên cứu: thực hiện trên tập dữ liệu video FairMOT [9] và một số tập dữ liệu video chọn lọc từ youtube khác. Các video dữ liệu chứa rất nhiều đối tượng được quay ở nhiều vị trí khung cảnh khác nhau (trên đường phố hoặc trong siêu thị,…)

5. Phương pháp nghiên cứu

- Phương pháp chuyên gia:

Tổng hợp các kiến thức đã biết về các mô hình học sâu – cụ thể là mạng xương sống (Backbone Network), Nhánh phát hiện đối tượng(Object Detection Branch) , Nhánh nhúng danh tính (Identity Embedding Branch), Dò tìm trực tuyến (Online Tracking) [8].

- Phương Pháp Thực Nghiệm:

Thực nghiệm trên tập dữ liệu video FairMOT [9] và một số tập dữ liệu video chọn lọc từ youtube khác và bộ dữ liệu tự xây dựng để tìm ra một mô hình cho độ chính xác (accuracy) cao và tốc độ chạy thời gian thực khi nhận dạng và theo vết nhiều đối tượng.

- Phương Pháp Tổng Kết Kinh Nghiệm:

Nghiên cứu và xem xét lại những thành quả thực tiễn đã có của các tập dữ liệu video đã thực hiện để rút ra kết luận giúp xây dựng mô hình vừa dò tìm và theo vết nhiều đối tượng đảm bảo đạt hiệu xuất cao và tốc độ nhanh.

II. NỘI DUNG

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1 Các phương pháp dò tìm đối tượng

Multi-Object Tracking (MOT) là một trong những mục tiêu lâu dài của thị giác máy tính [1] [2] [3] [4]. Mục tiêu là dự đoán đường đi của nhiều vật thể được chú ý trong các video. Các ứng dụng của nó sẽ có lợi ích cho rất nhiều ứng dụng khác như: dự đoán hành động, phân tích các video thể thao, robot trợ giúp người già và sự tương tác giữa người và máy tính. Các phương pháp state-of-the-art [1] [2] [3] [4] [5] [6] [7] cũ thường chia bài toán này thành hai model riêng biệt: detection đầu tiên sẽ định vị và khoanh vùng vật thể cần chú ý tới bằng bounding box trong tập các ảnh, sau đó sẽ đến association sẽ chiết xuất ra các đặc trưng Re-identification (Re-ID) cho mỗi bounding box và kết nối nó tới một trong những tuyến đường (tạo ra bởi vật thể) đã được xác định bởi các đặc trưng trước đó. Các model trên đã có những bước tiến đáng kinh ngạc khi tăng độ chính xác và tốc độ trong các năm gần đây. Tuy nhiên, khi kết hợp chúng thì lại không thể đủ tốc độ khi dùng ở 30FPS của video bởi vì các network đó không chia sẻ cùng một bộ đặc trưng (tức là muốn dùng được đặc trưng của detection thì Association phải qua một bước biến đổi nào đó – two-steps).

Với sự phát triển của việc học tập đa tác vụ [8], phương pháp one-shot để kết hợp phát hiện vật thể và các đặc trưng Re-ID được chú ý đến nhiều hơn [9] [10]. Do phần lớn các đặc trưng có thể được chia sẻ giữa hai model nên phương pháp này có khả năng làm giảm thời kết hợp (inteference time) hai model. Tuy nhiên sự chính xác (accuracy) của phương pháp one-shot hiện tại lại giảm đi rõ rệt khi so sánh với phương pháp two-steps, dựa vào cả thực nghiệm thì rõ ràng việc kết hợp hai model này không thể thực hiện một cách đơn giản được, mà phải chú ý một cách cẩn thận.

Thay vì sử dụng các trick trong máy học và học sâu để tăng độ chính xác thì chúng ta sẽ nghiên cứu lý do chính xác cho sự thất bại đó. Sau đây sẽ là 3 nhân tố quan trọng nhất ảnh hưởng đến độ chính xác:

- Anchors don’t fit Re-ID (Neo không phù hợp với Re-ID)
- Multi-Layer Feature Aggregation (Tổng hợp đặc trưng trên nhiều lớp)
- Dimensionality of the ReID Features (Kích thước của các đặc trưng Re-ID)

1.2 Phân tích vấn đề

Neo không phù hợp với Re-ID

Hiện tại thì với cách theo dõi one-shot [9] [10] đều dựa theo neo (anchor) vì chúng đều được thay đổi từ phát hiện vật thể, tuy nhiên các cái neo vật thể đó không phù hợp cho đặc trưng Re-ID với 2 lý do: Thứ nhất, khi mà có nhiều neo dựa trên các image patches, chúng có thể dự đoán chung một định danh cho cùng 1 vật thể (Bounding box trùng lên nhau). Việc này sẽ gây lên sự nhập nhằng cho mạng. Thứ hai, bản đồ đặc trưng thường được giảm độ lấy mẫu (down-sample) 8 lần để có thể điều hòa giữa tốc độ và độ chính xác cho việc nhận diện vật thể nhưng lại rất là thô cho RE-ID vì object center có thể không được căn chỉnh tốt với vị trí của neo do đó có thể làm sai khi dự đoán định danh của vật thể. Để xử lý vấn đề này chúng tôi dự đoán pixel-wise keypoint (object center) và định danh vật thể ở trên cùng của bản đồ đặc trưng high-resolution.

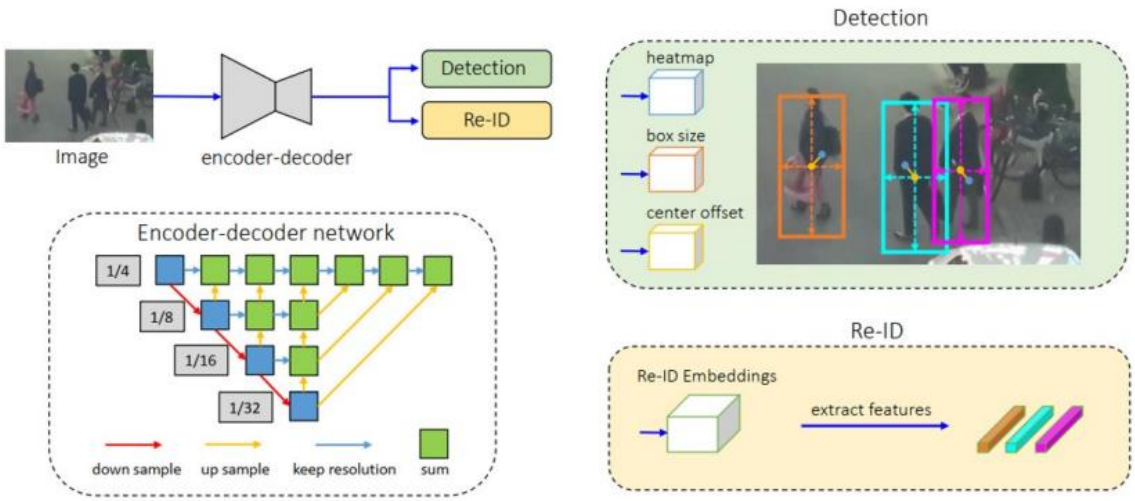
Tổng hợp đặc trưng trên nhiều lớp

Việc này quan trọng với MOT vì các đặc trưng Re-ID cần tận dụng cả các đặc trưng cấp thấp và cấp cao để thích nghi với vật thể khi bị phóng to và thu nhỏ. Trong thực nghiệm chúng ta thấy việc này rất có ích để giảm indentity switches cho phương pháp one-shot vì nó là kỹ năng để xử lý sự thay đổi tỷ lệ của vật thể. (Chú ý điều này sẽ không tác dụng mấy tới phương pháp two-steps do vật thể sẽ có cùng một tỷ lệ khi đã có bước cắt và thay đổi kích thước).

Kích thước của các đặc trưng Re-ID

Các phương pháp cũ dùng các đặc trưng Re-ID có kích thước lớn nhưng ở phương pháp này chúng ta sẽ tìm cách giảm kích thước của các đặc trưng Re-ID là do ảnh để huấn luyện cho MOT ít hơn ảnh để huấn luyện Re-ID, và cũng không thể dùng ảnh huấn luyện của Re-ID được vì bộ dữ liệu đó chỉ đưa ra các ảnh hình người bị cắt ra. Việc học các đặc trưng có kích thước nhỏ cũng giúp vượt qua được các mối nguy từ việc over-fitting khi học trên các tập dữ liệu nhỏ, và tăng tốc độ cho việc theo dõi vật thể.

1.3 Giải pháp



Hình 1.1: Tổng quát one-shot MOT. Ảnh đầu vào sẽ được cho vào mạng encoder-decoder để tạo ra bản đồ đặc trưng độ phân giải cao (stride = 4). Sau đó sẽ đưa vào hai đầu song song để dự đoán đặc trưng bounding box và Re-ID

Hình 1. 1

Ở đây chúng tôi sẽ giới thiệu một cách giải quyết cho các vấn đề ở chương 2. Một cách tổng quát, chúng tôi dùng kỹ thuật anchor-free (không neo) để dự đoán tâm vật thể ở trên bản đồ đặc trưng độ phân giải cao (high-resolution feature map), khi dùng kỹ thuật này chúng ta sẽ vượt qua được vấn đề nhập nhằng, từ đó mà các đặc trưng Re-ID sẽ căn chỉnh về đúng tâm của vật thể hơn. Sau đó chúng tôi sẽ thêm vào nhánh song song để dự đoán đặc trưng pixel-wise Re-ID (định danh vật thể). Ở mạng xương sống (backbone network) chúng tôi kết hợp với kỹ thuật Deep Layer Aggregation để có thể xử lý các vật thể trên các tỷ lệ khác nhau.

1.3.1 Giới thiệu hướng tiếp cận mới

Vấn đề của các mạng object detection thành công nhất hiện nay là chúng phải thực hiện lần qua tất cả các vị trí có thể có vật và thực hiện phân loại mỗi vị trí đó. Điều đó dẫn đến việc lãng phí tài nguyên tính toán, không hiệu quả và cần thực hiện các bước hậu xử lý (Non-maximum suppression).

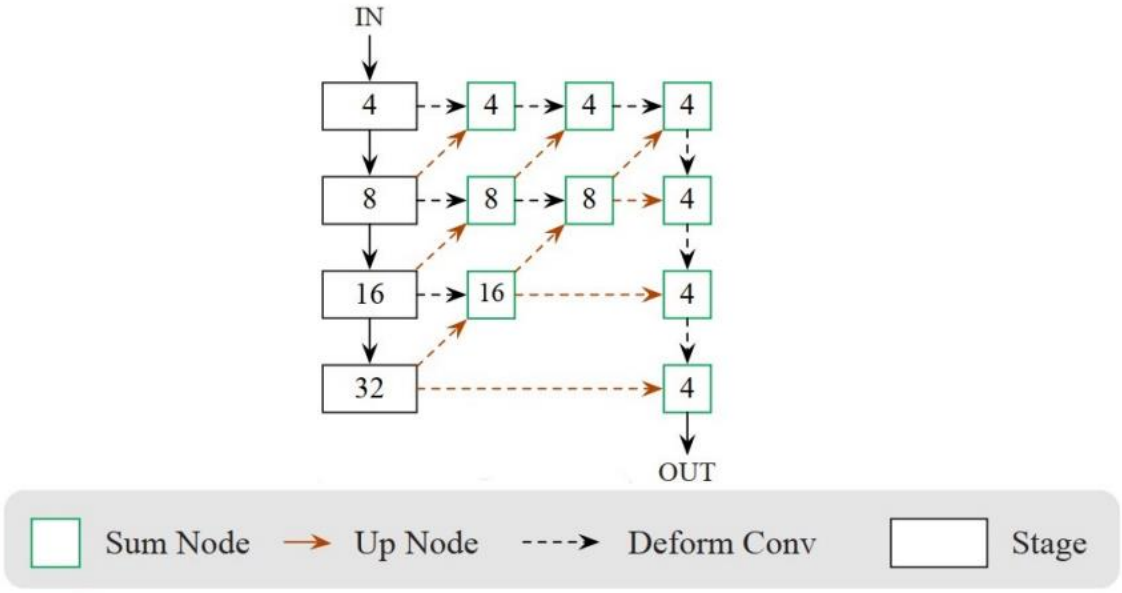
Hướng tiếp cận mới của luận văn là đưa bài toán phát hiện vật (object detection) về bài toán tìm điểm đặc trưng (keypoint estimation), từ đó cũng suy ra kích thước và tính toán được bounding box cho bài toán phát hiện vật.

Nó vượt qua các thuật toán 1 stage (One-shot MOT methods) phổ biến nhất hiện nay là YOLO v3, RetinaNet trong sự cân bằng giữa tốc độ và độ chính xác. Hơn nữa độ chính xác của nó còn ngang ngửa Faster RCNN - một mạng phát hiện vật 2 stage (Two-Step MOT methods).

- One-shot MOT methods: YOLO v3, RetinaNet, CenterNet...
- Two-Step MOT methods: RCNN, Fast-RCNN, Masked-RCNN,...

1.3.2 Mạng xương sống (Backbone Network)

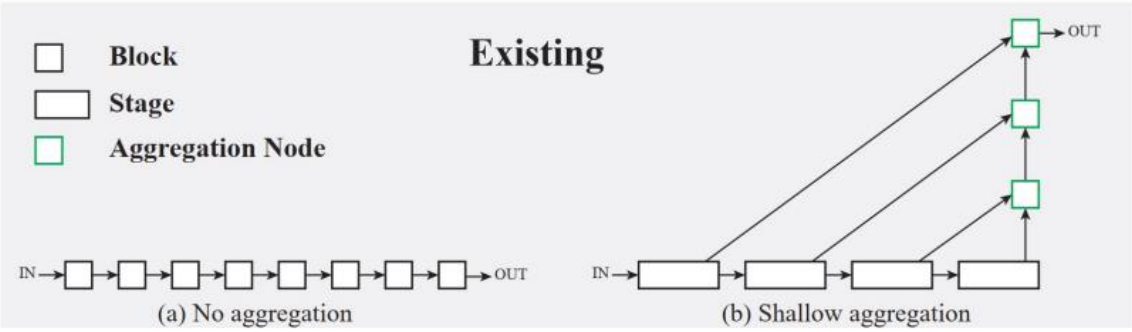
Luận văn chọn mạng Resnet-34 [11] làm mạng xương sống để có thể cân bằng giữa tốc độ và độ chính xác. Để vật thể thích nghi được với nhiều tỷ lệ khác nhau một biến thể của **Deep Layer Aggregation (DLA)** [12], sự khác biệt ở DLA này là nó có nhiều liên kết nhảy hơn giữa đặc trưng low-level và high-level, tương tự như **Feature Pyramid Network (FPN)** [13]. Ngoài ra tất cả các lớp tích chập up-sampling được thay thế bởi **deformable convolution layers** để chúng có thể linh hoạt trong việc thích nghi với dáng người và thay đổi tỷ lệ. Những thay đổi trên cũng rất có ích để làm giảm thiểu tác động của aligment issues. Kết quả ta đặt tên mạng là DLA-34, ảnh đầu vào có kích thước $H_{image} \times W_{image}$ thì bản đồ đặc trưng có kích thước $C \times H \times W$ là với $H = H_{image}/4$ và $W = W_{image}/4$.



Hình 1.2: Chi tiết mạng xương sống DLA 34

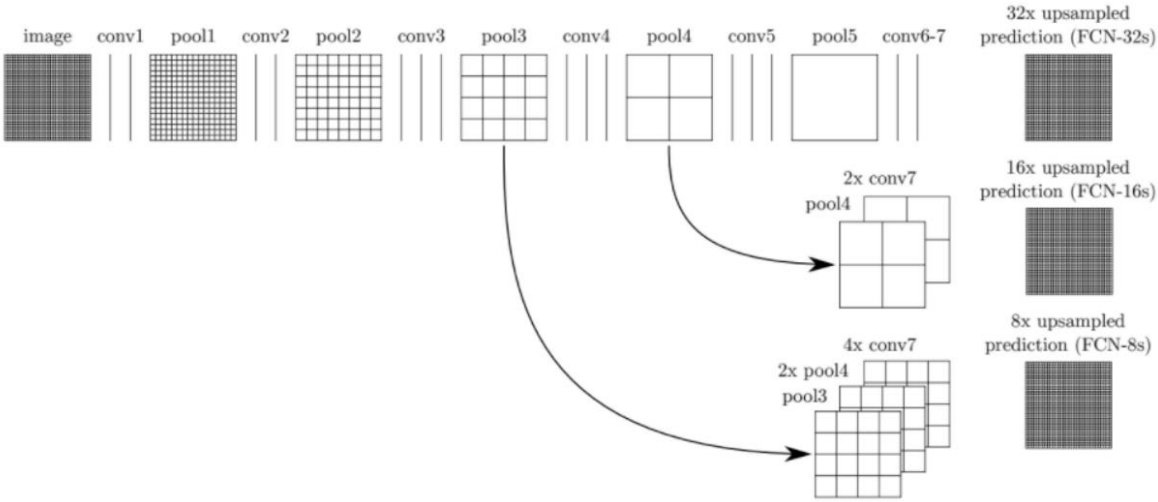
Deep Layer Aggregation

Deep Layer Aggregation bao gồm hai loại là: Iterative Deep Aggregation(IDA) và Hierarchical Deep Aggregation(HDA). Phần lớn các kết nối nhảy bước hiện tại vẫn khá là nông ví dụ như ResNet. IDA và HDA ra đời để phục vụ cho việc nhảy kết nối này có thể sâu hơn.



Hình 1.3: (a) là mạng CNN cơ bản như VGG (b) là mô tả kết nối nông như của Feature Pyramid

Tầng nhảy kết nối là gì?



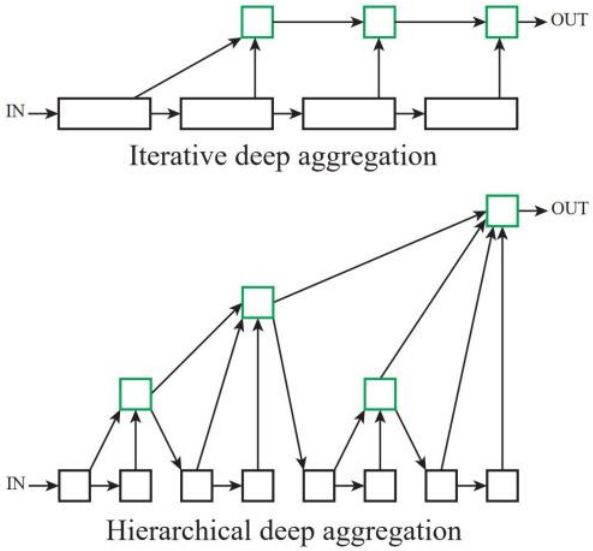
Hình 1.4: Fully Convolutional Networks for Semantic Segmentation

Nhảy kết nối (Skip connection) có nghĩa là phép ghép lại, ví dụ như hình 5 mô tả cho FCN thì nhảy kết nối từ "pool 4" đã nhảy qua pool 5 và 6 để kết hợp với "pool 7".

Tại sao nhảy kết nối quan trọng

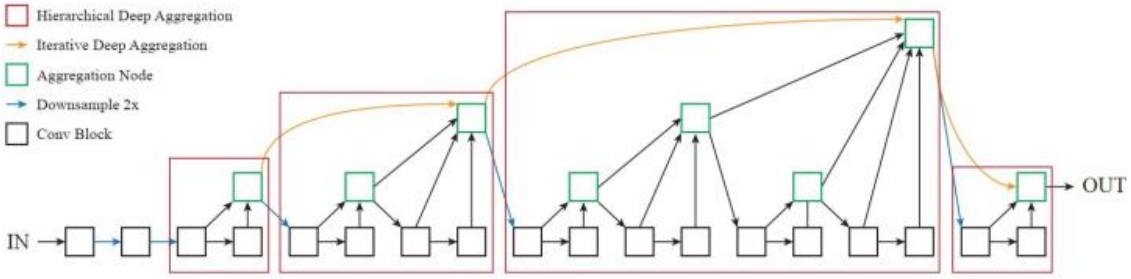
1. Kết hợp các đặc trưng cấp thấp với các đặc trưng cấp cao lại với nhau.
2. Muốn huấn luyện các mạng sâu hơn, thì ví dụ như các kết nối ngắn như ResNet có thể giúp tránh tình trạng vanishing gradient với mạng rất sâu.
3. Các nhảy kết nối dài có thể giúp phục hồi các thông tin đã bị mất khi downsampling. (Fully Convolutional Networks for Semantic Segmentation).

4. Tăng tốc độ hội tụ (Huấn luyện mạng). The Importance of Skip Connections in Biomedical Image Segmentation.



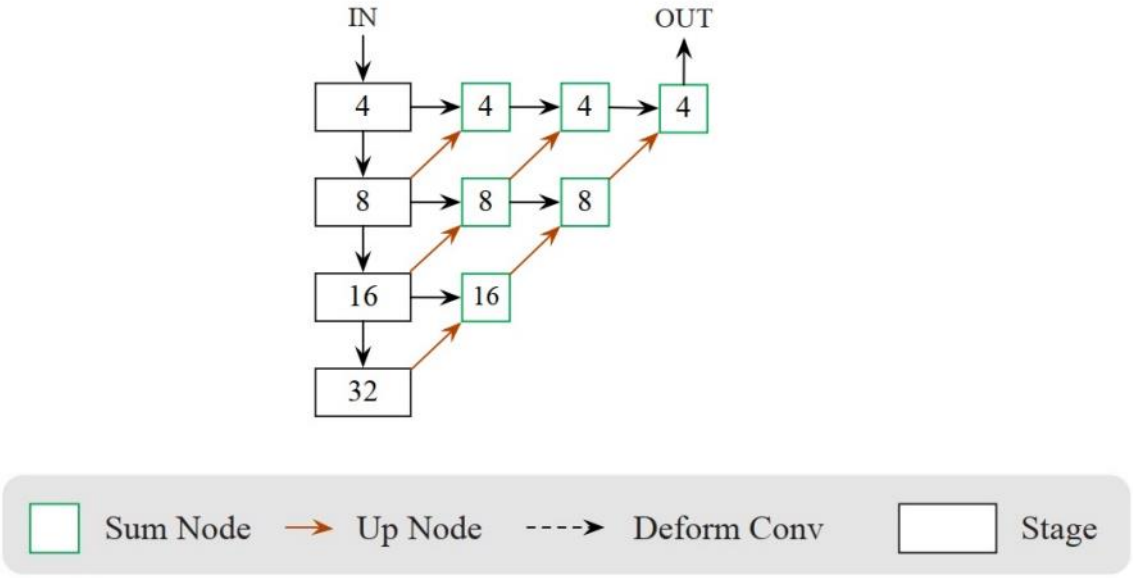
Hình 1.5: IDA hoặc HDA

IDA - Iterative Deep Aggregation tập trung vào giải quyết độ nét (resolution) và tỷ lệ (scale). **HDA - Hierarchical Deep Aggregation** tập trung vào việc kết hợp các đặc trưng cho toàn bộ các module và channel. Từ IDA và HDA chúng ta kết hợp lại thì đầu ra của mạng sẽ có cả ngữ nghĩa ở lớp cao và các thông tin không gian khác ở các lớp thấp.



Hình 1.6: Mạng kết hợp IDA và HDA

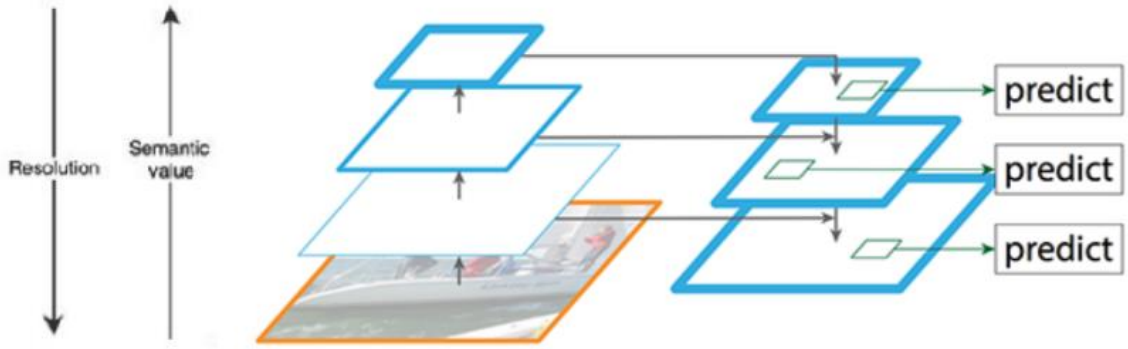
DLA-34 gốc tương đương với hình 8 sau:



Hình 1.7: DLA-34 gốc

Feature Pyramid Network

Dò tìm các đối tượng có kích thước nhỏ là một vấn đề đáng được giải quyết để nâng cao độ chính xác. Và FPN là mô hình mạng được thiết kế ra dựa trên khái niệm pyramid để giải quyết vấn đề này.



Hình 1.8: Feature Pyramid Network

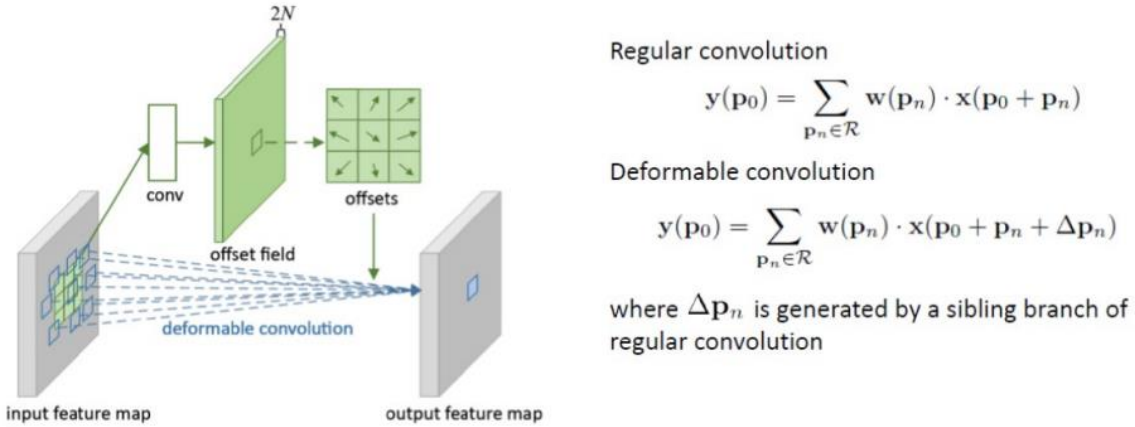
Mô hình FPN kết hợp thông tin của mô hình theo hướng bottom-up kết hợp với top-down để dò tìm đối tượng (trong khi đó, các thuật toán khác chỉ thường sử dụng bottom-up). Khi chúng ta ở bottom và đi lên (up), độ phân giải sẽ giảm, nhưng giá trị ngữ nghĩa sẽ tăng lên. Trong khi đó, FPN xây dựng thêm mô hình top-down, nhằm mục đích xây dựng các layer có độ phân giải cao từ các layer có ngữ nghĩa cao. Trong quá trình xây dựng lại các layer từ top xuống bottom, chúng ta sẽ gặp một vấn đề khá nghiêm trọng là bị mất mát thông tin của các đối tượng. Ví dụ một đối tượng nhỏ khi lên top sẽ không

thấy nó, và từ top đi ngược lại sẽ không thể tái tạo lại đối tượng nhỏ đó. Để giải quyết vấn đề này, chúng ta sẽ tạo các kết nối (skip connection) giữa các reconstruction layter và các feature map để giúp quá trình detector dự đoán các vị trí của đối tượng thực hiện tốt hơn (hạn chế tốt nhất việc mất mát thông tin).

Deformable Convolution Layers

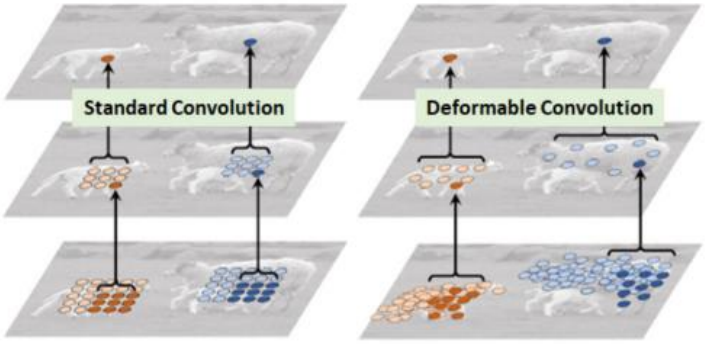
Với các cách tích chập thông thường sẽ tính toán trên một lưới ô vuông định trước cho ảnh đầu vào hoặc tập hợp các bản đồ đặc trưng dựa theo độ lớn của bộ lọc ()filter). Lưới này có thể là 3 × 3 hoặc 5 × 5 v.v. Tuy nhiên, có các vật thể chúng ta cần phát hiện và định danh có thể bị biến dạng, mắc kẹt (trùng với vật thể khác) hoặc thay đổi theo tỷ lệ, ví dụ trong bài toán này là khi theo dõi vật thể là con người với một camera, thì vật thể khi ở xa camera sẽ bị nhỏ lại, ở gần sẽ phóng lớn lên, hay như vật thể có thể bị che khuất bởi cây cối, cột đèn v.v.

Ở DCN, lưới này có thể biến dạng, có nghĩa là mỗi điểm lưới có thể di chuyển bởi môt độ lệch có thể học được. Và tích chập sẽ hoạt động trên các điểm lưới di chuyển này, do đó được gọi là tích chập có thể biến dạng, tương tự đối với trường hợp tổng hợp RoI (Region of Interest) có thể biến dạng. Bằng cách sử dụng hai mô-đun mới này, DCN cải thiện độ chính xác của DeepLab, Faster R-CNN, R-FCN và FPN, v.v.

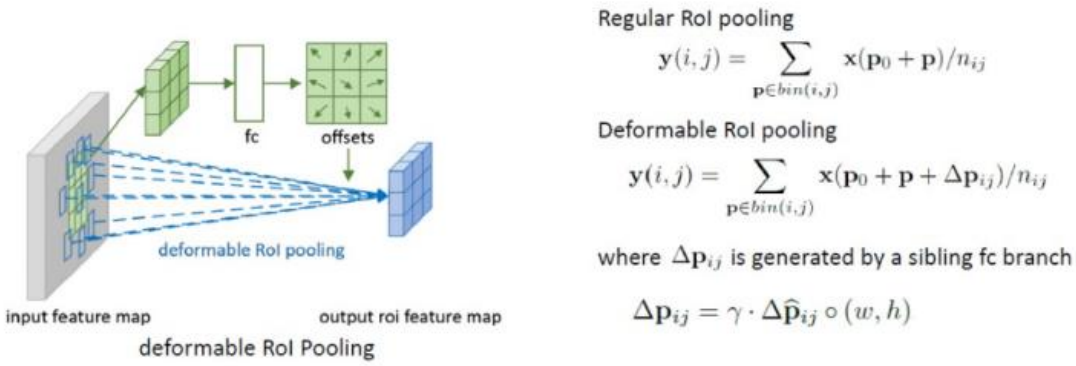


Hình 1.9: Deformable Convolution

- Tích chập thông thường sẽ hoạt động trên lưới vuông R.
- Tích chập biến dạng hoạt động trên R nhưng với mỗi điểm được thay đổi với một offset có thể học Δp_n.
- Tích chập dùng để tạo ra 2N số lượng bản đồ đặc trưng tương ứng với N điểm lệch 2D Δp_n (hướng x và hướng y cho mỗi offset).



Hình 1.10: Tích chập biến dạng có thể lấy các điểm có giá trị khác nhau tuỳ theo ảnh đầu vào, như ở hình này chúng tập trung vào hình ảnh của con vật thay vì phân tán như ở tích chập thường



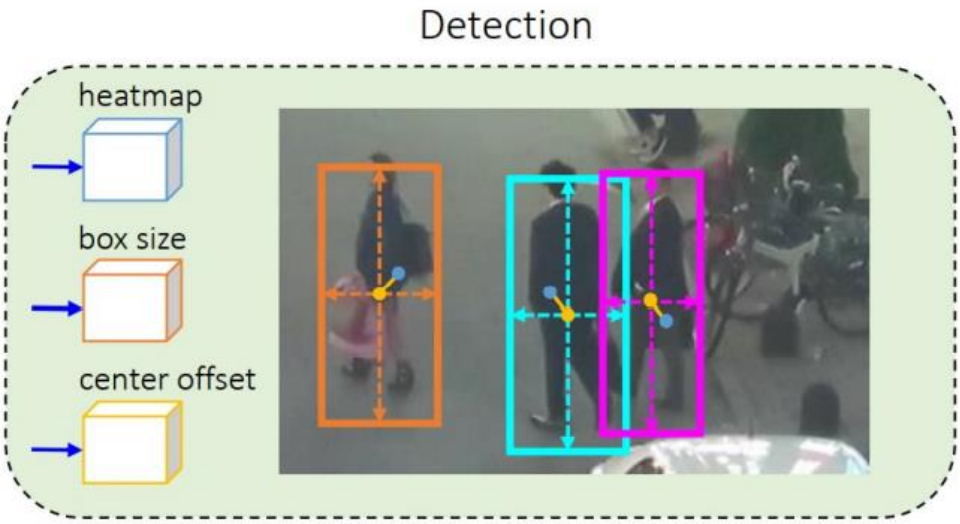
Hình 1.11: Deformable ROI

- ROI (Region of Interest) Thông thường sẽ chuyển hóa ảnh đầu vào là vùng ô vuông có kích thước cố định thành các đặc trưng có kích thước cố định.
- Ở ROI biến dạng ([Deformable ROI), đầu tiên ở đường trên hình 12 chúng ta vẫn gần ROI pooling thông thường để tạo ra bản đồ đặc trưng, sau đó một tầng kết nối đầy đủ sẽ tạo ra các offset được chuẩn hoá Δp^{ij} và rồi từ đó biến đổi bản đồ đặc trưng trên thành bản đồ đặc trưng với dựa theo offset đó ($\gamma = 0.1$).
- Việc chuẩn hoá offset Δp^{ij} là cần thiết để cho việc học kích thước của RoI bất biến.
- Cuối cùng, ở đường dưới hình 12, chúng ta sẽ biến đổi RoI pooling. bản đồ đặc trưng đầu ra sẽ được pool dựa theo offset đã học được ở trên.

1.3.3 Nhánh phát hiện vật thể

Phương pháp của luận văn này là coi việc phát hiện vật thể như center-based based bounding box regression task trên bản đồ đặc trưng có độ phân giải cao. Để làm việc này chúng tôi dùng 3 việc chạy song song được kết nối với đầu ra của mạng xương sống để

tính **heatmaps**, **object center offsets** và **bounding box sizes**, ở đây chúng tôi đang dùng kỹ thuật **Multitask Learning** . Trên mỗi việc song song đó chúng tôi áp dụng tích chập 3 × 3 (với 256 kênh) cho đặc trưng đầu ra của mạng xương sống và theo sau đó là tầng tích chập 1 × 1 để tạo ra đặc trưng cuối cùng.



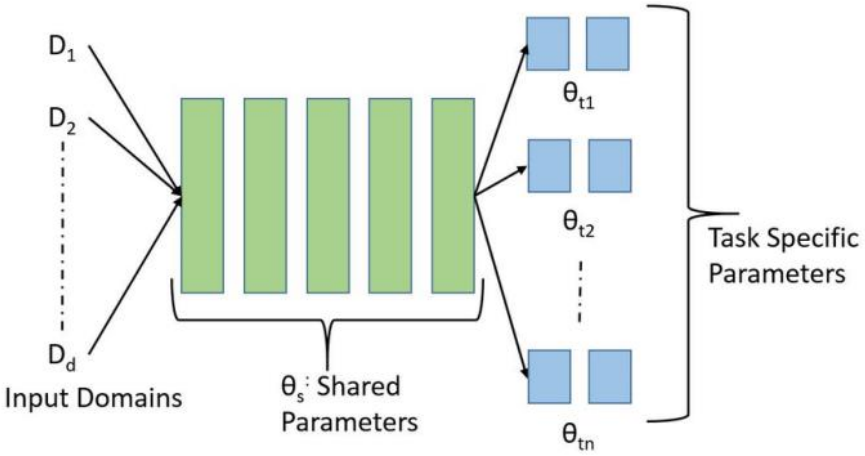
Multitask Learning - Multi Branch

Kiến trúc rẽ nhánh sẽ cho phép thuật toán học được nhiều tác vụ đồng thời nhưng không chia sẻ đặc trưng. Mô hình của chúng ta sử dụng chung một đầu vào là ảnh và phân nhánh thành nhiều mô hình con. Mỗi mô hình sẽ phụ trách dự báo cho một tác vụ một cách độc lập.

Ví dụ: Trong nhận diện khuôn mặt, chúng ta sẽ cần sử dụng rất nhiều các dự báo trên cùng một ảnh khuôn mặt như: giới tính, độ tuổi, chủng tộc, màu mắt, màu tóc,...

Những tác vụ trên không chia sẻ các đặc trưng để phân biệt. Ví dụ: Khi phân biệt giới tính chúng ta dựa trên các đặc trưng về độ dài tóc, râu, lông mày, mắt, cằm và quai hàm nhiều hơn nhưng phân biệt độ tuổi chúng ta chủ yếu dựa vào nếp nhăn trên khuôn mặt, màu da, màu tóc. Đây là những đặc trưng không hoàn toàn giống nhau. Do đó sử dụng kiến trúc multitask learning chia sẻ tham số cho bài toán này sẽ không hợp lý.

Một lựa chọn tốt hơn trong trường hợp này cho chúng ta đó là xây dựng một kiến trúc rẽ nhánh ngay từ input layer. Giữa các nhánh là độc lập, chỉ sử dụng chung một đầu vào mà không chia sẻ tham số.



Hình 1.12: Multi Branch - Kiến trúc rẽ nhánh

Heatmap Head

Đầu này chịu trách nhiệm ước tính vị trí của tâm vật thể. Biểu diễn của bản đồ nhiệt là tiêu chuẩn để thực hiện nhiệm vụ ước tính điểm tâm. Kích thước bản đồ nhiệt là 1 × H × W. Phản hồi tại vị trí của bản đồ nhiệt được mong đợi là trùng với vị trí của vật thể trong tập ground-truth. Phản hồi này sẽ giảm dần theo cấp số nhân tỉ lệ thuận khoảng cách giữa vị trí tâm vật thể và điểm cần tính.

Hình 1.13: Heatmap Flow

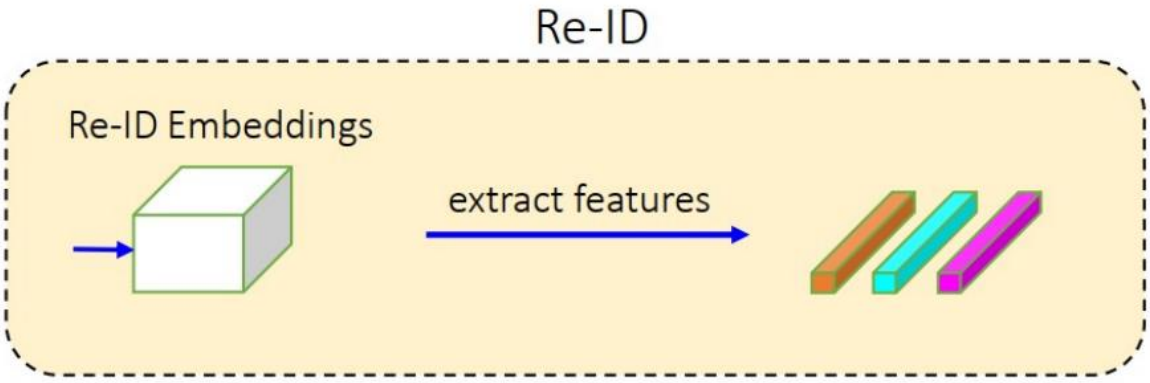
Center Offset Head

Đầu này chịu trách nhiệm khoanh vùng tâm vật thể đúng hơn. Nhớ lại rằng ở bước trước đó, bước trượt (stride) của bản đồ đặc trưng là 4 nên sẽ tạo ra một số lỗi khi nội suy vị trí của vật trên ảnh thật từ bản đồ đặc trưng. Đầu này không có lợi với việc phát hiện vật thể nhưng lại rất quan trọng với định danh vật thể Re-ID, khi mà đầu này sẽ xác định tâm vật thể tốt hơn.

Box Size Head

Đầu này chịu trách nhiệm ước tính chiều cao và chiều rộng của bounding box vật thể tại mỗi vị trí neo. Đầu này không liên quan trực tiếp đến các đặc trưng định danh Re-ID nhưng độ chính xác vị trí sẽ ảnh hưởng đến việc đánh giá hiệu suất phát hiện đối tượng. Đầu ra của size v trong **Box Size Head** là $S \in R^{W \times H \times 2}$.

1.3.4 Nhánh định danh vật thể



Hình 1.14: Nhánh định danh vật thể

Mục đích của nhánh định danh vật thể là tạo ra các đặc trưng để có thể phân biệt các vật khác nhau. Lý tưởng thì khoảng cách các vật thể khác nhau sẽ lớn hơn so với cùng một vật thể. Để có thể đạt được mục đích này chúng tôi sẽ áp dụng lớp tích chập với 128 Kernels trên đỉnh của đặc trưng mạng xương sống để trích xuất ra đặc trưng định danh cho mỗi điểm. Bản đồ đặc trưng sẽ là $E \in R^{128 \times W \times H}$. Đặc trưng Re-ID sẽ là $E_{x,y} \in R^{128}$ của vật thể tại điểm (x, y) sẽ được rút trích từ bản đồ đặc trưng trên.

1.4 Các kỹ thuật áp dụng

1.4.1 Hàm lỗi

Để huấn luyện bất kỳ mạng nào thì chúng ta đều cần phải định nghĩa hàm lỗi, huấn luyện mạng tương đương với việc tìm trọng số sao cho hàm lỗi là có giá trị nhỏ nhất, hay tương đương với việc độ lệch với tập huấn luyện là ít nhất.

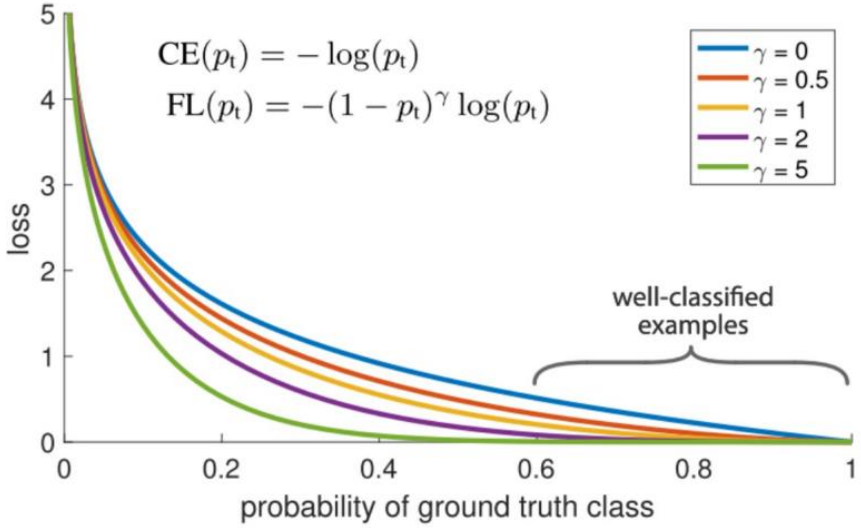
Kỹ thuật Focal Loss

Những mô hình nhận diện vật thể chính xác nhất đến nay được xây dựng dựa trên cách tiếp cận two-stage mà điển hình là R-CNN. Các mô hình này thường được dùng với một tập các object nằm khá thưa thớt và rải rác, trái ngược với phương pháp one-stage, thường được sử dụng cho các tập mẫu object có vị trí phân bố đồng đều và dày đặc. Mô hình sử dụng phương pháp one-stage thường nhanh và đơn giản hơn, tuy nhiên lại không

chính xác bằng two-stage. Lí do cho việc này là sự không cân bằng giữa các foreground và background class gặp phải trong quá trình huấn luyện. Trong phần này, chúng tôi sẽ trình bày một giải pháp để giải quyết vấn đề trên, đó chính là sử dụng Focal Loss.

Tổng quan: Focal loss được sử dụng bằng việc thay đổi một chút hàm cross-entropy nhằm giảm trọng số đối mất mát của các object được phân loại tốt. Thay vào đó, nó sẽ tập trung vào các trường hợp khó hơn, nhằm tránh việc các trường hợp dễ sẽ gây ảnh hưởng quá lớn đến mô hình, dẫn đến giảm hiệu quả khi huấn luyện.

Focal Loss được đưa ra để giải quyết trong trường hợp có sự mất cân bằng lớn giữa các foreground và background classes trong huấn luyện, chẳng hạn 1:1000.



Hình 1.15: So sánh giữa Focal loss và cross entropy loss

Cross Entropy : Để bắt đầu thì chúng ta nhắc lại định nghĩa hàm cross-entropy (CE) cho binary classification

$$CE_{p,y} = \begin{cases} -log(p), & \text{nếu } y = 1; \\ -log(1-p) & \text{còn lại} \end{cases}$$

Trong hàm trên thì y nhận giá trị 1 hoặc -1 biểu diễn ground-truth class và p nằm trong khoảng (0,1) là xác suất dự đoán cho class với y =1. Để cho thuận tiện ta định nghĩa lại hàm trên.

$$p_t = \begin{cases} p, & \text{nếu } y = 1; \\ 1-p & \text{còn lại} \end{cases} CE(p,y) = CE(p_t) = -log(p_t)$$

Cross-entropy có thể được biểu diễn bởi đường màu xanh da trời trong hình trên. Có thể dễ dàng nhận thấy là với các trường hợp được phân loại tốt (xác suất lớn hơn

hoặc bằng 0.6) thì hàm loss nhận giá trị với độ lớn lớn hơn 0, và khi tính tổng các số hạng này sẽ cho ra một số rất lớn so với loss của các trường hợp khó phân loại, và có thể làm ảnh hưởng đến quá trình huấn luyện. Ý tưởng chính của focal-lost là đối với các trường hợp được phân loại tốt (xác suất lớn hơn 0.5) thì focal lost sẽ làm giảm giá trị cross-entropy của nó xuống nhỏ hơn so với thông thường. Do đó, ta sẽ thêm trọng số cho hàm cross-entropy để biến thành hàm focal lost.

Định nghĩa hàm focal loss: Chúng ta sẽ thêm một nhân tử vào phía trước hàm cross-entropy, được gọi là modulating factor, với gamma lớn hơn hoặc bằng 0 được gọi là tham số focussing có thể điều chỉnh được

$$\mathbf{FL}(p_t) = -(1-p_t)^{\gamma} log(p_t)$$

Quay trở lại hình 17 ban đầu, hàm focal loss được mô tả với các giá trị khác nhau của gamma với các giá trị từ 0 đến 5, trong đó với 0 chính là hàm cross-entropy như đã được mô tả bên trên. Chúng ta chú ý đến 2 tính chất của hàm focal loss:

- Khi một mẫu bị phân loại sai và pt nhỏ, modulating factor gần 1 và loss sẽ không bị ảnh hưởng. Còn khi pt tiến tới 1, tức các trường hợp được phân loại tốt, moduling factor sẽ tiến tới 0 và hàm loss trong trường hợp này sẽ bị giảm trọng số xuống.
- Tham số focusing gamma sẽ điều chỉnh tỷ lệ các trường hợp được phân loại tốt được giảm trọng số. Khi gamma càng tăng thì ảnh hưởng của *modulating factor cũng tăng. Thực nghiệm cho thấy với gamma = 2 thì kết quả đạt được sẽ tốt nhất.

Heatmap Loss

Với mỗi GT box (Ground-truth box) $b^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ trong ảnh, chúng ta sẽ tính được tâm của vật thể (c_x^i, c_y^i) như sau:

$$c_x^i = \frac{x_1^i + x_2^i}{2}$$

$$c_y^i = \frac{y_1^i + y_2^i}{2}$$

Sau đó điểm ở trên bản đồ đặc trưng sẽ được định vị bằng việc lấy tâm chia cho độ trượt (stride = 4), do vị trí chỉ lấy phần nguyên nên ở công thức này ta chỉ lấy phần nguyên sau khi chia cho 4.

$$(\tilde{c}_x^i, \tilde{c}_y^i) = \left(\left\lfloor \frac{c_x^i}{4} \right\rfloor, \left\lfloor \frac{c_y^i}{4} \right\rfloor\right)$$

Tiếp đến phản hồi bản đồ nhiệt tại (x, y) sẽ được tính bằng cách áp phân phối chuẩn lên điểm đó (phân phối Gauss) như sau:

$$M_{xy} = \sum_{i=1}^{N} exp^{-\frac{(x-\tilde{c}_x^i)^2+(y-\tilde{c}_y^i)^2}{2\sigma_c^2}}$$

Trong đó N thể hiện cho số lượng vật thể trong ảnh còn σc thể hiện cho độ lệch chuẩn trong hàm Gauss. Vậy hàm lỗi cho bản đồ nhiệt sẽ được định nghĩa như sau với kỹ thuật pixel-wise logistic regression with focal loss như sau [14]:

$$\mathcal{L}_{heatmap} = -\frac{1}{N}\sum_{x,y}\begin{cases}(1-\hat{M}_{xy})^{\alpha} log(\hat{M}_{xy}), & \text{nếu } M_{xy}=1;\\ (1-M_{xy})^{\beta}(\hat{M}_{xy})^{\alpha} log(1-\hat{M}_{xy}) & \text{còn lại}\end{cases}$$

Trong đó $\hat{M}$ là bản đồ nhiệt ước lượng, α, β là các biến, có thể thay đổi tuỳ ý, trong thực nghiệm thì có thể chọn α = 2, β = 4 tương tự như CenterNet.

Offset and Size Loss

Chúng ta có đầu ra của size và offset trong **Box Size Head** là $\hat{S} \in R^{W\times H\times 2}$ và $\hat{O} \in R^{W\times H\times 2}$. Với mỗi tập GT box (ground-truth box) $\mathbf{b}^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ trên ảnh ta tính được size $\mathbf{s}^i = (x_2^i - x_1^i, y_2^i - y_1^i)$. tương tự GT offset $\mathbf{o}^i = \left(c_x^i,\ c_y^i\right) - \left(\left\lfloor \frac{c_x^i}{4}, \frac{c_y^i}{4} \right\rfloor\right)$, đoạn offset này chính là đoạn để sửa lỗi khi lấy phần nguyên ở **Heatmap head**. Do ước lượng size và offset sẽ phụ thuộc vào vị trí của $\hat{o}^i$ và $\hat{s}^i$ của đầu ra trong **Box Size Head**. Vậy chúng ta sẽ định nghĩa hàm lỗi với độ dài mahattan như sau để có thể cân bằng tốc độ tính toán và độ chính xác:

$$\mathcal{L}_{box} = \sum_{i=1}^{N} \|\mathbf{o}^i - \hat{\mathbf{o}}^i\|_{L_1} + \|\mathbf{s}^i - \hat{\mathbf{s}}^i\|_{L_1}$$

Identity Embedding Loss

Chúng ta coi định danh vật thể là nhiệm vụ phân biệt vật thể. Cụ thể là, tất cả các trường hợp vật thể trong lúc huấn luyện đều được coi là cùng một lớp. Với mỗi Với mỗi tập GT box (ground-truth box) $b^i = (x_1^i, y_1^i, x_2^i, y_2^i)$ trong ảnh, chúng ta sẽ lấy được điểm tâm của vật thể $(\tilde{c_x^i}, \tilde{c_y^i})$ trên bản đồ nhiệt. Từ đó ta sẽ rút trích véc tơ định danh đặc trưng là E_{x^i,y^i}, E_yi tại vị trí đó và học để đối chiếu nó vào véc tơ phân phối lớp P_K . Đặt biểu diễn one-hot cho lớp nhãn Ground-truth là Li(k). Vậy ta có hàm lỗi softmax như sau:

$$\mathbf{L_{identity}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} \mathbf{L}^i(k) log(\mathbf{p}(k))$$

Trong đó K là số lớp phân loại (number of classes). Một cách dễ hiểu thì mỗi điểm trên bản đồ nhiệt sau khi được huấn luyện xong sẽ có một giá trị nhất định và giá trị này sẽ khớp với một nhãn trong lớp nhãn Ground-truth, nên hai nhánh định danh Re-ID và xác định vật thể chạy song song với nhau và ra kết quả cùng lúc để cho bước tiếp theo Traking.

1.4.2 Online Tracking

Network Inference

Mạng lấy hình ảnh có kích thước 1088 × 608 làm đầu vào, giống như tác phẩm trước đó JDE [10]. Thực hiện kỹ thuật non-maximum suppression (NMS) lên trên bản đồ nhiệt điểm số để trích xuất các keypoint cao nhất. Chúng tôi giữ vị trí của các keypoint chính có điểm bản đồ nhiệt lớn hơn ngưỡng. Sau đó, chúng tôi tính toán các bounding box tương ứng dựa trên hiệu số ước tính và kích thước hộp. Chúng tôi cũng trích xuất các nhận dạng tại các trung tâm đối tượng ước tính cùng lúc.

Online Box Linking

Chúng tôi sử dụng thuật toán online tracking tiêu chuẩn để đạt được liên kết hộp. Chúng tôi khởi tạo một số tracklet dựa trên các hộp ước tính trong khung đầu tiên. Trong các khung tiếp theo, chúng tôi liên kết các hộp với các tracklet hiện có theo khoảng cách của chúng được đo bằng các tính năng Re-ID và IoU. Chúng tôi cũng sử dụng Kalman Filter để dự đoán vị trí của các tracklet trong khung hiện tại. Nếu nó ở quá xa so với phát hiện được liên kết, chúng tôi đặt chi phí tương ứng thành vô cùng, điều này ngăn cản hiệu quả việc liên kết các phát hiện với chuyển động lớn. Cập nhật các đặc trưng xuất hiện của tracker trong từng bước thời gian để xử lý các biến thể về giao diện [15] [16].

1.5 Kết luận chương 1

Chương này đã trình bày tổng quan về các phương pháp dò tìm đối tượng, đồng thời nêu lên những nhược điểm làm giảm độ chính xác của các phương pháp. Qua đó, đưa ra các giải pháp, hướng tiếp cận mới đến đề tài và các kỹ thuật áp dụng nhằm nâng cao hiệu suất, độ chính xác hơn nữa.

CHƯƠNG 2. CÁC CÔNG TRÌNH LIÊN QUAN

2.1 Phương pháp Two-Steps MOT

Các phương pháp MOT hiện đại như [2] [4] [6] [7] [5] thường coi việc phát hiện đối tượng và Re-ID là hai nhiệm vụ riêng biệt. Đầu tiên chúng áp dụng các bộ dò CNN như [17] [18] [19] để khoanh vùng tất cả các đối tượng được quan tâm trong hình ảnh theo các hộp (boxes). Sau đó, trong một bước riêng biệt, họ cắt hình ảnh theo các hộp và đưa chúng vào mạng nhúng nhận dạng để trích xuất các tính năng Re-ID và liên kết các hộp với tạo thành nhiều track. Các công trình thường tuân theo một phương pháp tiêu chuẩn để liên kết hộp, trước tiên tính toán ma trận chi phí theo các tính năng Re-ID và Intersection over Unions (IoU) (chỉ số đánh giá được sử dụng để đo độ chính xác của Object detector trên tập dữ liệu cụ thể) của các hộp giới hạn (bounding boxes), sau đó sử dụng Kalman Filter [20] và thuật toán Hungarian [21] để hoàn thành nhiệm vụ liên kết. Một số lượng nhỏ các công trình như [6] [5] [7] sử dụng các chiến lược liên kết phức tạp hơn như mô hình nhóm và RNNs.

Ưu điểm của phương pháp two-step là chúng có thể sử dụng mô hình phù hợp nhất cho từng nhiệm vụ tương ứng mà không cần thỏa hiệp. Ngoài ra, chúng có thể cắt hình ảnh theo các hộp giới hạn đã phát hiện và thay đổi kích thước của chúng thành cùng một kích thước trước khi dự đoán các tính năng Re-ID. Điều này giúp xử lý các biến thể tỷ lệ của đối tượng. Kết quả là, những cách tiếp cận này [4] đã đạt được hiệu suất tốt nhất trên các bộ dữ liệu công khai. Tuy nhiên, chúng thường rất chậm vì cả tính năng phát hiện đối tượng và nhúng tính năng Re-ID đều cần nhiều tính toán mà không cần chia sẻ giữa chúng. Vì vậy, thật khó để đạt được suy luận về tốc độ video vốn được yêu cầu trong nhiều ứng dụng.

2.2 Phương pháp One-Shot MOT

Với sự phát triển vượt bậc của tính năng học đa tác vụ [8] [22] [23] trong học sâu, one-shot MOT đã bắt đầu thu hút nhiều sự chú ý của công cuộc nghiên cứu. Ý tưởng cốt lõi là thực hiện đồng thời việc phát hiện đối tượng và nhúng danh tính (các tính năng Re-ID) trong một mạng duy nhất để giảm thời gian suy luận thông qua việc chia sẻ hầu hết các tính toán. Ví dụ: Track-RCNN [8] thêm đầu Re-ID ở trên cùng của Mask-RCNN [17] và hồi quy của một hộp giới hạn và tính năng Re-ID cho mỗi đề xuất. JDE [10] được giới

thiệu trên cùng của YOLOv3 [18] framework giúp đạt được suy luận tốc độ video gần bằng nhau.

Tuy nhiên, độ chính xác theo dõi của phương pháp one-shot thường thấp hơn so với phương pháp two-steps. Điều này là do các tính năng Re-ID đã học không tối ưu, dẫn đến số lượng lớn các công tắc ID. Để giải quyết vấn đề, chúng tôi đề xuất sử dụng các phương pháp tiếp cận không có mỏ neo cho cả phát hiện đối tượng và nhúng danh tính để cải thiện đáng kể độ chính xác theo dõi trên tất cả các điểm chuẩn.

2.3 Các công trình khác

Bài nghiên cứu của Peng Chu và cộng sự vào năm 2021 "TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking" [24]. Công trình này đề xuất một giải pháp có tên TransMOT, sử dụng các máy biến áp đồ thị mạnh mẽ để mô hình hóa hiệu quả các tương tác không gian và thời gian giữa các đối tượng. TransMOT mô hình hóa hiệu quả các tương tác của một số lượng lớn các đối tượng bằng cách sắp xếp quỹ đạo của các đối tượng được theo dõi dưới dạng một tập hợp các đồ thị có trọng số thưa thớt và xây dựng lớp mã hóa máy biến áp đồ thị không gian, lớp mã hóa biến áp thời gian và dựa trên lớp bộ giải mã biến áp đồ thị không gian trên đồ thị. TransMOT không chỉ hiệu quả hơn về mặt tính toán so với Transformer truyền thống mà còn đạt được độ chính xác theo dõi tốt hơn. Để cải thiện hơn nữa tốc độ theo dõi và độ chính xác, chúng tôi đề xuất một khuôn khổ liên kết tầng để xử lý các phát hiện điểm thấp và sai khớp cắn trong thời gian dài đòi hỏi nguồn lực tính toán lớn để lập mô hình trong TransMOT. Phương pháp đề xuất được đánh giá trên nhiều bộ dữ liệu điểm chuẩn bao gồm MOT15, MOT16, MOT17 và MOT20 và nó đạt được hiệu suất hiện đại trên tất cả các bộ dữ liệu.

Bài nghiên cứu của Yifu Zhang và cộng sự vào năm 2021 "ByteTrack: Multi-Object Tracking by Associating Every Detection Box" [25]. Nội dung công trình này trình bày về việc theo dõi đa đối tượng (MOT) nhằm ước tính các hộp giới hạn và danh tính của các đối tượng trong video. Hầu hết các phương pháp có được danh tính bằng cách liên kết các hộp phát hiện có điểm cao hơn ngưỡng. Các đối tượng có điểm phát hiện thấp, ví dụ: các đối tượng bị tắc nghẽn, chỉ đơn giản là bị ném đi, điều này mang lại các đối tượng thực sự không đáng kể bị thiếu và quỹ đạo bị phân mảnh. Để giải quyết vấn đề này, bài nghiên cứu trình bày một phương pháp liên kết đơn giản, hiệu quả để theo dõi bằng cách liên kết mọi ô phát hiện thay vì chỉ những ô có điểm cao. Đối với các hộp phát

hiện điểm thấp, sử dụng các điểm tương đồng của chúng với các tracklet để khôi phục các đối tượng thực và lọc ra các phát hiện nền. Khi được áp dụng cho 9 trình theo dõi hiện đại khác nhau, phương pháp đạt được sự cải thiện nhất quán về điểm IDF1, từ 1 đến 10 điểm. Để thúc đẩy hoạt động hiện đại của MOT, nhóm tác giả đã thiết kế một bộ theo dõi đơn giản và mạnh mẽ, có tên là ByteTrack. Kết quả, họ đạt được 80,3 MOTA, 77,3 IDF1 và 63,1 HOTA trong bộ thử nghiệm của MOT17 với tốc độ chạy 30 FPS trên một GPU V100 duy nhất.

Bài báo của Jialian Wu và cộng sự vào năm 2021 "Track to Detect and Segment: An Online Multi-Object Tracker" [26]. Trong công trình này, tác giả trình bày một mô hình theo dõi và phát hiện chung trực tuyến mới, TraDeS (TRAck to DEtect and Segment), khai thác các manh mối theo dõi để hỗ trợ phát hiện từ đầu đến cuối. TraDeS suy luận theo dõi đối tượng bù đắp bởi một khối lượng chi phí, được sử dụng để truyền bá các tính năng của đối tượng trước đó nhằm cải thiện khả năng phát hiện và phân đoạn đối tượng hiện tại. Tính hiệu quả và tính ưu việt của TraDeS được thể hiện trên 4 bộ dữ liệu, bao gồm MOT (theo dõi 2D), nuScenes (theo dõi 3D), MOTS và Youtube-VIS (theo dõi phân đoạn trường hợp).

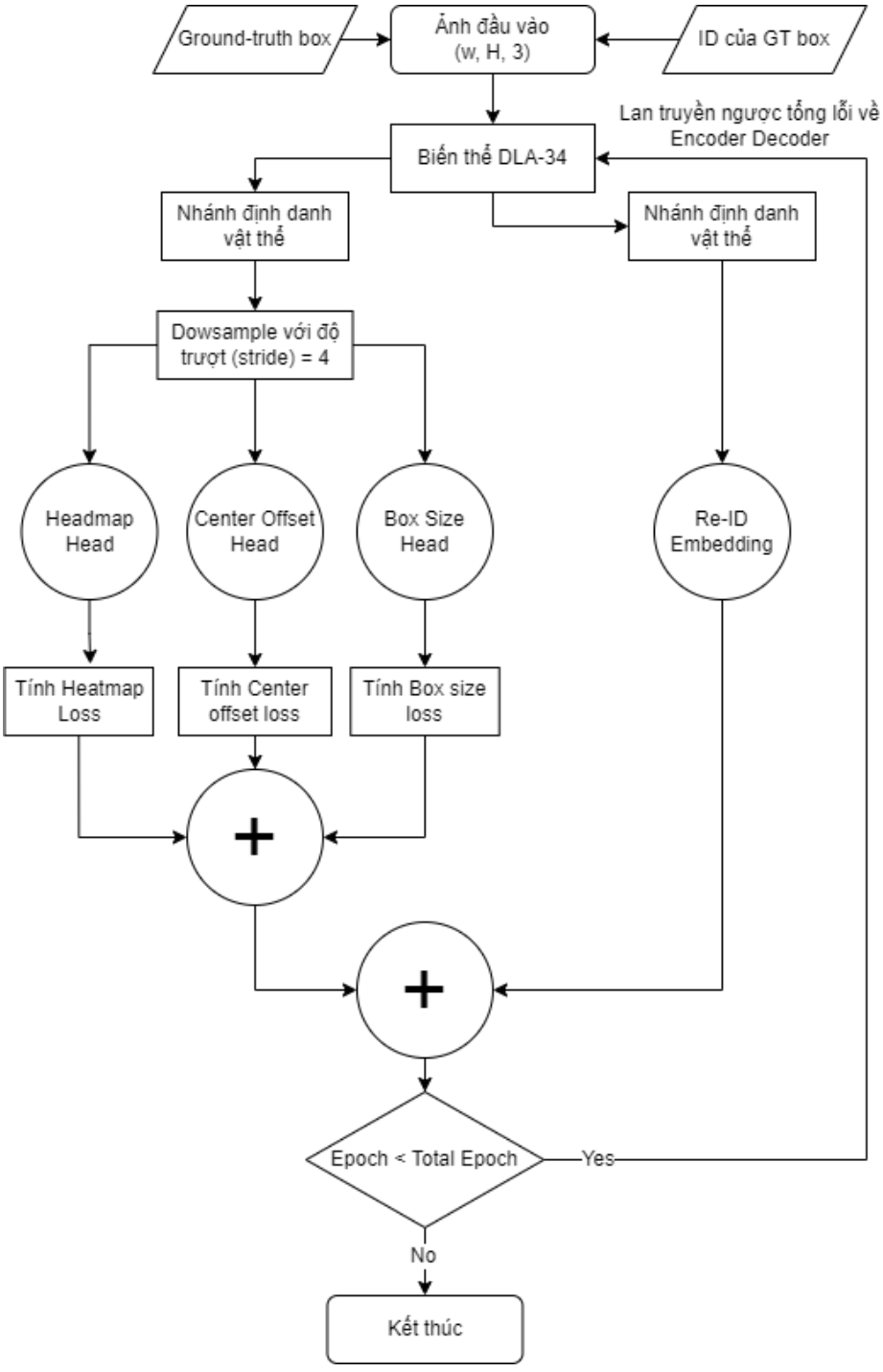
2.4 Kết luận chương 2

Chương này đã trình bày các công trình liên quan mật thiết, giúp củng cố cơ sở kiến thức cho đề tài. Từ đó góp phần giúp định hình hướng phát triển của đề tài. Tiếp theo, chương 3 sẽ trình bày quy trình thực hiện dò tìm và tái định danh đối tượng, đồng thời cũng nêu lên những chỉ số đánh giá sẽ được sử dụng nhằm đánh giá độ chính xác.

CHƯƠNG 3. QUY TRÌNH THỰC HIỆN DÒ TÌM VÀ TÁI ĐỊNH DANH ĐỐI TƯỢNG

3.1 Huấn luyện và nội suy ra đặc trưng

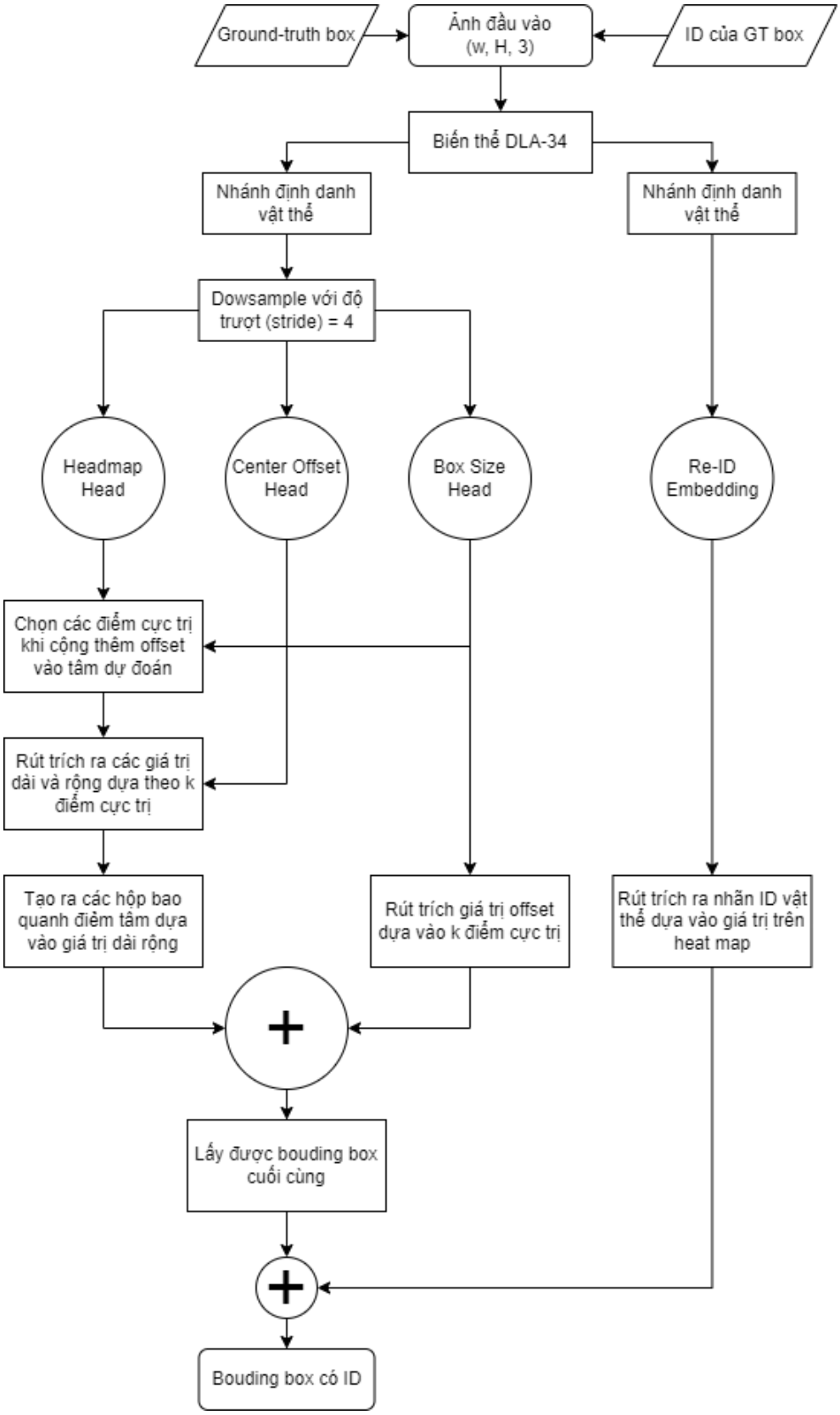
3.1.1 Huấn luyện



Hình 3.1: Flowchart huấn luyện

3.1.2 Nội suy đặc trưng

Đặc trưng ở đây bao gồm bounding box và ID của bounding box đó.



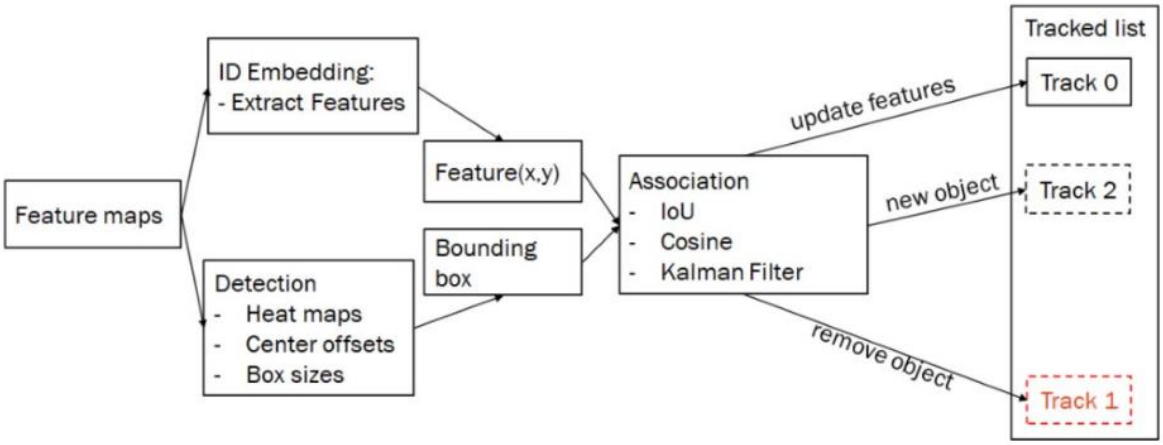
Hình 3.2: Flowchart mô tả cách nội suy đặc trưng

3.2 Theo vết online (Online Tracking)

1. Các phương pháp trước đây hiện thực trình theo dõi thường qua 4 bước chính:
2. Khung ban đầu của đoạn phim
3. Phát hiện đối tượng: Sử dụng các mạng phát hiện đối để trích xuất ô chứa đối tượng.
4. Định danh đối tương: Tính độ tương đồng giữa các ô chứa đối tượng ở khung trước đó và khung hiện tại (đối tượng giống nhau có khoảng cách nhỏ, khác nhau có khoảng cách lớn).
5. Liên kết đối tượng với ID được tạo từ khung trước, tạo mới nếu chưa từng phát hiện

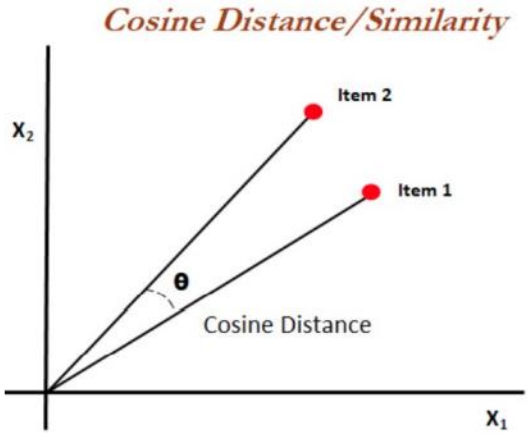
Điểm khác biệt chính là ở bước số 2 và 3 thay vì thực hiện từng bước riêng biệt, nhóm tác giả sử dụng Deep Layer Aggregation để tạo ra bộ đặc trưng có thể dùng cho cả bước Phát hiện đối tượng và Định danh đối tượng, từ đó cho phép trình theo dõi hoạt động với tốc độ cao hơn so với phương pháp truyền thống.

Luồng xử lý của trình theo dõi

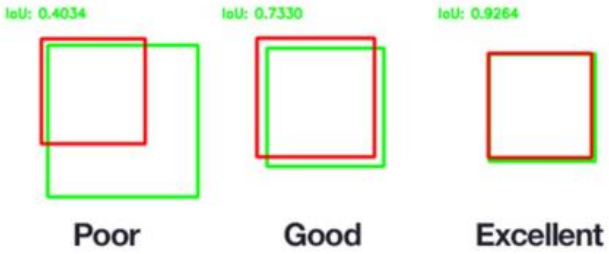


Hình 3.3: Luồng xử lý của trình theo dõi

Trình theo dõi của nhóm tác giả tương tự như Jointly learns the Detector and Embedding với hai thành phần chính là giải thuật Hungarian và Kalman Filter. Giải thuật Hungarian Dùng để so khớp hai vector đặc trưng của đối tượng mới phát hiện và đối tượng đang được theo dõi. sử dụng độ đo khoảng cách Cosine.



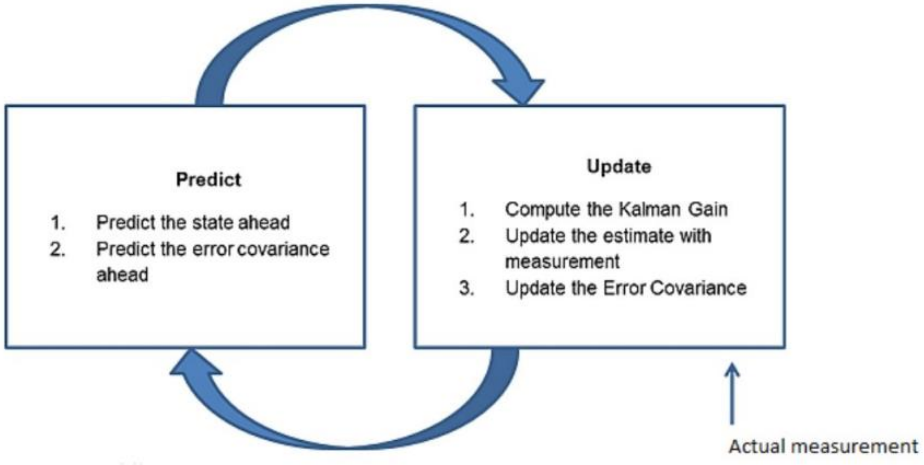
Hình 3.4: Khoảng cách Cosine giữa hai vector đặc trưng



Hình 3.5: Điểm IoU giữa hai vector đặc trưng

Kalman Filter

Kalman filter dùng để dự đoán vị trí của đối tượng từ vị trí hiện tại, được chia làm hai bước dự đoán và cập nhật. Thuật toán hoạt động tương tự như một bước huấn luyện trong mạng neural tương ứng với bước học(dự đoán) và lan truyền ngược(cập nhật). Dự đoán: Khi đối tượng di chuyển mô hình chuyển động của đối tượng(vị trí và vận tốc và ma trận hiệp phương sai) ở khung trước đó được dùng để tính vị trí và vận tốc ở khung hiện tại. Cập nhật: Dựa vào vị trí thực tế và vị trí dự đoán để cập nhật lại ma trận phương sai (covariance matrix) của đối tượng trong mô hình chuyển động.



Hình 3.6: Flow chart of the Iterative process

Với mỗi bounding box ta sử dụng điểm Intersection over Union để xác định vị trí mới của của đối tượng. Từ đó tách danh sách đối tượng phát hiện thành 3 phần gồm:

- Tracked detection: chứa đối tượng được phát hiện lại.
- Unmatched detection: chứa danh sách đối tượng mới hoặc có khoảng cách đặc trưng quá xa so với đặc trưng cũ.

Tracked detection: sẽ tiếp tục được áp dụng Kalman Filter để dự đoán và cập nhật lại mô hình chuyển động, nếu khoảng cách giữa vị trí hiện tại và vị trí dự đoán đối tượng sẽ được xóa khỏi danh sách tracked tạm thời để kiểm tra tiếp ở những khung sau, nếu quá số khung quy định đối tượng sẽ được xóa hoàn toàn khỏi trình theo dõi.

Untracked detection: Đối tượng sẽ được gắn với một ID mới và thêm vào danh sách tracked. Ứng với mỗi bounding box ta cũng có điểm x,y trung tâm tương ứng, từ đó trích xuất được vector đặc trưng có số chiều 128 thuộc R^{128} và sử dụng các độ đo khoảng cách cosine để đo khoảng cách đặc trưng giữa danh đối tượng được phát hiện tại khung hiện tại với đối tượng đang được theo dõi. Dựa trên khoảng cách cosine này ta có thể xác định được ID đối tượng đang được theo dõi ở vị trí nào trong Tracked detection. Ngoài ra đối với các đối tượng đã từng phát hiện nhưng không thể tìm thấy lại sau một số khung nhất định sẽ bị xóa khỏi trình theo dõi.

3.3 Đánh giá độ chính xác của mô hình

Trong vài năm gần đây, cộng đồng theo dõi đa đối tượng đã phát triển mạnh mẽ một phần là do đầu tư lớn từ ngành công nghiệp xe tự hành. Điều này đã dẫn đến một số lượng lớn các MOT benchmark mới được đề xuất. Nhiều trình theo dõi xếp hạng này đã

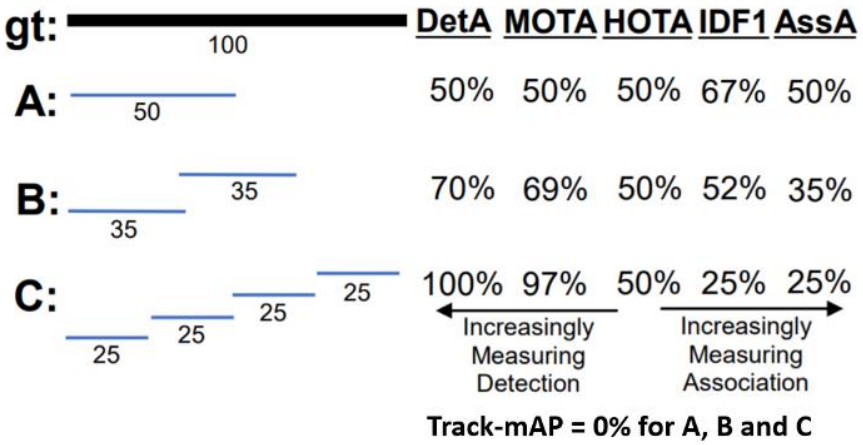
sử dụng chỉ số đánh giá MOTA (Multiple Object Tracking Accuracy) [9], [27],... Chỉ số này đo lường độ chính xác tổng thể của cả trình theo dõi và phát hiện. Nó xử lý cả đầu ra theo dõi và đầu ra phát hiện. Công thức tính chỉ số MOTA như sau:

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}$$

Trong đó m_t, fp_t và mme_t lần lượt là số lần bỏ sót, số lần dương tính giả và số lần không khớp đối với thời gian t.

Thước đo thứ hai được sử dụng gần đây được MOT benchmark áp dụng là IFD1 [28]. Chỉ số này được đề xuất đặc biệt để theo dõi các đối tượng sử dụng bởi 'multi-camera MOT'. IDF1 gần đây cũng đã được triển khai như một chỉ số phụ trên tiêu chuẩn MOTChallenge và đã được ưu tiên hơn MOTA để đánh giá bởi một số phương pháp theo dõi camera đơn. Chỉ số IDF1 là tỷ lệ giữa các phát hiện được xác định chính xác trên số lượng trung bình của các phát hiện xác thực và được tính toán.

Tiếp theo chỉ số HOTA (Higher Order Tracking Accuracy) được định nghĩa bởi [29] có thể đánh giá tất cả các khía cạnh của việc theo dõi. Chỉ số HOTA đo lường rõ ràng cả hai loại lỗi (nhấn mạnh quá mức đến việc phát hiện và liên kết) và kết hợp chúng một cách cân bằng. HOTA cũng tích hợp tỷ lệ đo lường độ chính xác bản địa hóa của các kết quả theo dõi không có trong MOTA hoặc IDF1. HOTA có thể được sử dụng như một chỉ số thống nhất duy nhất để xếp hạng các trình theo dõi, đồng thời phân tách thành một nhóm các chỉ số phụ có thể đánh giá các khía cạnh khác nhau của việc theo dõi riêng biệt và cho phép các trình theo dõi được điều chỉnh cho các yêu cầu khác nhau. Chỉ số HOTA cũng rất dễ hiểu. Có thể thấy rõ điều này trong hình dưới đây. Độ chính xác phát hiện, DetA, chỉ đơn giản là tỷ lệ phần trăm của các phát hiện căn chỉnh. Độ chính xác liên quan như AssA chỉ đơn giản là sự liên kết trung bình giữa các quỹ đạo phù hợp, được tính trung bình trên tất cả các lần phát hiện.



Hình 3.7: Ví dụ một theo dõi đơn giản nêu lên một trong những điểm khác biệt chính giữa các chỉ số đánh giá. Ba trình theo dõi khác nhau được hiển thị để tăng độ chính xác phát hiện và giảm độ chính xác liên kết. MOTA và IDF1 nhấn mạnh quá mức ảnh hưởng của việc

3.4 Kết luận chương 3

Chương này đã trình bày các quy trình thực hiện việc dò tìm cùng với tái định danh đối tượng, bên cạnh đó là các chỉ số đánh giá được áp dụng để tính toán độ chính xác của mô hình trên bộ dữ liệu được xây dựng của luận văn. Tiếp theo, chương 4 sẽ trình bày về bộ dữ liệu thực nghiệm được sử dụng trong bài nghiên cứu và đánh giá nó với các chỉ số đánh giá đã được nêu lên ở chương này.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ THỰC NGHIỆM

4.1 Tập dữ liệu thực nghiệm

Trong bài luận văn này đã sử dụng bộ dữ liệu MOT gồm 4 tập dữ liệu là MOT15, MOT16, MOT17, MOT20 cùng một bộ dữ liệu luận văn tự xây dựng đặt tên là MOT-25.

4.1.1 Tập dữ liệu đã công bố: Multiple Object Tracking Benchmark

Bộ dữ liệu chứa các chuỗi video trong môi trường không bị giới hạn được quay bằng cả máy ảnh tĩnh và máy ảnh chuyển động. Theo dõi và đánh giá được thực hiện trong các tọa độ hình ảnh. Thông tin chi tiết của từng bộ dữ liệu đề cập ở [30]. Trong luận văn này sẽ lấy tập dữ liệu training của mỗi bộ MOT bao gồm:

- **MOT15**: gồm **11** video: Venice-2, KITTI-17, KITTI-13, ADL-Rundle-8, ADL-Rundle-6, ETH-Pedcross2, ETH-Sunnyday, ETH-Bahnhof, PETS09-S2L1, TUD-Campus, TUD-Stadtmitte.
- **MOT16** gồm **7** video: MOT16-02, MOT16-04, MOT16-05, MOT16-09, MOT16-10, MOT16-11, MOT16-13.
- **MOT17**: gồm **21** video: MOT17-02-SDP, MOT17-04-SDP, MOT17-05-SDP, MOT17-09-SDP, MOT17-10-SDP, MOT17-11-SDP, MOT17-13-SDP, MOT17-02-FRCNN, MOT17-04-FRCNN, MOT17-05-FRCNN, MOT17-09-FRCNN, MOT17-10-FRCNN, MOT17-11-FRCNN, MOT17-13-FRCNN, MOT17-02-DPM, MOT17-04-DPM, MOT17-05-DPM, MOT17-09-DPM, MOT17-10-DPM, MOT17-11-DPM, MOT17-13-DPM.
- **MOT20**: gồm **4** video: MOT20-01, MOT20-02, MOT20-03, MOT20-05.

4.1.2 Tập dữ liệu xây dựng

Bộ dữ liệu này được xây dựng ở các khu vực công cộng ở Thành phố Hồ Chí Minh và Tây Ninh kết hợp với một video tìm kiếm được. Bộ video tập trung chủ yếu vào việc ghi lại cảnh hoạt động, di chuyển của người dân trong khu vực. Cụ thể bộ dữ liệu bao gồm 8 video với:

- Video 1: được cắt từ video gốc [31] với nội dung là nhảy múa ở công cộng.
- Video 3, 5: được quay ở khu vực đường lên núi và thánh thất Tây Ninh.

- Video 7, 9, 11, 13: được quay ở khu vực bệnh viện Ung Bướu thành phố Hồ Chí Minh.
- Video 15: được quay ở siêu thị Co.op Mark thành phố khu vực Hồ Chí Minh.

Các thông tin chi tiết của bộ dữ liệu

Thông tin chi tiết của bộ dữ liệu được trình bày trong bảng dưới đây:

Bảng 4.1: Thông tin của tập dữ liệu MOT25

Tên video	FPS	Độ phân giải	Độ dài (số frame, số giây)
MOT25-01	30	1920x1080	1800 (01:00)
MOT25-03	24	1920x1080	370 (00:15)
MOT25-05	25	1920x1080	220 (00:08)
MOT25-07	30	1920x1080	1762 (00:59)
MOT25-09	30	1920x1080	3846 (02:09)
MOT25-11	30	1920x1080	1410 (00:47)
MOT25-13	30	1920x1080	3936 (02:12)
MOT25-15	30	1920x1080	991 (00:33)

4.2 Xây dựng bộ dữ liệu MOT25 Chi tiết quá trình huấn luyện

4.2.1 Xây dựng tracker

Từ tập dữ liệu MOT25, tiến hành chạy tuần tự tập lệnh (được công bố bởi nghiên cứu của tác giả [32]) sau trên môi trường của Google Colaboratory [33] (bật chế độ sử dụng bộ tăng tốc phần cứng GPU) để xây dựng bộ dữ liệu tracker cho bài nghiên cứu. Tiến hành chạy mô hình để phát hiện các đối tượng trên các bộ dữ liệu đã chuẩn bị như sau:

Bước 1: Cài đặt một số thư viện hỗ trợ cần thiết và mô hình CenterNet.

```
!pip install -U torch==1.4 torchvision==0.5 -
f https://download.pytorch.org/whl/cu101/torch_stable.html

import os
from os.path import exists, join, basename, splitext

git_repo_url = 'https://github.com/xingyizhou/CenterNet.git
'
project_name = splitext(basename(git_repo_url))[0]
if not exists(project_name):
```

```
  # clone
  !git clone -q --depth 1 $git_repo_url
  # fix DCNv2
  !cd {project_name}/src/lib/models/networks && rm -
rf DCNv2 && git clone https://github.com/CharlesShang/DCNv2
.git && cd DCNv2 && ./make.sh
  # dependencies
  !cd $project_name && pip install -q -r requirements.txt
```

Bước 2: Cài đặt FairMOT chứa source code để chạy video

```
!git clone https://github.com/microsoft/FairMOT.git && cd F
airMOT && pip install -q -r requirements.txt
```

Sau khi đoạn lệnh được thực thi thành công, cấu trúc thư mục của Google Colab sẽ hiện thị thêm thư mục FairMOT. Lúc này ta sẽ vào thư mục videos và tiến hành tải lên những video đã được chuẩn bị từ trước để tiến hành tạo bộ dữ liệu tracker cho bài nghiên cứu.

Bước 3: Di chuyển đường dẫn vào thư mục models và tiến hành tải xuống mô hình Fairmot_d34 cho bài. Mô hình cơ sở FairMOT (xương sống DLA34) đã được huấn luyện trước trên mô hình CrowdHuman (xương sống DLA34) với số epoch là 60 cùng phương pháp học tập tự giám sát (self-supervised), sau đó được đào tạo trên tập dữ liệu MIX với epoch là 30.

```
!cd /content/FairMOT && mkdir models && cd models && gdown
'https://drive.google.com/u/0/uc?id=1pl_-
ael8wERdUREEnaIfqOV_VF2bEVRT'
```

Bước 4: Tiếp tục tải xuống mô hình ctdet_coco_dla_2x

```
!cd /content/FairMOT/models && gdown 'https://drive.google.
com/u/0/uc?id=1iqRQjsG9BawIl8SlFomMg5iwkb6nqSpi'
```

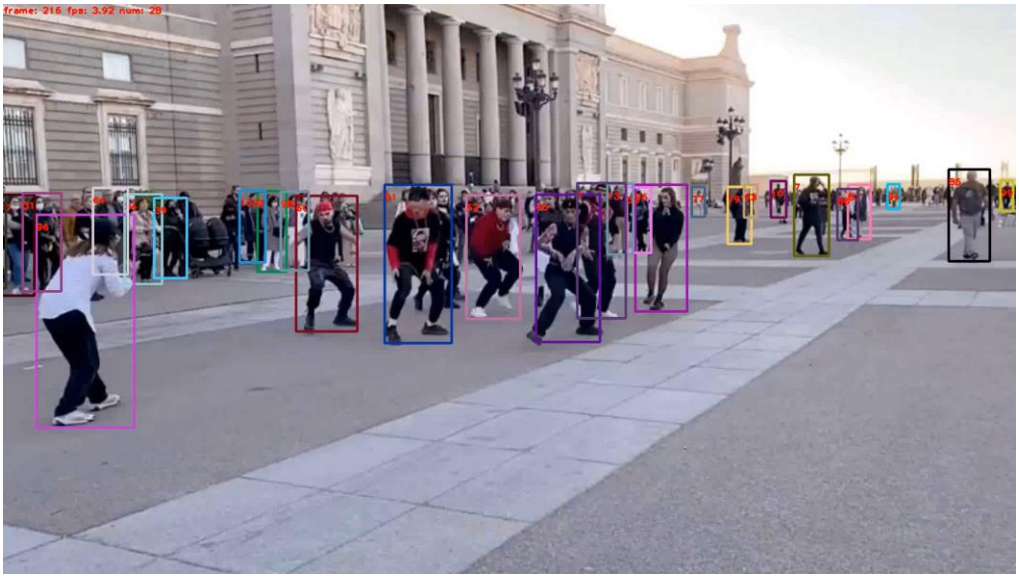
Bước 5: Ở đoạn lệnh này ta tiến hành thay đổi file mình cần chạy bằng cách thay thế chuỗi MOT25-01.mp4 bằng tên của file cần chạy và chuỗi result-MOT25-01 bằng tên thư mục sẽ chứa kết quả sau khi chạy xong.

```
!cd /content/FairMOT/src/ && python demo.py mot --
load_model ../models/fairmot_dla34.pth --conf_thres 0.4 --
input-video ../videos/MOT25-01.mp4 --output-root ../result-
MOT25-01/
```

Kết quả hiển thị trong thư mục result bao gồm video đã được detect (sẽ là file tracker của bài luận văn) và file chứa nội dung chi tiết các thông số của video này cùng

với thư mục chứa bộ hình ảnh là các frame được cắt ra từ video. Lặp lại bước 5 cho tới khi hoàn tất 8 file video đã chuẩn bị, thu được bộ data tracker.

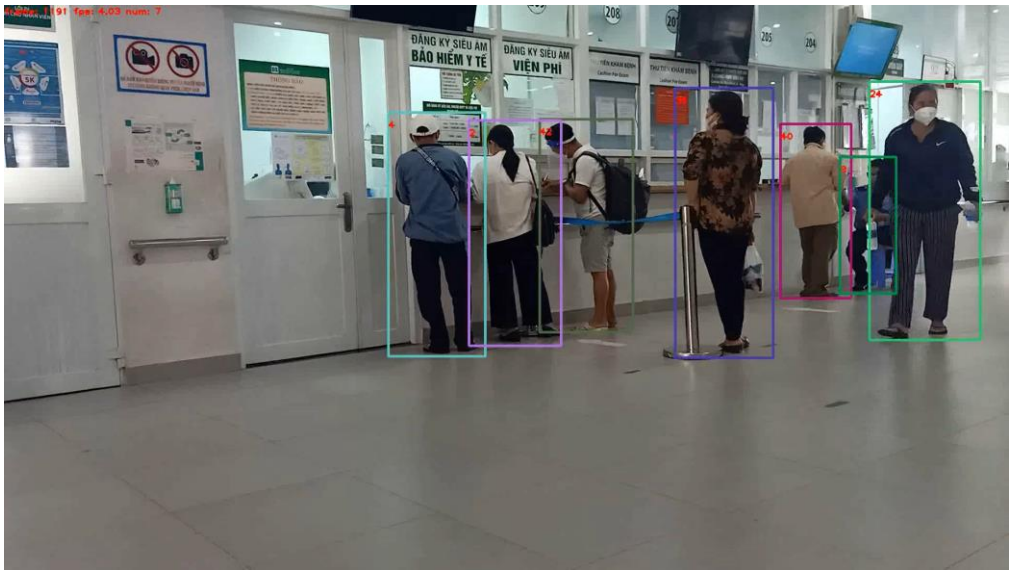
Dưới đây là một số hình ảnh được chụp lại từ những video sau khi chạy detect những người đi bộ.



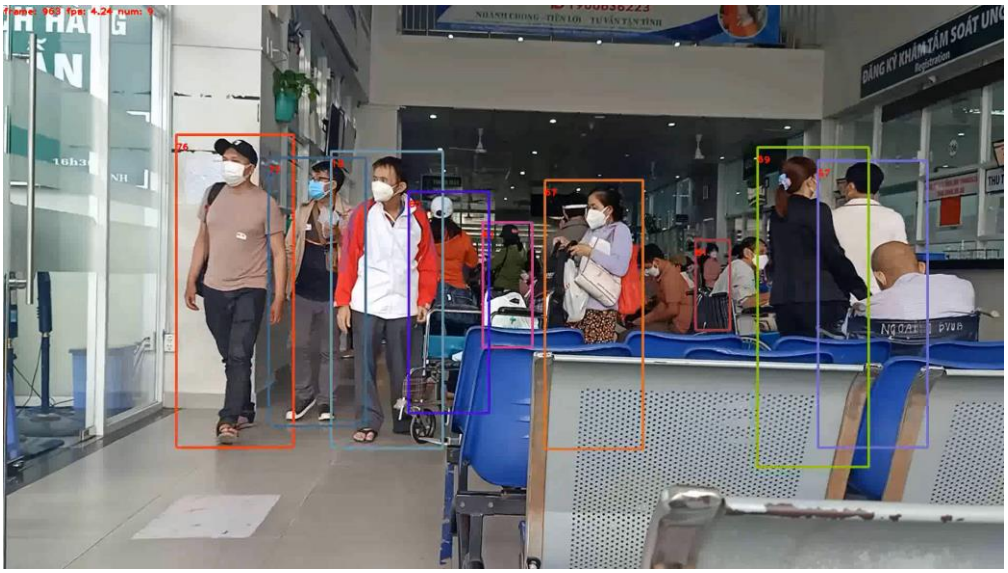
Hình 4.1: Detect người đi bộ trên đường phố ở video nhảy múa đường phố

Hình 4.2: Detect người đi bộ ở khu vực Thánh thất Tây Ninh

Hình 4.3: Detect người đi bộ trước cửa bệnh viện Ung Bướu



Hình 4.4: Detect người đi bộ khu vực khám bệnh của bệnh viện



Hình 4.5: Detect người đi bộ khu khám bệnh của bệnh viện

Hình 4.6: Detect người đi bộ khu vực mua sắm ở siêu thị

4.2.2 Xây dựng ground true

Với bộ data tracker thu được ở bước phía trên, từ các frame được xuất từ video, tiến hành đánh nhãn cho các frame này. Từ đó, ta thu được bộ ground true.

4.3 Đánh giá và so sánh các bộ dữ liệu với TrackEval

Để tiến hành so sánh và đánh giá các bộ dữ liệu, ta cũng cần thực thi tập lệnh với Google Colaboratory. Quá trình như sau:

Bước 1: Truy cập vào Google drive để có thể thao tác với các thư mục

```
from google.colab import drive
drive.mount('/content/drive')
```

Bước 2: di chuyển đến thư mục làm việc. Ở đây luận văn sử dụng thư mục MOT là nơi chứa source code của đề tài, sau đó tiến hành tải về source code của tác giả được công khai tại [29].

```
%cd /content/drive/My Drive/MOT
!git clone https://github.com/JonathonLuiten/TrackEval
```

Bước 3: sau khi hoàn thành việc tải về source code cần thiết, ta di chuyển đến thư mục TrackEval. Tệp dữ liệu data.zip được tác giả lưu trữ ở đường link này. Sau khi tải về, tiến hành upload bộ data vào thư mục trên để giải nén.

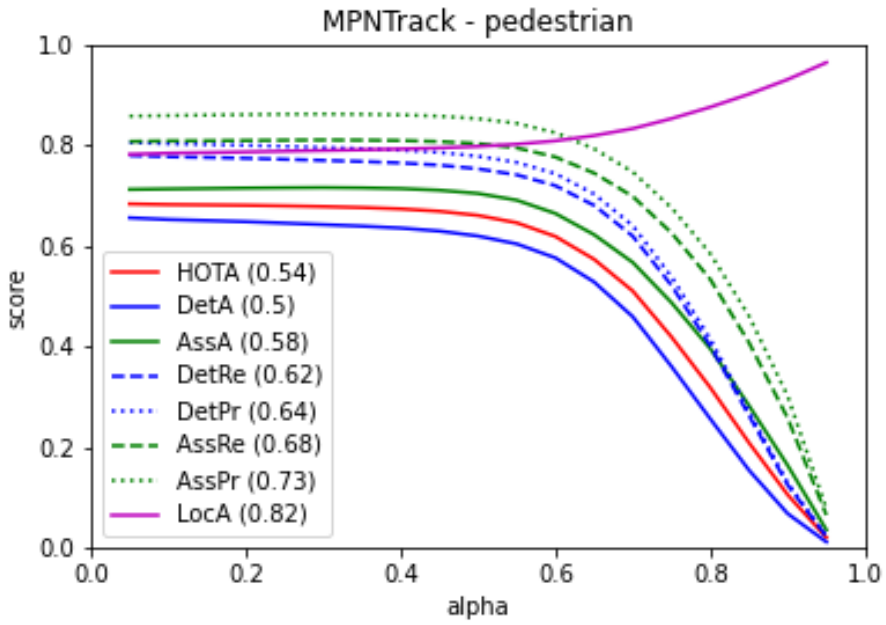
```
%cd /content/drive/My Drive/MOT/TrackEval
!unzip data.zip
```

Bước 4: Chuẩn bị bộ data tracker và ground true cùng thiết lập file code để chạy tương ứng với hướng dẫn của tác giả [29] sau đó tiến hành thực thi đoạn lệnh dưới đây:

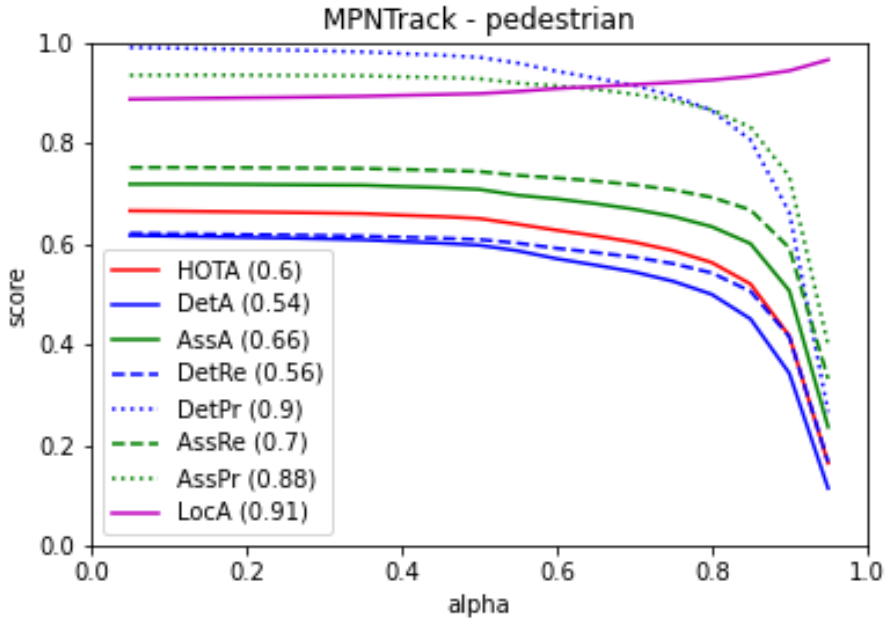
```
import sys
import os
import argparse
from multiprocessing import freeze_support

# sys.path.insert(0, os.path.abspath(os.path.join(os.path.d
irname(__file__), '..')))
import trackeval  # noqa: E402

freeze_support()

# Command line interface:
default_eval_config = trackeval.Evaluator.get_default_eval_
config()
default_eval_config['DISPLAY_LESS_PROGRESS'] = False
default_dataset_config = trackeval.datasets.MotChallenge2DB
ox.get_default_dataset_config()
default_metrics_config = {'METRICS': ['HOTA', 'CLEAR', 'Ide
ntity'], 'THRESHOLD': 0.5}
config = {**default_eval_config, **default_dataset_config,
**default_metrics_config}

# ==========================================================
==============================#
config['TRACKERS_FOLDER'] = '/content/drive/My Drive/MOT/Tr
ackEval/data/trackers/mot_challenge/'
config['GT_FOLDER'] = '/content/drive/My Drive/MOT/TrackEva
l/data/gt/mot_challenge/'
```

```
config['BENCHMARK'] = 'MOT25'
# ================================================================
==============================#

eval_config = {k: v for k, v in config.items() if k in defa
ult_eval_config.keys()}
dataset_config = {k: v for k, v in config.items() if k in d
efault_dataset_config.keys()}
metrics_config = {k: v for k, v in config.items() if k in d
efault_metrics_config.keys()}

# Run code
evaluator = trackeval.Evaluator(eval_config)
dataset_list = [trackeval.datasets.MotChallenge2DBox(datase
t_config)]
metrics_list = []
for metric in [trackeval.metrics.HOTA, trackeval.metrics.CL
EAR, trackeval.metrics.Identity]:
    if metric.get_name() in metrics_config['METRICS']:
        metrics_list.append(metric(metrics_config))
if len(metrics_list) == 0:
    raise Exception('No metrics selected for evaluation')
evaluator.evaluate(dataset_list, metrics_list)
```

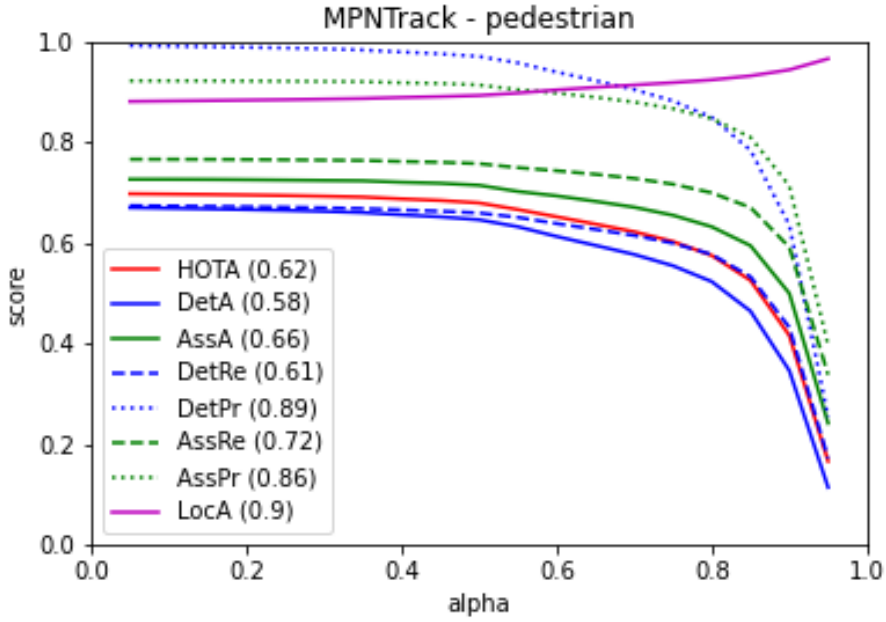
Tiến hành chạy đánh giá lần lượt với MOT15, MOT16, MOT17, MOT20 và MOT25 ta thu được kết quả như sau:



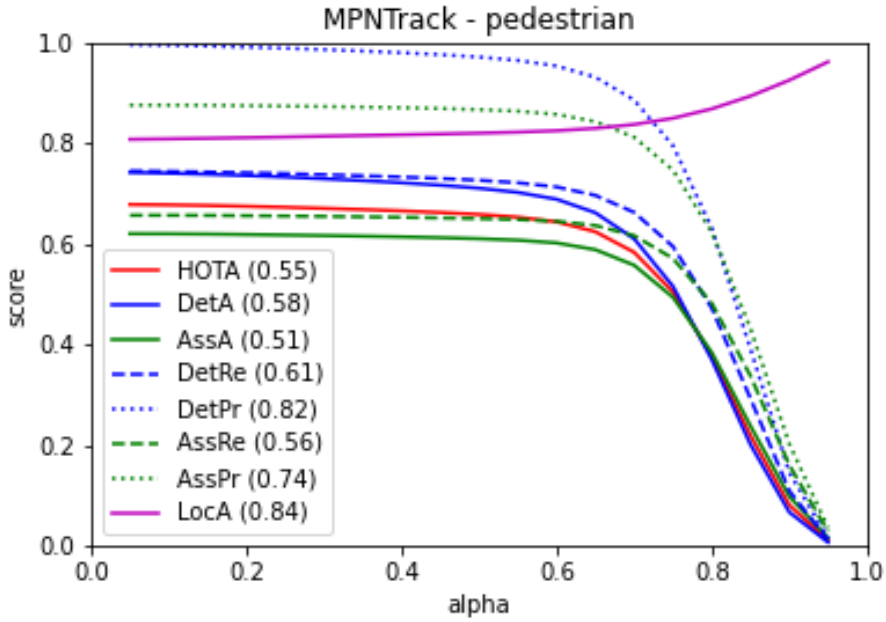
Hình 4.7: Kết quả chạy TrackEval của bộ MOT15



Hình 4.8: Kết quả chạy TrackEval của bộ MOT16

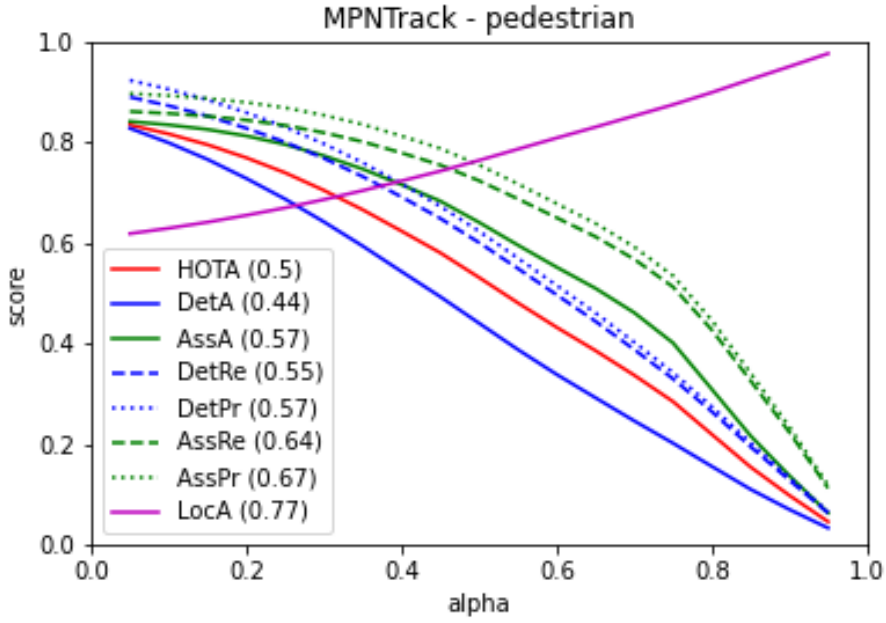


Hình 4.9: Kết quả chạy TrackEval của bộ MOT17



Hình 4.10: Kết quả chạy TrackEval của bộ MOT20

Từ biểu đồ của các bộ MOT 16 đến 20 trên ta thấy các thông đố, độ đo (metrics) đều trên 50%. Trong đó, tất cả các bộ dữ liệu đều đạt cao nhất là LocA dao động từ 82% đến 91%. Đối với MOT15,16,17 thấp nhất là chỉ số DetA với giá trị lần lượt là 50%, 54%, 58%, MOT20 là chỉ số AssA với 51%. Các chỉ số còn lại khá tương đồng dao động từ 55% đến 89% với tất cả các bộ MOT.



Hình 4.11: Kết quả chạy TrackEval của bộ MOT25

Từ biểu đồ của bộ MOT25 ta thấy chỉ số LocA vẫn đạt giá trị cao với 77%, giá trị DetA thấp nhất với 44%, các chỉ số còn lại dao động từ 50% đến 67%. Vì lý do ở Việt nam ít người đi bộ, thường di chuyển trên xe máy và tình trạng dịch bệnh COVID đang diễn biến phức tạp nên chưa thể quay được các video với mật độ người đi bộ cao. Cùng với đó bộ dữ liệu huấn luyện của mô hình chưa có huấn luyện các hình ảnh này (ngồi xe máy, đi xe máy,...) cho nên kết quả phát hiện đối tượng thấp hơn rất nhiều so với bộ dữ liệu công bố trên trang MOT Benchmark.

Kết quả chi tiết của các biểu đồ được mô tả trong bảng sau:

Bảng 4.2: Kết quả các chỉ số đánh giá của bộ data MOT25

Video	HOTA	MOTA	IDF1	MT	ML	IDs
MOT25-01	72.3%	49.5%	47.4%	84	72	371
MOT25-03	86.0%	83.2%	91.0%	9	10	13
MOT25-05	71.9%	15.7%	31.9%	8	12	34
MOT25-07	80.8%	21.6%	59.1%	34	20	115
MOT25-09	91.6%	32.9%	64.5%	33	11	145
MOT25-11	92.9%	48.5%	73.1%	11	3	39
MOT25-13	92.5%	51.2%	74.7%	70	21	210
MOT25-15	95.4%	53.9%	76.6%	11	3	40

Bảng 4.3: Kết quả tổng hợp các chỉ số đánh giá của các bộ data

Dataset	HOTA	MOTA	IDF1	MT	ML	IDs
MOT15	68.3%	53.9%	68.3%	237	138	413
MOT16	66.5%	59%	68.4%	160	131	404
MOT17	69.7%	64.4%	71.2%	649	360	1344
MOT20	67.7%	72.1%	67.8%	1070	236	2674
MOT25*	83.3%	23.4%	59.9%	252	142	967

4.4 Nhận xét

Nhìn chung tất cả các bộ dữ liệu đều nhận diện được đều trên 50%, tuy nhiên bộ dữ liệu ở Việt Nam còn khá yếu kém so với các bộ dữ liệu còn lại vì các lý do được liệt kê ở trên. Từ đó cũng nhận ra phương pháp luận văn xây dựng so sánh với bộ dữ liệu có

sẵn thì kết quả khá ổn định, còn bộ dữ liệu tự xây dựng thì chủ quan. Chỉ số MOTA thấp cho thấy độ chính xác của hệ thống khi detech trên bộ dữ liệu MOT25 kém so với các bộ dữ liệu khác.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả nghiên cứu của đề tài

Việc nghiên cứu bài toán theo vết đa đối tượng đã và đang được nhiều nhà nghiên cứu đặc biệt quan tâm trong thời gian gần đây. Mục tiêu của các công trình nghiên cứu này là tìm ra phương pháp hiệu quả để phát hiện chính xác đối tượng cũng như đánh giá chính xác được độ chính xác của từng mô hình cụ thể. Đồng thời các nghiên cứu cũng liên tục tổng hợp và đánh giá các kết quả của nhiều công trình khác nhau, góp phần cập nhật liên tục xu thế nghiên cứu trong lĩnh vực theo vết đa đối tượng. Với đề tài luận văn này, tác giả đã tập trung nghiên cứu các cách thức và ứng dụng học sâu để tiến hành theo vết đa đối tượng trên bộ dữ liệu của mình, cụ thể:

Xây dựng được bộ data tracker và ground true có tên MOT25 gồm tổng cộng 8 video với độ dài từ 20 đến 90 giây trên mỗi video tập trung ở khu vực công cộng Thành phố Hồ Chí Minh và Tây Ninh. Sau đó, chạy TrackEval để đánh giá với kết quả thu được trên bộ chỉ số đánh giá HOTA, MOTA, IDF1, MT, ML , IDs lần lượt là 83.3%, 23.4%, 59.9%, 252, 142, 967. Tuy nhiên với kết quả này thì luận văn vẫn còn có thể cải thiện thêm để có thể đạt được hiệu năng tốt hơn trên bộ data tốt hơn nữa.

5.2 Hạn chế của đề tài

Trong quá trình thực hiện bài luận luận này cũng không tránh khỏi thiếu sót:

- Bộ dữ liệu chưa tối ưu: số lượng người đi bộ trong mỗi video khá ít, chưa quay được video với tần số người di chuyển cao.
- Mô hình (với bộ dữ liệu training được sử dụng) chưa có nhiều tư thế và hành động của người như tư thế ngồi, ngồi xe máy, lái xe, cúi nhặt đồ,.. vốn phổ biến ở Việt Nam, chỉ phát hiện được số ít người với tư thế này.

5.3 Hướng phát triển của đề tài

Vấn đề kiến nghị và hướng đi tiếp theo của nghiên cứu:

- Xây dựng thêm nhiều bộ data mới với số lượng người nhiều hơn, khu vực quay video đa dạng hơn.
- Sử dụng thêm nhiều mô hình khác hoặc tự xây dựng một mô hình mới để thực hiện việc tracker phong phú hơn.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

[1] Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B., "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016.

[2] Wojke, N., Bewley, A., Paulus, D., " Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*, 2017.

[3] Chen, L., Ai, H., Zhuang, Z., Shang, C., "Real-time multiple people tracking with deeply learned candidate selection and person re-identification," in *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 2018.

[4] Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., Yan, J., "Poi: Multiple object tracking with high performance detection and appearance feature," in *European Conference on Computer Vision*, 2016.

[5] Fang, K., Xiang, Y., Li, X., Savarese, S., "Recurrent autoregressive networks for online multi-object tracking," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[6] Mahmoudi, N., Ahadi, S.M., Rahmati, M., "Multi-target tracking using cnn-based features: Cnnmtt," *Multimedia Tools and Applications*, p. 7077–7096, 2019.

[7] Zhou, Z., Xing, J., Zhang, M., Hu, W., "Online multi-target tracking with tensor-based high-order graph matching," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018.

[8] Kokkinos, I.: Ubernet, "Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," *CVPR*, p. 6129–6138, 2017.

[9] Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Wang, Z., Zheng, L., Liu, Y., Wang, S., "Towards real-time multi-object tracking.,"

arXiv preprint arXiv:1909.12605, 2019.

[11] He, K., Zhang, X., Ren, S., Sun, J., "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[12] Zhou, X., Wang, D., Kr¨ahenb¨uhl, P., "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[13] Lin, T.Y., Doll´ar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[14] Lin, T.Y., Goyal, P., Girshick, R., He, K., Doll´ar, P., "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[15] Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M., "Visual object tracking using adaptive correlation filters," in *2010 IEEE computer society conference on computer vision and pattern recognition*, 2010.

[16] Henriques, J.F., Caseiro, R., Martins, P., Batista, J., "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, p. 583–596, 2014.

[17] He, K., Gkioxari, G., Doll´ar, P., Girshick, R., "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[18] Redmon, J., Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[19] Ren, S., He, K., Girshick, R., Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015.

[20] Welch, G., Bishop, G., et al., "An introduction to the kalman filter," 1995.

[21] H. Kuhn, "The hungarian method for the assignment problem.," *Naval research logistics quarterly*, p. 83–97, 1955.

[22] Ranjan, R., Patel, V.M., Chellappa, R., "Hyperface: A deep multi-task learning

framework for face detection, landmark localization, pose estimation, and gender recognition," *T-PAMI,* p. 121–135, 2017.

[23] Sener, O., Koltun, V., "Multi-task learning as multi-objective optimization," *NIPS,* p. 527–538, 2018.

[24] Peng Chu, Jiang Wang, Quanzeng You, Haibin Ling, Zicheng Liu, "TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking," *arXiv:2104.00194,* 2021.

[25] Zhang, Yifu & Sun, Peize & Jiang, Yi & Yu, Dongdong & Yuan, Zehuan & Luo, Ping & Liu, Wenyu & Wang, Xinggang, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box.," *arXiv:2110.06864,* 2021.

[26] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, Junsong Yuan, "Track to Detect and Segment: An Online Multi-Object Tracker," *arXiv:2103.08808,* 2021.

[27] Wen, Longyin & Du, Dawei & Cai, Zhaowei & Lei, Zhen & Chang, Ming-Ching & Qi, Honggang & Lim, Jongwoo & Yang, Ming-Hsuan & Lyu, Siwei, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *Computer Vision and Image Understanding,* 2020.

[28] Ristani, E., Solera, F., Zou, R., Cucchiara, R., & Tomasi, C, "Performance measures and a data set for multi-target, multi-camera tracking," in *European conference on computer vision*.

[29] Luiten, Jonathon & Ošep, Aljoša & Dendorfer, Patrick & Torr, Philip & Geiger, Andreas & Leal-Taixé, Laura & Leibe, Bastian, "HOTA: A Higher Order Metric for Evaluating Multi-object Tracking," *International Journal of Computer Vision,* pp. 1-31, 2021.

[30] "Multiple Object Tracking Benchmark," [Online]. Available: https://motchallenge.net/.

[31] Pony Squad Official, "[KPOP IN PUBLIC SIDE CAM VER] Jessi (제시) - Cold Blooded [with SWF] + Original choreo by PS ONE TAKE," 22 November 2021. [Online]. Available: https://www.youtube.com/watch?v=Sjl7vTU9fbA.

[32] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, Wenyu Liu, "FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking," *arXiv:2004.01888*, 2020.

[33] "Google Colaboratory," [Online]. Available: https://colab.research.google.com/.

KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	Kỹ thuật học sâu cho bài toán theo vết đa đối tượng.pdf
Tác giả:	Trần Quốc Đạt
Điểm trùng lặp:	1
Thời gian tải lên:	11:28 09/12/2021
Thời gian sinh báo cáo:	11:29 09/12/2021
Các trang kiểm tra:	47/47 trang

Kết quả kiểm tra trùng lặp

Nguồn trùng lặp tiêu biểu

123doc.net tailieu.vn

Học viên **Người hướng dẫn Khoa học**

Trần Quốc Đạt **PGS.TS Lê Hoàng Thái**

III. BẢNG CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn qua **app.kiemtratailieu.vn** một cách trung thực và đạt kết quả mức độ tương đồng **1%** toàn bộ nội dung luận văn. Bản luận văn kiểm tra qua phần mềm là bản cứng luận văn đã nộp để bảo vệ trước hội đồng, Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

TP.Hồ Chí Minh, ngày 25 tháng 01 năm 2022

HỌC VIÊN CAO HỌC

Trần Quốc Đạt