

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**
KHOA KHOA HỌC MÁY TÍNH



**PHẠM MẠNH TIỀN - 18520166
NGUYỄN QUỐC CƯỜNG - 18520206**

KHÓA LUẬN TỐT NGHIỆP

**PHÁT HIỆN VÀ TRUY VẾT VẬT THỂ TRONG
VIDEO VÀ ỨNG DỤNG ƯỚC TÍNH TỐC ĐỘ
PHƯƠNG TIỆN GIAO THÔNG**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN

PGS.TS. VŨ ĐỨC LUNG

TP. HỒ CHÍ MINH, 2022

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**
KHOA KHOA HỌC MÁY TÍNH

**PHẠM MẠNH TIỀN - 18520166
NGUYỄN QUỐC CƯỜNG - 18520206**

KHÓA LUẬN TỐT NGHIỆP

**PHÁT HIỆN VÀ TRUY VẾT VẬT THỂ TRONG
VIDEO VÀ ỨNG DỤNG ƯỚC TÍNH TỐC ĐỘ
PHƯƠNG TIỆN GIAO THÔNG**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN

PGS.TS. VŨ ĐỨC LUNG

TP. HỒ CHÍ MINH, 2022

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(CÁN BỘ HƯỚNG DẪN)

Tên khóa luận:

**PHÁT HIỆN VÀ TRUY VẾT VẬT THỂ TRONG VIDEO VÀ ỨNG DỤNG ƯỚC
TÍNH TỐC ĐỘ PHƯƠNG TIỆN GIAO THÔNG**

Nhóm SV thực hiện:

Nguyễn Quốc Cường - 18520206

Phạm Mạnh Tiến - 18520166

Cán bộ hướng dẫn:

TS. Vũ Đức Lung

Đánh giá Khóa luận

1. Về cuốn báo cáo:

Số trang _____

Số chương _____

Số bảng số liệu _____

Số hình vẽ _____

Số tài liệu tham khảo _____

Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

.....
.....
.....
.....
.....

2. Về nội dung nghiên cứu:

.....
.....
.....

3. Về chương trình ứng dụng:

.....
.....

4. Về thái độ làm việc của sinh viên:

.....
.....
.....

Đánh giá chung:

.....

.....

Điểm từng sinh viên:

<Tên sinh viên 1>:...../10

<Tên sinh viên 2>:...../10

Người nhận xét

(Ký tên và ghi rõ họ tên)

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(CÁN BỘ PHẢN BIỆN)

Tên khóa luận:

**PHÁT HIỆN VÀ TRUY VẾT VẬT THỂ TRONG VIDEO VÀ ỨNG DỤNG ƯỚC
TÍNH TỐC ĐỘ PHƯƠNG TIỆN GIAO THÔNG**

Nhóm SV thực hiện:

Nguyễn Quốc Cường - 18520206

Phạm Mạnh Tiến - 18520166

Cán bộ phản biện:

Đánh giá Khóa luận

1. Về cuốn báo cáo:

Số trang _____
Số bảng số liệu _____
Số tài liệu tham khảo _____

Số chương _____
Số hình vẽ _____
Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

.....
.....
.....
.....

2. Về nội dung nghiên cứu:

.....
.....
.....

3. Về chương trình ứng dụng:

.....
.....
.....

4. Về thái độ làm việc của sinh viên:

.....
.....
.....

Đánh giá chung:

.....
.....

Điểm từng sinh viên:

<Tên sinh viên 1>:...../10

<Tên sinh viên 2>:...../10

Người nhận xét

(Ký tên và ghi rõ họ tên)

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỀ TÀI: PHÁT HIỆN VÀ TRUY VẾT VẬT THỂ TRONG VIDEO VÀ ỨNG DỤNG ƯỚC TÍNH TỐC ĐỘ PHƯƠNG TIỆN GIAO THÔNG

Cán bộ hướng dẫn: PGS.TS. Vũ Đức Lung

Thời gian thực hiện: Từ ngày 06/09/2021 đến ngày 22/01/2022.

Sinh viên thực hiện:

Nguyễn Quốc Cường - 18520206

Phạm Mạnh Tiến - 18520166

Nội dung đề tài:

A. Mục tiêu:

- Hiểu được các cơ sở lý thuyết các kiến trúc mạng học sâu cho bài toán Theo dõi nhiều vật thể (Multiple Object Tracking) như Deep SORT, FairMOT, CenterTrack, GSOT, IoUTracker, Tracktor
- Hiểu được cơ sở lý thuyết các giải thuật ước tính tốc độ vật thể qua camera giao thông. Rút ra được giải thuật phù hợp
- Đánh giá được độ chính xác, độ lớn của mô hình và tốc độ xử lý của các cài đặt.

B. Phạm vi:

- Các phương pháp được dùng để giải quyết bài toán xác định và gán nhãn nhiều vật thể trong video hoặc chuỗi các hình ảnh (Multiple Object Tracking) dựa trên các nghiên cứu kỹ thuật học sâu và xử lý ảnh từ năm 2017 cho tới nay
- Các giải thuật dùng để ước tính tốc độ vật thể trong dữ liệu video.

C. Đối tượng: Các phương tiện giao thông trong các video dữ liệu từ máy quay giám sát giao thông.

D. Phương pháp thực hiện: Sử dụng các kỹ thuật học sâu và xử lý ảnh để xác định vị trí, phân loại và theo dõi các vật thể trong hình ảnh. Từ kết quả trên tiến hành ước lượng tốc độ của phương tiện

E. Kết quả mong đợi:

- Kết quả tìm hiểu các nhóm ý tưởng dựa trên học sâu của các phương pháp truy vết đối tượng, kết quả đánh giá trên dữ liệu video giao thông nhằm tạo tiền đề cho các nghiên cứu lý thuyết trong tương lai.
- Xây dựng được hệ thống có độ chênh lệch sai số giữa tốc độ ước tính và tốc độ thực tế của các vật thể giao thông thấp nhất có thể nhằm đặt tiền đề cho việc ứng

dụng dụng hệ thống vào các thiết bị máy quay giám sát giao thông trong thực tế.

- Tham gia vào một cuộc thi học thuật cấp quốc tế, viết và đăng được một bài báo quốc tế uy tín.

Kế hoạch thực hiện:

A. Tóm tắt kế hoạch làm việc:

- Bước chuẩn bị: Thu thập các bộ dữ liệu chuẩn cho bài toán Multiple Object Tracking với đối tượng là phương tiện giao thông được các nghiên cứu gần đây sử dụng nhiều để đánh giá như UAVDT, UA-DETRAC. Thu thập dữ liệu có nhãn tốc độ phương tiện giao thông (BrnoCompSpeed)
- Bước 1: Tìm hiểu cơ sở lý thuyết các kiến trúc mạng học sâu từ năm 2017 đến nay cho bài toán theo dõi nhiều vật thể trong video (Multiple Object Tracking). Tổng kết các nhóm ý tưởng chính và các phương pháp đặc trưng của các nhóm ý tưởng.
- Bước 2: Tái hiện kết quả được công bố của các phương pháp. Chạy thử nghiệm các mô hình đã khảo sát trên dữ liệu giao thông và rút ra kết quả đánh giá. Lựa chọn mô hình cho bước tiếp theo.
- Bước 3: Tìm hiểu cơ sở lý thuyết các giải thuật ước tính tốc độ phương tiện giao thông qua video từ các nghiên cứu gần đây.
- Bước 4: Chạy thử nghiệm các giải thuật ước tính tốc độ đã khảo sát và rút ra kết quả đánh giá.
- Bước 5: Tổng kết kết quả đạt được và viết báo cáo tổng kết, viết báo cáo khóa luận tốt nghiệp.

B. Phân công công việc:

- Sinh viên 1: Chuẩn hóa dữ liệu, xây dựng Pipeline huấn luyện mô hình, thực hiện bước 1, 2, 3, 4 và 5 theo kế hoạch
- Sinh viên 2: Thu thập dữ liệu, xây dựng mã nguồn, cùng thực hiện bước 1, 2, 3, 4 và 5 theo kế hoạch

C. Dự kiến:

- Dự kiến bước 1 và 2 phải hoàn thành trước ngày 01/11/2021 (Thời gian báo cáo tiến độ)
- Dự kiến bước 3 và 4 phải hoàn thành trước ngày 27/12/2021 (Thời gian phản biện đề tài)

Xác nhận của CBHD (Ký tên và ghi rõ họ tên)	TP. HCM, ngày 30 tháng 12 năm 2021 Sinh viên Nguyễn Quốc Cường Phạm Mạnh Tiên
---	--

LỜI CẢM ƠN

Để hoàn thành khóa luận này, chúng tôi tỏ lòng biết ơn sâu sắc đến PGS. TS. Vũ Đức Lung đã hướng dẫn tận tình trong suốt quá trình nghiên cứu.

Chúng tôi chân thành cảm ơn quý thầy, cô trong khoa Khoa Học Máy Tính, Trường Đại học Công Nghệ Thông Tin - Đại học Quốc gia thành phố Hồ Chí Minh đã tận tình truyền đạt kiến thức trong những năm chúng tôi học tập ở trường. Với vốn kiến thức tích lũy được trong suốt quá trình học tập không chỉ là nền tảng cho quá trình nghiên cứu mà còn là hành trang để bước vào đời một cách tự tin.

Cuối cùng, chúng tôi xin chúc quý thầy, cô dồi dào sức khỏe và thành công trong sự nghiệp cao quý.

Mục lục

1 MỞ ĐẦU	1
1 Giới thiệu đề tài	1
2 Mục tiêu, đối tượng và phạm vi nghiên cứu	2
2.1 Mục tiêu của đề tài	2
2.2 Đối tượng và phạm vi nghiên cứu	2
3 Đóng góp của đề tài	3
4 Cấu trúc luận văn	3
2 TỔNG QUAN	4
1 Mô tả bài toán	4
2 Các hướng nghiên cứu hiện nay	6
2.1 Khảo sát các phương pháp phát hiện vật thể	7
2.2 Khảo sát các phương pháp truy vết vật thể	8
2.3 Khảo sát các phương pháp ước tính tốc độ phương tiện giao thông	10
3 Thách thức của bài toán	12
4 Vấn đề nghiên cứu	14
3 CƠ SỞ LÝ THUYẾT	15
1 Truy vết đa vật thể	15
1.1 Các kiến thức cơ sở	15

1.1.1	Giải thuật Hungary	15
1.1.2	Bộ lọc Kalman	19
1.2	Các phương pháp truy vết nhiều vật thể trong video	23
1.2.1	IoUTracker[1]	23
1.2.2	SORT[2]	25
1.2.3	DEEPSORT[3]	26
1.2.4	CenterTrack[4]	27
1.2.5	FairMOT[5]	28
1.2.6	Tracktor[6]	30
2	Ước tính tốc độ phương tiện giao thông	34
2.1	Chuyển từ tọa độ ảnh thành tọa độ thực tế - Mô hình camera tuyến tính (Linear camera model)	34
2.2	Phương pháp tự động hiệu chỉnh thông số camera (Automatic Camera Calibration)	39
2.2.1	Phương pháp phát hiện điểm biến mất (Vanishing Point)	41
2.2.2	Hiệu chỉnh thông số máy ảnh từ các vanishing point	50
2.2.3	Ước tính tốc độ phương tiện từ các thông số máy ảnh đã được hiệu chỉnh	52
4	KẾT QUẢ ĐẠT ĐƯỢC	53
1	Thí nghiệm đánh giá các phương pháp truy vết vật thể trong video . .	53
1.1	Tổng quan về bộ dữ liệu UA-DETRAC	54
1.2	Độ đo được sử dụng để đánh giá cho bài toán truy vết vật thể trong video	57
1.2.1	Độ đo <i>MOTA</i> [7]	59
1.2.2	Độ đo <i>IDFI</i> [8]	60
1.2.3	Độ đo <i>HOTA</i> [9]	61

1.3	Kết quả đánh giá	63
1.3.1	Thí nghiệm đánh giá độ chính xác	64
1.3.2	Thí nghiệm đo tốc độ	67
2	Thí nghiệm đánh giá thuật toán ước tính tốc độ giao thông	68
2.1	Tổng quan về bộ dữ liệu BronoCompSpeed	68
2.2	Độ đo được sử dụng để đánh giá thuật toán ước tính tốc độ . .	71
2.3	Kết quả đánh giá	72
5	KẾT LUẬN	76
6	HƯỚNG PHÁT TRIỂN	77

Danh sách bảng

4.1	Bảng dữ liệu thống kê thông tin các nhãn vị trí vật thể và nhãn tracking vật thể ở tập huấn luyện và tập kiểm thử	55
4.2	Cài đặt huấn luyện mô hình phát hiện vật thể	64
4.3	Kết quả độ chính xác các phương pháp SDE	65
4.4	Cài đặt huấn luyện mô hình JDE	66
4.5	Kết quả độ chính xác các phương pháp JDE	66
4.6	Kết quả tốc độ các mô hình	67
4.7	Số lượng phương tiện đi qua vùng quan tâm của mỗi video trong tập dữ liệu [10]	70
4.10	Số trường hợp dự đoán sai và recall của mô hình phát hiện và truy vết vật thể	73
4.8	Bảng kết quả sai số tốc độ tuyệt đối (absolute speed error) của thuật toán ước tính tốc độ phương tiện (đơn vị: km/h)	74
4.9	Bảng kết quả sai số tốc độ tương đối (relative speed error) của thuật toán ước tính tốc độ phương tiện (đơn vị: %)	75

Danh sách hình vẽ

2.1	Minh họa 3 phần chính của bài toán ước tính tốc độ bằng thuật toán truy vết	5
2.2	Các thành phần chính của hệ thống cảnh báo và xử phạt vượt quá tốc độ qua camera: Thông tin dữ liệu đầu vào, phát hiện và truy vết vật thể, ước tính khoảng cách và tốc độ, cảnh báo và xử phạt khi tốc độ vượt mức quy định [11]	6
2.3	Các cách biểu diễn vị trí của phương tiện. (a)Các điểm đặc trưng. (b)Tâm vùng bao. (c) Hộp giới hạn. (d)Biển số[11]	7
2.4	Các bước chính của phần truy vết đối tượng	9
2.5	Sơ đồ các khía cạnh của bài toán ước tính tốc độ giao thông hiện nay [11]	10
3.1	Đồ thị cặp ghép minh họa cho bài toán phân chia công việc	16
3.2	Quá trình dự đoán và cập nhật của mô hình xác suất của bộ lọc Kalman	20
3.3	Nguyên lý hoạt động của IoUTracker[1]	24
3.4	Nguyên lý của VIoUTracker. (a)Kết quả của IoUTracker. (b)Mô hình chuyển động được sử dụng để nối dài các chuỗi truy vết. (c)Kết quả của VIoUTracker[12]	25
3.5	Sơ đồ phương pháp SORT[13]	26
3.6	Sơ đồ phương pháp DEEPSORT[13]	27
3.7	Sơ đồ phương pháp CenterTrack[4]	28

3.8	Sơ đồ biểu diễn phương pháp FairMOT [5]	29
3.9	Sơ đồ mạng của Faster RCNN với backbone Resnet101-FPN[14]. Sơ đồ bao gồm 3 phần chính: mạng backbone, mạng đề xuất khu vực, mạng ROI head[15]	31
3.10	Tính kết quả vị trí chính xác của vật	32
3.11	Phương pháp Tracktor[6]	33
3.12	Mô hình camera tuyến tính chuyển đổi giữa tọa độ ảnh và tọa độ 3D thực tế[16]	35
3.13	Vanishing Point (Điểm biến mất)	39
3.14	Minh họa kết quả thuật toán tự động hiệu chỉnh thông số camera với 3 véc-tơ chỉ phương của đường thẳng đi qua ba vanishing point	40
3.15	Minh họa trường hợp không thể tham số đường thẳng bằng hệ tọa độ song song	42
3.16	Minh họa đường thẳng trong hệ tọa độ Đề-các ban đầu được mô hình hóa trong hai hệ tọa độ song song	42
3.17	Minh họa hai phép biến đổi Hough liên tiếp bằng tham số hóa PCLines để chuyển một điểm từ không gian vô hạn Đề-các thành một điểm trong không gian song song[17]	43
3.18	Minh họa tổ hợp bốn phép biến đổi Hough liên tiếp bằng tham số hóa PCLines [18]	44
3.19	Diamond space[17]	44
3.20	Minh họa vị trí vanishing point tìm được qua hai phép biến đổi Hough bằng tham số hóa PCLines	45
3.21	Minh họa các điểm đặc trưng của phương tiện được truy vết [18]	47
3.22	Tích lũy các đặc trưng cạnh của phương tiện trong không gian kim cương để tìm vanishing point thứ hai [18]	49
3.23	Pipeline thực hiện ước tính tốc độ tự kết quả truy vết	52
3.24	Một frame từ video demo	52

4.1	Minh họa các khung hình được gán nhãn của bộ dữ liệu UA-DETRAC. [19]	55
4.2	Thông số thống kê các đặt trưng của bộ dữ liệu UA-DETRAC trên bộ dữ liệu huấn luyện (training set) và bộ dữ liệu kiểm thử (testing set). [19]	56
4.3	Minh họa cách xác định giá trị S bằng hàm Jaccard Index (IoU)	58
4.4	Minh họa trường hợp nhãn truy vết vật thể (IDSW) thay đổi trong quỹ đạo di chuyển của vật thể.	60
4.5	Minh họa cách xác định TPA, FPA, FNA [9]	63
4.6	Mô hình hệ thống thu thập dữ liệu tốc độ. Với hai thiết bị phát sóng LIDARS được xác định tọa độ vị trí đặt và thời gian thực tế bằng GPS và ba máy ghi hình ở ba vị trí đặt có góc khác nhau [10]	69
4.7	Minh họa quá trình lấy mẫu của hệ thống gán nhãn tốc độ [10]	70
4.8	(6 biểu đồ trên) Biểu đồ tần suất phương tiện theo thời gian (xe/phút), 6 biểu đồ giữa Biểu đồ tần suất tốc độ phương tiện đo được bởi hệ thống LIDAR,[10]	70
4.9	Minh họa sai số vị trí phương tiện trong hai phương pháp ước tính khoảng cách bằng phép biến đổi đồng nhất (Hormography transform, hình a) và sử dụng đường ảo hình b) [11]	72
4.10	Sai số với hệ thống ước tính khoảng cách sử dụng hai khung hình liên tiếp [11]	72

Bảng dịch thuật

Deep Learning	Học sâu
FPS	Frame per second
mAP	mean Average Precision
vanishing point	Điểm biến mất (điểm ảo)
polyline	Tập hợp các đường thẳng liền nhau tạo thành một khối
bounding box	Hộp giới hạn
anchor box	Hộp anchor
tracking trajectory	Chuỗi truy vết

TÓM TẮT KHÓA LUẬN

Nội dung chính của khóa luận nhằm tìm hiểu, nghiên cứu xây dựng hệ thống một camera cố định và truy vết nhiều phương tiện giao thông, đồng thời ước tính tốc độ di chuyển của các phương tiện chỉ với hình ảnh thu được từ một camera. Trong quá trình nghiên cứu, nhóm chúng tôi đã tiến hành tổng hợp, đánh giá ưu và nhược điểm của các công trình, công nghệ đã và đang được nghiên cứu, sử dụng, đồng thời tìm hiểu và tiếp cận nhiều công trình nghiên cứu mới được công bố những năm gần đây. Để hoàn thành nội dung của đề tài, nhóm chúng tôi đã tiến hành nghiên cứu, khảo sát các phương pháp theo dõi vật thể trong dữ liệu video để từ đó đặt nền móng cho việc thực hiện ước tính tốc độ giao thông của các phương tiện giao thông. Phần còn lại của khóa luận tập trung vào việc đánh giá tính chính xác và hiệu quả của mô hình theo dõi vật thể trong video và thuật toán ước tính tốc độ đã được thực hiện, kết quả đạt được đánh giá trên bộ dữ liệu đánh giá đã được đảm bảo bởi các công trình nghiên cứu liên quan, thông qua các độ đo dành riêng cho từng bài toán được triển khai trong hệ thống và được sử dụng rộng rãi bởi cộng đồng nghiên cứu, đồng thời phân tích ưu nhược điểm của các phương pháp đã thực hiện và thảo luận những vấn đề mà mô hình phát hiện, theo dõi vật thể và thuật toán ước tính tốc độ còn gặp phải. Cuối cùng, nhóm chúng tôi đề xuất hướng phát triển tiếp theo của đề tài.

Chương 1

MỞ ĐẦU

1 Giới thiệu đề tài

Ngày nay các thiết bị công nghệ thông minh như điện thoại thông minh, máy tính, máy tính bảng... ngày càng đóng vai trò to lớn trong cuộc sống của con người. Chúng không chỉ hỗ trợ con người trong việc thông tin liên lạc mà còn giúp tính toán, xử lý tác vụ nhanh và chính xác, hỗ trợ con người đưa ra quyết định, thu thập thông tin, hỗ trợ giám sát và điều khiển hoạt động của máy móc. Bên cạnh đó chúng còn phục vụ nhu cầu giải trí của con người. Vì tầm ảnh hưởng và ứng dụng rộng rãi đó mà các ngành khoa học nghiên cứu về công nghệ thông minh như Trí tuệ nhân tạo (AI), Dữ liệu lớn (Big Data), Điện toán đám mây (Cloud),... ngày càng được quan tâm nghiên cứu. Trong đó, nhiều nghiên cứu đã và đang nỗ lực phát triển các thiết bị thông minh mô phỏng khả năng quan sát của con người bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh và được gọi chung lĩnh vực Thị giác máy tính.

Áp dụng các phương pháp xử lý ảnh của lĩnh vực Thị giác máy tính vào dữ liệu giám sát giao thông là một trong lớp các bài toán có nhiều thách thức và được quan tâm phát triển trong những năm gần đây. Hình ảnh từ máy quay được trang bị trên các hệ thống giám sát giao thông hay các thiết bị thông minh như điện thoại, máy tính được thu thập. Sau đó là sử dụng các phương pháp xử lý ảnh và đặc biệt là sự phát triển và thành công những năm gần đây của các phương pháp Máy học (Machine Learning) và Học sâu (Deep Learning) giúp máy tính có khả năng nhận dạng và truy vết sự chuyển động của các phương tiện giao thông chính xác và đáng tin cậy hơn.

Những nghiên cứu này tạo cơ sở cho các hệ thống giám sát giao thông hoạt động tự động, phân tích và đếm số lượng phương tiện tham gia tại một điểm nút giao thông, hỗ trợ cảnh báo sớm tình trạng ùn tắc và phát hiện các điểm bất thường trong video giám sát để phát hiện kịp thời các tình huống xảy ra tai nạn.

Trong hệ thống giám sát giao thông hiện nay, hệ thống xác định tốc độ phương tiện giao thông là một trong những bài toán mang đến nhiều thách thức như chi phí lắp đặt, bảo trì. Khả năng vận hành còn dựa nhiều vào sức của con người và điều kiện thời tiết. Việc cảnh báo sớm, hoặc xử phạt kịp thời phương tiện tham gia giao thông chạy quá tốc độ sẽ góp phần đáng kể làm giảm thiểu tai nạn giao thông, cũng như góp phần dự đoán trước các điểm ùn tắc giao thông, hỗ trợ cơ quan chức năng điều tiết giao thông một cách hiệu quả.

Nếu một hệ thống chỉ dựa vào một máy ghi hình có thể nhận dạng, xác định, truy vết vật thể và tự động xác định tốc độ của các phương tiện giao thông sẽ giúp làm giảm chi phí lắp đặt, vận hành, bảo trì, từ đó giúp tiết kiệm cho xã hội. Nhận thấy tầm quan trọng và ứng dụng thực tế như trên, nhóm chúng tôi đã tiến hành nghiên cứu, hiện thực và đánh giá các mô hình truy vết vật thể và xác định tốc độ phương tiện giao thông trong đề tài.

2 Mục tiêu, đối tượng và phạm vi nghiên cứu

2.1 Mục tiêu của đề tài

Mục tiêu của đề tài là tìm hiểu, nghiên cứu và đánh giá một số phương pháp phát hiện truy vết nhiều vật thể trong dữ liệu video. Qua đó đề tài tạo tiền đề để xây dựng hệ thống ước tính tốc độ giao thông qua video hoặc dữ liệu trực tuyến từ chỉ một camera ghi hình.

2.2 Đối tượng và phạm vi nghiên cứu

Trong phạm vi khóa luận, các nghiên cứu sẽ xoay quanh các kiến thức trong lĩnh vực xử lý ảnh liên quan tới bài toán phát hiện, truy vết và ước tính tốc độ vật thể. Chúng tôi tập trung nhiều hơn vào các phương pháp dựa trên học sâu cho bước phát hiện và truy vết vật thể. Đối với bước ước tính tốc độ, một hướng tiếp cận đang nhận

được nhiều sự quan tâm là tự động hóa toàn bộ quy trình tính tốc độ qua camera sẽ được tìm hiểu.

3 Đóng góp của đề tài

Các đóng góp chính của đề tài bao gồm:

- Tìm hiểu về bài toán ước tính tốc độ dựa trên kỹ thuật truy vết đối tượng.
- Đánh giá một số phương pháp tiêu biểu cho bài toán truy vết đối tượng trên dữ liệu giao thông.
- Đánh giá phương pháp tự động ước tính tốc độ phương tiện giao thông.

4 Cấu trúc luận văn

Phần còn lại của luận văn được tổ chức như sau:

- **Chương 2:** Trình bày, khảo sát các phương pháp và hệ thống truy vết vật thể (Multiple Object Tracking) và ước tính tốc độ giao thông (Speed Estimation).
- **Chương 3:** Trình bày lý thuyết các phương pháp được khảo sát, đặt biệt tập trung các phương pháp mới đang thu hút sự quan tâm của cộng đồng nghiên cứu và cho hiệu quả cao trong những năm trở lại đây.
- **Chương 4:** Trình bày những kết quả đạt được khi đánh giá các mô hình phát hiện, truy vết vật thể và ước tính tốc độ giao thông bằng các độ đo tiêu chuẩn được cộng đồng nghiên cứu sử dụng rộng rãi để đánh giá những mô hình và hệ thống này.
- **Chương 5:** Tổng kết những kết quả đạt được và phân tích ưu, nhược điểm của các phương pháp phát hiện và truy vết vật thể và ước tính tốc độ giao thông
- **Chương 6:** Từ những hạn chế của các phương pháp phát hiện và truy vết vật thể và ước tính tốc độ, nhóm chúng tôi sẽ định hướng những vấn đề cần nghiên cứu cải thiện và kế hoạch phát triển của đề tài trong thời gian sắp tới.

Chương 2

TỔNG QUAN

1 Mô tả bài toán

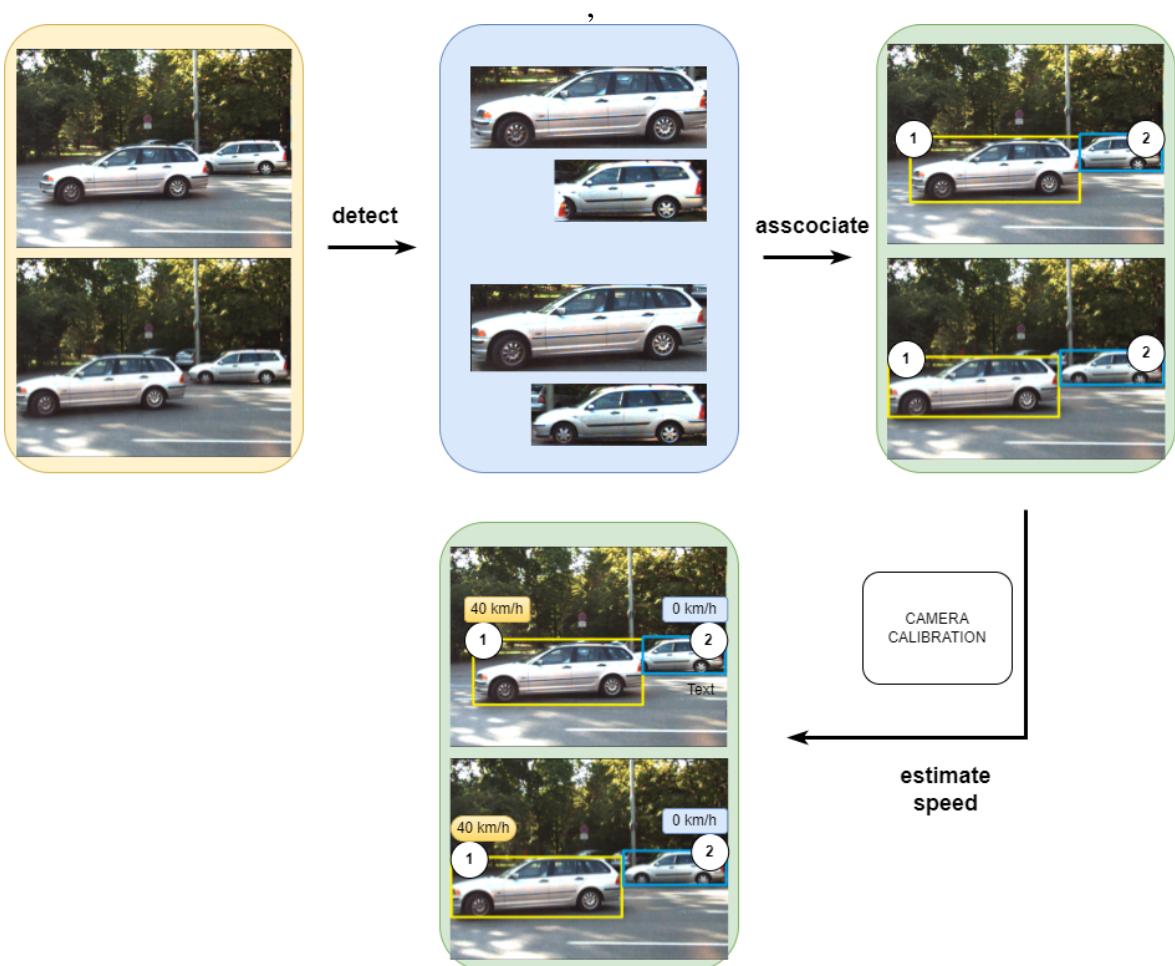
Ước tính tốc độ qua camera giao thông có nhiệm vụ tính toán giá trị ước lượng vận tốc của các phương tiện di chuyển bên trong video. Hệ thống ước tính tốc độ sẽ có đầu vào và đầu ra như sau:

- Đầu vào là dữ liệu dưới dạng file video hoặc video stream
- Đầu ra là định danh và tốc độ ước tính của các phương tiện trong từng frame.

Một hệ thống ước tính tốc độ qua camera hoàn chỉnh gồm có 3 phần chính như minh họa ở hình 2.1

- Phát hiện các đối tượng là phương tiện giao thông trong các frame. Bước này đòi hỏi xác định vị trí và đoi khi là cả loại phương tiện. Vị trí của các đối tượng có thể được biểu diễn bằng nhiều hình thức. Trong đó hình thức biểu diễn vị trí phổ biến nhất là thông qua các hộp giới hạn 2 chiều. Xét riêng với trường hợp ứng dụng cho bài toán tính tốc độ, có một số cách biểu diễn vị trí khác như dựa trên keypoint, biến số. Các cách biểu diễn này sẽ được nói rõ hơn ở phần 2
- Truy vết hay còn gọi là theo dõi các đối tượng. Ở phần này hệ thống cần xác định 2 vật thể trong 2 frame bất kỳ có thuộc cùng một đối tượng hay không. Nhiệm vụ này thường được thực hiện thông qua việc gán một nhãn định danh cho từng vật thể thuộc từng frame. Hai vật thể ở 2 frame khác nhau nhưng có cùng nhãn định danh sẽ được dự đoán là thuộc cùng một đối tượng

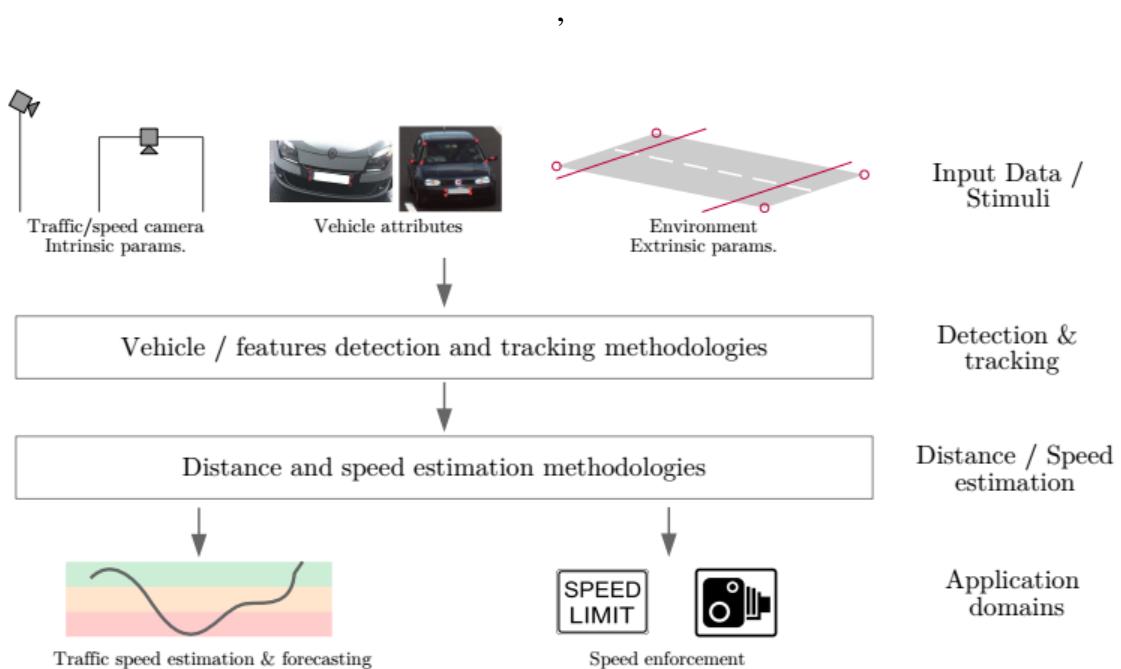
- Ước lượng tốc độ vật thể tự động. Tọa độ ảnh của các vật thể sẽ được chuyển về tọa độ 3 chiều bằng các kỹ thuật xử lý ảnh. Cuối cùng thông tin về tọa độ các vật thể trong từng frame, cũng như các thông tin khác như số FPS, sẽ được sử dụng để ước tính tốc độ di chuyển.



HÌNH 2.1: Minh họa 3 phần chính của bài toán ước tính tốc độ bằng thuật toán truy vết

Một trong các ứng dụng có thể tích hợp ước tính tốc độ là hệ thống dự báo, cảnh báo, xử lý vi phạm tốc độ bằng camera giao thông. Các thành phần chính của một hệ thống như vậy được miêu tả trong hình 2.2. Intrinsic, extrinsic parameter là các thông tin dùng để chuyển đổi tọa độ ảnh và tọa độ 3 chiều. Những thông tin này có thể được đo đặc và cung cấp sẵn khi lắp đặt camera. Tuy nhiên, với sự phát triển của các kỹ thuật xử lý ảnh, công đoạn tính toán các thông số này đã có thể được xử lý tự động.

Ngoài ra, còn có một số thành phần khác như hệ thống nhận diện biển số xe, cơ chế cảnh báo, xử phạt khi tốc độ của phương tiện vượt mức quy định, ...

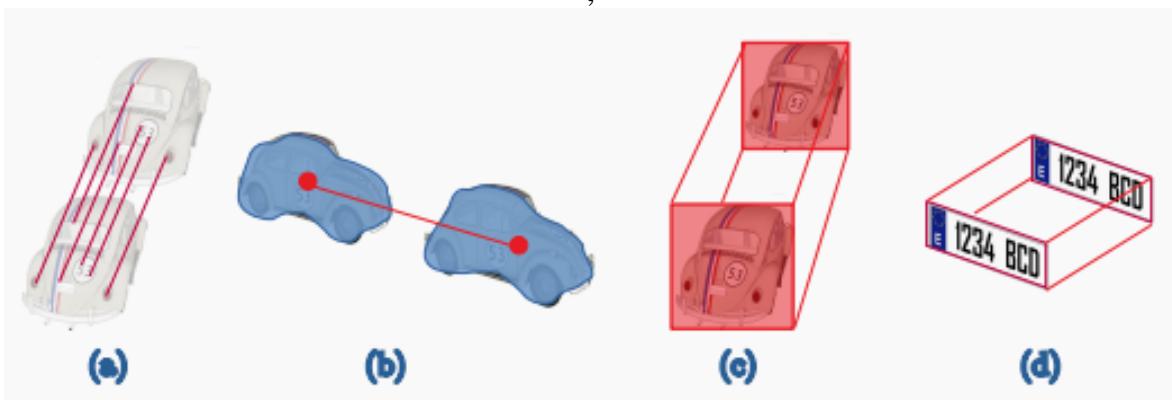


HÌNH 2.2: Các thành phần chính của hệ thống cảnh báo và xử phạt vượt quá tốc độ qua camera: Thông tin dữ liệu đầu vào, phát hiện và truy vết vật thể, ước tính khoảng cách và tốc độ, cảnh báo và xử phạt khi tốc độ vượt mức quy định [11]

2 Các hướng nghiên cứu hiện nay

Bài báo khảo sát gần đây [11] đã chỉ ra rằng có nhiều hơn 1 cách để biểu diễn vị trí phương tiện nhằm truy vết và tính tốc độ. Một số phương thức biểu diễn chính có thể liệt kê ra như sau:

- Các điểm đặc trưng: Đây là các vị trí nhất định thuộc vùng ảnh có chứa phương tiện. Các điểm đặc trưng nằm ở các vị trí đặc biệt, dễ dàng nhận biết và có thể sử dụng để phân biệt các vật thể khác nhau.
- Điểm nằm trên biển số xe
- Điểm tâm vùng bao của vật thể hoặc 1 điểm đặc biệt (tâm, góc) của hộp giới hạn



HÌNH 2.3: Các cách biểu diễn vị trí của phương tiện. (a)Các điểm đặc trưng. (b)Tâm vùng bao. (c) Hộp giới hạn. (d)Biển số[11]

Trong khóa luận này, chúng tôi tập trung nghiên cứu phương thức biểu diễn thông qua hộp giới hạn. Các phương pháp phát hiện và truy vết vật thể được khảo sát đều sử dụng cách biểu diễn vị trí này.

2.1 Khảo sát các phương pháp phát hiện vật thể

Hiện nay các phương pháp phát hiện vật thể bằng hộp giới hạn có thể được phân loại theo 2 cách chính: phương pháp 1 pha và 2 pha, phương pháp có sử dụng hộp anchor và phương pháp không sử dụng hộp anchor. Dưới đây sẽ trình bày sơ lược một số họ phương pháp nổi bật

- Họ phương pháp YOLO - You Only Look Once[20][21][22][23][24]: Đây là một nhóm các phương pháp 1 pha sử dụng hộp anchor. Kết quả tọa độ tâm vật thể, độ lệch tâm, nhãn phân lớp được dự đoán qua một mô hình mạng học sâu duy nhất. Vì các tác vụ được thực hiện trong chỉ một mạng, YOLO có thể tận dụng tốt khả năng tính toán song song của GPU và có tốc độ cao.
- Họ phương pháp R-CNN[25][26][27]: Các phương pháp thuộc nhóm này dự đoán thông qua mô hình 2 pha sử dụng hộp anchor. Khác với phương pháp 1 pha, họ RCNN dự đoán độ lệch tâm và nhãn phân loại tại pha thứ 2. Pha thứ nhất chịu trách nhiệm tìm ra các vùng có khả năng xuất hiện vật thể. Việc tách biệt các tác vụ trong 2 pha riêng biệt giúp nhóm phương pháp này đạt độ chính xác cao (đặc biệt là khi các vật nằm gần nhau và có kích thước nhỏ), nhưng đồng thời cũng hạn chế tốc độ thực thi.

-
- RetinaNet[28] và SSD[29] là những phương pháp 1 pha vẫn đang được sử dụng rộng rãi. Đặc biệt trong bài báo RetinaNet đã đề xuất hàm mất mát Focal giúp giải quyết vấn đề mất cân bằng dữ liệu trong các phương pháp 1 pha.
 - CenterNet[30] là một mạng phát hiện đối tượng có thiết kế đơn giản nhưng lại đạt được cân bằng tốt giữa tốc độ và độ chính xác. Thay vì dùng hộp anchor làm trung gian đánh giá kết quả, CenterNet so sánh kết quả dự đoán với nhãn dữ liệu bằng cách sử dụng bản đồ nhiệt.
 - Họ EfficientDet[31] kế thừa thành công của mạng backbone EfficientNet[32][33]. Các mạng backbone mới này được sử dụng để thay thế các mạng cũ như Resnet[14], MobileNet[34],... trong kiến trúc mạng phát hiện đối tượng
 - DETR[35] là 1 trong những phương pháp đầu tiên áp dụng mô hình Transformer[36] cho bài toán phát hiện đối tượng.

2.2 Khảo sát các phương pháp truy vết vật thể

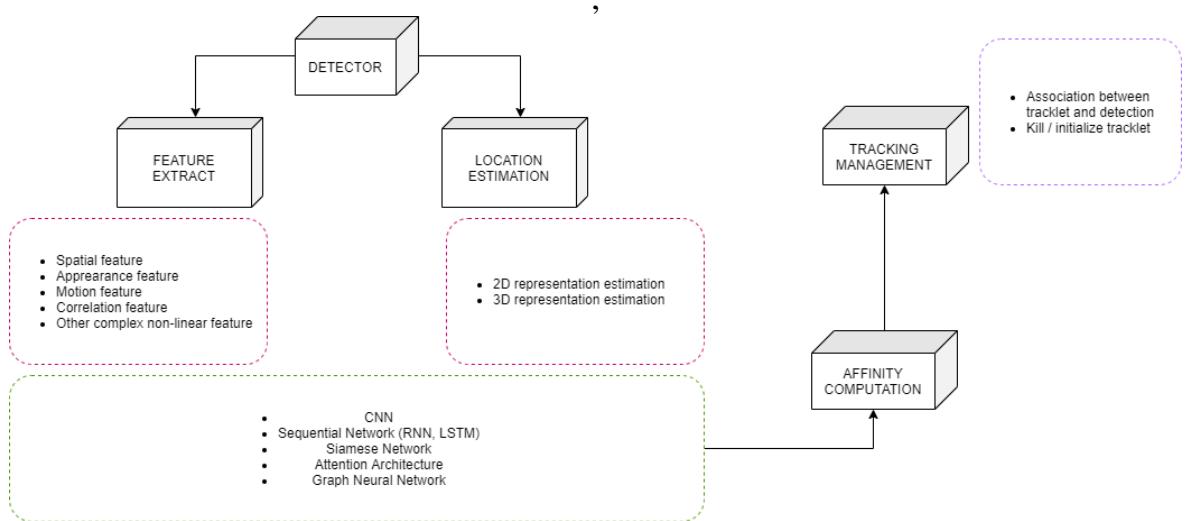
Truy vết vật thể bao gồm 2 nhánh chính:

- Truy vết đơn vật thể: tập trung vào việc theo dõi một đối tượng duy nhất trong toàn bộ video.
- Truy vết đa vật thể phát hiện đồng thời theo dõi tất cả các đối tượng trong các khung hình, kể cả các đối tượng mới xuất hiện. Đây là nhánh có thể ứng dụng trong bài toán ước tính tốc độ nên khóa luận sẽ khảo sát và tìm hiểu các phương pháp thuộc nhánh này.

Hình 2.4 dưới đây mô tả các bước chính của bài toán truy vết đa đối tượng. Ta có thể xem đây như một phần mở rộng của phần phát hiện đối tượng khi bên cạnh thông tin về vị trí, phương pháp truy vết truy vết cần gán một nhãn định danh cho mỗi đối tượng.

Phần trích xuất đặc trưng sẽ trích xuất các thông tin đặc trưng nhằm phục vụ cho việc định danh về sau. Với sự phát triển của các thuật toán truy vết trong thời gian gần đây, rất nhiều loại thông tin trùu tượng đã được đề xuất khai thác. Ta có thể liệt kê các loại thông tin này như: thông tin vị trí, kích cỡ (spatial feature), thông tin trực quan (appearance feature), thông tin chuyển động (motion feature), thông tin tương quan (correlation feature), và một số loại thông tin khác. **Phần dự đoán chuyển động** ước

lượng vị trí tại frame kế tiếp của mỗi vật thể. **Phần tính toán độ liên quan** tính mức độ giống nhau giữa từng cặp đối tượng ở các frame liên tiếp. **Phần truy vết** có thể dựa trên kết quả mức độ giống nhau được tính ở bước trước để tiến hành liên kết các đối tượng ở các frame khác biệt



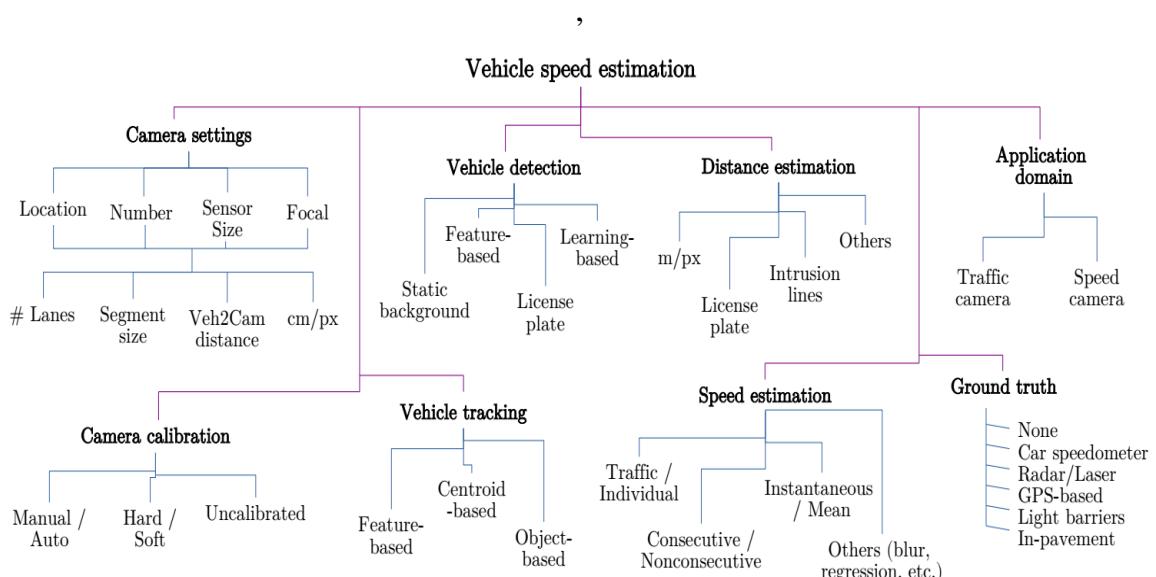
HÌNH 2.4: Các bước chính của phần truy vết đối tượng

Hầu hết các phương pháp truy vết gần đây đều nghiên cứu cách ứng dụng các kỹ thuật học sâu. Các bài báo khảo sát đề xuất nhiều cách để phân loại phương pháp truy vết. Ta có thể chia thành 3 nhóm:

- Các phương pháp tách biệt phát hiện và truy vết đối tượng - có tên tiếng anh là Separated Detection and Tracking(SDE): Các phương pháp này sử dụng kết quả từ các phương pháp như YOLO, RCNN. Nói cách khác, phần phát hiện và truy vết được tách biệt riêng rẽ.
- Các phương pháp tích hợp phát hiện và trích xuất đặc trưng, dự đoán chuyển động trong một mạng học sâu. Bước liên kết các đối tượng trong các frame bằng một kỹ thuật tối ưu tùy thuộc vào mỗi phương pháp
- Các mô hình tích hợp toàn bộ quá trình truy vết từ bước phát hiện đối tượng, trích đặc trưng, dự đoán chuyển động, liên kết đối tượng trong một mạng duy nhất.

2.3 Khảo sát các phương pháp ước tính tốc độ phương tiện giao thông

Một phương pháp thường được hiện nay là sử dụng súng bắn tốc độ (speed gun) dựa trên nguyên lý hoạt động như các thiết bị LIDAR hoặc RADAR. Các tia sáng hoặc hạt nguyên tử sẽ được phát ra, va đập vào phương tiện và phản xạ tới máy thu để ước tính quãng đường và thời gian. Nhược điểm của hệ thống này là chi phí lắp đặt súng bắn tốc độ và bảo trì hệ thống lớn. Với chi phí sản xuất và lắp đặt camera giám sát ngày càng giảm, việc chỉ sử dụng một camera giám sát để ước tính chính xác tốc độ phương tiện giao thông ngày càng được quan tâm nghiên cứu. Vì vậy trong phần này nhóm chúng tôi sẽ khảo sát các phương pháp ước tính tốc độ phương tiện dựa trên hệ thống chỉ một camera giám sát.



HÌNH 2.5: Sơ đồ các khía cạnh của bài toán ước tính tốc độ giao thông hiện nay [11]

Để ước tính được tốc độ các phương tiện, các hệ thống giám sát qua camera thường dựa trên ba bài toán nhỏ là: Phát hiện, truy vết vật thể - cụ thể ở đây là các phương tiện giao thông - và ước tính khoảng cách, từ đó ước tính tốc độ các phương tiện giao thông. Sơ đồ tổng quan của bài toán, các bài toán con và các nhóm phương pháp tương ứng được minh họa ở hình 2.5. Bài toán phát hiện và truy vết vật thể đã được khảo sát ở phần trước thì phần này chúng tôi sẽ khảo sát các yếu tố ảnh hưởng quá trình

ước tính tốc độ phương tiện giao thông và các phương pháp ước tính khoảng cách di chuyển cùng tốc độ phương tiện giao thông.

Các yếu tố ảnh hưởng đến bước ước tính tốc độ:

- Chiều cao của điểm đặt camera: hiện nay tùy thuộc vào mục đích của từng hệ thống mà có chiều cao đặt camera khác nhau. Ví dụ đối với hệ thống camera đặt trên các máy bay tự lái để giám sát giao thông có độ cao lớn. Ta có thể chia các loại hệ thống theo chiều cao như sau: hệ thống các camera giám sát giao thông với điểm đặt cao từ 5 mét trở lên, các hệ thống có có điểm đặt thấp nhỏ hơn 5 mét và cuối cùng là các hệ thống có camera sát mặt đường.
- Độ phân giải của hình ảnh thu được, cấu hình vật lý của camera với độ phân giải thấp nhất là 640 x 480 pixel (VGA). Độ phân giải sẽ ảnh hưởng lớn đến việc ước tính khoảng cách và tốc độ . Bởi lẽ việc chuyển từ các pixel ảnh về kích thước thực tế sẽ gấp khó khăn với tốc độ di chuyển của phương tiện rất nhanh mà độ phân giải thấp sẽ làm vật thể bị mờ gây lỗi trong việc xác định vị trí của phương tiện.
- Tiêu cự của camera, thông thường các camera đang được lắp đặt trên các hệ thống có tiêu cự nhỏ hơn 25 mi-li-mét, nhưng nghiên cứu [11] đã cho thấy rằng tiêu cự càng lớn thì sai số tốc độ và khoảng cách càng thấp. Do đó với các hệ thống camera được lắp đặt khác nhau sẽ ảnh hưởng đến đến việc ước tính khoảng cách và tốc độ. Vì vậy một hệ thống camera đạt yêu cầu để các thuật toán ước tính tốc độ phương tiện có sai số trong mức cho phép cần được xem xét kĩ lưỡng để giảm thiểu chi phí lắp đặt.
- Kỹ thuật ước tính (hiệu chỉnh) các thông số camera (Camera calibration): Việc tính toán chính xác các phép đo trong thế giới thực qua các tọa độ pixel trong hình ảnh phụ được gọi là hiệu chỉnh các thông số camera. Nói một cách chi tiết hơn thì đây là quá trình thông qua các mối quan hệ và phép tính để chuyển tọa độ 2D của ảnh (u, v) về tọa độ 3D trên thực tế (x_w, y_w, z_w) .

Kỹ thuật hiệu chỉnh thông số camera:

Để có thể ước tính tốc độ từ kết quả truy vết theo tọa độ ảnh, ta cần chuyển các tọa độ ảnh này về tọa độ thực. Thao tác này được thực hiện qua việc tính toán các tham số hiệu chỉnh camera. Kỹ thuật hiệu chỉnh camera được sử dụng sẽ quyết định liệu hệ thống có thể thực thi hoàn toàn tự động và ở góc nhìn bất kỳ hay không. Đây là

hai yêu cầu quan trọng để có thể triển khai hệ thống ở quy mô lớn. Các kỹ thuật hiệu chỉnh camera trong trường hợp có 1 camera tĩnh có thể chia thành 2 nhóm chính[10]:

- Tính các intrinsic parameter và extrinsic parameter dựa trên một số giá trị đo đạc thực tế. Những giá trị này có thể là khoảng cách thực tế giữa các cặp điểm mốc, kích thước một số vật thể, ...
- Tính các intrinsic parameter và extrinsic parameter thông qua các vanishing point (sẽ được nói rõ hơn ở phần 2.2). Các phương pháp này thường giúp loại bỏ bớt nhu cầu đo đạc bằng sức người và tiến gần hơn đến mục tiêu tự động hóa hoàn toàn quá trình hiệu chỉnh camera.

Trong khóa luận này, chúng tôi hướng tới tìm hiểu một phương pháp ước tính khoảng cách cụ thể thuộc nhóm phương pháp thứ hai.

Các phương pháp ước tính tốc độ bằng một máy quay giám sát:

- Ước tính tốc độ dựa trên khoảng cách mà phương tiện di chuyển và thời gian ghi hình của máy quay ở hai khung hình t và $t + 1$
- Ước tính tốc độ dựa trên khoảng cách và thời gian mà phương tiện di chuyển qua các đường ảo (virtual line) mô tả vùng được quan tâm. Thuật toán tính số khung hình để phương tiện di chuyển hết khoảng cách giữa hai đường ảo đó và tính trung bình cộng các tốc độ được ước tính giữa các đường ảo.

3 Thách thức của bài toán

Về bài toán

Một số vấn đề có thể phát sinh với các phương pháp phát hiện và truy vết vật thể trong video:

- Mô hình có thể sẽ phát hiện thiếu vật thể trong trường hợp các đối tượng bị che lấp hoặc video thu được trong trạng thái thời tiết xấu, ánh sáng kém, ...
- Nếu một phương tiện có tốc độ di chuyển cao, đổi hướng đột ngột hoặc vì lý do máy ghi hình có độ phân giải thấp, trường hợp này có thể bị hiểu nhầm là phương tiện này đã ra khỏi khung hình. Ở lần xuất hiện tiếp theo, phương tiện này sẽ bị gán nhầm một nhãn định danh mới.

-
- Việc phân biệt các vật thể có các đặc trưng trực quan gần giống nhau có thể gây khó khăn cho thuật toán truy vết.

Một số vấn đề đặt ra đối với bài toán ước tính tốc độ phương tiện giao thông qua dữ liệu được ghi từ camera giám sát giao thông:

- Làm thế nào hệ thống có thể được triển khai ở nhiều điều kiện môi trường, đặc biệt trong điều kiện thời gian và điều kiện thời tiết đặc biệt như buổi tối, trời mưa,... và góc đặt camera khác nhau.
- Làm thế nào để xây dựng hệ thống ước tính tốc độ giao thông tiêu tốn ít tài nguyên nhất có thể để tích hợp hệ thống lên các thiết bị IoT?
- Làm cách nào để ước tính tốc độ với sai số thấp nhất có thể đáp ứng yêu cầu thực tế?

Về phương pháp

Hiện nay, cộng đồng nghiên cứu đã đề xuất rất nhiều phương pháp với những ý tưởng cải tiến nhằm giải quyết các khó khăn của bài toán này. Thêm vào đó, bài toán ước tính tốc độ được chia ra làm 3 phần chính, kết quả nhận được của từng phần sau khi kết hợp lại với nhau sẽ ảnh hưởng đến kết quả tổng quát. Việc lựa chọn thuật toán và phương pháp phù hợp trong số rất nhiều phương pháp cho mỗi phần sẽ cần được lựa chọn cẩn trọng.

Ở bước truy vết vật thể, các phương pháp hiện tại chủ yếu được đề xuất để xử lý dữ liệu video người đi bộ. Liệu các phương pháp này có thể hoạt động tốt trên một loại dữ liệu đầy tính thách thức khác như dữ liệu video về các phương tiện giao thông? Chủ đề này vẫn chưa thu hút được nhiều sự quan tâm của các nghiên cứu từ trước đến nay.

Ở bước ước tính tốc độ, các thông số hiệu chỉnh camera thường được tính bằng cách sử dụng các đặc điểm hình học tĩnh ở trên mặt phẳng đường; sau đó tính toán phép biến đổi đồng nhất bao gồm quay, dịch và tỷ lệ. Kích thước, khoảng cách, chiều dài,... của các đặc điểm tĩnh này phải được cung cấp trước bằng cách đo đạc thủ công trực tiếp hoặc gián tiếp bằng bằng việc sử dụng máy quét lazer hoặc hệ thống định vị của Google Map. Việc này có thể dẫn đến phải tạm dừng lưu thông đường, gây cản trở, mất thời gian và công sức.

Về dữ liệu

Chúng ta vẫn chưa có thực sự nhiều các tập dữ liệu giao thông được kiểm tra kỹ lưỡng và được sử dụng rộng rãi. Ngoài ra, các tập dữ liệu có sẵn hầu hết đều chỉ được sử dụng để đánh giá một phần của bài toán. Có những tập dữ liệu chuyên dùng để đánh giá phần phát hiện và truy vết vật thể. Ngoài ra cũng có những tập dữ liệu chuyên dùng để đánh giá phần ước tính tốc độ. Đặc biệt tập dữ liệu có chứa dữ liệu chính xác của các phương tiện để đánh giá tốc độ giao thông được xây dựng và công bố rất ít, theo tác giả David và các cộng sự [11], hiện nay chỉ có hai bộ dữ liệu được công bố có thể đánh giá cho bài toán ước tính tốc độ giao thông. Bộ dữ liệu thứ nhất dựa trên biển số phương tiện để phát hiện, truy vết và ước tính các thông số camera. Vì vậy bộ dữ liệu này không phù hợp cho các phương pháp truy vết vật thể được nhóm sử dụng. Chỉ có bộ dữ liệu BronoCompSpeed [10] là bộ dữ liệu duy nhất hiện nay có thể sử dụng để ước tính tốc độ giao thông dựa trên các phương pháp phát hiện và truy vết vật thể được nghiên cứu gần đây đã được đề cập đến ở mục 2.2.

4 Vấn đề nghiên cứu

Khóa luận tập trung giải quyết 2 vấn đề chính:

- Hiện có rất nhiều các phương pháp truy vết đối tượng. Nhưng các phương pháp này chưa được đánh giá đầy đủ trên dữ liệu giao thông. Chúng tôi sẽ thực hiện tìm hiểu và đánh giá một số phương pháp nổi bật trên tập dữ liệu UA-DETRAC (sẽ được miêu tả ở phần 1.1)
- Chưa có nhiều các nghiên cứu tìm hiểu các phương pháp truy vết tiên tiến gần đây cho bài toán ước tính tốc độ. Từ kết quả đánh giá phương pháp truy vết ở phần trước, chúng tôi thử nghiệm áp dụng một phương pháp truy vết có áp dụng các tiên bộ của kỹ thuật học sâu gần đây. Chúng tôi cũng tiến hành so sánh kết quả này với việc sử dụng các phương pháp truy vết kiểu cũ đã được áp dụng trên các hệ thống ước tính tốc độ đã có.

Chương 3

CƠ SỞ LÝ THUYẾT

1 Truy vết đa vật thể

1.1 Các kiến thức cơ sở

Ở mục này nhóm chúng tôi sẽ trình bày các giải thuật và bộ lọc được sử dụng rộng rãi như một phần của nhiều phương pháp truy vết.

1.1.1 Giải thuật Hungary

Giải thuật Hungarian[37] (Hungarian algorithm) là một thuật toán tối ưu hóa tổ hợp để giải quyết bài toán phân chia công việc (assignment problem), ghép cặp (Biprivate Matching) trong đồ thị với thời gian đa thức, được phát triển và công bố năm 1955 bởi Harold Kuhn.

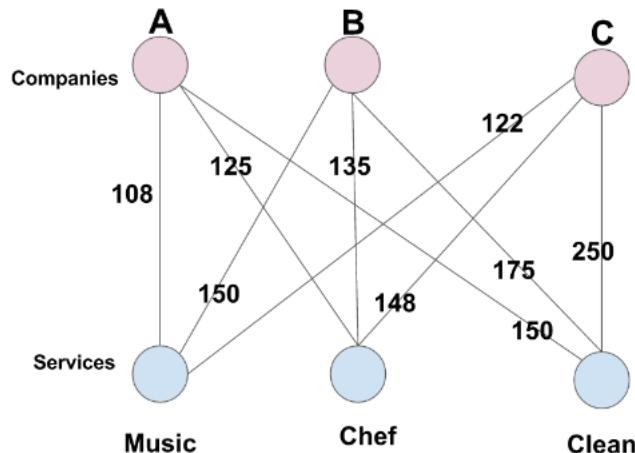
Phát biểu bài toán phân chia công việc: Có n người ($i = 1, 2, \dots, n$) và m công việc ($j = 1, 2, \dots, m$). Để giao cho người i thực hiện một công việc j cần một chi phí c_{ij} . Yêu cầu của bài toán là tìm cách giao cho mỗi người duy nhất một công việc sao cho chi phí bỏ ra tổng cộng là nhỏ nhất. **Liên hệ đến bài toán trên với bài toán đặt ra trong việc truy vết nhiều đối tượng từ khung hình thứ t và $t - 1$:** Có n đối tượng ($i = 1, 2, \dots, n$) ở khung hình thứ t và m đối tượng đã được dự đoán nhãn truy vết ở khung hình thứ $t - s$ tới $t - 1$ ($j = 1, 2, \dots, m, t - s \geq 0$). Để liên kết một đối tượng đã được phát hiện i ở khung hình t với một đối tượng đã dự đoán ở s khung hình trước đó, giả sử ta sử dụng một độ đo tương đồng D (Cosine Similarity, F1, F2,...) để đo khoảng cách giữa hai véc-tơ biểu diễn cho i và j trong không gian véc-tơ. Bài toán

đặt ra là cần liên kết một đối tượng i với một đối tượng j đã được phát hiện và gán nhãn truy vết trước đó, tương ứng sao cho sai số của phép đo tương đồng D giữa i và j là nhỏ nhất dựa trên độ đo tương đồng D . Ta thấy bài toán phát biểu trên có tương tự với một bài toán phân chia công việc và có thể áp dụng thuật giải Hungary để tìm đối tượng i ở khung hình t giống nhất với đối tượng đã được truy vết ở s khung hình trước đó.

Để trình bày thuật toán, ví dụ ta có bài toán một bữa tiệc muốn thuê một nhạc công biểu diễn, một đầu bếp chuẩn bị thức ăn và một dịch vụ dọn dẹp giúp dọn dẹp sau bữa tiệc. Có ba công ty cung cấp ba dịch vụ này, nhưng một công ty chỉ có thể cung cấp một dịch vụ tại một thời điểm (tức là Công ty B không thể cung cấp cả người dọn dẹp và đầu bếp). Ta đang quyết định mua từng dịch vụ của công ty nào để giảm thiểu chi phí cho bữa tiệc.

Company	Cost for Musician	Cost for Chef	Cost for Cleaners
Company A	\$108	\$125	\$150
Company B	\$150	\$135	\$175
Company C	\$122	\$148	\$250

Từ mô tả bài toán ta có đồ thị cặp ghép như hình dưới:



HÌNH 3.1: Đồ thị cặp ghép minh họa cho bài toán phân chia công việc

Để giải quyết bài toán trên ta quy bài toán về bài toán phân chia công việc. Ta cần mô hình hóa bài toán thành ma trận kề (Adjacency Matrix) tương ứng với bảng giá thuê dịch vụ tương ứng của mỗi công ty đã cho ở trên. Ma trận $C(n,m)$ có n hàng

tương ứng với n công ty và m cột tương ứng với m dịch vụ (công việc). Mỗi giá trị trong ma trận $C[i, j]$ ($1 \leq i \leq n, 1 \leq j \leq m$) là chi phí khi thuê dịch vụ j của công ty i ($C[i, j] \geq 0$).

Từ điều kiện của bài toán ta rút ra nhận xét: Giả sử ma trận chi phí của bài toán giao việc là không âm. Nếu ta có thể đưa một phần tử $C[i, j] = 0$ bằng cách cộng hoặc trừ một số $\alpha \neq 0$ vào hàng i hoặc cột j . Thì cách phân chia n công việc tương ứng với mỗi giá trị $C[i, j] = 0$ là cách phân chia công việc tối ưu của ma trận ban đầu.

Dựa vào hai nhận xét trên ta có các bước của thuật toán Hungarian để giải quyết bài toán phân chia công việc trên như sau:

- Bước 1: Trừ giá trị nhỏ nhất trong mỗi hàng với tất cả các giá trị khác trong hàng. Điều này sẽ làm cho giá trị $C[i, j] = 0$ nhỏ nhất trong hàng bây giờ bằng 0.
- Bước 2: Trừ giá nhỏ nhất trong mỗi cột khỏi tất cả các giá trị khác trong cột. Điều này sẽ làm cho giá trị nhỏ nhất trong cột bây giờ bằng 0.
- Bước 3: Vẽ các đường qua hàng và cột có giá trị $C[i, j] = 0$ sao cho vẽ được ít dòng nhất có thể.
- Bước 4: Nếu có n dòng được vẽ, các giá trị 0 là cách phân công việc của n người cho m công việc và thuật toán đã kết thúc. Nếu số dòng nhỏ hơn n thì chưa tìm được lời giải tối ưu và đến bước tiếp theo.
- Bước 5: Tìm giá trị $C[i, j]$ nhỏ nhất không bị kẻ bởi bất kỳ đường nào. Trừ giá trị này khỏi mỗi hàng chưa bị gạch bỏ, rồi cộng thêm giá trị đó vào mỗi cột đã bị gạch bỏ. Sau đó, quay lại Bước 3

Áp dụng giải thuật Hungarian cho bài toán trên để tìm lời giải:

Ma trận ban đầu:

108	125	150
150	135	175
122	148	250

Bước 1: Trừ giá trị nhỏ nhất trong mỗi hàng với các giá trị khác cùng hàng tương ứng:

0	17	42
15	0	40
0	26	128

Bước 2: Trừ giá trị nhỏ nhất trong mỗi cột với tất cả các giá trị khác cùng cột tương ứng:

0	17	2
15	0	0
0	26	88

Bước 3: Vẽ các đường qua hàng và cột có giá trị 0 sao cho số dòng ít dòng nhất có thể (chọn hàng và cột có nhiều giá trị 0 nhất):

0	17	2
15	0	0
0	26	88

Bước 4: Có 2 đường kẻ được vẽ và 2 nhỏ hơn $n = 3$ nên chưa có lời giải tối ưu.

Bước 5: Tìm giá trị nhỏ nhất chưa bị gạch bởi bất kỳ dòng nào. Trừ giá trị này khỏi mỗi hàng chưa gạch bỏ.

-2	15	0
15	0	0
-2	24	86

Cộng thêm nó vào mỗi cột đã bị gạch bỏ. Sau đó, quay lại Bước 3.

0	15	0
17	0	0
0	24	86

Bước 3: Vẽ các đường qua hàng và cột có giá trị 0 sao cho số dòng ít nhất có thể:

0	15	0
17	0	0
0	24	86

Bước 4: Có 3 dòng và cột có giá trị 0 đã được kẻ nên ta đã tìm được lời giải tối ưu.
Kết thúc thuật toán.

0	15	0
17	0	0
0	24	86

Lời giải tối ưu:

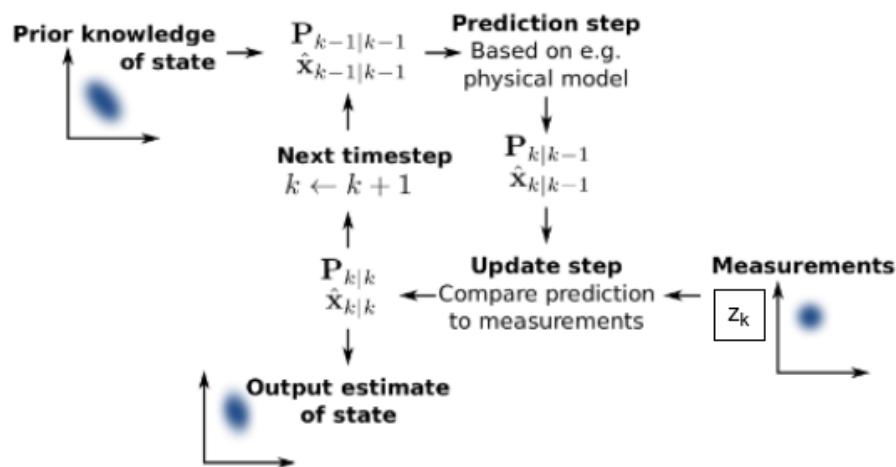
108	125	150
150	135	175
122	148	250

1.1.2 Bộ lọc Kalman

Bộ lọc Kalman (Kalman Filter) là một mô hình Linear-Gaussian State Space Model, được giới thiệu lần đầu năm 1960 và được ứng dụng giải quyết các bài toán trong nhiều

lĩnh vực như: Thống kê, điều kiểm tối ưu, xe tự lái, thực tế ảo và đặc biệt là trong bài toán truy vết vật thể trong video.

Trong bài toán truy vết nhiều vật thể (Multiple Object Tracking), bộ lọc Kalman [38] (Kalman Filter) được sử dụng để dự đoán các trạng thái của đối tượng hiện tại. Quá trình tính toán này dựa vào tập các đối tượng đã được truy vết trong quá khứ và cập nhật lại các nhãn vị trí (bounding box), véc-tơ đặc trưng sau khi đã được thuật toán Hungarian (mục 1.1.1) liên kết với các tập đối tượng đã được gán nhãn truy vết trước đó.



HÌNH 3.2: Quá trình dự đoán và cập nhật của mô hình xác suất của bộ lọc Kalman

Tiếp theo là mô tả toán học của bộ lọc Kalman. Để hiểu rõ cách hoạt động của bộ lọc này trong bài toán truy vết nhiều vật thể (Multiple Object Tracking), chúng tôi sẽ mô tả bốn phương trình xử lý đại diện cho hai giai đoạn của Kalman Filter trong bài toán truy vết nhiều vật thể (Multiple Object Tracking) là dự đoán trạng thái và cập nhật sự thay đổi trạng thái của vật thể:

- **Phương trình xử lý (Process equation):** Trong hệ thống truy vết vật thể gọi x_k là véc-tơ trạng thái biểu diễn chuyển động của vật thể, với k là tập các thời điểm rời rạc. Mục tiêu của bài toán là dự đoán thay đổi trạng thái x_k của vật thể do chuyển động qua việc đo lường z_k . Ta có công thức xác định x_k tại thời điểm k là:

$$x_k = \mathbf{A}x_{k-1} + w_{k-1} (x_k \in R^n) \quad (3.1)$$

Trong đó ma trận chuyển tuyến (Markov) \mathbf{A} là một ma trận vuông mô tả các xác suất chuyển từ trạng thái này sang trạng thái khác trong một hệ thống động, còn có thể kí hiệu là $\mathbf{A}(x_k|x_{k-1})$ với \mathbf{A} là ma trận, x_k là trạng thái tại thời điểm k và x_{k-1} là trạng thái từ thời điểm $k-1$. Véc-tơ w_{k-1} là nhiễu của Gausian process [39] theo xác xuất phân phối chuẩn $p(w), p(w) \sim N(0, Q)$. Và x_k tuân theo xác suất phân phối chuẩn $p(x_k)$ (xem k là giá trị rời rạc). Ta có $p(x_k) \sim N(\hat{x}_k, \delta)$

Hàm mật độ xác suất:

$$p(x_k) = \frac{1}{\delta\sqrt{2\pi}} \exp -\frac{(x_k - \hat{x}_k)^2}{2\delta^2}$$

- **Phương trình đo lường (Measurement equation):**

$$z_k = \mathbf{H}x_k + v_k (z_k \in R^m) \quad (3.2)$$

Với \mathbf{H} là ma trận đo lường, z_k là giá trị đo lường nhận được từ thời điểm $k-1$ đến k tương ứng và v_k là nhiễu của phép đo Gausian (Gausian measurement) tuân theo phân phối chuẩn $p(v), p(v) \sim N(0, R)$

- **Các phương trình cập nhật thời gian (Time update equations):** Công thức (3.1) và (3.2) mô tả một mô hình tuyến tính ở thời điểm k , do đó giá trị z_k có được từ sự đo lường thông tin được sử dụng để cập nhật những trạng thái chưa biết của x_k . Để dự đoán các giá trị trạng thái trong không gian phân phối Gaussian, ta dự đoán các giá trị trạng thái của biến cố ngẫu nhiên trong không gian xác suất của phân phối chuẩn bằng cách sử dụng giá trị kỳ vọng biểu diễn cho xác suất tiên nghiệm \hat{x}_k^{pos} của biến cố và ma trận hiệp phương sai P_k^{pri} với biến ngẫu nhiên, ta có:

$$\hat{x}_k^{pri} = \mathbb{E}[x_k] = \mathbf{A}_{k-1}\hat{x}_{k-1} + E[w_{k-1}] \quad (3.3)$$

$$P_k^{pri} = Var(x_k) = \mathbf{A}_{k-1}P_{k-1}\mathbf{A}_{k-1}^\top + Q_{k-1} \quad (3.4)$$

Vì vậy trong bài toán truy vết vật thể: \hat{x}_k^{pri} và P_k^{pri} tương ứng dùng để dự đoán đặc trưng trạng thái của vật thể tại thời điểm hiện tại và ước tính ma trận hiệp phương sai cho giai đoạn cập nhật trạng thái.

- **Các phương trình đo lường cập nhật thời gian (Measurement update equations):** Các phương trình dưới đây được tính toán khi nhận thông tin thay đổi của hệ thống

(ở bài toán truy vết vật thể là thông tin liên kết giữa ví trí vật thể tại thời điểm k và các đối tượng đã được truy vết ở những khung hình trước đó). Với mục tiêu ước tính xác suất hậu nghiệm x_k bằng tổ hợp xác suất tiên nghiệm đã được được tính ở công thức (3.3), (3.4) và giá trị đo lường mới z_k tại thời điểm k . Áp dụng định lý Bayes cho không gian xác suất Gaussian, ta có:

$$P_k^{pos} = (P_k^{pri-1} + \mathbf{H}_k^\top R_k^{-1} \mathbf{H})^{-1}$$

$$\hat{x}_k^{pos} = (P_k^{pri-1} + \mathbf{H}_k^\top R_k^{-1} \mathbf{H})^{-1} [\mathbf{H}_k^\top R_k^{-1} (z_k - \mathbf{H}_k) \hat{x}_k^{pri}] + P_k^{pri-1} \hat{x}_k^{pri}$$

Gọi K_k là hệ số Kalman tại thời điểm k , ta đặt K_k theo công thức dưới:

$$K_k = P_k \mathbf{H}^\top (\mathbf{H}_k P_k^{pri} \mathbf{H}_k^\top + R_k)^{-1} \quad (3.5)$$

Áp dụng đồng nhất thức ma trận Woodbury (Matrix Inversion Lemma) vào phương trình P_k^{pos} và \hat{x}_k^{pos} ở trên, khai triển và rút gọn ta được 2 công thức như sau:

$$\hat{x}_k^{pos} = \hat{x}_k^{pri} + K_k (z_k - \mathbf{H}_k \hat{x}_k^{pri}) \quad (3.6)$$

$$P_k^{pos} = (1 - K_k \mathbf{H}_k) P_k^{pri} \quad (3.7)$$

Ta thấy xác suất hậu nghiệm x_k và P_k được tính bởi giá trị đo lường z_k . Phương trình cập nhật và đo lường thời gian cho phép đệ quy sử dụng xác suất hậu nghiệm của thời điểm hiện tại để ước tính xác xuất tiên nghiệm mới ở bước kế tiếp.

Do đó trong ở giai đoạn cập nhật (đo lường) đặc trưng trạng thái các vật thể sau khi được liên kết quỹ đạo của bài toán truy vết vật thể: K_k dùng để cập nhật hệ số Kalman, \hat{x}_k^{pos} và P_k^{pos} giúp cập nhật thay đổi trạng thái của vật thể ở thời điểm hiện tại, để chuẩn bị dự đoán cho đặc trưng trạng thái cho những đối tượng ở khung hình kế tiếp.

Ứng dụng cụ thể của bộ lọc Kalman trong bài toán truy vết vật thể: Như đã đề cập ở phần đầu bộ lọc Kalman sử dụng để dự đoán các đặc trưng hình học của vật thể từ những đối tượng đã được truy vết trước đó như vị trí hộp giới hạn, hình dạng, tâm của vật thể,... nhằm giúp thuật toán Hungary liên kết vật thể được phát hiện trong khung hình hiện tại với kết quả liên kết chính xác hơn. Sau đó những thay đổi về các đặc trưng hình học tại khung hình hiện tại sẽ được bộ lọc Kalman cập nhật vào ma

trận **A** và ma trận **H** nhằm dự đoán các đặc trưng trạng thái ở khung hình kế tiếp.

1.2 Các phương pháp truy vết nhiều vật thể trong video

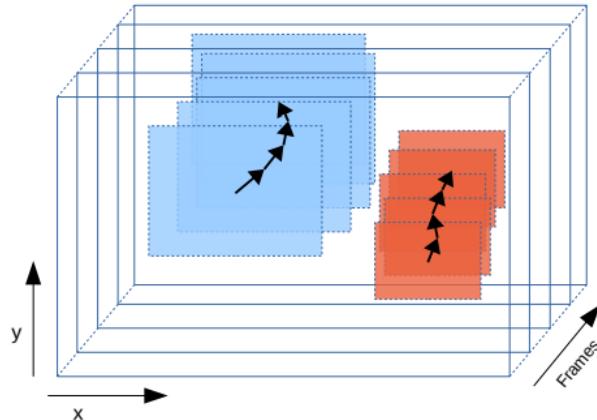
Phần 2.2 đã trình bày 3 nhóm phương pháp truy vết nhiều vật thể theo mức độ tích hợp các bước chính của bài toán. Với mục tiêu tìm hiểu các phương pháp thông dụng và nổi bật, trong khóa luận sẽ tìm hiểu một số phương pháp của 2 nhóm đầu tiên là SDE và JDE. Nhóm các phương pháp tích hợp toàn bộ quá trình truy vết chưa có nhiều nghiên cứu và hầu như chưa có đại diện nào được sử dụng phổ biến.

1.2.1 IoUTracker[1]

Đây là một phương pháp truy vết có nguyên lý đơn giản và đạt được tốc độ xử lý rất cao. IoUTracker mở rộng chuỗi truy vết của một vật thể bất kỳ bằng cách tính mức độ chồng khớp của vị trí xuất hiện cuối cùng của vật đó với toàn bộ các vật thể được phát hiện tại frame hiện tại. Vật có mức độ chồng khớp lớn nhất và lớn hơn một ngưỡng độ tin tưởng sẽ được sử dụng như vị trí kế tiếp cho chuỗi truy vết này. Vị trí của các vật thể tại mỗi frame được biểu diễn bởi các hộp giới hạn hình chữ nhật.

Gọi A và B là 2 hộp giới hạn cần tính, $area(A)$ và $area(B)$ lần lượt là 2 diện tích tương ứng. Mức độ chồng khớp của các hộp giới hạn được tính theo công thức sau.

$$IoU(A, B) = \frac{area(A) \cap area(B)}{area(A) \cup area(B)} \quad (3.8)$$

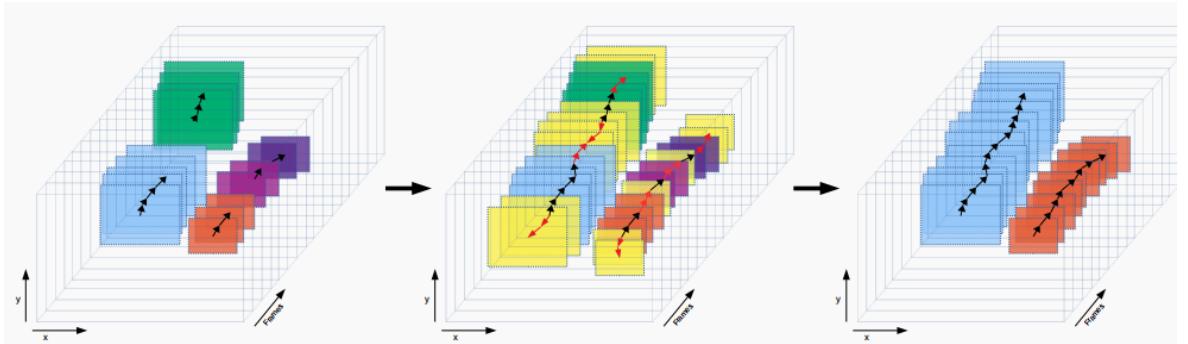


HÌNH 3.3: Nguyên lý hoạt động của IoUTracker[1]

Một chuỗi truy vết không được nối với bất kỳ vật thể ở frame hiện tại(các giá trị chòng khớp đều nhỏ hơn ngưỡng tin cậy) sẽ được xem như vật thể tương ứng chuỗi này đã rời khỏi khung hình. Một vật ở frame hiện tại không được nối với bất kỳ chuỗi nào trước đó sẽ được xem như một vật vừa đi vào khung hình. Có nhận xét rằng IoUTracker chỉ có 3 trong số 5 bước chính của thuật toán truy vết: phát hiện, tính độ liên quan(ở đây là IOU) và truy vết.

Một mở rộng mang tên VIoUTracker được đề xuất sau đó nhằm cải thiện tình trạng truy vết sai khi mô hình có False Negative lớn. Nói cách khác, khi mô hình phát hiện vật thể dự đoán thiếu, một chuỗi truy vết có thể bị kết thúc ngoài mong đợi. Để giải quyết vấn đề này, VIoUTracker sử dụng các phương pháp mô hình chuyển động[40],[41]. Đây là các phương pháp cho phép dự đoán vị trí của một vật thể tại các frame kế tiếp.

Nếu một chuỗi truy vết không nối được ở frame hiện tại sẽ vẫn được mở rộng thông qua các phương pháp mô hình chuyển động. Quá trình mở rộng chuỗi theo cách này sẽ được thực hiện trong tối đa ttl frame. Nếu trong ttl frame này chuỗi có thể nối được với một vật thể vừa được phát hiện, quy trình IoUTracker sẽ được tiếp tục. Ngược lại, chuỗi sẽ bị dừng lại. Ở chiều ngược lại, một chuỗi vừa mới xuất hiện sẽ được mở rộng qua mô hình chuyển động về ttl frame trước đó. Nếu một chuỗi đã bị dừng trước đó có độ chòng khớp thỏa mãn, 2 chuỗi này sẽ được hợp nhất với nhau. Với siêu tham số ttl , phương pháp như trên sẽ cho phép lấp đầy các khoảng trống có độ dài tối đa $2ttl$ từ kết quả của IoUTracker gốc.



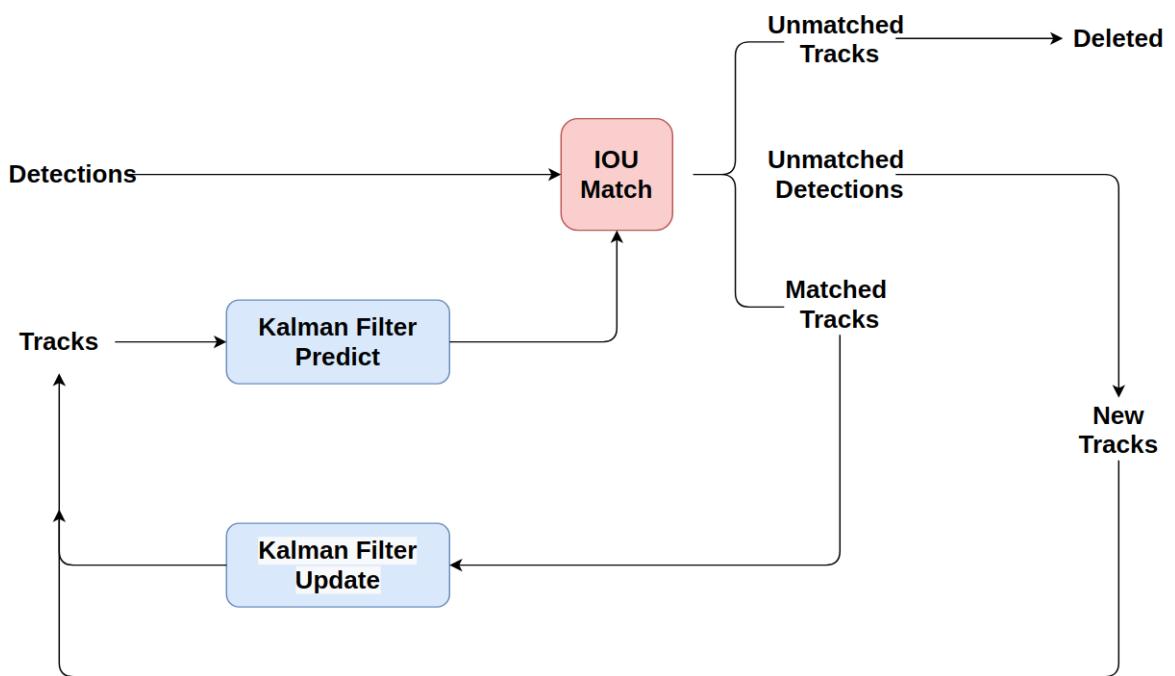
HÌNH 3.4: Nguyên lý của VIoUTracker. (a) Kết quả của IoUTacker.
 (b) Mô hình chuyển động được sử dụng để nối dài các chuỗi truy vết.
 (c) Kết quả của VIoUTracker[12]

1.2.2 SORT[2]

Phần này sẽ trình bày về SORT, một phương pháp cũng thuộc nhóm Separated Detection and Tracking. So với IoUTacker, SORT có thêm 1 trong số các bước chính của bài toán truy vết là dự đoán chuyển động. Các vật thể đã xuất hiện trong các frame trước đó được dự đoán vị trí mới ở frame hiện tại. Các vật thể vừa mới được phát hiện được so sánh với vị trí dự đoán mới này thay vì là vị trí xuất hiện cuối cùng như trong IoUTacker. Mô hình dự đoán chuyển động được sử dụng là Kalman Filter.

Hình 3.5 mô tả lại quy trình hoạt động của SORT. Tương tự IoUTacker, các vật thể mới xuất hiện được so sánh với các chuỗi truy vết đã có thông qua độ chồng khớp (IoU). Các chuỗi không được nối sẽ bị ngắt và các vật thể mới phát hiện không được nối sẽ tạo thành chuỗi mới. Khác biệt chủ yếu nằm ở việc SORT sử dụng Kalman Filter để dự đoán vị trí mới của các chuỗi và các vị trí mới này được dùng để so sánh với các vật thể vừa phát hiện. Ngoài ra, thay vì một chiến lược liên kết đơn giản dựa trên ngưỡng độ tin tưởng, SORT ứng dụng thuật toán Hungarian để liên kết các chuỗi và đối tượng mới phát hiện.

Vấn đề của SORT (và IoUTacker) là chỉ sử dụng thông tin IoU để liên kết các chuỗi với vật thể mới. Hướng thực hiện này chỉ có thể hoạt động hiệu quả khi video có độ phân giải tốt, IoU của các hộp giới hạn thuộc cùng một đối tượng lớn. Các giải pháp như SORT và IoUTacker sẽ kém hiệu quả khi xảy ra tình trạng vật bị chướng ngại vật che khuất hoặc nhiều vật có quỹ đạo tương đồng.



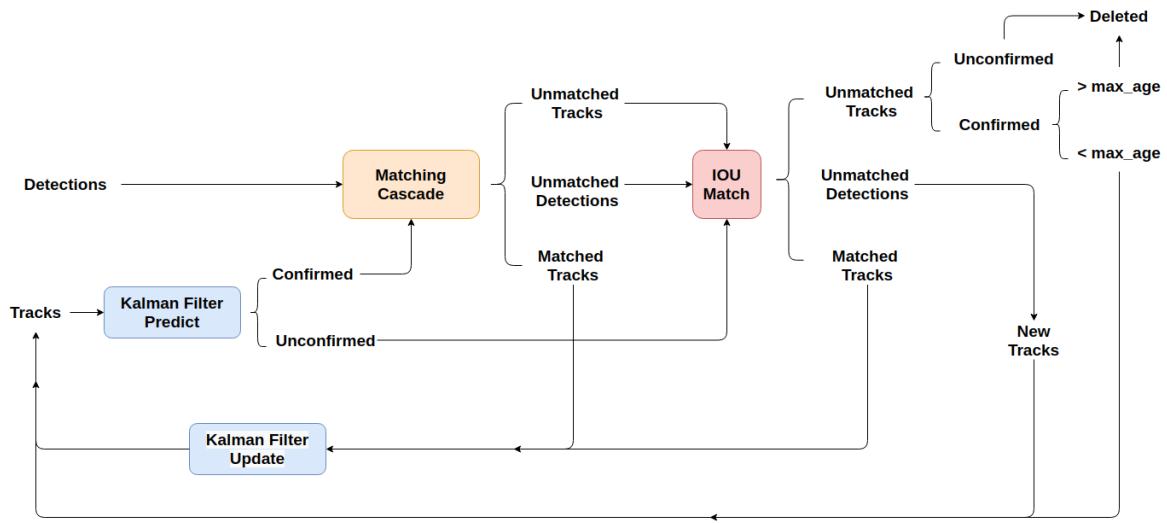
HÌNH 3.5: Sơ đồ phương pháp SORT[13]

1.2.3 DEEPSORT[3]

Để giải quyết các vấn đề còn tồn tại của SORT, tác giả của bài báo DEEPSORT[3] đã sử dụng thêm các đặc trưng về không gian và đặc trưng trực quan để phân biệt các đối tượng. Ngoài ra, một chiến lược liên kết theo tầng (Matching Cascade) và một cách quản lý mới cho vòng đời của chuỗi truy vết được xây dựng nhằm nâng cao độ chính xác liên kết.

Cải tiến thứ nhất của DEEPSORT là sử dụng thêm các độ đo khác ngoài IoU để liên kết các đối tượng. Các độ đo mới này bao gồm khoảng cách Mahalanobis giữa chuỗi truy vết và vật thể mới phát hiện trong không gian véc-tơ, và khoảng cách Cosine giữa 2 véc-tơ đặc trưng trực quan tương ứng 2 hộp giới hạn. Các đặc trưng trực quan này được trích xuất thông qua mạng học sâu, cụ thể là Wide Residual Net[42] trong bài báo gốc của tác giả.

Cải tiến thứ 2 của DEEPSORT là thay đổi cách quản lý vòng đời chuỗi truy vết. Thay vì loại ngay một chuỗi truy vết khi nó không được kết nối, tác giả của DEEPSORT đề xuất gán cho mỗi chuỗi 1 trong 3 loại trạng thái(tentative, confirmed, deleted):



HÌNH 3.6: Sơ đồ phương pháp DEEPSORT[13]

- Các chuỗi vừa xuất hiện sẽ được gán 1 giá trị mang tính thăm dò (tentative)
- Các chuỗi tentative sau khi được kết nối liên tục sau t_1 frame sẽ chuyển sang trạng thái xác nhận(confirmed). Các chuỗi đã được xác nhận (là một vật thể thật sự được truy vết chứ không phải là một vật thể nhiễu) sẽ không bị loại ngay lập tức nếu không được nối. Chuỗi này sẽ vẫn được cố gắng kết nối trong t_2 frame kế tiếp. Ở đây ta có 2 siêu tham số quan trọng là t_1 và t_2 . Trong bài báo gốc, chúng lần lượt là 3 và 30.
- Các chuỗi chưa được xác nhận mà bị mất dấu, hoặc một chuỗi đã được xác nhận nhưng bị mất dấu quá t_2 frame sẽ bị loại bỏ.

Cải tiến thứ 3 của DEEPSORT nằm ở chiến lược liên kết theo tầng. Như đã trình bày ở trên, 1 chuỗi đã được xác nhận sẽ không bị xóa trong t_2 frame. Tuy nhiên, khi liên kết các chuỗi này với kết quả của mô hình phát hiện vật thể trong frame hiện tại, các chuỗi bị mất dấu càng trễ sẽ càng được ưu tiên liên kết trước.

1.2.4 CenterTrack[4]

Đây là phương pháp thuộc nhóm Joint Detection and Tracking được tìm hiểu đầu tiên trong phạm vi khóa luận. CenterTrack[4] thực hiện một số thay đổi lên mô hình CenterNet[30] để có thể ứng dụng trực tiếp cho bài toán truy vết đa đối tượng.



HÌNH 3.7: Sơ đồ phương pháp CenterTrack[4]

Phương pháp CenterNet nhận đầu vào là 1 ảnh $I \in \mathbb{R}^{W \times H \times 3}$ với kích thước chiều rộng và chiều cao là W và H . Kết quả đầu ra của mô hình là 1 bản đồ nhiệt $Y \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times C}$ và 1 ma trận $S \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$. Ta có C là số lượng nhãn phân loại và R là mức giảm kích thước mỗi chiều của đầu vào khi đi qua mạng. Với $R = 4$ như trong bài báo CenterNet[30], kích thước của bản đồ nhiệt là $Y \in \mathbb{R}^{\frac{W}{4} \times \frac{H}{4} \times C}$. Mỗi vị trí trong bản đồ nhiệt Y là xác suất tâm vật thể nằm trong vùng 4×4 mà vị trí đó đại diện. Tương tự, mỗi vị trí của S tương ứng với 2 giá trị kích thước chiều rộng và chiều cao hộp giới hạn của vật thể.

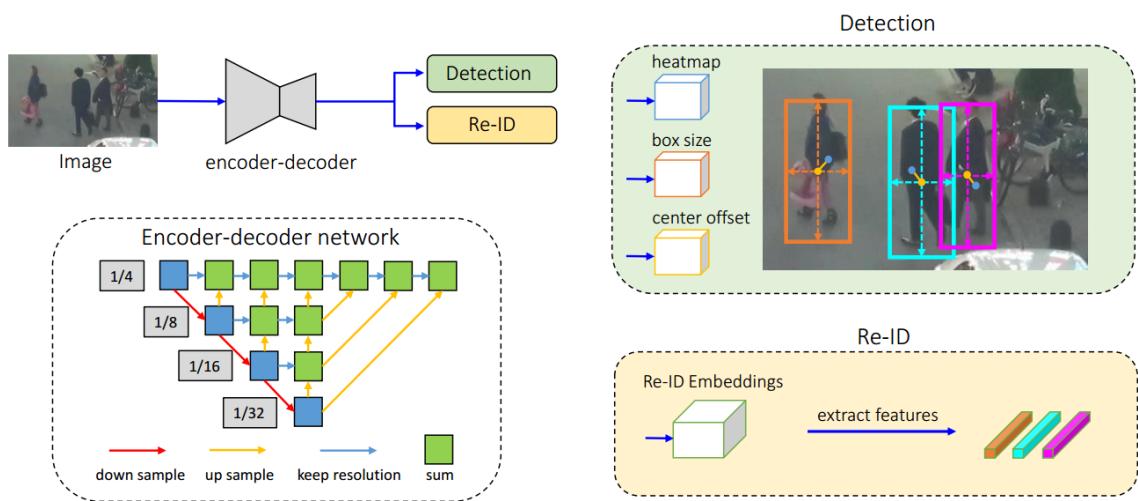
Được xây dựng dựa trên CenterNet, CenterTrack dùng thêm ảnh và bản đồ nhiệt từ 1 frame liền trước để bổ sung thông tin cho frame hiện tại. Hai thành phần này cùng với frame hiện tại sẽ tạo thành đầu vào cho mạng. Lập luận cho cách làm này, bài báo cho rằng CenterNet tập tung phát hiện vật thể trong 1 ảnh duy nhất. Nếu một vật thể bị che khuất và không thể nhìn thấy ở frame hiện tại, CenterNet sẽ không thể phát hiện được. Bằng cách dùng frame phía trước như đầu vào bổ sung, CenterTrack hướng đến việc phát hiện được các vật nhìn thấy được từ quá khứ nhưng bị khuất ở hiện tại. Đầu ra của mạng ngoại trừ Y và S như phiên bản gốc sẽ có thêm một ma trận $D \in \mathbb{R}^{\frac{W}{R} \times \frac{H}{R} \times 2}$ biểu diễn véc-tơ độ lệch của tâm vật thể so với giá trị từ frame phía trước.

Chiến lược liên kết của CenterTrack có thể miêu tả như sau. Mỗi vật thể vừa được phát hiện p với độ lệch tâm d_p sẽ được nối với vật thể có khoảng cách gần nhất với $p - d_p$ nếu khoảng cách này nhỏ hơn ngưỡng k . Nếu vật p không nối được với vật nào từ frame trước, một chuỗi truy vết mới ở vị trí của p sẽ được tạo ra.

1.2.5 FairMOT[5]

FairMOT [5] thuộc nhóm phương pháp JDE. Vì thực hiện cùng một lúc hai bài toán con là phát hiện và theo dõi vật thể trong bài toán MOT, bài toán đã trở thành một bài toán multi-task learning. Để giải quyết vấn đề này, nhóm tác giả đã xây dựng

một mô hình gồm hai nhánh cho phép rút trích véc-tơ đặc trưng riêng biệt cho hai bài toán phát hiện vật thể (detection) và định danh vật thể (re-ID embedding) như sơ đồ hình 3.8. Véc-tơ đặc trưng của vật thể cùng với thông tin vị trí tâm của vật thể (được xác định ở bước detection) giúp cho việc gán nhãn tracking cho các vật thể hiệu quả hơn, đồng thời giúp hạn chế sai sót các trường hợp một bounding box xuất hiện nhiều vật thể (vật thể chồng lấp lên nhau) hoặc nhiều bounding box chứa một vật thể (do sai số trong việc xác định bounding box vật thể của quá trình detection).



HÌNH 3.8: Sơ đồ biểu diễn phương pháp FairMOT [5]

Để rút trích đặc trưng vật thể, FairMOT sử dụng cấu trúc mạng Encoder-Decoder (hình 3.8) với encoder sử dụng kiến trúc Resnet-34 làm backbone để rút trích đặc trưng, sau đó sử dụng kiến trúc DLA[43] hoặc HRNet [44] giúp tổng hợp các véc-tơ đặc trưng từ nhiều lớp (layer) từ backbone (encoder). Thêm vào đó, các lớp Convolution đảm nhận vai trò kết hợp véc-tơ đặc trưng có chiều thấp hơn với chiều cao hơn (up-sampling modules) được thay thế bởi lớp Deformable Convolution [45] giúp véc-tơ đặc trưng được rút trích được điều chỉnh (dynamically adjusted) nhằm thu được thông tin theo tỉ lệ bounding box (scale) và hình dáng (pose) của vật thể hiệu quả.

Ở nhánh phát hiện vị trí vật thể (detection branch), nhóm tác giả dựa trên kiến trúc mạng của CenterNet [30], sử dụng ba lớp Convolution song song lên đầu ra của kiến trúc Encoder-Decoder với kích thước kernel 3x3, rồi 1x1 để rút trích ra ba véc-tơ đặc trưng là heatmap, box size và center offset. Heatmap có vai trò ước tính vị trí trung tâm của các vật thể, có kích thước $N \times H \times W$ với N là số lượng vật thể trong khung hình, H và W là kích thước của ảnh ban đầu. Boxsize có nhiệm vụ ước tính

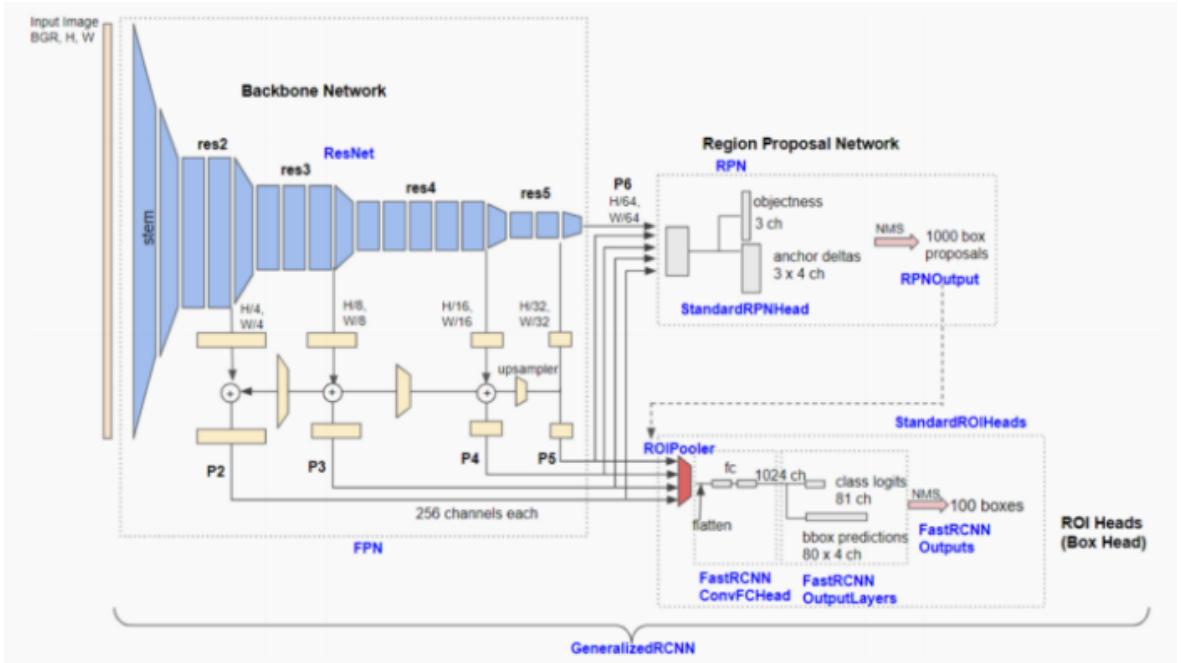
kích thước của bounding box của vật thể, Boxsize có kích thước $2 \times H \times W$ mỗi bounding box của vật thể $b^i = (x_1^i, y_1^i, x_2^i, y_2^i)$. Với x và y biểu diễn cho tọa độ của góc trên trái và góc dưới phải được biểu diễn tương ứng với véc-tơ có chiều $1 \times H \times W$, nhằm giúp xác định kích thước bounding box của vật thể. Kích thước bounding box của vật thể sau đó có thể tính bởi công thức đơn giản $s^i = (x_1^i - x_2^i, y_1^i - y_2^i)$. Center offset đảm nhiệm vai trò ước tính độ dời xung quanh vị trí tâm của vật thể, có kích thước $2 \times H \times W$ với mỗi bounding box giá trị center offset được tính dựa trên vị trí tâm của bounding box $c^i = (c_x^i, c_y^i)$ $o^i = (\frac{c_x^i}{4}, \frac{c_y^i}{4}) - (\lfloor \frac{c_x^i}{4} \rfloor, \lfloor \frac{c_y^i}{4} \rfloor)$.

Ở nhánh rút trích đặc trưng của vật thể (re-ID embedding), các véc-tơ đặc trưng từ backbone (Resnet34) được đi qua một lớp Convolution với 128 kernel nhằm rút trích đặc trưng cho mỗi vị trí bounding box của vật thể phát hiện được, đầu ra có cả véc-tơ re-ID có kích thước $E \in \mathbb{R}^{128 \times H \times W}$

Sau khi vị trí bounding box của vật thể đại diện bởi ba đặc trưng là heatmap, box size, và center offset và véc-tơ thể hiện đặc trưng của mỗi vật thể (re-ID) trong một khung hình được rút trích, FairMOT sẽ kết hợp với tập các véc-tơ đặc trưng tương ứng ở các khung hình trước, sau khi đã được dự đoán sự thay đổi qua bộ lọc Kalman Filter và sử dụng thuật toán Hungarian để gán nhãn tracking cho vật thể.

1.2.6 Tracktor[6]

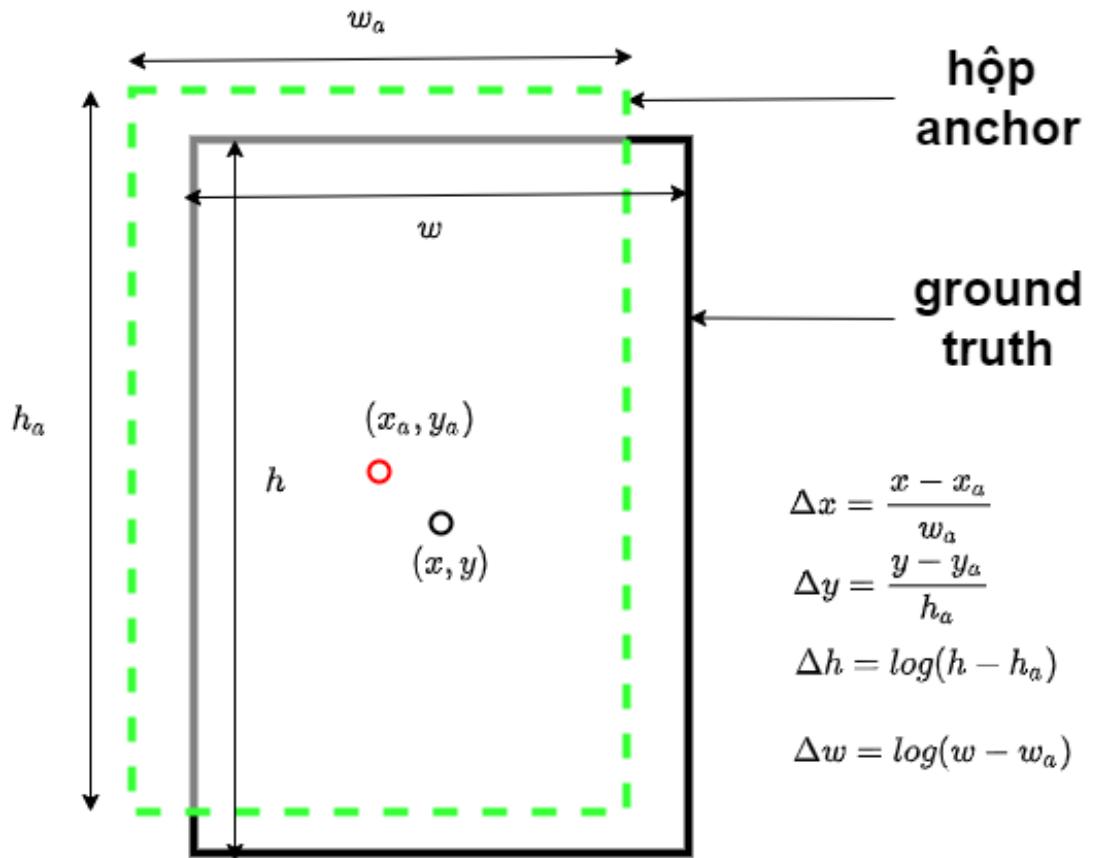
Trong khi CenterTrack[4] được xây dựng dựa trên mạng phát hiện vật thể CenterNet. Tracktor lại là một phương pháp JDE được xây dựng dựa trên Faster RCNN[27]



HÌNH 3.9: Sơ đồ mạng của Faster RCNN với backbone Resnet101-FPN[14]. Sơ đồ bao gồm 3 phần chính: mạng backbone, mạng đề xuất khu vực, mạng ROI head[15]

Ta cùng nhìn lại một chút về lý thuyết của Faster RCNN. Đây là một phương pháp phát hiện vật thể 2 pha sử dụng cơ chế hộp anchor. Đặc trưng của ảnh đầu vào $I \in \mathbb{R}^{H \times W \times 3}$ trước hết sẽ được trích xuất thông qua một mạng backbone. Cơ chế của FPN cho phép lấy ma trận đặc trưng từ nhiều lớp tích chập trung gian. Ta gọi các ma trận đặc trưng này là $P_i \in \mathbb{R}^{\frac{W}{R_i} \times \frac{W}{R_i} \times C}$ với C là số channel và R_i độ giảm kích thước của mỗi lớp P_i (trong hình 3.9 ta có R_i tương ứng P_1 đến P_6 là 4,8,16,32,64).

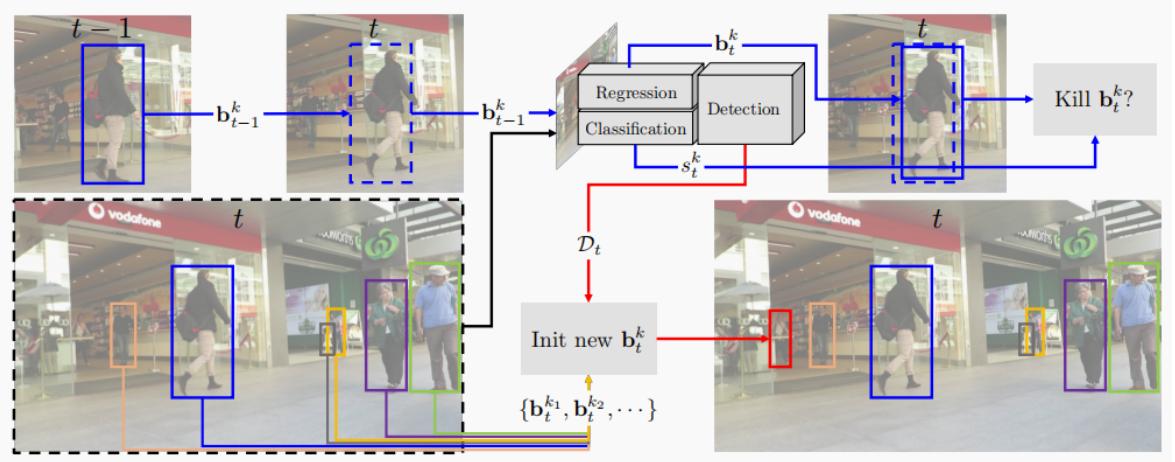
Các ma trận đặc trưng tiếp tục được dùng làm đầu vào cho mạng đề xuất khu vực một cách song song. Mạng này tách thành 2 nhánh gồm những lớp tích chập chồng nhau. Kết quả của 2 nhánh lần lượt là ma trận dự đoán xác suất có vật xuất hiện $Y_i \in \mathbb{R}^{\frac{W}{R_i} \times \frac{W}{R_i} \times B}$ và ma trận hiệu chỉnh vị trí $O_i \in \mathbb{R}^{\frac{W}{R_i} \times \frac{W}{R_i} \times B \times 4}$ với B là số lượng hộp anchor được đặt lên mỗi ô của các ma trận đặc trưng.



HÌNH 3.10: Tính kết quả vị trí chính xác của vật

Bởi vì các kết quả này thu được từ ma trận đặc trưng có kích thước thu nhỏ. Trong khi đó, kết quả vị trí của vật thể là tọa độ chính xác trên kích thước ảnh ban đầu. Vì thế mô hình sử dụng 4 kết quả của O_i là $\Delta x, \Delta y, \Delta w, \Delta h$ để tìm chính xác tọa độ tâm và kích thước chiều rộng chiều cao của vật.

Thuật toán chọn ra những hộp anchor có giá trị xác suất xuất hiện vật lớn bằng phương pháp NMS và dùng chúng làm đầu vào cho mạng ROI head. Vì các hộp anchor này có kích thước khác nhau, một tầng ROI Pooling[27] được sử dụng để thay đổi các hộp anchor về cùng kích cỡ. Mạng ROI head gồm 1 số lớp Fully connected và đưa ra kết quả gồm 2 thành phần: kết quả nhãn phân loại và kết quả hiệu chỉnh vị trí tương ứng với mỗi hộp anchor được đề xuất từ trước đó. Kết quả hiệu chỉnh vị trí ở bước này có ý nghĩa giống với ma trận hiệu chỉnh vị trí ở mạng đề xuất khu vực.



HÌNH 3.11: Phương pháp Tracktor[6]

Tracktor tận dụng khả năng dự đoán ma trận hiệu chỉnh vị trí ở mạng ROI head để dự đoán vị trí kế tiếp của chuỗi truy vết trong frame hiện tại. Cụ thể hơn, ở bước ROI Pooling, các vùng được đề xuất ở frame $t - 1$ được áp lên ma trận đặc trưng của frame t . Kết quả hiệu chỉnh vị trí tương ứng sẽ là dự đoán cho vị trí của vật thể đó trong frame t (tham khảo hình 3.11).

Tracktor quản lý vòng đời của mỗi chuỗi truy vết như sau: các chuỗi truy vết được dự đoán vị trí xuất hiện kế tiếp qua mạng ROI head. Kết quả của ROI head gồm ma trận hiệu chỉnh vị trí và ma trận kết quả nhãn phân loại. Nếu như xác suất của nhãn có giá trị lớn nhất nhỏ hơn ngưỡng σ_{active} , chuỗi này sẽ bị loại bỏ. Ngoài ra, các chuỗi truy vết cũng được lọc lại một lần nữa bằng NMS với ngưỡng chồng khớp λ_{active} . Các vật thể vừa được phát hiện của frame hiện tại không bị chồng khớp quá λ_{new} sẽ tạo thành một chuỗi truy vết mới. Tracktor sử dụng tổng cộng 3 siêu tham số σ_{active} , λ_{active} và λ_{new} để quản lý vòng đời các chuỗi.

Tác giả của Tracktor sau đó đề xuất thêm 2 cải tiến để gia tăng độ chính xác. Cải tiến thứ nhất là việc sử dụng một mô hình chuyển động đơn giản để dự đoán vị trí của hộp giới hạn trong quá khứ tại frame hiện tại. Vị trí mới dự đoán này sẽ được áp lên ma trận đặc trưng của frame hiện tại khi thực hiện ROI Pooling. Cải tiến thứ 2 là dùng một mạng trích xuất đặc trưng trực quan. Các chuỗi truy vết đã bị loại bỏ sẽ vẫn được giữ lại để xem xét trong F_{reid} frame. Nếu trong khoảng thời gian này, khoảng cách giữa véc-tơ đặc trưng của chuỗi với một vật thể mới xuất hiện lớn hơn một ngưỡng λ_{reid} , 2 chuỗi này sẽ được xem là cùng 1 vật và được nối với nhau.

2 Ước tính tốc độ phương tiện giao thông

Lưu ý: Để thuận tiện cho việc theo dõi các công thức, trong phần này chúng tôi sử dụng quy ước cho các tên biến như sau:

- Số thực được ký hiệu bởi chữ in thường
- Vec-tơ được ký hiệu bởi chữ thường in đậm.
- Ma trận được ký hiệu bởi chữ hoa in đậm.

Cách ký hiệu chỉ số trên và chỉ số dưới có thể khác nhau giữa các phần. Ví dụ tọa độ ảnh trong phần 2.1 được ký hiệu là (u, v) nhưng ở phần 2.2.1.5 được ký hiệu là (u_x, u_y) . Ý nghĩa của mỗi tên biến sẽ được giải thích ở đầu mỗi phần.

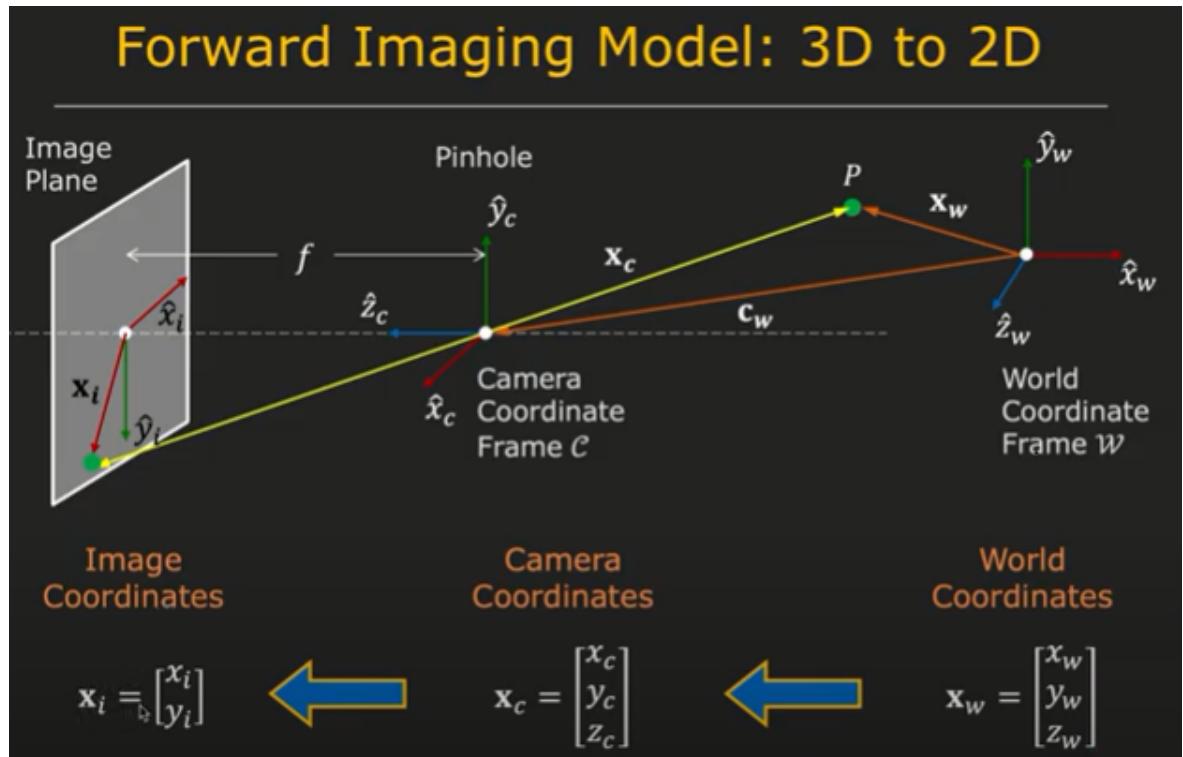
2.1 Chuyển từ tọa độ ảnh thành tọa độ thực tế - Mô hình camera tuyến tính (Linear camera model)

Một trong các vấn đề quan trọng của lĩnh vực thị giác máy tính là khôi phục cấu trúc 3 chiều từ không gian ảnh. Thủ tướng tượng chúng ta có một hệ tọa độ 3 chiều với gốc tọa độ đặt ở 1 vị trí bất kỳ. Từ một bức ảnh, chúng ta muốn biết mỗi 1 pixel trên ảnh nằm ở tọa độ nào trên không gian ba chiều; nói một cách khác là chuyển tọa độ 2D của ảnh (u, v) về tọa độ 3D trên thực tế (x_w, y_w, z_w) . Để thực hiện việc này, có 2 vấn đề cần được xác định:

- Vị trí, góc độ của camera so với tọa độ 3 chiều trong thực tế được gọi chung là extrinsic parameter. Các thông số này được biểu diễn thông qua ma trận xoay và ma trận dịch chuyển (\mathbf{R}, \mathbf{T} trong công thức 2.1)
- Cách camera ánh xạ các điểm chiếu lên mặt phẳng ảnh. Vấn đề này được quy định bởi tiêu cự camera, kích thước hình ảnh, tỉ lệ quy đổi giữa pixel và độ đo thực tế (chẳng hạn mét).

Tại đây chúng ta sẽ xây dựng một góc nhìn toàn cảnh về mô hình camera tuyến tính. Ta có một hệ tọa độ 3 chiều thực tế w và ta xem xét một điểm p nằm ở tọa độ \mathbf{x}_w bất kỳ. Trong hệ tọa độ w , camera nằm ở tọa độ x_c và được định nghĩa bởi một hệ tọa độ 3D camera c . Mô hình pinhole camera sẽ ánh xạ điểm p lên một mặt phẳng ảnh i tương ứng với camera. Các trục tọa độ của 3 hệ tọa độ trên lần lượt là $(\hat{x}_w, \hat{y}_w, \hat{z}_w), ((\hat{x}_c, \hat{y}_c, \hat{z}_c), ((\hat{x}_i, \hat{y}_i))$. Cần lưu ý thêm rằng trục \hat{z}_c sẽ trùng với trục quang

học của camera và có phương vuông góc với mặt phẳng ảnh. Tiêu cự của camera được ký hiệu bởi f và tọa độ của camera là \mathbf{c}_w . Tọa độ \mathbf{x}_w được chuyển đổi về hệ tọa độ c để trở thành \mathbf{x}_c , và sau đó là về tọa độ ảnh \mathbf{x}_i .



HÌNH 3.12: Mô hình camera tuyến tính chuyển đổi giữa tọa độ ảnh và tọa độ 3D thực tế[16]

Để chuẩn bị cho bước giải thích về công thức của mô hình camera tuyến tính, ta sẽ tìm hiểu về khái niệm *homogenous*. Đây là một dạng biểu diễn của các véc-tơ sang không gian nhiều chiều hơn. Ví dụ dạng biểu diễn homogenous 3D của 1 véc-tơ $\begin{bmatrix} u \\ v \end{bmatrix}$

là $\begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix}$. Trong đó \tilde{w} là một giá trị ảo thỏa mãn:

$$u = \frac{\tilde{u}}{\tilde{w}} \quad v = \frac{\tilde{v}}{\tilde{w}}$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} = \tilde{\mathbf{u}}$$

Ta bắt đầu với công thức chuyển đổi từ hệ tọa độ 3D của camera sang hệ tọa độ ảnh.

$$\frac{x_i}{f} = \frac{x_c}{z_c} \quad \frac{y_i}{f} = \frac{y_c}{z_c}$$

$$\rightarrow x_i = f \frac{x_c}{z_c} \quad y_i = f \frac{y_c}{z_c}$$

Bởi vì (x_i, y_i) có cùng đơn vị với 2 hệ tọa độ còn lại (chẳng hạn mét) trong khi tọa độ ảnh là pixel nên ta cần tỉ lệ chuyển đổi giữa pixel sang độ đo thực tế. Tỉ lệ chuyển đổi theo 2 phương x và y là m_x và m_y .

$$u = m_x x_i = m_x f \frac{x_c}{z_c} \quad v = m_y y_i = m_y f \frac{y_c}{z_c}$$

Trong thực tế, ta thường xem gốc tọa độ theo đơn vị pixel nằm ở góc trái trên trong khi gốc tọa độ theo độ đo thực tế nằm ở giữa ảnh. Vì vậy, công thức bên trên sẽ phụ thuộc thêm vào tọa độ pixel (p_x, p_y) của điểm giữa ảnh. Ngoài ra, ta biết rằng tiêu cự ảnh không phụ thuộc vào chiều x hay y , nhưng khi được nhân với tỉ lệ chuyển đổi theo 2 chiều m_x, m_y , ta xem như có 2 giá trị tiêu cự f_x, f_y

$$u = m_x f \frac{x_c}{z_c} + p_x \quad v = m_y f \frac{y_c}{z_c} + p_y$$

$$u = f_x \frac{x_c}{z_c} + p_x \quad v = f_y \frac{y_c}{z_c} + p_y$$

Lúc này ta đã có thể chuyển đổi các công thức trên về dạng của các phép nhân ma trận. Dạng biểu diễn homogenous sẽ được sử dụng ở bước này

$$\begin{aligned} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} &\equiv \begin{bmatrix} \tilde{w}u \\ \tilde{w}v \\ \tilde{w} \end{bmatrix} \equiv \begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x x_c + z_c p_x \\ f_y y_c + z_c p_y \\ z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \\ &= [\mathbf{K}|0] \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = [\mathbf{K}|0] \tilde{\mathbf{x}}_c \quad (3.9) \end{aligned}$$

Ta gọi ma trận $\mathbf{K} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}$ là ma trận của các tham số intrinsic. Tiếp theo ta đến với công thức chuyển đổi giữa hệ tọa tọa độ thực tế và hệ tọa độ camera. Các tham số extrinsic bao gồm 2 ma trận xoay \mathbf{R} và ma trận chuyển vị \mathbf{T} . Trong đó $\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ có mỗi hàng là véc-tơ chỉ phương của $\hat{x}_c, \hat{y}_c, \hat{z}_c$ trong hệ w và $\mathbf{T} = -\mathbf{R}\mathbf{c}_w$

$$\mathbf{x}_c = \mathbf{R}(\mathbf{x}_w - \mathbf{c}_w) = \mathbf{R}\mathbf{x}_w - \mathbf{R}\mathbf{c}_w = \mathbf{R}\mathbf{x}_w + \mathbf{T} \quad (3.10)$$

$$\Leftrightarrow \mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3.11)$$

$$\Leftrightarrow \tilde{\mathbf{x}}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (3.12)$$

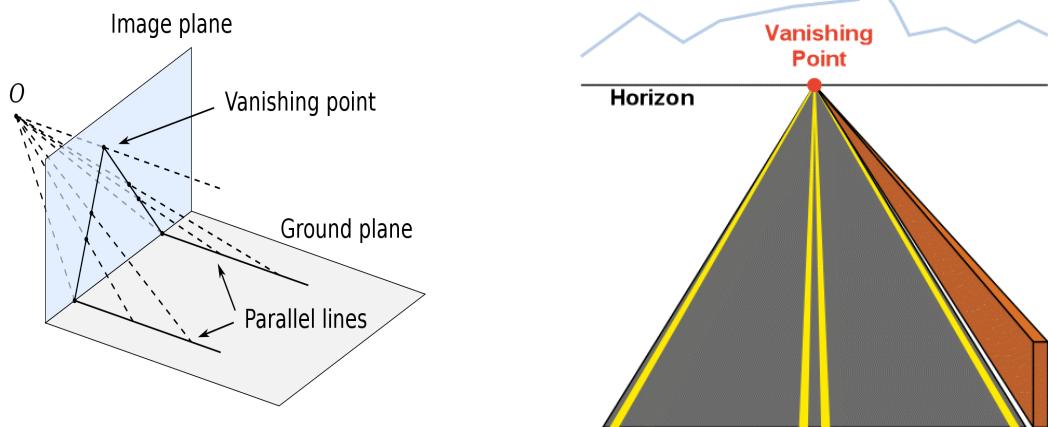
$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{x}}_w \quad (3.13)$$

Kết hợp 3.9 và 3.13, ta có:

$$\begin{aligned} \tilde{\mathbf{u}} &= \begin{bmatrix} f_x & 0 & p_x & 0 \\ 0 & f_y & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{\mathbf{x}}_w = [\mathbf{K}|0] [\mathbf{R}|\mathbf{T}] \tilde{\mathbf{x}}_w \\ &= \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \tilde{\mathbf{x}}_w = \mathbf{P} \tilde{\mathbf{x}}_w \end{aligned} \quad (3.14)$$

Nhìn lại công thức 3.9, 2 dấu đầu tiên không phải là dấu bằng của phương trình. Bởi lẽ đây là phép chuyển đổi homogeneous. Có 1 biến không xác định là z_c . Để có thể thực sự chuyển đổi giữa các hệ tọa độ, giá trị z_c này cần được xem xét. Lúc này, để có thể hoàn thành dạng phương trình, công thức 2.1 sẽ được viết lại:

$$z_c \mathbf{u} = \mathbf{P} \tilde{\mathbf{x}}_w \quad (3.15)$$



(A) Minh họa vanishing point trong phép chiếu phối (B) Minh họa vị trí vanishing point trong mặt phẳng đường (ground plane)

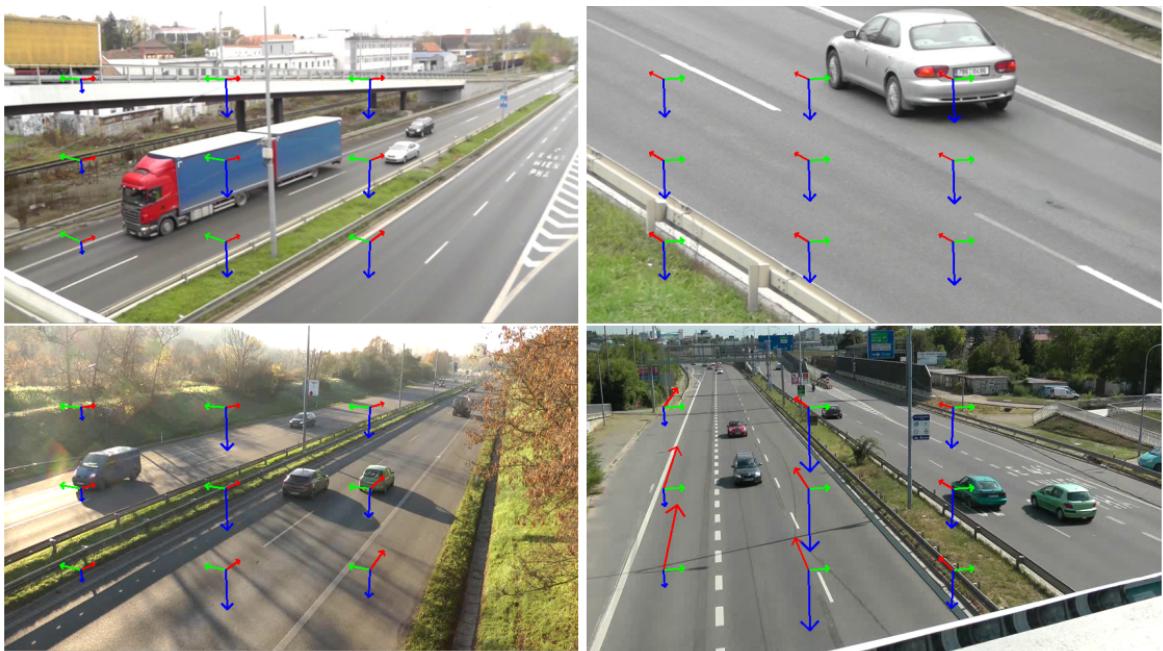
HÌNH 3.13: Vanishing Point (Điểm biến mất)

Giá trị của z_c được gọi là scale factor và là duy nhất với mỗi điểm trong không gian.

2.2 Phương pháp tự động hiệu chỉnh thông số camera (Automatic Camera Calibration)

Trong phần này, chúng tôi trình bày lý thuyết của thuật toán hiệu chỉnh thông số máy ảnh tự động mà nhóm đã sử dụng để giải quyết bài toán xác định các tham số bên trong (intrinsic) và bên ngoài (extrinsic) của máy ảnh. Mục tiêu của phương pháp này là phát hiện ba điểm biến mất trực giao (orthogonal vanishing point) từ một góc đặt của máy ảnh dựa trên phương pháp "Automatic roadsidecamera calibration" [18] do tác giả Dubská và các cộng sự phát triển và công bố.

Vanishing point: là một điểm trên mặt phẳng ảnh của phép chiếu phối cảnh (hình ảnh thu được trên màn chắn của máy ảnh lỗ kim) nơi các hình chiếu phối cảnh hai chiều (hoặc hình vẽ) của các đường thẳng song song trong không gian ba chiều hội tụ tại một điểm.



HÌNH 3.14: Minh họa kết quả thuật toán tự động hiệu chỉnh thông số camera với 3 véc-tơ chỉ phương của đường thẳng đi qua ba vanishing point

Xét trường hợp cảnh quay trên đoạn đường với các phương tiện di chuyển theo một hướng chính, ta có thể xác định tọa độ mặt đường (Ground plane) so với hệ tọa độ của máy ảnh và hướng di chuyển của phương tiện qua ba vanishing point. Các điểm vanishing point được xác định:

- **Vanishing point thứ nhất** là giao điểm của các véc-tơ thể hiện hướng di chuyển của phương tiện được biểu diễn bằng véc-tơ màu đỏ ở hình 3.14 và các vạch kẻ đường.
- **Vanishing point thứ hai** là giao điểm của các véc-tơ song song với mặt phẳng đường và vuông góc với hướng di chuyển của phương tiện được biểu diễn bằng véc-tơ màu xanh lục ở hình 3.14
- **Vanishing point thứ ba** là giao điểm của các véc-tơ vuông góc với mặt phẳng đường (ground plane) được biểu diễn bằng véc-tơ màu xanh lam ở hình 3.14

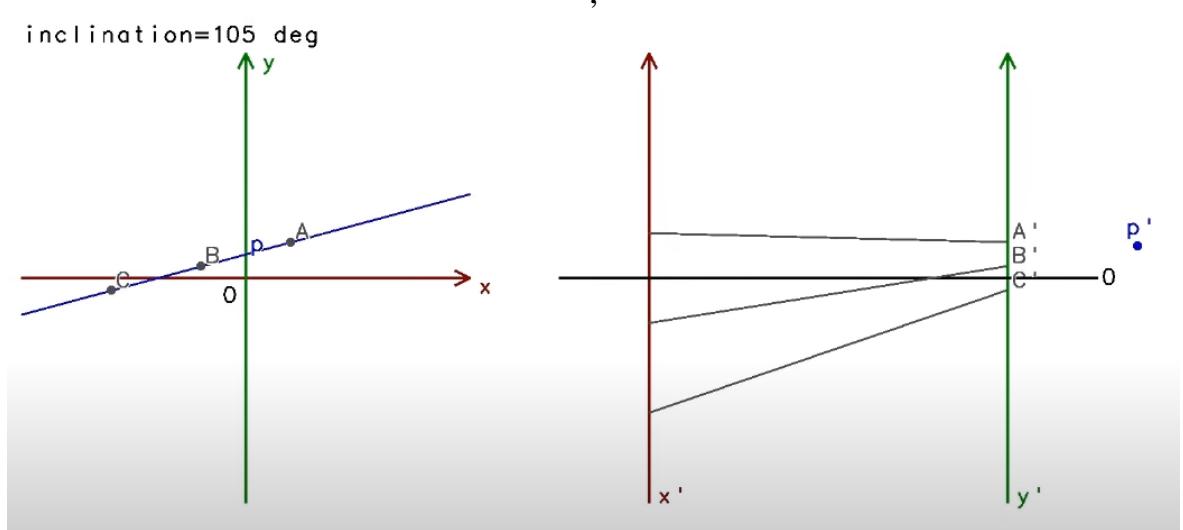
Điểm nổi bật của phương pháp này là ba vanishing point có thể phát hiện tự động từ bất kì góc đặt camera nào miễn là thỏa điều kiện đặt ra. Do đó ta có thể áp dụng phương pháp này cho nhiều hệ thống trong thực tế.

2.2.1 Phương pháp phát hiện điểm biến mất (Vanishing Point)

2.2.1.1 Không gian kim cương (Diamond space) Hệ tọa độ song song là hệ tọa độ trong đó các trục tọa độ song song và không gian của hệ tọa độ song song (dual space) bị giới hạn bên trong hai trục tọa độ song song đó. Do đó tọa độ một điểm (x,y) trong hệ tọa độ 2D được biểu diễn là một đường thẳng cắt hai trục tọa độ x và y .

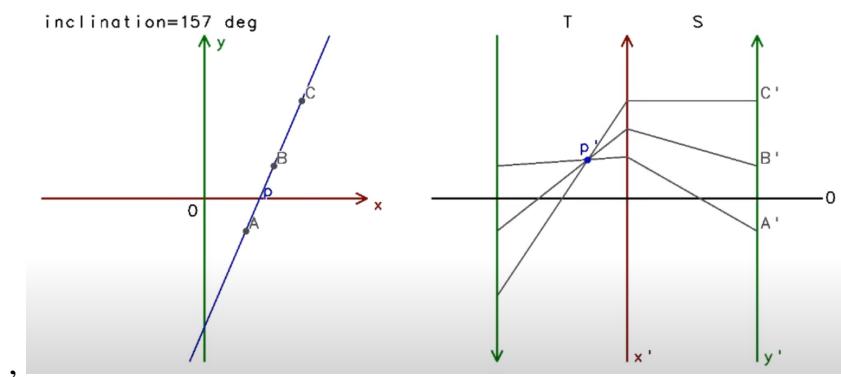
Từ các điểm góc cạnh của vật thể được xác định bằng các phương pháp phát hiện điểm như HOG (Histogram of gradient), Canny edge detection,... Để tham số hóa các đường thẳng trong ảnh cho bài toán xác định đường thẳng thì phương pháp nhận được quan tâm và sử dụng nhiều nhất là phép biến đổi Hough. Phép biến đổi Hough được mô tả là một phép biến đổi điểm thành đường (point to line) giữa hai không gian mà mỗi điểm trong không gian Đề-các (2D) được ánh xạ thành một đường thẳng trong không gian của tọa độ song song, được gọi là không gian Hough (Hough space). Trong đó, các điểm thuộc cùng một đường thẳng cắt nhau tại một điểm trong không gian của hệ tọa độ song song và điểm đó sẽ biểu diễn cho một đường thẳng trong hệ tọa độ 2D (Minh họa hình 3.17 a, b).

Bởi vì không gian song song (dual space) bị giới hạn bởi hai trục tọa độ song song nên việc tham số đường thẳng thành điểm sẽ gặp vấn đề khi các điểm của tọa độ Đề-các thuộc một đường thẳng được biểu diễn thành các đường thẳng trong hệ tọa độ song song không hội tụ trong giới hạn của không gian song song như hình 3.15.



HÌNH 3.15: Minh họa trường hợp không thể tham số đường thẳng bằng hệ tọa độ song song

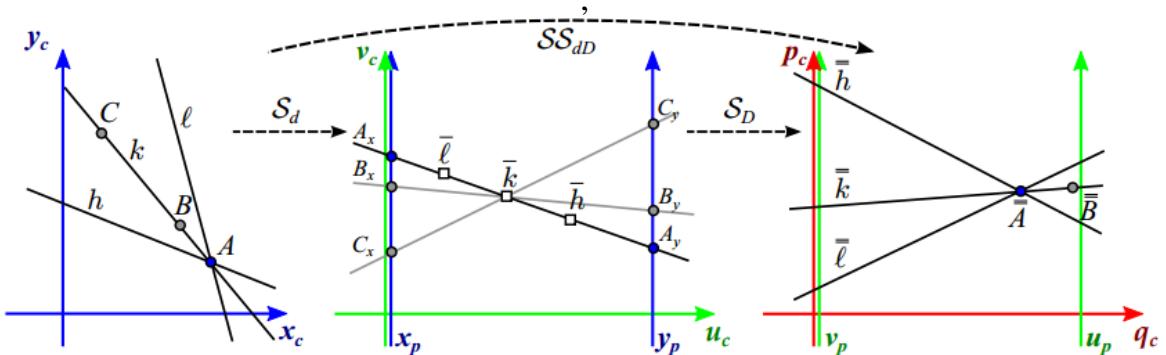
Để giải quyết vấn đề trên tác giả Dubská và các cộng sự đã đề xuất giải thuật PClines [46] tích hợp thêm một hệ trục tọa độ song song (không phải hệ trục tạo bởi không gian Hough) thứ hai bên cạnh hệ trục tọa độ song song thứ nhất (biểu diễn cho phép biến đổi Hough), có trục tọa độ ngược hướng với trục tọa độ còn lại. Hình 3.16 minh họa cho việc sử dụng hệ tọa độ thứ hai để giải quyết trường hợp ba đường thẳng đại diện cho ba điểm trong mặt phẳng ảnh không giao nhau trong không gian Hough thứ nhất. Giải thuật PCLines giúp tham số hóa tất cả đường thẳng trong toàn bộ hình chiêu phối cảnh ban đầu (mặt phẳng ảnh).



HÌNH 3.16: Minh họa đường thẳng trong hệ tọa độ Đề-các ban đầu được mô hình hóa trong hai hệ tọa độ song song

Sau khi dùng PCLines để tham số hóa các đường thẳng thành điểm ở hai tọa độ

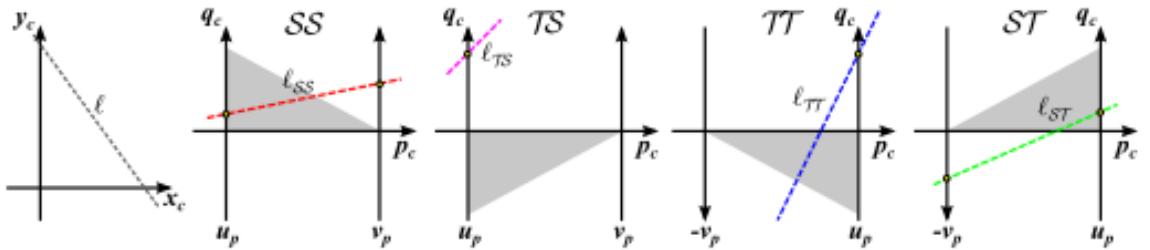
song song, nhằm xác định vị trí hình học của các đường thẳng trong hệ tọa độ song song, ta định nghĩa một hệ tọa độ Đề-các mới với hai trục tọa độ (u_c, v_c) như hình 3.17b. (p là ký hiệu cho trục tọa độ trong hệ tọa độ song song và c là ký hiệu cho trục tọa độ Đề-các) Lấy ý tưởng của từ phương pháp Cascade Hough Transform [47], ta sử dụng phép biến đổi Hough transform thứ hai để chuyển các điểm trở lại thành đường thẳng (point to line). Tác giả Dubská và các cộng sự đã sử dụng phép biến đổi Hough lần thứ hai và PCLines nhằm chuyển các đường thẳng đại diện cho các điểm trong không gian vô hạn Đề-các thành điểm ở không gian hữu hạn song song tạo bởi hai phép biến đổi Hough, mà điểm này ở không gian song song này là giao điểm của các đường thẳng trong ảnh, *tạo cơ sở cho việc tìm vanishing point*.



HÌNH 3.17: Minh họa hai phép biến đổi Hough liên tiếp bằng tham số hóa PCLines để chuyển một điểm từ không gian vô hạn Đề-các thành một điểm trong không gian song song[17]

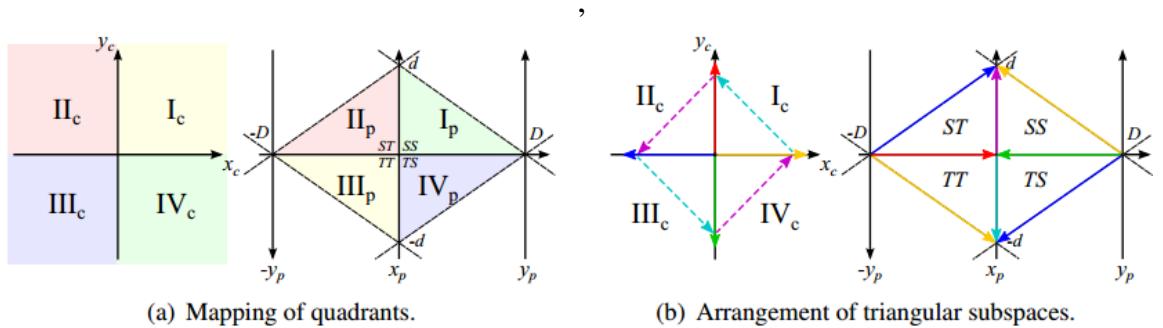
Hình 3.17a) biểu diễn hệ tọa độ Đề-các ban đầu (hệ tọa độ trong mặt phẳng ảnh) với các điểm và đường. Hình 3.17b) biểu diễn các điểm và đường tương ứng trong hình (a) ở hệ tọa độ song song sau phép biến đổi Hough thứ nhất. Hình 3.17c) biểu diễn hệ tọa độ Đề-các mới với hai trục tọa độ (q_c, p_c) , được biểu diễn là hai trục màu đỏ ở hình c), cho phép biến đổi Hough thứ hai với tham số hóa PCLines.

Gọi S (Straight) là phép biến đổi từ hệ tọa độ Đề-các về hệ tọa độ song song có hai trục tọa độ cùng hướng và T (Twisted) là phép biến đổi từ hệ tọa độ Đề-các về hệ tọa độ song song có hai trục tọa độ ngược hướng. Qua hai phép biến đổi Hough liên tiếp ta có tổ hợp bốn phép biến đổi: $S \circ S$, $S \circ T$, $T \circ S$, $T \circ T$, minh họa ở hình 3.18



HÌNH 3.18: Minh họa tổ hợp bốn phép biến đổi Hough liên tiếp bằng tham số hóa PCLines [18]

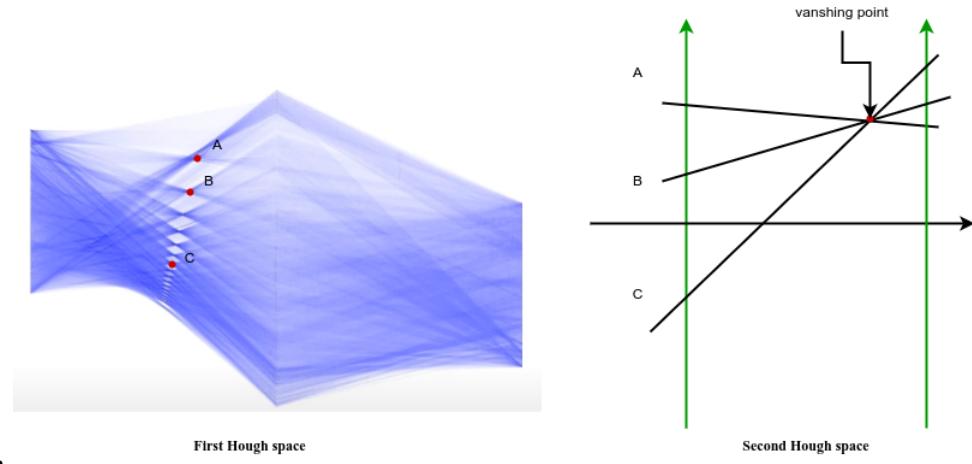
Tổ hợp bốn phép biến đổi biểu diễn cho ở hình 3.17 có thể được xem xét gộp thành một không gian gọi là *không gian kim cương* (*Diamond space*) như hình 3.19. Tác giả Dubská và cộng sự đã chứng minh [17] được rằng một điểm trong không gian Đề-các có thể liên kết tương ứng với một điểm trong không gian kim cương. Ví dụ ở hình 3.19 tương ứng với một điểm ở góc phần tư IV_c ta có thể tìm được một điểm tương ứng trong không gian giới hạn kim cương (*Diamond space*) ở góc phần tư I_p tương ứng với phép biến đổi $S \circ S$.



HÌNH 3.19: Diamond space[17]

2.2.1.2 Xác định vanishing point trong không gian kim cương Việc xây dựng không gian giới hạn kim cương (*Diamond space*) từ hai phép biến đổi Hough, sẽ chuyển một điểm trong không gian Đề-các thành một điểm trong không gian kim cương (*Diamond space*). Phép biến đổi Hough thứ nhất tìm các điểm cực đại (đại diện cho một đường thẳng trong hệ tọa độ Đề-các) được tích lũy bởi số đường thẳng đi qua của không gian Hough thứ nhất. Phép biến đổi Hough thứ hai biểu diễn một không gian Hough thứ hai là tập hợp các đường thẳng thể hiện cho tập điểm cực đại

ở không gian Hough thứ nhất, giao nhau tại một điểm, vanishing point, là điểm giao của các một tập các đường thẳng (Minh họa ở hình 3.20)



HÌNH 3.20: Minh họa vị trí vanishing point tìm được qua hai phép biến đổi Hough bằng tham số hóa PCLines

Đối với mỗi lần tìm được điểm cực đại, các đường thẳng có liên quan sẽ được phát hiện và tích lũy ở không gian kim cương. Giả sử một đường thẳng được xác định trong hệ tọa độ đồng nhất (homogeneous coordinate) được ký hiệu là (a, b, c) , xác định một công thức đường thẳng trong hệ tọa độ Đề-các của mặt phẳng ảnh (2D) là $ax + by + c = 0$. Đường thẳng này có thể được tích lũy thành một polyline (tập hợp các đoạn thẳng liền nhau tạo thành một khối) vào không gian giới hạn kim cương bằng việc áp dụng liên tiếp hai phép biến đổi Hough với phương pháp tham số hóa PCLines. Số lượng đoạn thẳng của polyline tương ứng với số lượng góc phần tư trong hệ tọa độ Đề-các mà đường thẳng đi qua. Các điểm giới hạn của polyline có thể được xác định bằng công thức sau:

$$\alpha = \text{sgn}(ab), \beta = \text{sgn}(bc), \gamma = \text{sgn}(ac)$$

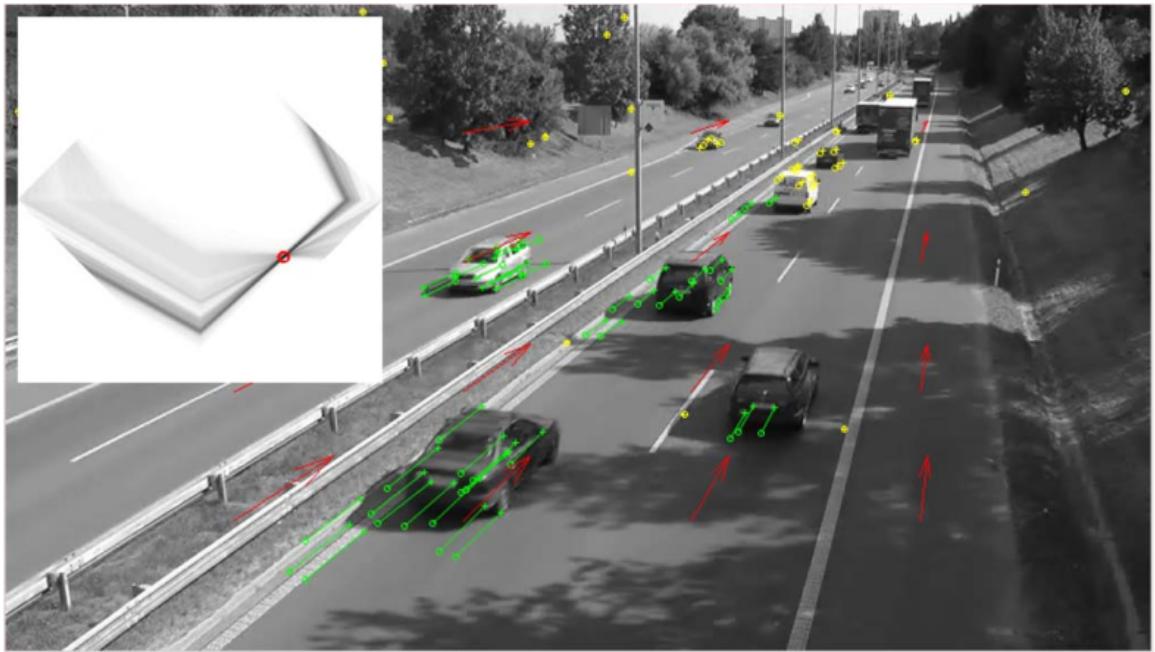
$$(a, b, c) \rightarrow \left[\frac{\alpha a}{c + \gamma a}, \frac{-\alpha c}{c + \gamma a} \right], \left[\frac{b}{c + \beta b}, 0 \right], \left[0, \frac{b}{a + \alpha b} \right], \left[\frac{-\alpha a}{c + \gamma a}, \frac{\alpha c}{c + \gamma a} \right] \quad (3.16)$$

Trong đó, $\text{sgn}(x)$ là hàm signum khác không. Khi đường thẳng trong tọa độ ảnh chỉ đi qua hai góc phần tư của hệ tọa độ Đề-các (đường thẳng ngang song song với trục hoành và đường thẳng dọc song song với tung độ, đường thẳng đi qua góc tọa độ), một đoạn của polyline sẽ trở thành một điểm. Sau các polyline được tích lũy vào không

gian, giá trị mỗi pixel biểu diễn một tọa độ của không gian kim cương tăng lên bởi các polyline cắt qua. Do đó ta sẽ tìm kiếm cực đại toàn cục trong không gian kim cương đã được tham số hóa, điểm được nhiều polyline cắt qua là vanishing point cần tìm. Sau khi tìm được vanishing point có tọa độ đồng nhất (homogeneous) $[p, q, 1]$ từ không gian kim cương chiếu trở về tọa độ 2D của mặt phẳng ảnh ban đầu bằng công thức:

$$[p, q, 1] \rightarrow [q, \text{sgn}(p)p + \text{sgn}(q)q - 1, p] \quad (3.17)$$

2.2.1.3 Phương pháp xác định vanishing point thứ nhất Như được đề cập ở đầu phần này, vanishing point thứ nhất là giao điểm của các đường thẳng biểu diễn cho hướng di chuyển của phương tiện, song song với mặt phẳng đường (ground plane). Dựa trên phương pháp được đề xuất bởi tác giả Dubská [18]. Để tìm vanishing point thứ nhất sử dụng hai phép biến đổi Hough và tham số hóa PCLines được mô tả ở phần 2.2.1.2. Trước tiên ta cần xác định các điểm đặc trưng (feature point) biểu diễn góc cạnh của phương tiện giao thông, nhóm đã sử dụng thuật toán giá trị riêng tối thiểu (the minimum eigenvalue algorithm) được tác giả Shi và Tomasi đề xuất [48] để phát hiện các điểm đặc trưng trên phương tiện qua từng khung hình của một chuỗi các khung hình trong video. Sau khi các điểm là đặc trưng góc trên phương tiện được phát hiện, nhóm tiếp tục sử dụng phương pháp truy vết Kanade-Lucas-Tomasi (KLT) [49] nhằm liên kết các đặc trưng góc cạnh qua chuỗi các khung hình nhằm thể hiện quỹ đạo chuyển động của phương tiện. Ta thấy, các đường đi qua điểm có tọa độ x_t và x_{t+1} được phát hiện và theo dõi thể hiện chuyển động ở khung hình t và $t+1$ được xem như kéo dài thành các đường thẳng vô hạn song song với hướng di chuyển của vật thể. Các đường biểu diễn quỹ đạo phương tiện truy vết được sẽ giao nhau tại một điểm, điểm đó là vanishing point thứ nhất. Do đó, ta có thể sử dụng hai phép biến đổi Hough nhằm đưa các đường quỹ đạo này hội tụ tại điểm cực đại cục bộ trong không gian kim cương để xác định vị trí vanishing point thứ nhất trong mặt phẳng ảnh.



HÌNH 3.21: Minh họa các điểm đặc trưng của phương tiện được truy vết
[18]

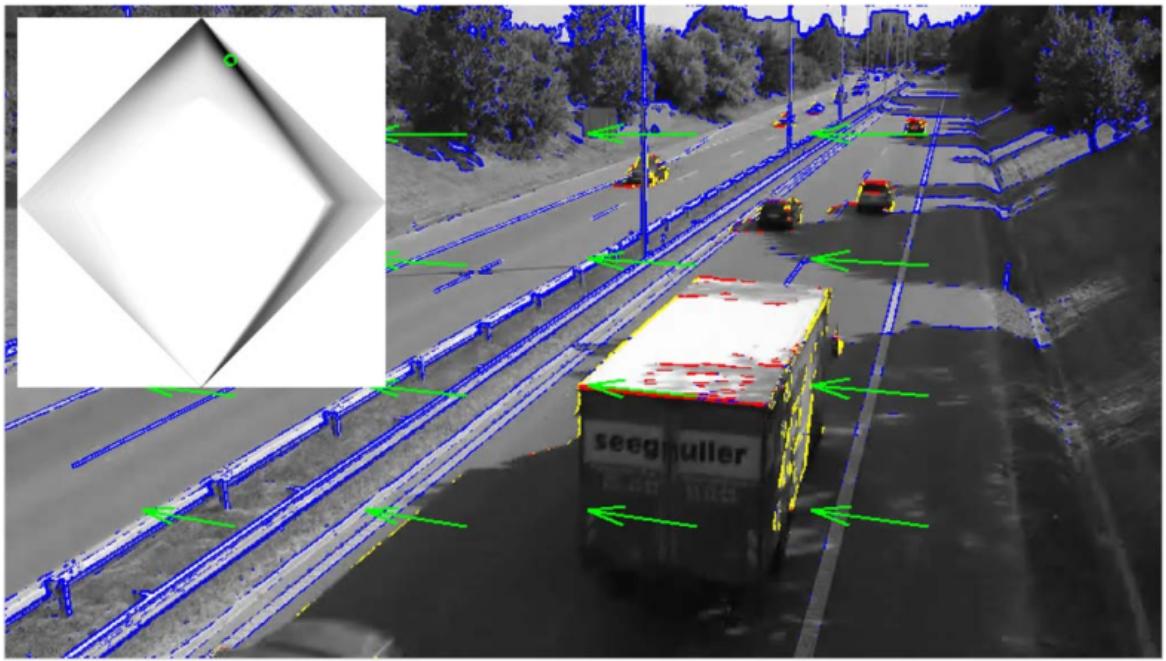
Hình 3.21 thể hiện các đặc trưng góc trên phương tiện được truy vết. Điểm màu xanh biểu diễn các đặc trưng góc được phát hiện và truy vết theo quỹ đạo chuyển động của phương tiện. Quỹ đạo chuyển động được thể hiện bằng các đường thẳng màu xanh nối đi qua hai điểm t và $t + 1$. Các điểm màu vàng là các điểm không thay đổi qua chuỗi khung hình nên sẽ có số lượng đường cắt (bình chọn) trong không gian kim cương thấp. Không gian kim cương ở góc trên trái biểu diễn điểm hội tụ của vanishing point.

2.2.1.4 Phương pháp xác định vanishing point thứ hai Vanishing point thứ hai là điểm tương ứng trên hướng song song mặt phẳng đường (ground plane) và vuông góc với hướng của vanishing point thứ nhất. Rất nhiều cạnh của phương tiện (đặc biệt là các cạnh ngang vuông góc với hướng di chuyển của phương tiện) trùng với hướng của vanishing point thứ hai. Do đó chúng ta có thể tích lũy các đường thẳng biểu diễn bởi cạnh của phương tiện, có cùng hướng với vanishing point thứ hai vào không gian kim cương và chọn (vote) ra điểm cực đại toàn cục trong không gian kim cương, thể hiện cho vanishing point thứ hai.

Để phát hiện các cạnh của phương tiện đang di chuyển một mô hình phát hiện cạnh nền (Background egde model) đã được sử dụng. Mô hình được cập nhật qua từng khung hình để tránh các hiện tượng đổ bóng và ánh sáng qua các khung hình kế tiếp thay đổi chậm. Mô hình phát hiện cạnh nền lưu trữ ở mỗi pixel tương ứng của kích thước ảnh một giá trị độ tin cậy biểu hiện một hướng của một cạnh. Với mỗi pixel (i, j) của khung hình kế tiếp, được ký hiệu là I_t có kích thước $w \times h$. Giá trị đạo hàm m và hướng của đạo hàm được xác định qua một bộ lọc cạnh dọc và cạnh ngang (phương pháp phát hiện cạnh Canny Edge detection [50]), với giá trị hướng của đạo hàm được biểu diễn thành 8 hướng. Một ảnh H_t có kích thước $w \times h \times 8$ lưu trữ của giá trị đạo hàm của mỗi pixel được tạo. Mỗi ma trận $w \times h \times 1$ của H_t lưu giữ một giá trị đạo hàm của pixel (i, j) , trong đó $H_t(i, j, k)$ lưu giữ hướng của đạo hàm theo hướng thứ k , nếu không giá trị $H_t(i, j, k) = 0$. Ban đầu ta khởi tạo mô hình phát hiện cạnh nền $B_t = H_t$ và được cập nhật qua mỗi khung hình theo công thức 3.18 với hệ số mịn α (smoothing coefficient) gần với 1 (thường giá trị α được chọn là 0.95).

$$B_t = \alpha B_{t-1} + (1 - \alpha) H_t \quad (3.18)$$

Một phép kiểm tra pixel có thuộc một cạnh của một đối tượng trong ảnh hay không được thực hiện cho tất cả các pixel. Nếu giá trị đạo hàm của pixel ở khung hình kế tiếp I_t lớn hơn một cận dưới (low threshold) τ_1 và pixel đó vượt qua phép lọc để trở thành điểm thuộc cạnh nếu sự khác nhau giữa giá trị đạo hàm theo các hướng tương ứng của pixel sau khi được cập nhật bởi mô hình B_t nhỏ hơn cận τ_2 . Các cạnh trên phương tiện sau khi được lọc nhưng có hướng kéo dài đến gần vanishing point thứ nhất hoặc có hướng xấp xỉ cùng hướng với vanishing point thứ nhất sẽ bị loại trừ khỏi quá trình tích lũy cạnh tới không gian kim cương tiếp theo. Hình 3.22 biến diễn một kết quả của phép lọc cạnh và tích lũy các cạnh vào không gian kim cương để tìm ra cực đại toàn cục tương ứng với vanishing point thứ hai.



HÌNH 3.22: Tích lũy các đặc trưng cạnh của phương tiện trong không gian kim cương để tìm vanishing point thứ hai [18]

2.2.1.5 Phương pháp xác định vanishing point thứ ba và ước tính tiêu cự camera

Vanishing point thứ ba nằm trên đường thẳng có hướng vuông góc với mặt đường (grond plane). Nhưng việc phát hiện vanishing point thứ ba rất khó khăn vì trong cảnh quay giao thông của một cung đường, số lượng cạnh tối thiểu để hỗ trợ phép biến đổi Hough có thể xác định vanishing point thứ ba rất hạn chế. Vì vậy thay vì tìm vanishing point thứ ba bằng phép biến đổi Hough, ta có thể tìm tọa độ vanishing point thứ ba trong hệ tọa độ thực tế bằng hai vanishing point và thông qua việc tìm tiêu cự của máy ảnh. Sau đó dựa vào các mối liên hệ hình học trong không gian 3D để tính tọa độ của vanishing point thứ ba.

Giả sử vị trí của gốc tọa độ của mặt phẳng ảnh nằm ở trung tâm của ảnh, gọi vanishing point thứ nhất là \mathbf{u} , vanishing point thứ hai là \mathbf{v} và gốc tọa độ là \mathbf{p} là các tọa độ trong mặt phẳng ảnh 2D, khi đó tiêu cự của máy ảnh được tính như sau:

$$\mathbf{u} = (u_x, u_y), \mathbf{v} = (v_x, v_y), \mathbf{p} = (p_x, p_y)$$

$$f = \sqrt{-(\mathbf{u} - \mathbf{p}) \cdot (\mathbf{v} - \mathbf{p})} \quad (3.19)$$

Sau khi tính được tiêu cự của máy ảnh f , vanishing point thứ ba ký hiệu \mathbf{w} có thể được tính như sau:

$$\begin{aligned}\mathbf{u}' &= (u_x, u_y, f), \mathbf{v}' = (v_x, v_y, f), \mathbf{p}' = (p_x, p_y, 0) \\ \mathbf{w}' &= (\mathbf{u}' - \mathbf{p}') \times (\mathbf{v}' - \mathbf{p}')\end{aligned}\quad (3.20)$$

Trong đó $\mathbf{u}', \mathbf{v}', \mathbf{w}', \mathbf{p}'$ đại diện cho tọa độ của vanishing point thứ nhất, thứ hai, thứ ba tương ứng và gốc tọa độ trong không gian 3D thực tế.

Một trường hợp đặc biệt khi hai vanishing point là điểm vô hạn nằm ngoài giới hạn của ảnh, khi đó ta không thể tính được tiêu cự và vanishing point thứ ba. Tuy nhiên, với việc sử dụng hệ tọa độ đồng nhất (homogeneous) và không gian kim cương để tham số hóa toàn bộ mặt phẳng vô hạn của ảnh thành không gian hữu hạn và từ đó ta có thể tìm được các điểm lý tưởng (điểm ở vô hạn) này.

2.2.2 Hiệu chỉnh thông số máy ảnh từ các vanishing point

Cần nhắc lại rằng việc hiệu chỉnh thông số thực chất là tính các tham số intrinsic (ma trận \mathbf{K}) và extrinsic (ma trận \mathbf{R} và \mathbf{T}). Phần này trình bày cách các ma trận được tính toán dựa trên kết quả tọa độ điểm biến mất (vanishing point) ở phần trên.

Với giả thuyết gốc tọa độ của ảnh nằm ở chính giữa, giá trị skew bằng 0, giá trị cần tính toán duy nhất ở \mathbf{K} là f_x và f_y . Trong đa số các phương pháp hiệu chỉnh camera, 2 giá trị f_x và f_y được giả định là bằng nhau và bằng tiêu cự f . Giá trị f này đã được tính ở phần 2.2.1.5. Gọi $(u_x, u_y), (v_x, v_y), (w_x, w_y)$ là tọa độ ảnh của 3 vanishing point và $\lambda_u, \lambda_v, \lambda_w$ là 3 scale factor tương ứng. Ma trận \mathbf{R} được quy định bởi phương trình:

$$\begin{bmatrix} \lambda_u & 0 & 0 \\ 0 & \lambda_v & 0 \\ 0 & 0 & \lambda_w \end{bmatrix} \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} u_x \lambda_u & v_x \lambda_v & w_x \lambda_w \\ u_y \lambda_u & v_y \lambda_v & w_y \lambda_w \\ \lambda_u & \lambda_v & \lambda_w \end{bmatrix} = \mathbf{KR} \quad (3.21)$$

$$\rightarrow \mathbf{R} = \begin{bmatrix} \frac{\lambda_u(u_x-p_x)}{f_x} & \frac{\lambda_u(v_x-p_x)}{f_x} & \frac{\lambda_u(w_x-p_x)}{f_x} \\ \frac{\lambda_u(u_y-p_y)}{f_y} & \frac{\lambda_u(v_y-p_y)}{f_y} & \frac{\lambda_u(w_y-p_y)}{f_y} \\ \lambda_u & \lambda_v & \lambda_w \end{bmatrix} \quad (3.22)$$

Ta biết rằng \mathbf{R} là một ma trận trực giao. Vì thế, ta có

$$\mathbf{R}\mathbf{R}^T = \mathbf{I} \quad (3.23)$$

Lấy các giá trị tại $\mathbf{I}_{13}, \mathbf{I}_{23}, \mathbf{I}_{33}$, ta có các phương trình

$$\frac{\lambda_u^2(u_x - p_x)}{f_x} + \frac{\lambda_v^2(v_x - p_x)}{f_x} + \frac{\lambda_w^2(w_x - p_x)}{f_x} = 0 \quad (3.24)$$

$$\frac{\lambda_u^2(u_y - p_y)}{f_y} + \frac{\lambda_v^2(v_y - p_y)}{f_y} + \frac{\lambda_w^2(w_y - p_y)}{f_y} = 0 \quad (3.25)$$

$$\lambda_u^2 + \lambda_v^2 + \lambda_w^2 = 1 \quad (3.26)$$

Giải 3 phương trình vừa rồi, ta có:

$$\lambda_u^2 = \frac{(p_y - w_y)(v_x - w_x) - (p_x - w_x)(v_y - w_y)}{(u_y - w_y)(v_x - w_x) - (u_x - w_x)(v_y - w_y)} \quad (3.27)$$

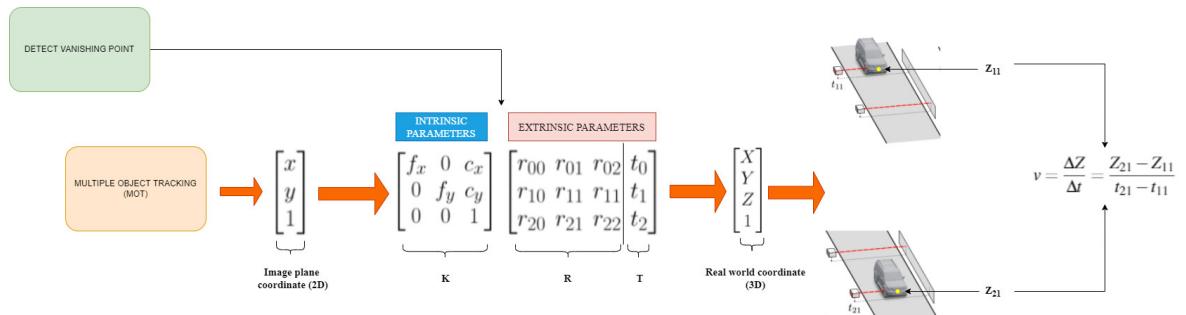
$$\lambda_v^2 = \frac{(u_y - w_y)(p_x - w_x) - (u_x - w_x)(p_y - w_y)}{(u_y - w_y)(v_x - w_x) - (u_x - w_x)(v_y - w_y)} \quad (3.28)$$

$$\lambda_w^2 = 1 - \lambda_u^2 - \lambda_v^2 \quad (3.29)$$

Có thêm giá trị $\lambda_u, \lambda_v, \lambda_w$, ta đã có thể giải phương trình 3.21 để thu được ma trận R . Để tính ma trận chuyển vị \mathbf{T} , ta cần tính thêm chiều cao H của camera so với mặt đường cũng như tọa độ ảnh (j_x, j_y) của gốc tọa độ thực tế.

$$-\mathbf{R}^{-1}\mathbf{T} = \begin{bmatrix} j_x \\ j_y \\ H \end{bmatrix} \quad (3.30)$$

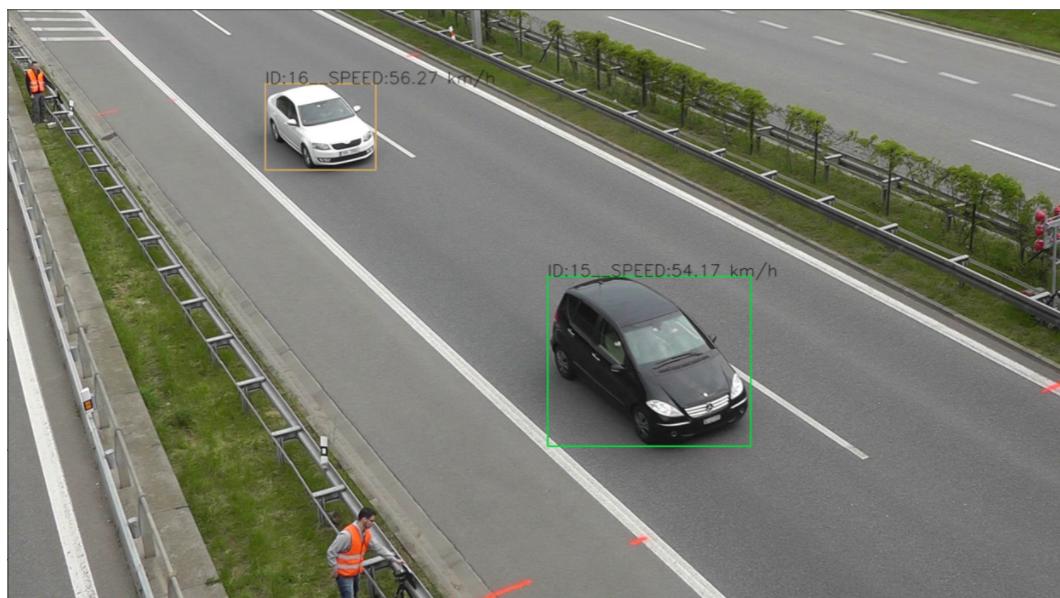
2.2.3 Ước tính tốc độ phương tiện từ các thông số máy ảnh đã được hiệu chỉnh



HÌNH 3.23: Pipeline thực hiện ước tính tốc độ tự kết quả truy vết

Quá trình ước tính tốc độ từ kết quả truy vết vật thể có thể tóm tắt như sau:

- Với mỗi hộp giới hạn trong mỗi chuỗi truy vết, lấy tọa độ ảnh điểm chính giữa cạnh dưới của hộp.
- Dùng mô hình camera tuyến tính với các ma trận **K**, **R**, **T** tính được ở trên để chuyển các tọa độ ảnh sang tọa độ không gian thực.
- Tính khoảng cách di chuyển trong tọa độ không gian thực. Lấy khoảng cách này chia cho thời gian di chuyển (dựa trên số FPS). Ta thu được tốc độ ước tính của vật thể từ khi đi vào đến khi rời khỏi khung hình.



HÌNH 3.24: Một frame từ video demo

Chương 4

KẾT QUẢ ĐẠT ĐƯỢC

1 Thí nghiệm đánh giá các phương pháp truy vết vật thể trong video

Các phương pháp truy vết vật thể trong video (Multiple Object Tracking) hiện nay đều được đánh giá và công bố chủ yếu trên bộ dataset **MOT challenge** bao gồm 4 bộ dataset MOT15[51], MOT16 [52], MOT17 (mở rộng của MOT16), MOT20 [53], đối tượng vật thể mà các phương pháp truy vết là người đi bộ di chuyển trên đường hoặc ở những địa điểm đông người như trong khu thương mại,... vì vậy đối tượng truy vết trong bộ dữ liệu này có tốc độ di chuyển thấp, đa số đối tượng có thể bắt trọn so với kích thước một kích thước của khung hình, số lượng đối tượng bị chồng lấp với đối tượng khác trong một khung hình xảy ra thường xuyên với đa số video có điều kiện môi trường ánh sáng có cường độ cao. Tuy nhiên với mục tiêu xây dựng hệ thống truy vết vật thể trong hệ thống giám sát giao thông, dữ liệu của bộ dữ liệu MOT challenge không phù hợp để đánh giá các phương pháp xác định và truy vết vật thể, vì đối tượng truy vết trong dữ liệu video giao thông đa dạng hơn như xe ô tô, xe buýt, xe container,... các đối tượng di chuyển với tốc độ cao, những đối tượng có kích thước lớn như xe container thì kích thước của xe có thể vượt qua kích thước khung hình cố định của máy ghi hình và các phương pháp truy vết vật thể phải giải quyết khi dữ liệu được ghi hình ở nhiều môi trường thực tế như trời nắng, trời nhiều mây, hoặc trời tối. Qua tìm hiểu và khảo sát các bộ dữ liệu đánh giá phương pháp truy vết vật thể (MOT) đã được công bố khoa học nhóm chúng tôi nhận thấy bộ dữ liệu UA-DETRAC[19]

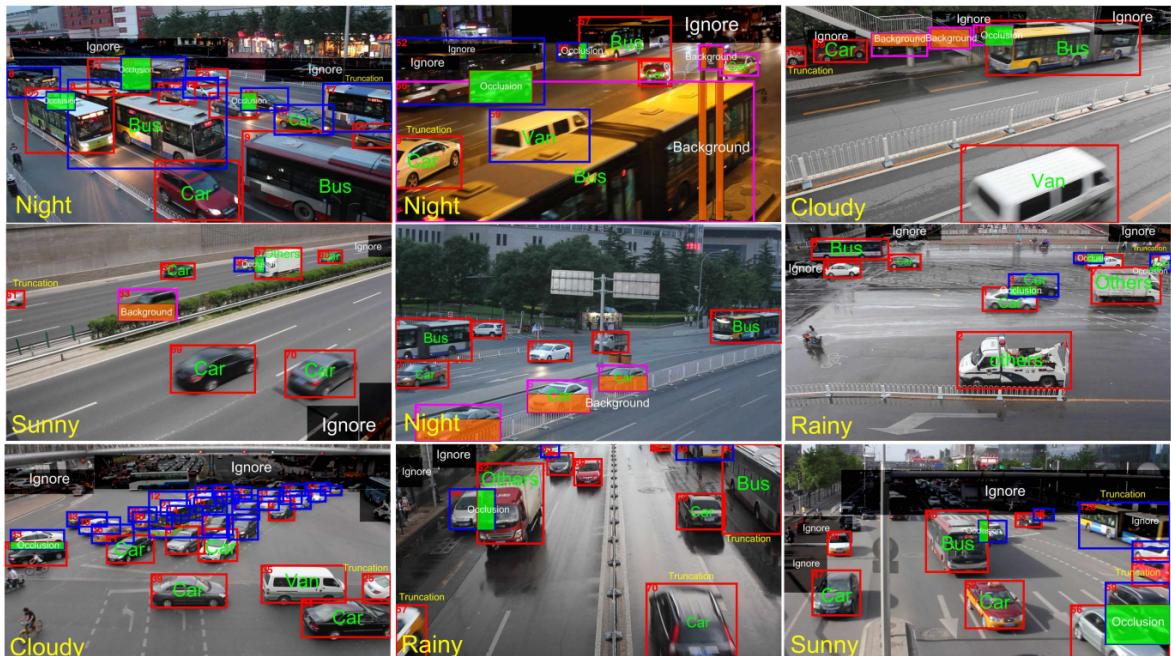
phù hợp với điều kiện thực tế của môi trường nên nhóm chúng tôi đã sử dụng bộ dữ liệu này để đánh giá các phương pháp truy vết vật thể (MOT) mà nhóm đã nghiên cứu.

1.1 Tổng quan về bộ dữ liệu UA-DETRAC

Bộ dữ liệu UA-DETRAC bao gồm 100 video được thu thập bởi camera Canon EOS 550D ở 24 vị trí đặt camera khác nhau. Các vị trí ghi hình thể hiện đa dạng đoạn đường và điều kiện lưu thông như trên đường cao tốc, tại các nút giao giao thông nơi có mật độ phương tiện cao. Để đảm bảo việc mô phỏng môi trường giao thông gần nhất với thực tế để đánh giá mô hình truy vết vật thể chính xác nhất có thể, bộ dữ liệu được ghi hình trong nhiều điều kiện thời tiết khác nhau với điều kiện ánh sáng khác nhau và ở nhiều vị trí góc đặt camera khác nhau. Các đoạn video được ghi hình ở tốc độ 25 khung hình trên giây (25 fps), mỗi khung hình được trích xuất thành ảnh với kích thước 960 x 540 pixels. Toàn bộ video của bộ dữ liệu UA-DETRAC được chia thành 60 video cho tập huấn luyện và 40 video cho tập kiểm thử. Các video trong tập huấn luyện và kiểm thử đã được nhóm tác giả đảm bảo tương tự nhau về sự đa dạng tình trạng giao thông và điều kiện môi trường.

Để đảm bảo bounding box và nhãn của vật thể được gán nhãn chính xác, 10 chuyên gia gán nhãn đã cùng tiến hành gán nhãn và các nhãn được gán được trải qua nhiều vòng kiểm tra chéo nhằm phát hiện lỗi sai trong quá trình gán nhãn. Qua đó với 140000 khung hình của dữ liệu UA-DETRAC đã gán nhãn cho 8250 phương tiện giao thông và có tổng cộng hơn một triệu bounding box đã được gán nhãn. Tuy nhiên một số vùng trong khung hình không xét trong đoạn đường quan tâm và các vật thể có kích thước bounding box quá nhỏ không thể gán nhãn sẽ bị bỏ qua (minh họa ở nhãn "ignore" ở hình 4.1). Các nhãn bounding box và nhãn tracking đảm bảo cho bộ dữ liệu có thể đánh giá các mô hình, thuật toán cho hai bài toán, phát hiện (Object Detection) và truy vết (Object Tracking) vật thể. Phương tiện được gán nhãn tác giả gán nhãn chủ yếu là xe có 4 bánh trở lên như xe ô-tô, xe buýt,... Nhìn vào hình 4.1, minh họa về các trường hợp xảy ra trong quá trình gán nhãn. Màu của bounding box cho biết mức độ chồng lấp của vật thể: màu đỏ là không bị chồng lấp và có thể thấy vật thể hoàn toàn, màu xanh lam biểu thị cho phương tiện bị che lấp một phần bởi phương tiện khác, màu hồng biểu thị cho phương tiện bị che lấp bởi các vật cản tầm nhìn trên đường; vùng tô kín màu xanh lục thể hiện cho vùng diện tích vật thể bị chồng lấp bởi

phương tiện khác, vùng tô kín màu cam thể hiện vùng diện tích vật thể bị chồng lấp bởi các vật cản trên đường điều kiện thời tiết được hiển thị ví dụ ở góc trái dưới màn hình



HÌNH 4.1: Minh họa các khung hình được gán nhãn của bộ dữ liệu UA-DETRAC. [19]

	Số khung hình	Số nhãn bounding box	Số nhãn tracking
Training set	84000	578000	5900
Testing set	56000	632000	2300

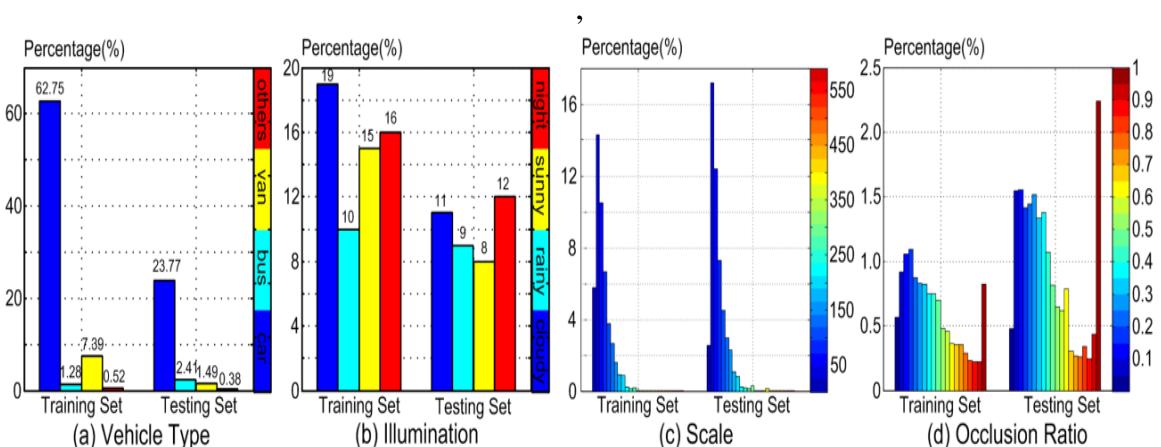
BẢNG 4.1: Bảng dữ liệu thống kê thông tin các nhãn vị trí vật thể và nhãn tracking vật thể ở tập huấn luyện và tập kiểm thử

Để phân tích, đánh giá tính chính xác của các mô hình, thuật toán xác định vị trí vật thể (Object Detection) và gán nhãn truy vết (Object Tracking), nhóm tác giả đã gán nhãn bộ dữ liệu có một số đặc điểm để phù hợp với bộ dữ liệu giao thông là:

- Loại phương tiện (Vehicle type): Nhãn được gán bao gồm 4 loại: *car*, *bus*, *van* và *others* (*loại khác*), nhãn *others* bao gồm các loại xe như xe tải, xe bồn,... Phân phối của các loại phương tiện trong bộ dữ liệu được thể hiện ở hình 4.2(a).
- Điều kiện ánh sáng (Illumination): Bộ dữ liệu được thu thập ở 4 điều kiện sáng khác nhau là *cloudy* (*trời nhiều mây*), *night* (*trời tối*), *sunny* (*trời nắng*), *rainy*

(trời mưa). Phân phối điều kiện sáng xuất hiện trong bộ dữ liệu thể hiện ở hình 4.2(b).

- Tỉ lệ kích thước bounding box (Scale): Mỗi bounding box được gán nhãn là một hình chữ nhật bao quanh vùng pixel của vật thể. Và nhãn tỉ lệ cho mỗi vùng diện tích bounding box được nhóm tác giả chia thành 3 loại: *small*(0 - 50 pixels), *medium scale* (50 - 150 pixels), *large scale* (lớn hơn 150 pixel). Phân phối diện tích của các vật thể trong bộ dữ liệu thể hiện ở hình 4.2(c).
- Tỉ lệ diện tích các bounding box bị chồng chéo lên nhau (Occulasion ratio): Phần diện tích bị chồng lấp của các bounding box được chia làm 3 loại nhãn: *no occlusion* (không bị chồng lấp), *partial occlusion* (bị chồng lấp một phần), và *heavy occlusion* (đa số diện tích của vật thể bị chồng lấp). Trong đó nhãn *partial occlusion* được gán khi phần diện tích bị chồng lấp từ 1% - 50 % và nhãn *heavy occlusion* được gán nhãn khi phần diện tích bị chồng lấp trên 50%. Phân phối tỉ lệ diện tích của bounding box trong bộ dữ liệu thể hiện ở hình 4.2(d).
- Tỉ lệ bounding box của vật thể nằm ngoài kích thước khung hình (Truncation ratio): Bởi vì tới một thời điểm khi vật thể di chuyển ra ngoài khung hình hoặc kích thước của vật thể quá lớn kích thước khung hình không thể chứa toàn bộ phần diện tích của vật thể vì vậy khi phần diện tích của vật thể không xuất hiện trong khung hình lớn hơn 50% (Truncation ratio > 50%) thì nhãn của vật thể đó bị bỏ qua.



HÌNH 4.2: Thông số thống kê các đặt trưng của bộ dữ liệu UA-DETRAC trên bộ dữ liệu huấn luyện (training set) và bộ dữ liệu kiểm thử (testing set). [19]

1.2 Độ đo được sử dụng để đánh giá cho bài toán truy vết vật thể trong video

Để so sánh sự chính xác của các mô hình, thuật toán nhóm chúng tôi đã sử dụng độ đo **MOTA**[7], **IDF1**[8] và đặc biệt là độ đo **HOTA**[9] để đánh giá tính chính xác của mô hình, thuật toán truy vết vật thể. Bài toán truy vết vật thể (Multiple Object Tracking) tồn tại hai khía cạnh cần được đánh giá độ chính xác của mô hình là phát hiện vật thể (Object detection) và gán nhãn vật thể phát hiện được. Các độ đo trên hiện nay đang được sử dụng rộng rãi để đánh giá các mô hình truy vết vật thể trong nghiên cứu, tuy nhiên độ đo **MOTA** và **IDF1** có khuynh hướng (bias) chỉ đánh giá một khía cạnh của bài toán hơn khía cạnh còn lại cần được đánh giá, nên nhóm chúng tôi đã sử dụng đã sử dụng thêm độ đo **HOTA** để đánh giá hiệu năng của các mô hình, độ đo này được nghiên cứu và công bố cuối năm 2020 nhằm mục đích cân bằng việc đánh giá hai khía cạnh của bài toán, mặc dù độ đo này hiện tại được sử dụng là độ đo chính cho **MOT challenge** [51, 52, 53] nhưng vẫn chưa được sử dụng nhiều bằng hai độ đo trước, nên nhóm chúng tôi sẽ báo cáo song song cả ba độ đo ở mục 1.3.

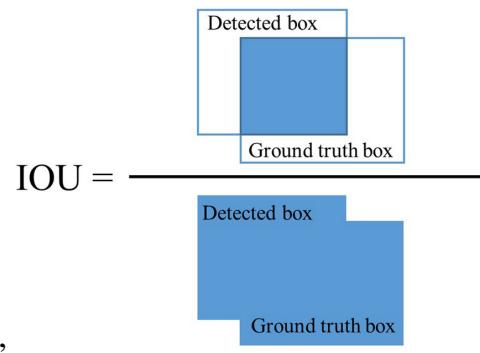
Một số loại nhãn vật thể ở tập kiểm thử (MOT Ground-Truth) và kết quả dự đoán của mô hình truy vết vật thể: Nhãn truy vết vật thể cho ở tập kiểm thử (ground truth tracks) ở mỗi khung hình (frame) đại diện cho hai đại lượng là tập các nhãn vị trí vật thể (Detections - gtDets) và tập các nhãn truy vết vật thể, mỗi nhãn vật thể được đánh số khác nhau từ 1 tới hết số lượng vật thể quan tâm (Tracking index - gtIDs) và nhãn truy vết vật thể qua nhiều khung hình sẽ xác định một quỹ đạo di chuyển duy nhất của vật thể (Trajectories index - gtTrajs) vì vậy nhãn quỹ đạo là duy nhất trong video. Tương tự với nhãn ở tập kiểm thử ở tập dự đoán cũng bao gồm ba loại nhãn, ở mỗi khung hình là nhãn vị trí vật thể (prDets) và nhãn truy vết vật thể (prIDs), xét với toàn bộ video là nhãn quỹ đạo di chuyển của vật thể.

Các loại lỗi của bài toán truy vết vật thể (MOT) trong video (Tracking Errors): Các lỗi có thể xảy ra khi các mô hình truy vết vật thể dự đoán nhãn tracking vật thể thường được được chia làm 3 loại và được sử dụng rộng rãi trong các nghiên cứu để đánh giá tính chính xác của mô hình truy vết vật thể: lỗi phát hiện (Detection error), lỗi vị trí bounding box (Localisation errors), lỗi liên kết (Association errors).

- Lỗi phát hiện (Detection error) xảy ra khi mô hình dự đoán nhãn vị trí (prDet) vật thể không xuất hiện trong tập kiểm thử (ground truth) hoặc dự đoán sai bounding

box của vật thể.

- Lỗi liên kết (Association errors) xảy ra khi mô hình dự đoán cùng một nhãn truy vết (prID) cho hai bounding box có nhãn truy vết (gtID) khác nhau trong tập kiểm thử (ground truth).
- Lỗi cục bộ (Localisation error) xảy ra khi vị trí của bounding box (prDet) dự đoán không chính xác hoàn toàn với vị trí bounding box (gtDet) ở tập kiểm thử (ground truth).



HÌNH 4.3: Minh họa cách xác định giá trị S bằng hàm Jaccard Index (IoU)
Nguồn: internet

So sánh song ánh (Bijective Matching): là cách tiến hành so sánh lần lượt (one-to-one) sự tương đồng giữa tập nhãn vị trí vật thể ở tập kiểm thử (gtDets) và tập nhãn vị trí vật thể ở tập dự đoán (prDets), nhằm đảm bảo một nhãn gtDet chỉ được tương ứng với một nhãn prDet và ngược lại. Gọi sự tương đồng giữa nhãn vị trí vật thể ở hai tập dự đoán và kiểm thử là S để so sánh sự tương đồng giữa hai nhãn ta sử dụng thuật toán Hungarian (mục 1.1.1) để tìm ra cặp nhãn có S nhỏ nhất thỏa mãn điều kiện $S > \alpha$, với α là điểm chặn dưới (threshold) của phần diện tích giao nhãn của hai bounding box. Có hai cách thường được dùng để tìm giá trị tương đồng của cặp nhãn S :

- So sánh sự tương đồng giữa hai vec-tơ đặc trưng (feature vector) của vật thể sau khi xác định vị trí bounding box của vật thể bằng các độ đo như cosine similarity, L1, L2,...
- Sử dụng độ đo *Jacard Index* hay còn gọi là *IoU* để xác định diện tích giao nhau giữa nhãn vị trí (bounding box) của từng cặp nhãn prDet và gtDet. Ta có công

thức của độ đo IoU (Minh họa ở hình 4.3):

$$IoU = \frac{|TP|}{|TP| + |FN| + |FP|} \quad (4.1)$$

Qua quá trình so sánh từng cặp nhãn trong tập prDets và gtDets bằng độ đo IoU , ta sẽ tìm được các cặp nhãn prDet và gtDet có vị trí bounding box giống nhau nhất và gọi số lượng các cặp đó là true positives (TP). Các nhãn ở trong tập kiểm thử gtDet không tìm được nhãn phù hợp được coi là false negatives (FN). Bất kỳ nhãn nào ở tập dự đoán prDet không tìm được nhãn phù hợp ở tập kiểm thử thì được coi là false positives (FP). Các nhãn được coi là FN hoặc FP thể hiện độ lỗi trong việc dự đoán của mô hình.

1.2.1 Độ đo MOTA[7]

Độ lỗi trong việc xác định vị trí bounding box của vật thể được xác định bằng số lượng nhãn bounding box là FN và FP được mô tả ở phần so sánh song ánh ở trên (Bijective Matching).

Độ lỗi trong việc liên kết các vật thể nhằm tạo thành một quỹ đạo được tính bằng số lần chuyển đổi nhãn truy vết của vật thể (Identity Switch - IDS). Một IDSW xảy ra khi mô hình truy vết vật thể trao đổi nhãn của hai vật thể hay có thể nói cách khác một IDSW là một TP của nhãn vị trí vật thể nhưng nhãn truy vết prID khác với nhãn truy vết của vật thể trước. Ví dụ ở khung hình thứ t vật thể A có nhãn truy vết (Tracking ID) là 1, và vật thể B được gán nhãn là 2, nhưng ở khung hình thứ $t+1$ vật thể A ở khung hình t lại được gán nhãn là 2, còn vật thể B lại được gán nhãn 1, xem hình minh họa 4.4



HÌNH 4.4: Minh họa trường hợp nhãn truy vết vật thể (IDSW) thay đổi trong quỹ đạo di chuyển của vật thể.

Công thức độ đo MOTA được tính dựa trên hai loại lỗi của bài toán truy vết vật thể được mô tả ở trên với, lỗi phát hiện vị trí vật thể (detection error) đại diện bởi số lượng nhãn vị trí bounding box mà trong tập kiểm thử và tập dự đoán không tìm được nhãn phù hợp là số lượng nhãn FN và số lượng nhãn FP. Lỗi liên kết được xác định bằng tổng số lần IDSW của các nhãn quỹ đạo trong tập kiểm thử. Kết quả của độ đo MOTA được tính bằng cách tổng số lần xảy ra ba sai số mô tả ở trên và chia cho số lượng nhãn vị trí vật thể và nhãn truy vết vật thể thuộc các khung hình trong video ở tập kiểm thử, rồi lấy 1 trừ đi. Công thức độ đo **MOTA**:

$$MOTA = 1 - \frac{|FN| + |FP| + |IDSW|}{|gtDet|} \quad (4.2)$$

1.2.2 Độ đo IDF1[8]

Khác với độ đo MOTA thực hiện phép đo song ánh giữa tập các nhãn vị trí bounding box của tập kiểm thử với đầu ra của mô hình và sai số liên kết quỹ đạo di chuyển xác định bởi số lượng nhãn truy vết (prID) được gán sai giữa khung hình đang xét hiện tại và khung hình liền trước nó. Độ đo IDF1 thực hiện so sánh lần lượt (Bijective matching) từng nhãn quỹ đạo của tập kiểm thử (gtTraj) và nhãn quỹ đạo dự đoán của mô hình (prTraj) giống với gtTraj qua việc so sánh số lượng khung hình gtTraj và số lượng khung hình của prTraj. Ví dụ một vật thể ở tập kiểm thử có gtTraj = 1 và có độ dài các khung hình có nhãn gtTraj = 10, nếu một vật thể được mô hình dự đoán nhãn truy vết prID = 1 và có tổng cộng 8 khung hình xuất hiện nhãn prID = 1, nếu ta chọn threshold $\alpha = \text{len(gtTraj)}/2$, ta có độ tương đồng $S = 10 - 8 > 5$ nên ta có thể coi nhãn quỹ đạo dự đoán của vật thể là prTraj = 1 và nhãn đó là một

đại lượng quỹ đạo dự đoán đúng (ID true positive - IDTP). Tương tự với nhãn FN và FP của so sánh nhãn vị trí vật thể (Detections) như trên ta có hai đại lượng ID false negative - IDFN ở tập kiểm thử có tập nhãn gtID không xuất hiện trong tập dự đoán và đại lượng ID false porisitive - IDFP khi tập nhãn prID mà mô hình dự đoán không tồn tại trong tập kiểm thử (Ground truth).

Từ đó ta có công thức tính độ đo IDF1:

$$IDF1 = \frac{|IDTP|}{|IDTP| + 0.5|IDFN| + 0.5|IDFP|} \quad (4.3)$$

1.2.3 Độ đo HOTA[9]

Đánh giá kết quả mô hình dự đoán vị trí bounding box của vật thể với vị trí của bounding box của vật thể trong tập kiểm thử: Tương tự với cách đo của độ đo MOTA, độ đo HOTA bằng phép so sánh song ánh và phép đo tương đồng IoU để đo sự chính xác ở mức phát hiện vị trí vật thể (Detections) ở tập nhãn được dự đoán và tập nhãn ở trong tập kiểm thử bằng cách xác định ba đại lượng TP, FP và FN. Một true positive (TP) xuất hiện khi độ tương đồng của cặp nhãn gtDet và prDet, $S > \alpha(DetectionThreshold)$. Một dự đoán được gán nhãn false positive (FP) khi nhãn gtDet không giống với nhãn prDet trong tập kiểm thử và một false negative (FP) xảy ra khi nhãn prDet không liên kết được với nhãn nào trong tập nhãn dự đoán của mô hình. TP, FP, FN được sử dụng để đánh giá độ chính xác vị trí của vật thể.

Đánh giá độ chính xác của các nhãn truy vết vật thể (prID) được gán cùng một nhãn quỹ đạo Tương tự nhãn vị trí bounding box của vật thể, HOTA cũng đo độ lỗi nhãn quỹ đạo di chuyển của vật thể mà các mô hình truy vết vật thể đã dự đoán bằng ba đại lượng, True Positive Association (TPA), False Positive Association (FPA), False Negative Asociation (FNA). TPA, FPA và FNA được xác định từ tập được các nhãn vị trí bounding box mà mô hình dự đoán chính xác khi so sánh với tập kiểm thử (TPs).

- Xét với một tập nhãn bounding box chính xác (TP) trong toàn bộ một 1 video, ký hiệu là c , nếu mỗi nhãn truy vết dự đoán (prID) giống tương ứng với một nhãn truy vết ở tập kiểm thử (gtID) thì nhãn dự đoán vết của vật thể đó tại khung hình đang xét là 1 TPA, tập hợp các TPA trong c ta được một tập nhãn vật thể được liên kết chính xác (TPAs), công thức 4.4:

$$TPA(c) = \{k \mid k \in \{TP \mid prID(k) = prID(c) \wedge gtID(k) = gtID(c)\}\} \quad (4.4)$$

- Xét với một tập nhãn bounding box chính xác (TP) trong toàn bộ 1 video, ký hiệu là c , nếu mỗi nhãn truy vết dự đoán (prID) khác với toàn bộ nhãn truy vết ở tập kiểm thử (gtIDs) hoặc không xuất hiện thì nhãn dự đoán vết của vật thể đó tại khung hình đang xét là 1 FNA, tập hợp các FNA trong c ta được một tập nhãn vật thể liên kết không chính xác (FNAs), công thức 4.5:

$$FNA(c) = \{k\} \quad (4.5)$$

$$k \in \{TP \mid prID(k) \neq prID(c) \wedge gtID(k) = gtID(c)\} \cup \{FN \mid gtID(k) = gtID(c)\}$$

- Xét với một tập nhãn bounding box chính xác (TP) trong toàn bộ 1 video, ký hiệu là c , nếu mỗi nhãn truy vết trong tập kiểm thử (gtID) khác với toàn bộ nhãn truy vết ở tập kiểm thử (gtIDs) n thì nhãn dự đoán vết của vật thể đó tại khung hình đang xét là 1 FPA, tập hợp các FPA trong c ta được một tập nhãn vật thể liên kết không chính xác (FPAs), công thức 4.6:

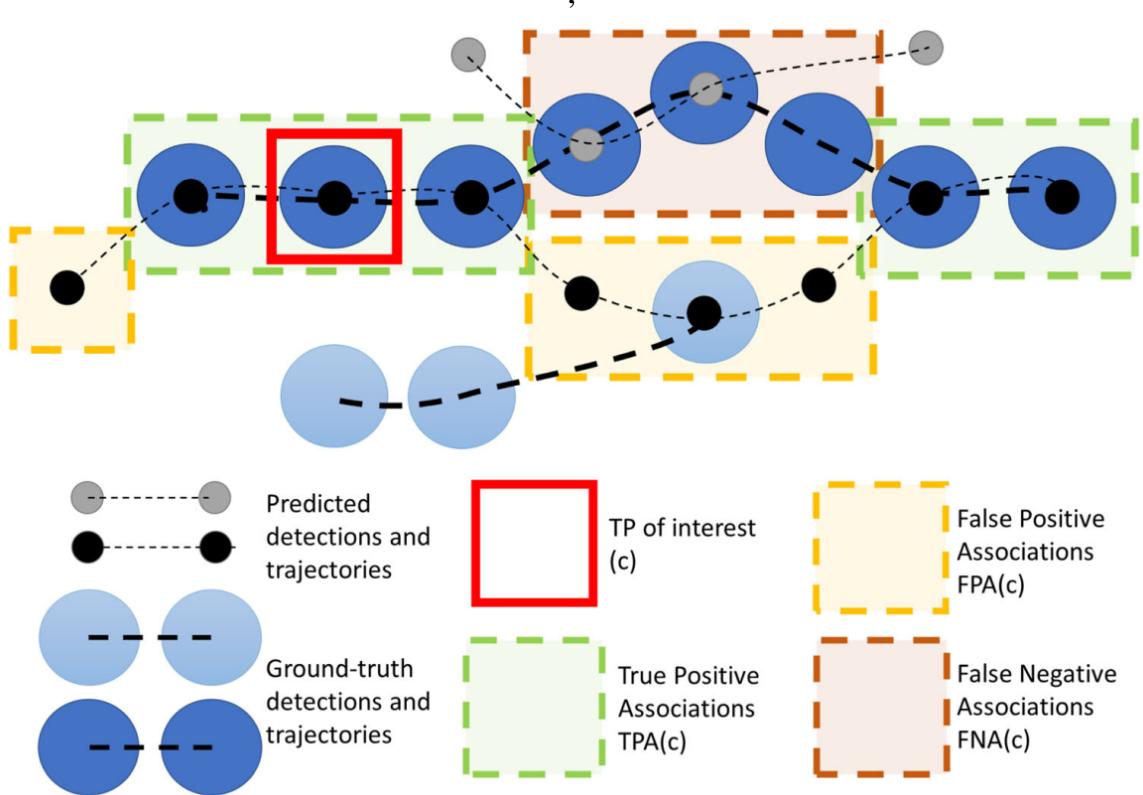
$$FPA(c) = \{k\} \quad (4.6)$$

$$k \in \{TP \mid prID(k) = prID(c) \wedge gtID(k) = gtID(c)\} \cup \{FP \mid prID(k) = prID(c)\}$$

Độ lỗi cục bộ của toàn bộ vật thể trong toàn bộ video trong tập dữ liệu được tính bởi công thức 4.7, 4.8 với chặn dưới α (threshold).

$$\mathcal{A}(c) = \frac{|TPA(c)|}{|TPA(c)| + |FNA(c)| + |FPA(c)|} \quad (4.7)$$

$$HOTA_\alpha = \sqrt{\frac{\sum_{c \in TP} \mathcal{A}(c)}{|TP| + |FN| + |FP|}} \quad (4.8)$$



HÌNH 4.5: Minh họa cách xác định TPA, FPA, FNA [9]

Để cân bằng và đánh giá ảnh hưởng của việc lựa chọn α , ta có lấy trung bình các giá trị α của công thức HOTA cục bộ, ta được công thức độ đo HOTA tổng quát như công thức 4.9:

$$HOTA = \int_0^1 HOTA_\alpha \, d\alpha \approx \frac{1}{19} \sum_{\alpha \in \{0.05, 0.1, \dots, 0.9, 0.95\}} HOTA_\alpha \quad (4.9)$$

1.3 Kết quả đánh giá

Các phương pháp truy vết vật thể được đánh giá trên tập kiểm thử của bộ dữ liệu UA-DETRAC[19]. Độ chính xác của các mô hình được đánh giá bởi 3 độ đo chính: HOTA[9], MOTA[7], IDF1[8]. Ngoài ra, 3 thông số khác là False Positive, False Negative, ID Switch cũng được cung cấp để bổ sung thông tin đánh giá. Các kết quả độ chính xác sẽ tương ứng với các mô hình và bộ trọng số cụ thể. Kết quả này không dùng để so sánh bản thân các phương pháp.

1.3.1 Thí nghiệm đánh giá độ chính xác

Separated Detection and Tracking (SDE) Ở bước phát hiện vật thể chúng tôi sử dụng 2 phương pháp là Faster RCNN[54] và YOLOV5[24]. Mỗi phương pháp được đánh giá bằng 2 phiên bản: mô hình huấn luyện trên tập dữ liệu COCO[55] và mô hình huấn luyện lại trên tập huấn luyện của UA-DETRAC[19] (tổng cộng 4 mô hình phát hiện vật thể). Với YOLOV5, chúng tôi sử dụng cài đặt trên Pytorch của phiên bản YOLOV5L6; với Faster RCNN, chúng tôi sử dụng cài đặt của torchvision. Mô hình Resnet50-FPN được dùng làm mạng backbone cho Faster RCNN. Chúng tôi huấn luyện lại 2 phương pháp trên UA-DETRAC trong 1 epoch với các thông số được mô tả ở bảng 4.2

Phương pháp	YOLOV5L6	Faster RCNN
Kích cỡ đầu vào	640x640	960x540
Learning rate gốc	0.01	0.0001
Thuật toán tối ưu hóa	SGD	SGD
Tăng cường ảnh	Random Horizontal Flip, Random Translation, Random HSV, Mosaic	Random Horizontal Flip
Kích cỡ batch	4	4

BẢNG 4.2: Cài đặt huấn luyện mô hình phát hiện vật thể

Ở bước truy vết vật thể, các phương pháp được đánh giá bao gồm: IoUTracker[1], VIOUTracker[12], SORT[2], DEEPSORT[3]. Phương pháp truy vết ảo được sử dụng cho VIOUTracker là medianflow[41]. Mô hình trích xuất đặc trưng cho DEEPSORT là một mạng Resnet gồm 2 lớp tích chập và 6 lớp residual. Mô hình Resnet này được huấn luyện trên bộ dữ liệu MARS[56]. Với 4 mô hình phát hiện và 4 mô hình truy vết vật thể, ta thu được kết quả của 16 tổ hợp cho nhóm phương pháp Separated Detection and Tracking. Các kết quả độ đo nêu trên được tính theo cài đặt của tác giả bài báo HOTA[9]¹.

¹<https://github.com/JonathonLuiten/TrackEval>

Tên phương pháp phát hiện vật thể	Tên phương pháp truy vết vật thể	HOTA	MOTA	IDF1	FP	FN	IDSW
YOLOV5L6 (COCO)	IoUTracker	42.1	26.64	42.18	13693	481967	65
YOLOV5L6 (COCO)	VIoUTracker	45.5	29.36	46.54	18020	459277	26
YOLOV5L6 (COCO)	SORT	48.63	43.04	53.96	45414	337098	2397
YOLOV5L6 (COCO)	DEEPSORT	51.83	44.9	61.03	56053	315284	1002
YOLOV5L6 (UA-DETRAC)	IoUTracker	56.25	35.72	66.27	341400	92277	678
YOLOV5L6 (UA-DETRAC)	VIoUTracker	55.48	27.4	66.26	411860	78304	394
YOLOV5L6 (UA-DETRAC)	SORT	52.22	11.11	60.31	523070	76339	1230
YOLOV5L6 (UA-DETRAC)	DEEPSORT	50.36	0.97	59.38	590523	77497	1139
FRCNN (COCO)	IoUTracker	35.06	-177.11	39.17	1622547	32196	3161
FRCNN (COCO)	VIoUTracker	29.48	-290.2	28.42	2294720	31023	8756
FRCNN (COCO)	SORT	30.63	-176.75	26.81	1803466	56251	10478
FRCNN (COCO)	DEEPSORT	43.96	-49.29	49.9	943583	63056	2255
FRCNN (UA-DETRAC)	IoUTracker	57.64	48.11	69.46	274151	75348	1161
FRCNN (UA-DETRAC)	VIoUTracker	57.7	42.73	70.07	320994	65266	694
FRCNN (UA-DETRAC)	SORT	56.05	37.59	67.1	349296	71097	1299
FRCNN (UA-DETRAC)	DEEPSORT	57.29	45.75 65	70.83	283064	82525	1007

BẢNG 4.3: Kết quả độ chính xác các phương pháp SDE

Joint Detection and Tracking Có 4 phương pháp thuộc nhóm này được tìm hiểu và đánh giá: TransTrack[57], CenterTrack[4], FairMOT[5], Tracktor[6]. Bởi vì các mô hình có sẵn của các phương pháp này đều được huấn luyện trên dữ liệu người đi bộ nên chúng bắt buộc phải được huấn luyện lại trên dữ liệu giao thông UA-DETRAC. Riêng với Tracktor[6], chúng tôi đánh giá 4 phiên bản bao gồm Tracktor, Tracktor kết hợp dự đoán chuyển động, Tracktor kết hợp trích xuất đặc trưng, Tracktor++ (kết hợp cả 2 mở rộng ở trên).

Phương pháp	CenterTrack	FairMot	Tracktor
Kích cỡ đầu vào	960x544	1088x608	960x540
Learning rate gốc	0.000125	0.0001	0.0001
Thuật toán tối ưu hóa	Adam[58]	Adam	SGD
Tăng cường ảnh	Random Horizontal Flip, random Resized Crop, Color Jittering	Transform RGB to HSV	Random Horizontal Flip, Random Crop
Kích cỡ batch	8	8	4

BẢNG 4.4: Cài đặt huấn luyện mô hình JDE

Tên mô hình	HOTA	MOTA	IDF1	FP	FN	IDSW
CenterTrack	57.01	45.59	68.97	184227	182992	451
FairMot	45.69	29.59	54.28	106803	368360	587
Tracktor	60.28	54.24	70.89	219011	87540	2662
Tracktor + Alignment	60.29	54.23	70.88	219116	87567	2610
Tracktor + Reid	61.91	54.56	75.2	219011	87540	509
Tracktor++	61.88	54.54	75.08	219116	87567	512

BẢNG 4.5: Kết quả độ chính xác các phương pháp JDE

Nhận xét:

- Kết quả thí nghiệm cho thấy những cải tiến của việc sử dụng mô hình Reid (DEEPSORT > SORT, Tracktor + Reid > Tracktor).
- Với các phương pháp thuộc nhóm SDE, độ hiệu quả của mô hình phát hiện đối tượng có thể ảnh hưởng rất lớn đến kết quả truy vết. Bằng việc huấn luyện lại YOLOV5L6 và Faster RCNN trên dữ liệu giao thông, kết quả của các phương pháp SDE đều được cải thiện đáng kể.

-
- Các phương pháp đơn giản như IoUTracker, SORT vẫn cho kết quả có thể so sánh với các phương pháp hiện đại hơn. Điều này là hợp lý khi UA-DETRAC có tỉ lệ FPS cao và ít có hiện tượng che lấp.

1.3.2 Thí nghiệm đo tốc độ

Nhằm cung cấp một cái nhìn tổng quan về mức độ khả dụng của các phương pháp nêu trên khi áp dụng vào ứng dụng thực tế, chúng tôi đã tiến hành đo đạc kết quả tốc độ xử lý. Mỗi phương pháp được tiến hành đo trên 2 loại phần cứng khác nhau: Intel(R) Xeon(R) CPU @ 2.20GHz và GPU Tesla P100. Ngoài ra, nhằm giảm bớt sự phụ thuộc của tốc độ xử lý vào cách cài đặt của các tác giả, thời gian sẽ chỉ tính với tác vụ xử lý của vi xử lý và các công đoạn lưu, đọc dữ liệu sẽ không được tính vào kết quả. Bởi vì trong tương lai, ta có thể tổng hợp các phương pháp này vào một dự án và thống nhất cách lưu, đọc dữ liệu. Bằng cách này, sự khác biệt về thời gian xử lý cho tác vụ lưu, đọc dữ liệu có thể bỏ qua.

Tên mô hình	Framework học sâu	Tốc độ trên CPU(FPS)	Tốc độ trên GPU(FPS)
YOLOV5L6	Pytorch 1.10.0	0.57	12
FRCNN	Pytorch 1.10.0	0.08	10.8
IoUTracker	Không sử dụng	2426	2666
VIOUTracker	Không sử dụng	3	3
SORT	Không có	167	209
DEEPSORT	Tensorflow 1.15	5	36
CenterTrack	Pytorch 1.4.0	0.11	14
Tracktor	Pytorch 1.10.0	0.07	13
Tracktor+Alignment	Pytorch 1.10.0	0.06	13
Tracktor+Reid	Pytorch 1.10.0	0.05	12
Tracktor++	Pytorch 1.10.0	0.05	11

BẢNG 4.6: Kết quả tốc độ các mô hình

Các kết quả tốc độ được tính trung bình trên toàn bộ tập kiểm thử của UA-DETRAC. Đối với các phương pháp SDE gồm IoUTracker, VIOUTracker, SORT, DEEPSORT, chúng ta cần cung cấp sẵn đầu vào là kết quả từ bước phát hiện vật thể. Trong khóa luận này, thí nghiệm đo tốc độ trên các phương pháp trên được tiến hành bằng cách

sử dụng bộ kết quả phát hiện vật thể của mô hình YOLOV5L6 huấn luyện trên UA-DETRAC.

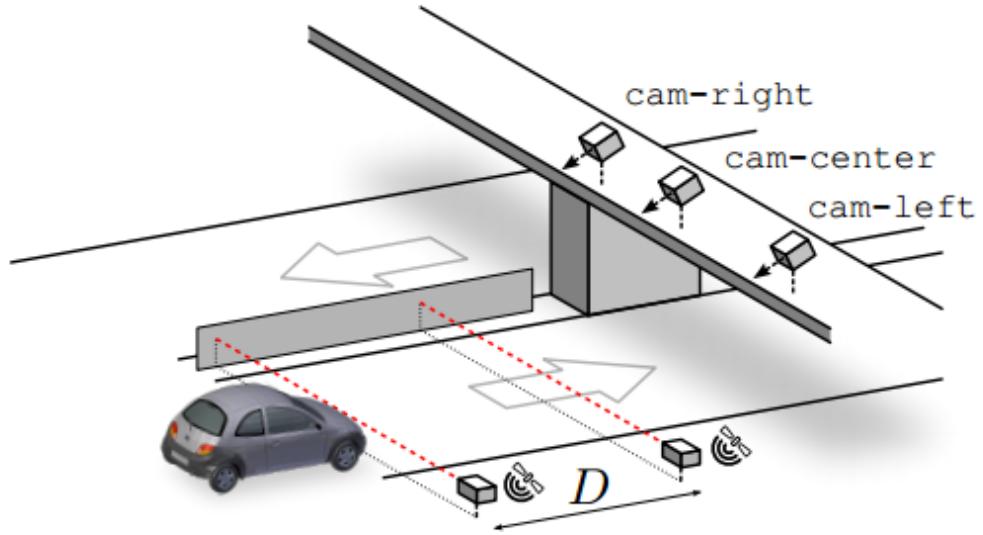
Nhận xét: Các phương pháp không ứng dụng mạng học sâu như SORT và IoUTracker có thể đạt tốc độ truy vết cao hơn nhiều các phương pháp khác.

2 Thí nghiệm đánh giá thuật toán ước tính tốc độ giao thông

2.1 Tổng quan về bộ dữ liệu BrnoCompSpeed

Bộ dữ liệu BrnoCompSpeed bao gồm 18 video được thu thập tại 6 địa điểm khác nhau, tại mỗi địa điểm ghi hình đặt camera ở 3 vị trí khác nhau là góc trái, góc phải và chính giữa làn đường (minh họa ở hình 4.6). Mỗi video có độ dài 60 phút với tốc độ khung hình là 50 fps. Độ phân giải khung hình là full-HD với kích thước khung hình 1920 x 1080 pixels. Để gán nhãn tốc độ cho phương tiện, nhóm tác giả đã sử dụng bộ công cụ gồm:

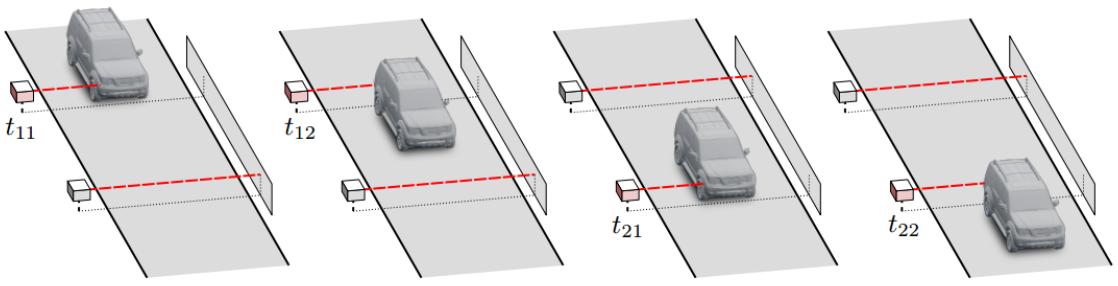
- Hai máy LIDAR sử dụng tia Lazer, được đặt vuông góc với hướng di chuyển của phương tiện và được đảm bảo có độ cao giống nhau so với mặt đường, hai điều kiện này được đảm bảo nhằm tránh sai số khi phát hiện thời điểm xe tới, giao và rời đi khỏi tia lazer từ máy LIDAR, tốc độ lấy mẫu là 1 kHz và khoảng cách lớn nhất của tia lazer là 300m.
- Một thiết bị GPS kết nối với máy tính giúp ước tính chính xác thời gian phương tiện giao nhau và rời khỏi phương tiện.



HÌNH 4.6: Mô hình hệ thống thu thập dữ liệu tốc độ. Với hai thiết bị phát sóng LIDARS được xác định tọa độ vị trí đặt và thời gian thực tế bằng GPS và ba máy ghi hình ở ba vị trí đặt có góc khác nhau [10]

Với khoảng cách giữa hai máy phát LIDAR được đo thủ công bằng thước đo lazer nhằm giảm sai số khoảng cách, ta được khoảng cách $D = 28m$. Quá trình ước tính tốc độ giao thông cho một phương tiện bắt đầu khi phương tiện chạm vào tia lazer của máy LIDAR thứ nhất ở thời điểm t_{11} và thời điểm t_{22} là thời điểm cuối cùng phương tiện giao với tia lazer của máy LIDAR thứ hai (hình minh họa 4.7). Những phương tiện không xác định được thời gian t_{11} và t_{12} do chồng lấp sẽ được bỏ qua và không được gán nhãn tốc độ. Tốc độ của phương tiện được tính dựa bằng cách lấy thương của D với hiệu hai thời điểm thời điểm t_{11} và t_{21} . Để kiểm tra tốc độ tính được nhóm tác giả cũng thực hiện phép tính trên với hiệu hai thời điểm t_{12} và t_{22} .

$$v = \frac{D}{t_{21} - t_{12}} \quad (4.10)$$

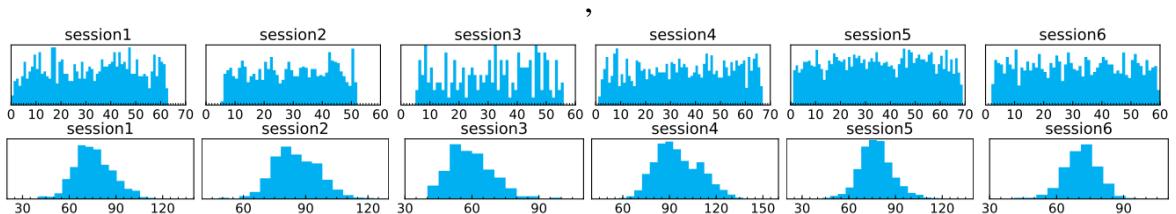


HÌNH 4.7: Minh họa quá trình lấy mẫu của hệ thống gán nhãn tốc độ [10]

Một số thông tin thống kê về bộ dữ liệu BrnoCompSpeed được thể hiện ở bảng 4.7 và hình 4.8:

	left	center	right
session 1	845	848	849
session 2	1163	1258	1583
session 3	193	193	193
session 4	1188	1192	1177
session 5	2021	2027	2030
session 6	1358	1358	1358

BẢNG 4.7: Số lượng phương tiện đi qua vùng quan tâm của mỗi video trong tập dữ liệu [10]



HÌNH 4.8: (6 biểu đồ trên) Biểu đồ tần suất phương tiện theo thời gian (xe/phút), 6 biểu đồ giữa Biểu đồ tần suất tốc độ phương tiện đo được bởi hệ thống LIDAR,[10]

Theo những kiến thức tìm hiểu được, bộ dữ liệu BrnoComp lớn hơn và được sử dụng nhiều hơn đáng kể so với các bộ dữ liệu khác [10]. Tuy nhiên, bộ dữ liệu này chưa thực sự đa dạng về đặc điểm ánh sáng và thời tiết. Đa phần các video(trừ 1 phần nhỏ của session 3) được quay ở điều kiện có mây. Các điều kiện thời tiết khác như trời có mưa, sương mù, ... chưa được ghi lại trong tập dữ liệu này.

2.2 Độ đo được sử dụng để đánh giá thuật toán ước tính tốc độ

Xét trường hợp bình thường, tốc độ phương tiện được ước tính dựa trên hai vị trí khác nhau Z_1 và Z_2 tại hai thời điểm phương tiện được phát hiện và truy vết t_1 và t_2 tương ứng. Ta có tốc độ trung bình của phương tiện sẽ được ước tính bằng công thức (giá trị này thường là giá trị tốc độ chính xác của phương tiện mà bộ dữ liệu cung cấp để đánh giá):

$$v = \frac{\Delta Z}{\Delta t} = \frac{Z_{21} - Z_{11}}{t_{21} - t_{11}} \quad (4.11)$$

Nếu tính thêm các sai số vị trí do việc phát hiện vị trí vật thể là Z_{1err} và Z_{2err} của hệ thống ước tính tốc độ giao thông, trong trường hợp xấu nhất ta có tốc độ trung bình của phương tiện được tính theo công thức:

$$v' = \frac{\Delta Z + Z_{1err} + Z_{2err}}{\Delta t} \quad (4.12)$$

Dẫn đến sai số tốc độ tuyệt đối (absolute speed error) được tính theo công thức:

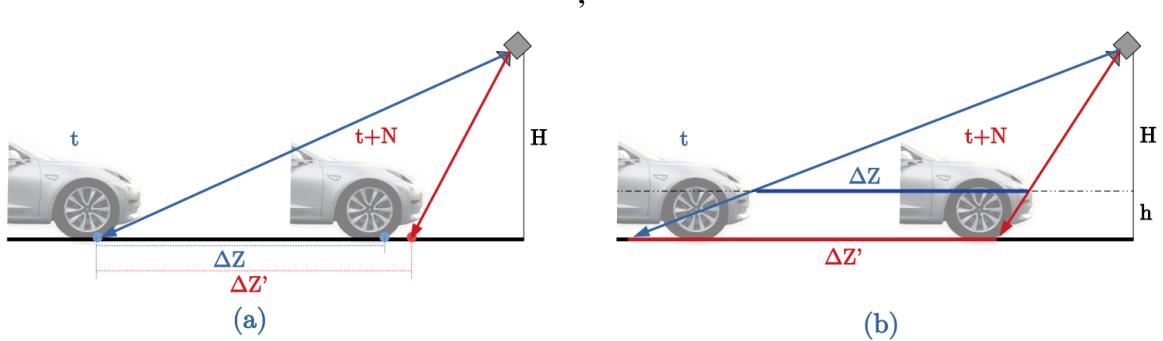
$$v_{err} = |v' - v| = \frac{Z_{1err} + Z_{2err}}{\Delta t} = \frac{Z_{1err} + Z_{2err}}{\Delta Z} v \quad (4.13)$$

và sai số tương đối tốc độ tương đối relative speed error:

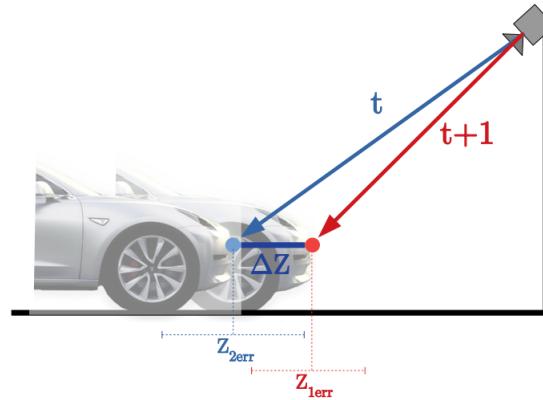
$$\frac{v_{err}}{v} = \frac{Z_{1err} + Z_{2err}}{\Delta Z} \quad (4.14)$$

Một số trường hợp thường xảy ra sai số vị trí của phương tiện của phương pháp ước tính khoảng cách được khảo sát ở mục 2.3 chương 2:

- Trong hệ thống sử dụng phương pháp ước tính khoảng cách bằng phép biến đổi đồng nhất và sử dụng đường ảo, sai số trong việc xác định vị trí của điểm dùng để lấy mẫu khoảng cách trên phương tiện do góc chiếu của phép thay đổi ở thời điểm t và thời điểm $t + N$ gây sai số khoảng cách $\Delta Z'$ so với khoảng cách ΔZ mà phương tiện thực sự di chuyển. (Hình 4.9)
- Kích thước của phần đường phương tiện di chuyển được dùng để ước tính tốc độ liên quan trực tiếp đến các thời điểm đo. Vì vậy, khi sử dụng hệ thống ước tính tốc độ dựa trên hai khung hình liên tiếp, khi tốc độ khung hình (FPS) thấp, giá trị ΔZ càng giống với sai số ước tính Z_{err} (Hình 4.10)



HÌNH 4.9: Minh họa sai số vị trí phương tiện trong hai phương pháp ước tính khoảng cách bằng phép biến đổi đồng nhất (Hormograph transform, hình a) và sử dụng đường ảo hình b) [11]



HÌNH 4.10: Sai số với hệ thống ước tính khoảng cách sử dụng hai khung hình liên tiếp [11]

2.3 Kết quả đánh giá

Dựa trên kết quả đánh giá các phương pháp phát hiện và truy vết vật thể được đánh giá ở bộ dữ liệu UA-DETRAC và kết quả khảo sát dữ liệu, nhóm chúng tôi chọn mô hình có kết quả truy vết cân bằng giữa tốc độ và độ chính xác để thực hiện thuật toán ước tính tốc độ. Cụ thể hơn, nhận xét rằng dữ liệu BronoComp được ghi hình ở FPS cao (50 FPS) và ít trường hợp bị che khuất. Đây là các đặc điểm tương tự như dữ liệu UA-DETRAC. Vì vậy, các thuật toán đơn giản và có tốc độ cao như IoUTracker vẫn có thể cho kết quả khả quan. Kết quả độ chính xác của IoUTracker không vượt trội so với các phương pháp khác như DeepSort, CenterTrack, Tracktor. Tuy nhiên khi tổ hợp với một mô hình có tốc độ cao trên nhiều loại cấu hình như Yolov5, tổ hợp này thể

hiện được sự cải tiến về tốc độ. Đây là một yếu tố quan trọng khi lựa chọn mô hình truy vết bởi vì trong thực tế, đa phần mô hình được thực thi với phần cứng hạn chế.

Ở đây, IoUTracker được sử dụng để truy vết và YOLOV5L6 huấn luyện trên dữ liệu UA-DETRAC được dùng để phát hiện vật thể. Điểm đại diện cho hộp giới hạn được chọn là điểm chính giữa cạnh dưới của hộp giới hạn 2D.

Ngoài ra chúng tôi so sánh kết quả đánh giá này với kết quả đánh giá của các phương pháp truy vết kiểu cũ. Kết quả truy vết kiểu cũ này được mô tả trong bài báo của Dubská [59]. Bước phát hiện vật thể được thực hiện bằng thuật toán background subtraction(BS)[60]. Sau đó, Kalman Filter [38] được dùng để truy vết. Điểm đại diện cho hộp giới hạn là góc trái dưới của hộp giới hạn 3D. Cách xây dựng hộp giới hạn 3D được trình bày chi tiết hơn trong bài báo gốc[59].

Thuật toán tính tốc độ được đánh giá là phương pháp được mô tả ở mục 2.2.1 để các tìm điểm biến mất (vanishing point) và tự động hiệu chỉnh các thông số máy ảnh để chuyển tọa độ hai chiều từ mặt phẳng ảnh thành tọa độ ba chiều trong thực tế. Nhóm chúng tôi sử dụng phương thức đánh giá sai số tốc độ tương tự trong bài báo BrunoCompSpeed [10]. Cách chia tập kiểm thử được tham khảo từ bài báo trên và tập kiểm thử bao gồm 18 video từ session1_left đến session6_right.

Chú thích các giá trị thống kê:

- Trung bình (mean): là giá trị trung bình cộng các điểm dữ liệu trong tập dữ liệu.
- Trung vị (median): là giá trị nằm ở giữa tập dữ liệu được sắp xếp tăng dần.
- Số phân vị (percentile): là một giá trị mà tại đó nhiều nhất có P% số trường hợp quan sát trong tập dữ liệu có giá trị thấp hơn giá trị này và nhiều nhất là (100-P)% số trường hợp có giá trị lớn hơn giá trị này.
- Trường hợp tệ nhất (worst): là giá trị sai số lớn nhất ghi nhận được.

Phương pháp	False Positive	False Negative	Precision	Recall
IoU Track (Yolov5)	2028	1674	0.89660	0.91283
BS + Kalman Filter feature method	10308	2467	0.69782	0.87225

BẢNG 4.10: Số trường hợp dự đoán sai và recall của mô hình phát hiện và truy vết vật thể

Trong bảng 4.10 là kết quả tính số False Positive, False Negative, Precision và Recall của 2 phương pháp truy vết khác nhau. Số False Positive là số lượng chuỗi truy

Video	Yolo+IouTracker				BS + Kalman Filter			
	mean	median	95 percentile	worst	mean	median	95 percentile	worst
session 1 left	12.37	12.35	14.97	17.90	11.18	11.00	13.70	51.44
session 1 center	17.62	4.28	38.39	55.24	10.15	9.12	17.81	77.84
session 1 right	14.71	14.58	21.19	35.84	6.8	6.65	11.22	31.53
session 2 left	16.07	15.88	19.46	61.69	16.24	16.14	19.72	36.91
session 2 center	11.92	11.61	15.42	34.68	11.14	10.69	14.72	33.22
session 2 right	0.56	0.49	1.33	6.72	0.9	0.77	1.45	21.03
session 3 left	14.45	14.04	19.56	27.53	14.28	13.99	19.75	27.55
session 3 center	13.77	13.57	17.33	22.08	13.96	13.90	17.58	22.23
session 3 right	8.56	8.22	12.04	16.05	7.54	7.27	10.85	15.57
session 4 left	16.39	16.42	19.72	23.82	15.88	15.82	18.95	22.48
session 4 center	0.76	0.55	1.78	74.82	1.10	0.62	2.61	27.59
session 4 right	7.90	7.78	12.20	18.09	2.05	1.81	3.79	25.88
session 5 left	11.39	11.31	15.45	23.85	6.99	7.08	8.87	15.53
session 5 center	9.31	9.10	12.19	74.23	9.64	9.11	14.38	119.32
session 5 right	17.31	16.40	25.00	36.65	12.02	11.83	19.39	68.10
session 6 left	8.37	8.32	10.55	24.04	7.14	7.23	8.95	10.84
session 6 center	1.42	1.21	3.46	8.89	2.89	2.74	4.17	29.49
session 6 right	11.92	11.92	14.29	37.92	12.56	12.64	15.04	55.90
TOTAL	10.16	10.78	18.98	74.82	8.59	8.45	17.14	119.32

BẢNG 4.8: Bảng kết quả sai số tốc độ tuyệt đối (absolute speed error) của thuật toán ước tính tốc độ phương tiện (đơn vị: km/h)

vết xe dự đoán thừa và False Negative là số lượng chuỗi truy vết xe dự đoán thiếu. False Positive và False Negative được tính tổng trên toàn bộ các video. Ngược lại, Precision và Recall được tính bằng cách lấy trung bình giữa các video.

Ta có thể nhận thấy việc ứng dụng thuật toán phát hiện và truy vết vật thể mới đã cải thiện đáng kể các giá trị False Positive và Recall. Điều này có ý nghĩa rằng phát hiện và theo dõi chính xác và đầy đủ hơn phương pháp cũ. Trung bình và phương sai của sai số tốc độ không quá khác biệt với kết quả của phương pháp cũ. Việc ước tính tốc độ các phương tiện phát hiện được vẫn chưa có cải tiến. Vấn đề này xảy ra bởi lẽ phương pháp được sử dụng để ước tính tốc độ là giống nhau.

Video	Yolo+IouTracker				BS + Kalman Filter			
	mean	median	95 percentile	worst	mean	median	95 percentile	worst
session 1 left	16.32	16.73	19.92	44.22	14.59	15.08	16.25	53.54
session 1 center	26.50	18.65	66.30	119.33	13.08	12.14	20.99	92.29
session 1 right	19.18	19.68	23.09	88.53	8.83	8.86	12.85	35.87
session 2 left	18.87	18.86	20.40	63.56	18.96	19.00	20.32	42.23
session 2 center	13.89	13.84	15.42	42.67	12.95	12.64	14.42	34.58
session 2 right	0.65	0.55	1.51	8.97	1.11	0.90	1.51	28.52
session 3 left	24.09	24.07	26.73	27.77	23.78	23.71	26.06	27.59
session 3 center	23.13	23.08	24.27	26.81	23.44	23.37	24.58	27.00
session 3 right	14.20	14.14	16.61	20.08	12.49	12.44	14.78	17.43
session 4 left	17.32	17.52	19.24	21.30	16.60	17.02	18.13	20.02
session 4 center	0.80	0.59	1.84	79.74	1.10	0.66	2.72	30.97
session 4 right	8.45	9.01	12.87	15.29	2.08	1.96	3.32	28.46
session 5 left	14.64	14.70	18.64	32.77	9.09	9.73	10.92	28.12
session 5 center	11.94	11.91	13.12	94.14	12.32	11.89	16.66	153.14
session 5 right	22.07	21.51	26.69	32.26	15.28	15.82	20.57	89.27
session 6 left	11.83	11.70	14.54	37.92	10.04	10.18	11.72	14.57
session 6 center	2.05	1.71	5.20	16.15	4.07	3.83	5.87	43.31
session 6 right	16.80	16.80	18.14	51.12	17.67	17.79	19.18	96.93
TOTAL	12.86	13.89	23.18	119.33	10.89	11.41	19.84	153.14

BẢNG 4.9: Bảng kết quả sai số tốc độ tương đối (relative speed error) của thuật toán ước tính tốc độ phương tiện (đơn vị: %)

Chương 5

KẾT LUẬN

Tổng kết lại quá trình thực hiện cả đề tài chúng tôi xin rút ra một vài kết luận như sau:

- Trong đề tài này, chúng tôi đã tìm hiểu ứng dụng các phương pháp truy vết tiên tiến cho bài toán ước tính tốc độ phương tiện giao thông. Đồng thời, chúng tôi xây dựng video demo cho kết quả của bài toán ước tính tốc độ.
- Trong điều kiện dữ liệu có số FPS cao, ít bị che lấp và mô hình phát hiện vật thể được huấn luyện phù hợp, các kỹ thuật đơn giản có tốc độ thực thi cao như IoUTracker vẫn đạt được độ chính xác truy vết tốt.
- Việc sử dụng các kỹ thuật truy vết mới có thể cải tiến đáng kể kết quả phát hiện và theo dõi. Các phương tiện được phát hiện một cách đầy đủ và chính xác hơn.
- Bởi vì phương pháp ước tính tốc độ được sử dụng là giống nhau nên kết quả sai số ước tính không khác biệt giữa việc dùng phương pháp truy vết cũ và phương pháp truy vết mới.

Chương 6

HƯỚNG PHÁT TRIỂN

Qua quá trình phân tích và đánh giá phương pháp truy vết và ước tính tốc độ phương tiện giao thông mà nhóm chúng tôi đã sử dụng, chúng tôi nhận thấy một số mặt hạn chế như sau:

- Tốc độ mô hình truy vết còn hạn chế và chưa đáp ứng yêu cầu để triển khai thực tế.
- Sai số tuyệt đối trong bước ước tính tốc độ còn lớn.
- Dữ liệu BrunoComp chưa mô tả các trường hợp khó trong thực tế như xe đổi hướng hoặc video được quay trong điều kiện thiếu sáng.

Từ những hạn chế nêu trên, nhóm chúng tôi đề xuất một số hướng cải tiến trong tương lai:

- Cải thiện tốc độ xử lý của mô hình truy vết bằng cách: tối ưu mã nguồn, cắt bỏ kiến trúc mô hình học sâu sử dụng để phát hiện vật thể. Ngoài ra, thực hiện thêm các thí nghiệm đánh giá các mô hình phát hiện và truy vết mới được công bố, có tốc độ xử lý và độ chính xác thỏa mãn với yêu cầu của một hệ thống ước tính tốc độ phương tiện giao thông cơ bản.
- Xây dựng hộp giới hạn 3D của vật thể từ kết quả hộp giới hạn 2D của vật thể, nhằm giảm sai số khi ước tính khoảng cách di chuyển của vật thể.

Tài liệu tham khảo

- [1] E. Bochinski, V. Eiselein, and T. Sikora, “High-speed tracking-by-detection without using image information,” in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2017.
- [2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and realtime tracking,” in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, IEEE, 2016.
- [3] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649, IEEE, 2017.
- [4] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in *European Conference on Computer Vision*, pp. 474–490, Springer, 2020.
- [5] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “Fairmot: On the fairness of detection and re-identification in multiple object tracking,” *International Journal of Computer Vision*, vol. 129, no. 11, pp. 3069–3087, 2021.
- [6] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 941–951, 2019.
- [7] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” *EURASIP J. Image Video Process.*, vol. 2008, 2008.
- [8] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” *CoRR*, 2016.

-
- [9] J. Luiten, A. Osep, P. Dendorfer, P. H. S. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, “HOTA: A higher order metric for evaluating multi-object tracking,” *CoRR*, vol. abs/2009.07736, 2020.
 - [10] J. Sochor, R. Juránek, J. Spanhel, L. Marsík, A. Siroký, A. Herout, and P. Zemcík, “Brnocompspeed: Review of traffic camera calibration and comprehensive dataset for monocular speed measurement,” *CoRR*, vol. abs/1702.06441, 2017.
 - [11] D. F. Llorca, A. H. Martínez, and I. G. Daza, “Vision-based vehicle speed estimation for ITS: A survey,” *CoRR*, vol. abs/2101.06159, 2021.
 - [12] E. Bochinski, T. Senst, and T. Sikora, “Extending iou based multi-object tracking by visual information,” in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, 2018.
 - [13] B. T. Tung, “Sort - deep sort : Một góc nhìn về object tracking (phần 1),” Dec 2021.
 - [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
 - [15] H. Honda, “Digging into detectron 2 (part 2),” Jun 2020.
 - [16] S. Nayar, “Linear camera model | camera calibration,” Apr 2021.
 - [17] M. Dubská and A. Herout, “Real projective plane mapping for detection of orthogonal vanishing points,” pp. 90.1–90.11, 01 2013.
 - [18] M. Dubská, A. Herout, R. Juranek, and J. Sochor, “Fully automatic roadside camera calibration for traffic surveillance,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, pp. 1162–1171, 06 2015.
 - [19] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu, “DETRAC: A new benchmark and protocol for multi-object tracking,” *CoRR*, vol. abs/1511.04136, 2015.
 - [20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

-
- [21] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.
 - [22] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
 - [23] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.
 - [24] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, “ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements,” Oct. 2020.
 - [25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.
 - [26] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
 - [27] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.
 - [28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
 - [29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*, pp. 21–37, Springer, 2016.
 - [30] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as points,” *arXiv preprint arXiv:1904.07850*, 2019.

-
- [31] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790, 2020.
 - [32] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.
 - [33] M. Tan and Q. V. Le, “Efficientnetv2: Smaller models and faster training,” *arXiv preprint arXiv:2104.00298*, 2021.
 - [34] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.
 - [35] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” *arXiv preprint arXiv:2010.04159*, 2020.
 - [36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.
 - [37] H. W. Kuhn, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
 - [38] X. Li, K. Wang, W. Wang, and Y. Li, “A multiple object tracking method using kalman filter,” in *The 2010 IEEE international conference on information and automation*, pp. 1862–1866, IEEE, 2010.
 - [39] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.
 - [40] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.
 - [41] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.

-
- [42] N. Wojke and A. Bewley, “Deep cosine metric learning for person re-identification,” in *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 748–756, IEEE, 2018.
 - [43] F. Yu, D. Wang, and T. Darrell, “Deep layer aggregation,” 2017.
 - [44] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, “Deep high-resolution representation learning for visual recognition,” 2019.
 - [45] X. Zhu, H. Hu, S. Lin, and J. Dai, “Deformable convnets v2: More deformable, better results,” 2018.
 - [46] M. Dubská, A. Herout, and J. Havel, “Pclines — line detection using parallel coordinates,” pp. 1489 – 1494, 07 2011.
 - [47] T. Tuytelaars, M. Proesmans, and L. Van Gool, “The cascaded hough transform,” in *Proceedings of International Conference on Image Processing*, vol. 2, pp. 736–739 vol.2, 1997.
 - [48] J. Shi and Tomasi, “Good features to track,” in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.
 - [49] C. Tomasi and T. Kanade, “Detection and tracking of point,” *Int J Comput Vis*, vol. 9, pp. 137–154, 1991.
 - [50] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, pp. 679 – 698, 12 1986.
 - [51] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, “MOTChallenge 2015: Towards a benchmark for multi-target tracking,” *arXiv:1504.01942 [cs]*, Apr. 2015.
 - [52] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “MOT16: A benchmark for multi-object tracking,” *arXiv:1603.00831 [cs]*, Mar. 2016.
 - [53] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, “Mot20: A benchmark for multi object tracking in crowded scenes,” *arXiv:2003.09003[cs]*, Mar. 2020.
 - [54] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

-
- [55] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, pp. 740–755, Springer, 2014.
 - [56] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, “Mars: A video benchmark for large-scale person re-identification,” in *European Conference on Computer Vision*, pp. 868–884, Springer, 2016.
 - [57] P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, and P. Luo, “Trantrack: Multiple-object tracking with transformer,” *arXiv preprint arXiv:2012.15460*, 2020.
 - [58] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
 - [59] M. Dubská, J. Sochor, and A. Herout, “Automatic camera calibration for traffic understanding,” *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 01 2014.
 - [60] M. Piccardi, “Background subtraction techniques: a review,” in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 4, pp. 3099–3104, IEEE, 2004.