

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN TIN



NGUYỄN XUÂN TUẤN - NGUYỄN THÁI THÔNG

GIÁM SÁT GIAO THÔNG TRÊN
CAMERA MẮT CÁ BẰNG PHƯƠNG
PHÁP TỐI ƯU YOLOv9

LUẬN VĂN TỐT NGHIỆP CỦ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

TP. HCM, 2024

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA TOÁN TIN

NGUYỄN XUÂN TUẤN - 20280112
NGUYỄN THÁI THÔNG - 20280092

GIÁM SÁT GIAO THÔNG TRÊN
CAMERA MẮT CÁ BẰNG PHƯƠNG
PHÁP TỐI ƯU YOLOv9

LUẬN VĂN TỐT NGHIỆP CỦ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN
TS. HUỲNH THẾ ĐĂNG

TP. HCM, 2024

Lời cam đoan

Chúng tôi xin cam đoan các kết quả trong luận văn này là do chúng tôi thực hiện dưới sự hướng dẫn của TS Huỳnh thế Đăng.

Tất cả các kiến thức liên quan được sử dụng trong luận văn đều được trích dẫn nguồn gốc một cách rõ ràng tại danh mục tài liệu tham khảo trong luận văn.

Luận văn không sao chép tài liệu, công trình nghiên cứu của người khác mà không chỉ rõ về mặt tài liệu tham khảo. Các kết quả thực nghiệm trong luận văn đều thực sự được tiến hành thực tế.

Nếu có bất kỳ sự gian lận nào, chúng tôi xin hoàn toàn chịu trách nhiệm trước hội đồng, cũng như kết quả luận văn tốt nghiệp của mình.

Tp.Hồ Chí Minh, ngày tháng năm 2024

Sinh viên ký tên

Lời cảm ơn

Đầu tiên, chúng tôi muốn gửi lời cảm ơn chân thành nhất tới TS. Huỳnh Thé Dăng là người đã luôn tận tình hướng dẫn chúng tôi nghiên cứu đề tài này. Nếu không có sự định hướng, những lời dạy bảo của thầy thì luận văn này của chúng tôi rất khó có thể hoàn thiện được.

Chúng tôi xin cảm ơn Khoa toán tin, Trường Đại học Khoa học tự nhiên đã tạo điều kiện, môi trường thuận lợi cho học viên trong quá trình học tập, nghiên cứu và hoàn thiện luận văn cử nhân.

Chúng tôi xin bày tỏ lòng kính trọng và biết ơn sâu sắc tới các thầy, cô, bạn bè trong khoa Toán tin, ngành Khoa học dữ liệu, đã luôn nhiệt tình giúp đỡ trong suốt quá trình học tập và nghiên cứu.

Cuối cùng, chúng tôi muốn gửi lời cảm ơn tới gia đình, người thân, những người luôn quan tâm, động viên để giúp tôi có động lực học tập, nghiên cứu và hoàn thiện đề tài nghiên cứu này.

Chúng tôi nhận thức rằng trong quá trình nghiên cứu, tìm hiểu các vấn đề cho việc “Giám sát giao thông trên camera mắt cá bằng phương pháp tối ưu YOLOv9”, luận văn có thể còn thiếu sót do kiến thức hạn chế. Do đó, chúng tôi rất mong nhận được ý kiến đóng góp từ các thầy cô và bạn bè để có thể hoàn thiện luận văn này hơn nữa.

Chúng tôi xin chân thành cảm ơn!

Mục lục

Lời cam đoan	i
Lời cảm ơn	ii
Mục lục	iii
Danh sách hình	vi
Danh sách bảng	viii
Danh mục các từ viết tắt	ix
Trang thông tin luận văn	x
1 Giới thiệu	1
1.1 Tổng quan về bài toán	1
1.2 Các thách thức và vấn đề cần giải quyết	2
1.3 Bố cục luận văn	2
2 Tổng quan	5
2.1 Deep learning là gì ?	5
2.1.1 Giới thiệu về deep learning.	5
2.1.2 Các thành phần của mạng học sâu.	6
2.1.3 Các loại Neural Network	6
2.2 Thị giác máy tính và phát hiện đối tượng.	7
2.3 Quy trình hoạt động của các mô hình phát hiện đối tượng	8
2.4 Các hệ số đánh giá hiệu suất phát hiện đối tượng và loại bỏ không tối đa (Non-Maximum Suppression)	9
2.4.1 Các hệ số đánh giá	9
2.4.2 Cách tính mAP	10
2.4.3 Hàm mất mát - Loss function	11
2.4.4 Loại bỏ sự chồng chéo không tối đa (Non-maximum Suppression)	11
2.5 Mạng Nơ-ron tích chập	12
2.6 Các mô hình hai giai đoạn	13
2.6.1 Mạng Nơ-ron Tích chập Dựa trên Vùng (R-CNN)	13
2.6.2 Mask R-CNN	14
2.6.3 Faster R-CNN	14
2.7 Các mô hình dựa trên huật toán YOLO	15

2.7.1	Các giai đoạn của YOLO	15
2.7.2	Xuất phát điểm - YOLOv1	15
2.7.3	Tăng cường cải thiện - YOLOv2	16
2.7.4	Nâng cấp - YOLOv3	17
2.7.5	Mô-đun hóa các thành phần của mô hình YOLO	18
2.7.6	Tối ưu tốc độ và độ chính xác - YOLOv4	19
2.7.7	YOLOv5	21
2.7.8	Thiết kế thân thiện với phần cứng - YOLOv6	22
2.7.9	Trainable Bag-of-Freebies (BoF) - YOLOv7	23
2.7.10	Hiệu suất và độ chính xác tối đa - YOLOv8	24
2.7.11	Bước nhảy vọt đáng kể - YOLOv9	26
2.8	Các thuật toán theo dõi vật thể	28
2.8.1	Tổng quan về các thuật toán theo dõi vật thể	28
2.8.2	DeepSORT	29
2.8.3	StrongSORT	30
3	Phương pháp đề xuất	31
3.1	Tập dữ liệu FISHEYE 8K	31
3.1.1	Tổng quan về tập dữ liệu	31
3.1.2	Quá trình Thu thập Video và Chuẩn bị Dữ liệu	32
3.1.3	Điều kiện môi trường	32
3.1.4	Chú thích và Xác thực	33
3.1.5	Benchmark của tập dữ liệu	33
3.2	Phân tích tập dữ liệu FISHEYE 8K	33
3.2.1	Phân tích tổng quan	33
3.2.2	Sự phân bố của các thực thể	34
3.3	Lựa chọn mô hình cơ sở	34
3.4	Phân tích các nghiên cứu liên quan	35
3.4.1	Phép tích chập biến dạng (Deformable Convolution)	35
3.4.2	Large Selective Kernel Network	37
3.4.3	Programmable Gradient Information	40
3.4.4	Generalized ELAN (GELAN)	41
3.5	Đề xuất các kiến trúc mô hình cải tiến	43
3.5.1	RepNDCNELAN4 - GELAN với Deformable Convolution	43
3.5.2	RepNLSKELAN4 - GELAN với Large Selective Kernel Network	45
3.6	Đề xuất cải tiến hàm mắt mèo	46
3.6.1	Các hàm mắt mèo của YOLOv9	46
3.6.2	Đề xuất hàm mắt mèo tối ưu cho mô hình phát hiện vật thể trên camera fisheye	49
3.7	Triển khai mô hình sau huấn luyện	52
3.7.1	Tích hợp mô hình đã được huấn luyện với thuật toán theo dõi	52
3.7.2	Nền tảng cung cấp môi trường triển khai ứng dụng	52

3.7.3	Framework hỗ trợ xây dựng ứng dụng	53
4	Thực nghiệm, đánh giá và triển khai ứng dụng mô hình	54
4.1	Chi tiết quá trình triển khai thử nghiệm	54
4.1.1	Mục đích thử nghiệm	54
4.1.2	Các triển khai thử nghiệm	54
4.2	Môi trường thử nghiệm	55
4.3	Các mô hình đề xuất	55
4.3.1	Hiệu suất của mô hình YOLOv9-e và YOLOv9-c huấn luyện với tập dữ liệu FishEye8K	55
4.3.2	RepNDCNELAN4	56
4.3.3	RepNLSKELAN4	58
4.3.4	Kết hợp RepNDCNELAN4 và RepNLSKELAN4	60
4.3.5	Tích hợp Warp Loss	60
4.4	Mô hình đề xuất có hiệu năng tốt nhất	65
4.4.1	Hiệu năng các mô hình đề xuất với SOTA	65
4.4.2	Kiến trúc mô hình đề xuất có hiệu năng tốt nhất	65
4.4.3	Trực quan hoá và so sánh các kết quả trên tập thử nghiệm	67
4.5	Triển khai ứng dụng mô hình	67
4.5.1	Môi trường triển khai ứng dụng	67
4.5.2	Thiết kế giao diện người dùng	68
4.5.3	Thử nghiệm với dữ liệu thực tế	70
4.5.4	Tốc độ phản hồi và đáp ứng của ứng dụng	71
5	Kết luận	72
5.1	Kết luận chung	72
5.2	Hướng phát triển	72
Tài liệu tham khảo		74

Danh sách hình

2.1	Sơ đồ Deep Learning	5
2.2	mAP flowchart	11
2.3	NMS	12
2.4	Sự phát triển của YOLO qua các năm	15
2.5	Kiến trúc mô hình YOLOv1	16
2.6	Cách thức hoạt động của YOLOv1	16
2.7	hộp giới hạn neo trong YOLOv2	17
2.8	Kiến trúc mô hình YOLOv3	18
2.9	Mạng đặc trưng kim tự tháp FPN trong YOLOv3	18
2.10	Liên kết dư được tích hợp trong YOLOv3	18
2.11	Kiến trúc tổng quan của mô hình YOLO kể từ phiên bản v4	19
2.12	Liên kết CSP trong YOLOv4	19
2.13	Phép gộp không gian kim tự tháp SPP trong YOLOv4	20
2.14	PAN (Path Aggregation Network) framework	20
2.15	Các phương pháp cải thiện mô hình BoF và BoS	21
2.16	Kiến trúc mô hình YOLOv5	22
2.17	Coupled head và Decoupled head	22
2.18	EfficientRep	23
2.19	Kiến trúc của mạng ELAN và E-ELAN	24
2.20	Kiến trúc mô hình YOLOv7	25
2.21	Kiến trúc của YOLOv8	25
2.22	Kiến trúc khối GELAN và ELAN	26
2.23	Kiến trúc mô hình YOLOv9-e	27
2.24	Kiến trúc mô hình YOLOv9-c	28
2.25	Các hoạt động của thuật toán DeepSORT	29
2.26	Các hoạt động của thuật toán StrongSORT	30
3.1	Tập dữ liệu FISHEYE8K	31
3.2	Hình ảnh ghi lại các lớp trong bộ dữ liệu Fisheye8K	32
3.3	Sự phân bổ thực thể của các lớp trong tập dữ liệu Fisheye8K	34
3.4	Sự phân bố về vị trí của các lớp trên toàn bộ tập dữ liệu	34
3.5	Hiệu suất vượt trội của YOLOv9 so với các mô hình khác trên tập COCO	35
3.6	Các biến thể về mặt hình học của vật thể	36
3.7	Vị trí lấy mẫu của kernel trong phép tích chập biến dạng	36
3.8	Cách hoạt động của phép tích chập biến dạng	37

3.9	Cách hoạt động của khối LSK	38
3.10	Phân rã phép tích chập với kernel lớn thành chuỗi phép tích chập với kernel nhỏ hơn	38
3.11	Kiến trúc mô hình tích hợp PGI	40
3.12	Kiến trúc mạng Generalized ELAN (GELAN)	41
3.13	Cross stage partial operation	42
3.14	Sự kết hợp từ các vị trí của Gradient flow truncate operation	43
3.15	Mạng ELAN với đường dẫn của gradient	44
3.16	Kiến trúc khối RepNCSP	44
3.17	Kiến trúc khối RepNBottleNeck	45
3.18	Kiến trúc khối RepNBottleNeckDCNv2	45
3.19	Kiến trúc mạng RepNBottleNeckLSK	47
3.20	Người đi bộ ở các vị trí khác nhau trong tập dữ liệu Fisheye8K	50
3.21	Sơ đồ hoạt động của ứng dụng khi tích hợp thuật toán theo dõi	52
4.1	Khả năng trích xuất thông tin của RepNDCNELAN4 và GELAN.	58
4.2	Khả năng trích xuất thông tin của RepNLSKELAN4 và GELAN.	59
4.3	Khả năng trích xuất thông tin của YOLOv9-e kết hợp Warp Loss và YOLOv9-e.	61
4.4	Hiệu năng của mô hình và phương pháp đề xuất của chúng tôi với các mô hình SOTA	63
4.5	Giá trị loss của mô hình YOLOv9-e so với mô hình đề xuất cỡ lớn.	64
4.6	Điểm số MAP của mô hình YOLOv9-e so với mô hình đề xuất cỡ lớn.	64
4.7	Kiến trúc mô hình cỡ lớn của chúng tôi dựa trên YOLOv9-e 2.23	66
4.8	Kiến trúc mô hình rút gọn của chúng tôi dựa trên YOLOv9-c 2.24	66
4.9	Trực quan hóa suy luận của 2 mô hình	67
4.10	Cấu hình sử dụng trên HuggingFace Space	68
4.11	Phần dữ liệu đầu vào	68
4.12	Kết quả đầu ra sau khi áp dụng nhận dạng đối tượng	69
4.13	Ví dụ về biểu đồ đa đường trực quan cho dữ liệu là video	69
4.14	Ví dụ về biểu đồ cột trực quan cho dữ liệu là hình ảnh	70
4.15	Màu trực quan của từng lớp trong tập dữ liệu.	70
4.17	Các ví dụ trong ứng dụng.	70
4.16	Phần cài đặt mô hình và thuật toán tracking	71
4.18	Thời gian thực hiện các suy luận trên ứng dụng	71

Danh sách bảng

3.1	Kết quả của các mô hình phát hiện đối tượng YOLO được huấn luyện trên tập dữ liệu FishEye8K với hai kích thước đầu vào là 1280×1280 và 640×640	33
4.1	Kết quả của mô hình YOLOv9-e base	56
4.2	Kết quả của mô hình YOLOv9-c base	56
4.3	Bảng kết quả thực nghiệm trên mô hình YOLOv9 kết hợp với RepNDCNELAN4 . . .	57
4.4	Kết quả của mô hình YOLOv9-e kết hợp với RepNDCNELAN4 ở layer thứ 9	57
4.5	Bảng kết quả thực nghiệm trên mô hình YOLOv9 kết hợp với RepNLSKELAN4	58
4.6	Kết quả của mô hình YOLOv9-e kết hợp với RepNLSKELAN4 ở các layer 22-25-28 .	59
4.7	Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với RepNLSKELAN4 và RepNDCNELAN4	60
4.8	Chi tiết về các layer được tích hợp giữa RepNLSKELAN4 và RepNDCNELAN4	60
4.9	Bảng kết quả thực nghiệm trên mô hình YOLOv9 kết hợp với RepNDCNELAN4 và sử dụng hàm Warp Loss.	61
4.10	Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với RepNLSKELAN4 và sử dụng hàm Warp Loss.	61
4.11	Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với RepNLSKELAN4, RepNDCNELAN4 và sử dụng hàm Warp Loss.	61
4.12	Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với Network B và sử dụng hàm Warp Loss với các thông số khác nhau.	62
4.13	Kết quả của network B sử dụng Warp Loss	62
4.14	Kết quả mô hình YOLOv8x trên ảnh đầu vào 640×640	62
4.15	Kết quả mô hình rút gọn đề xuất với Warp Loss.	63
4.16	Bảng kết quả hiệu năng các mô hình đề xuất với SOTA.	65

Danh mục các từ viết tắt

STT	Ký hiệu chữ viết tắt	Chữ viết đầy đủ
1	YOLO	You Look Only One
2	CNN	Convolution Neural Network
3	SOTA	State Of The Art
4	ML	Machine Learning
5	AI	Artifical Intelligent
6	FNN	Feedfoward Neural Network
7	ANN	Artifitial Neural Network
8	NLP	Natural Language Processing
9	RNN	Recurrent Neural Network
10	CSP	Cross Stage Partial
11	PANet	Path Aggregation Network
12	NMS	Non Maximum Suppression
13	ELAN	Efficient Long-Range Attention Network
14	E-ELAN	Extended Efficient Long-Range Attention Network
15	FPN	Feature Pyramid Network
16	PGI	Programmable Gradient Information
17	GELAN	Generalized Efficient Layer Aggregation Network
18	LSK	Large Selective Kernel
19	SPP	Spatial Pyramid Pooling

Trang thông tin luận văn

Tên đề tài luận văn: Giám sát giao thông trên camera mắt cá bằng phương pháp tối ưu YOLOv9

Ngành: Khoa học dữ liệu

Mã số ngành:

Họ và tên sinh viên: Nguyễn Xuân Tuấn và Nguyễn Thái Thông

Khóa đào tạo: 2020

Người hướng dẫn luận văn: TS. Huỳnh Thế Đăng

Cơ sở đào tạo: Trường Đại học Khoa học Tự Nhiên, DHQG.HCM

Tóm tắt nội dung luận văn

Những năm gần đây, tiến bộ trong thị giác máy tính nhờ học sâu đã ứng dụng vào nhiều lĩnh vực như phân loại tài liệu, phân tích y tế và phát hiện vật thể, đặc biệt trong giao thông thông minh và lái xe tự động. Hệ thống phát hiện phương tiện hiệu quả rất quan trọng để đảm bảo an toàn và tối ưu hóa luồng giao thông. Các phương pháp truyền thống gặp hạn chế với môi trường biến đổi, đòi hỏi kỹ thuật phát hiện chính xác hơn.

Ông kính mắt cá, khác với máy ảnh truyền thống, cung cấp phạm vi quét toàn cảnh, giảm chi phí và hữu ích cho giám sát giao thông tại khu vực đông đúc. Tuy nhiên, cần các thuật toán chuyên biệt để xử lý hình ảnh bị bẻ cong của camera fisheye, cải thiện quản lý luồng giao thông và ngăn ngừa tai nạn.

Luận án này khám phá phương pháp phát hiện đối tượng cho camera fisheye trong giám sát giao thông, nhằm tăng cường an toàn và khả năng di chuyển đô thị.

Những kết quả mới của luận văn

Sau khi hoàn thành luận văn, kết quả đã cho thấy mô hình YOLOv9 kết hợp với những đề xuất của chúng tôi cho bài toán phát hiện phương tiện đã đạt được một kết quả vượt trội so với các mô hình SOTA đang có hiện nay.

Các ứng dụng/ khả năng ứng dụng trong thực tiễn hay những vấn đề còn bỏ ngõ cần tiếp tục nghiên cứu

Nghiên cứu này được ứng dụng trong hệ thống giám sát giao thông đường phố và có thể được áp dụng rộng rãi trong thực tế.

Mặc dù đã tích hợp các thuật toán theo dõi theo sau mô hình phát hiện vật thể tuy nhiên chúng tôi vẫn chưa có các đề xuất tinh chỉnh thuật toán để tối ưu cho việc theo dõi đối tượng trên camera

fisheye.

Tập thể cán bộ hướng dẫn
(Ký tên, họ tên)

Sinh viên
(Ký tên, họ tên)

Xác nhận của cơ sở đào tạo
Hiệu trưởng

Thesis information

Thesis title: Traffic surveillance On FishEye Camera Using Improved YOLOv9

Speciality: Data Science

Code:

Name of Student: Nguyen Xuan Tuan and Nguyen Thai Thong

Academic year: 2020

Supervisor: PhD. Huynh The Dang

At: VNUHCM - University of Science

Summary

In recent years, advancements in computer vision through deep learning have been applied to various fields such as document classification, medical analysis, and object detection, especially in intelligent transportation and autonomous driving. Efficient vehicle detection systems are crucial for ensuring safety and optimizing traffic flow. Traditional methods face limitations in dynamic environments, requiring more accurate detection techniques.

Fisheye cameras, unlike traditional cameras, offer a panoramic view, reducing costs and proving useful for traffic monitoring in crowded areas. However, specialized algorithms are needed to process the distorted images from fisheye cameras, improving traffic flow management and preventing accidents.

This thesis explores object detection methods for fisheye cameras in traffic monitoring, aiming to enhance urban safety and mobility.

Novelty of Thesis

Upon completing the thesis, the results demonstrated that our proposed enhancements to the YOLOv9 model for vehicle detection achieved significantly better performance compared to SOTA and traditional models. This advancement highlights the effectiveness of our approach in handling the complexities of fisheye camera imagery for traffic monitoring.

Application/Applicability/Perpective

This research is applied in street traffic monitoring systems and has the potential for widespread real-world application.

Although we have integrated tracking algorithms following the object detection model, we have not yet proposed specific algorithm refinements to optimize object tracking for fisheye cameras.

Supervisor
(Signature)

Student
(Signature)

Certification
University of Science
President

Chương 1

Giới thiệu

1.1 Tổng quan về bài toán

Trong những năm gần đây, những tiến bộ đáng kể trong thị giác máy tính đã được thúc đẩy bởi sự phát triển nhanh chóng của các phương pháp học sâu. Những tiến bộ này đã được ứng dụng trên nhiều lĩnh vực khác nhau, chẳng hạn như phân loại tài liệu, phân tích y tế và đặc biệt là phát hiện vật thể, vốn có tầm quan trọng then chốt trong nhiều lĩnh vực khác nhau như hệ thống giao thông thông minh, lái xe tự động, giám sát giao thông và giám sát môi trường.

Sự phổ biến và đa dạng của các phương tiện giao thông trong cuộc sống hàng ngày của chúng ta nhấn mạnh tầm quan trọng của các hệ thống phát hiện phương tiện hiệu quả trong việc đảm bảo an toàn, tối ưu hóa luồng giao thông và cho phép các hệ thống giao thông thông minh hoạt động. Các phương pháp truyền thống để phát hiện phương tiện thường phụ thuộc vào các đặc điểm được tạo thủ công và các thuật toán dựa trên quy tắc, điều này hạn chế khả năng thích ứng của chúng đối với các điều kiện môi trường biến đổi. Hơn nữa, khi sự phức tạp của các tình huống thực tế tăng lên, nhu cầu về các kỹ thuật phát hiện chính xác và tinh vi hơn cùng với việc đáp ứng được thời gian thực trở nên cần thiết hơn bao giờ hết.

Trong lĩnh vực giám sát giao thông, sự xuất hiện của các camera fisheye đã cách mạng hóa hệ thống giám sát, hệ thống giao thông và thay đổi cách chúng ta quan sát và phân tích hoạt động của phương tiện và người đi bộ. Khác với máy ảnh truyền thống, camera fisheye được đặc trưng bởi khả năng cung cấp phạm vi quét toàn cảnh đưa ra một giải pháp hạn chế chi phí với ít máy ảnh hơn, đặc biệt có lợi cho việc giám sát các khu vực có giao lộ và khu vực đô thị đông đúc, phức tạp. Các tính năng độc đáo của camera fisheye đòi hỏi các thuật toán chuyên biệt có khả năng phát hiện chính xác các đối tượng trong các khung hình bị bẻ cong. Điều này quan trọng cho nhiều ứng dụng, từ quản lý luồng giao thông đến ngăn ngừa tai nạn. Khả năng phát hiện và theo dõi đối tượng một cách đáng tin cậy trong thời gian thực có thể cải thiện đáng kể hiệu suất của các hệ thống giám sát giao thông, góp phần tạo ra các thành phố thông minh và an toàn hơn.

Luận án này khám phá các phương pháp tiên tiến trong việc phát hiện đối tượng dành riêng cho camera fisheye, tập trung vào ứng dụng của chúng trong giám sát giao thông. Nó nhằm mục tiêu thu gọn khoảng cách giữa khả năng bao phủ rộng lớn của camera fisheye và độ chính xác cần thiết cho việc giám sát hiệu quả. Qua phân tích toàn diện và thực nghiệm, công trình này mong muốn đóng góp vào những nỗ lực liên tục trong việc cải thiện hệ thống giám sát giao thông và qua đó tăng cường an toàn và khả năng di chuyển trong đô thị.

1.2 Các thách thức và vấn đề cần giải quyết

Việc sử dụng ống kính fisheye trong giám sát giao thông đặt ra một loạt thách thức mới lạ xuất phát từ các đặc trưng vốn có của camera fisheye.

Vấn đề nổi bật nhất là đặc tính bẻ cong điểm ảnh của ống kính fisheye, gây ra sự biến dạng không tuyến tính của hình ảnh. Khác với máy ảnh truyền thống, ống kính fisheye cung cấp góc nhìn toàn cảnh, bao phủ một góc rộng của môi trường, dẫn đến các vật thể nằm ở các vị trí khác nhau trong khung hình sẽ bị bẻ cong và thu phóng độ lớn theo các cấp độ khác nhau. Sự bẻ cong này đặc biệt rõ ràng ở rìa của hình ảnh, dẫn đến những thách thức đáng kể trong việc phát hiện và định vị đối tượng, cụ thể là việc nhận dạng các vật thể trở nên khó khăn hơn so với hình ảnh được cung cấp bởi camera với ống kính truyền thống.

Ngoài ra, các thuật toán và mô hình phát hiện đối tượng có sẵn hiện nay được thiết kế cho máy ảnh với ống kính truyền thống có góc nhìn và dựa vào tiêu cự và khẩu độ tiêu chuẩn, không thể bao quát cho các đối tượng bị bẻ cong trong hình ảnh được cung cấp bởi camera fisheye. Sự bẻ cong này ảnh hưởng đến hình dạng, kích thước và vị trí xuất hiện của đối tượng, làm cho việc áp dụng trực tiếp các phương pháp và mô hình phát hiện thông thường trở nên khó khăn.

Một thách thức khác là sự đa dạng của các vị trí đặt camera tạo ra các góc nhìn, khoảng cách và tỷ lệ vật thể khác nhau dẫn đến các biến đổi về hình ảnh của đối tượng cũng khác nhau. Hơn nữa, tầm nhìn rộng của camera fisheye có khả năng ghi lại một số lượng lớn đối tượng vào khung hình, làm tăng độ phức tạp của việc phát hiện và theo dõi. Vì vậy, việc đòi hỏi các thuật toán và mô hình phải đáp ứng được khả năng xử lý các biến đổi phi tuyến do đặc trưng của ống kính và đồng thời mang tính tổng quát cho các góc nhìn khác nhau, không phụ thuộc vào vị trí đặt camera.

Vấn đề càng trở nên phức tạp hơn do tính biến động của môi trường ảnh hưởng đến hình ảnh mà camera thu được. Vì điều kiện ánh sáng, thời tiết và sự che khuất vật thể có thể ảnh hưởng nghiêm trọng đến khả năng nhìn thấy và phát hiện các đối tượng có trong khung hình. Do đó việc phát triển các hệ thống phát hiện đối tượng hiệu quả cho máy ảnh fisheye đòi hỏi phải giải quyết những vấn đề này, đảm bảo phát hiện và theo dõi chính xác trong các điều kiện giám sát đa dạng và đầy thách thức.

Luận án này nhằm định nghĩa và giải quyết các vấn đề, thách thức của việc phát hiện đối tượng trong hình ảnh được thu từ camera fisheye, với trọng tâm là vượt qua những thách thức như vật thể bị bẻ cong và mật độ vật thể dày đặc trong bối cảnh giám sát giao thông.

1.3 Bố cục luận văn

Luận văn chia thành 5 phần cụ thể như sau:

Chương 1: Giới thiệu và tổng quan về bài toán.

Chương này đóng vai trò là một phần giới thiệu quan trọng của luận văn, cung cấp cái nhìn tổng quan sâu sắc về các vấn đề cần nghiên cứu, giải quyết và xác định nền tảng cho cả bài luận văn. Trọng tâm của chương là sự phát triển trong các phương pháp phát hiện vật thể, đặc biệt là phân tích dữ liệu hình ảnh thu thập từ các camera đặt trên đường phố. Từ đó nêu ra về vấn đề và mục tiêu cần nghiên cứu, nguyên nhân ra đời của dự án và nhấn mạnh vào sứ mệnh của nó trong lĩnh vực

phát hiện phương tiện thông qua ống kính fisheye. Đồng thời, phần này cũng nhấn mạnh vào động lực đằng sau việc tiến hành nghiên cứu và những cống hiến tiềm năng của nó đối với lĩnh vực thị giác máy tính.

Ngoài ra, nội dung chương cũng bao gồm những phân tích về các hạn chế tiềm ẩn có thể gặp phải trong quá trình nghiên cứu, đặc biệt là trong việc sử dụng ống kính fisheye và đề xuất các giải pháp để vượt qua những hạn chế đó. Cấu trúc và phương pháp được sử dụng trong luận văn cũng được trình bày chi tiết, nhấn mạnh vào tính hiệu quả và tính khả thi của chúng trong việc đạt được mục tiêu nghiên cứu.

Chương 2: Các nghiên cứu, kiến thức liên quan đến luận văn.

Chương này tập trung vào việc cung cấp một cái nhìn tổng quan và phân tích toàn diện về nghiên cứu hiện có liên quan đến đề tài của luận văn. Mục tiêu là thể hiện sự hiểu biết của tác giả về lĩnh vực và những khoảng trống trong kiến thức hiện tại mà luận văn nhằm giải quyết.

Ngoài ra, phần này sẽ cung cấp kiến thức về các thuật toán và mô hình phát hiện đối tượng trong thời gian thực khác nhau, bao gồm các phương pháp phát hiện đối tượng một giai đoạn như YOLO (You Only Look Once), đặc biệt là Yolov9 và các phương pháp phát hiện đối tượng hai giai đoạn như R-CNN, Faster-RCNN. Nghiên cứu về các hàm mất mát và các kỹ thuật nâng cao trong mạng nơ ron của bài toán nhận diện đối tượng cũng sẽ được đề cập một cách cẩn thận trong chương này.

Trong phần này, chúng tôi cũng giới thiệu các chỉ số đánh giá khác nhau để đo lường và so sánh hiệu suất của các phương pháp và mô hình phát hiện đối tượng. Các chỉ số như độ chính xác (Accuracy), độ phủ (Recall), độ chính xác tổng thể (Precision), và độ chính xác trung bình cộng (mAP) thường được sử dụng để đo lường hiệu suất của các phương pháp.

Chương 3: Phương pháp đề xuất.

Chương này sẽ tiến hành một bước đầu tiên quan trọng trong quá trình nghiên cứu bằng việc giới thiệu tổng quan và phân tích về tập dữ liệu được sử dụng trong luận văn, đó là FISHEYE8K. Trước hết, chúng ta cần hiểu rõ về tập dữ liệu này, bao gồm các đặc điểm chính, cách thức thu thập, điều kiện môi trường. Xem xét các nghiên cứu có liên quan đến bài toán như Large Selective Kernel, GELAN và các nghiên cứu khác, giúp chúng ta hiểu rõ hơn về các phương pháp và kỹ thuật đã được áp dụng để xử lý dữ liệu ảnh fisheye và giải quyết các bài toán cụ thể. Các nghiên cứu này cung cấp cái nhìn sâu sắc vào tiến triển của lĩnh vực và có thể cung cấp những hướng dẫn quý báu cho việc lựa chọn kiến trúc mô hình phù hợp nhất cho nghiên cứu của chúng ta. Dựa trên các kiến thức này, chúng ta có thể đề xuất các kiến trúc mô hình tiềm năng có thể giải quyết bài toán từ FISHEYE8K [1] một cách hiệu quả nhất.

Chương 4: Thực nghiệm và đánh giá.

Chương này tập trung vào việc trình bày chi tiết các khía cạnh thực tiễn của nghiên cứu, sau đó rút ra các đánh giá khách quan cho các kết quả đạt được từ các phương pháp tiếp cận và đề xuất đã nêu.

Chúng tôi tiến hành một loạt các thử nghiệm để đánh giá hiệu suất của các mô hình phát hiện đối tượng dựa trên thuật toán YOLO, đặc biệt là phiên bản mới nhất, YOLOv9. Các thử nghiệm này được thiết kế để kiểm tra khả năng nhận diện và theo dõi đối tượng trong các tình huống thực tế khác nhau. Phương pháp huấn luyện các mô hình được thực hiện trong môi trường thiết lập Hugging Face, một nền tảng mạnh mẽ và linh hoạt hỗ trợ các công cụ và tài nguyên cần thiết cho việc huấn

luyện mô hình AI. Chúng tôi cung cấp chi tiết về quy trình huấn luyện, từ việc chuẩn bị dữ liệu, cài đặt các tham số huấn luyện, đến việc tối ưu hóa mô hình để đạt hiệu suất cao nhất. Các kết quả thu được từ các thử nghiệm được trình bày một cách chi tiết và minh bạch.

Chúng tôi so sánh hiệu suất của mô hình trong bài báo [1] và YOLOv9 với các mô hình cải tiến đề xuất, sử dụng cùng một môi trường thiết lập để đảm bảo tính công bằng và nhất quán trong đánh giá. Các chỉ số hiệu suất bao gồm độ chính xác, tốc độ suy luận, và khả năng theo dõi đối tượng qua thời gian.

Chương 5: Kết luận.

Phần này bao gồm các phát hiện chính và tóm tắt kết quả nghiên cứu. Chương tái khẳng định các điểm chính được thảo luận xuyên suốt luận án và cung cấp một phát biểu ngắn gọn về kết luận rút ra từ nghiên cứu. Nó trả lời các câu hỏi nghiên cứu hoặc mục tiêu được đề ra ở đầu luận án và phản ánh liệu chúng đã được giải quyết thành công hay không. Nó thảo luận về cách các phát hiện nghiên cứu góp phần vào kiến thức hiện có trong lĩnh vực và làm nổi bật tác động tiềm năng của chúng đối với lý thuyết, thực hành, hoặc ứng dụng thực tế. Phần này cũng có thể đề cập đến bất kỳ phát hiện bất ngờ hoặc đáng chú ý nào được thực hiện trong quá trình nghiên cứu. Chương mô tả các lĩnh vực có thể được hưởng lợi từ việc điều tra thêm dựa trên các hạn chế và khoảng trống được xác định trong nghiên cứu hiện tại. Nó có thể đề xuất mở rộng nghiên cứu hiện tại, các phương pháp mới, hoặc các hướng mới để khám phá trong các lĩnh vực liên quan.

Chương 2

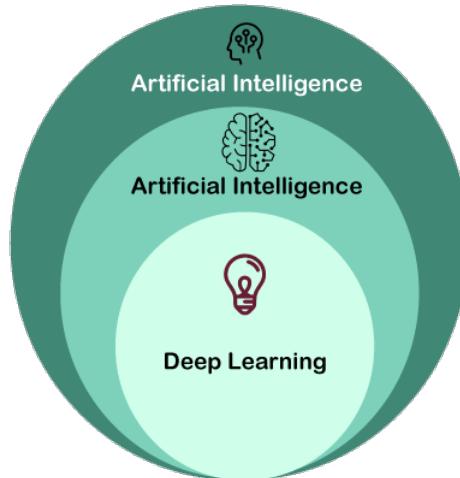
Tổng quan

Chương này tập trung trình bày về tình hình nghiên cứu cho đến thời điểm hiện tại của các bài toán nhận dạng đối tượng. Hiện nay, với sự phát triển của các thuật toán học sâu đối với các tác vụ trong thị giác môtô tính, luận văn sẽ tập trung vào khảo sát các phương pháp học sâu.

2.1 Deep learning là gì ?

2.1.1 Giới thiệu về deep learning.

Deep learning là một phương pháp, một nhánh của Machine Learning (ML) và trí tuệ nhân tạo (AI), được coi là một công nghệ cốt lõi của cuộc cách mạng Công nghiệp lần Thứ Tư. Nó cho phép huấn luyện một tập dữ liệu để dự đoán kết quả đầu ra từ một kết quả đầu vào nhờ vào khả năng học dữ liệu dựa trên kiến trúc của mạng nơ-ron nhân tạo (Artificial Neural Networks - ANNs) [2]. Từ "deep" trong deep learning đề cập đến độ sâu của mạng. Một mạng neuron nhân tạo cũng có thể rất cạn. Mạng nơ-ron nhân tạo mô phỏng cấu trúc mạng lưới thần kinh của con người. Mức cơ bản được gọi là perceptron là một biểu diễn toán học của một neuron sinh học. Giống như vỏ não, có thể có nhiều lớp perceptron kết nối với nhau.



Hình 2.1: Sơ đồ Deep Learning

2.1.2 Các thành phần của mạng học sâu.

Một mạng nơ-ron chuyên sâu có các thành phần sau.

Lớp đầu vào

Một mạng nơ-ron nhân tạo sẽ có một số nút để nhập dữ liệu đầu vào. Các nút này tạo nên lớp đầu vào của hệ thống.

Lớp ẩn

Lớp đầu vào xử lý và chuyển dữ liệu đến các lớp sâu hơn trong mạng nơ-ron. Các lớp ẩn này xử lý thông tin ở các cấp độ khác nhau, thích ứng với hành vi của mình khi nhận được thông tin mới. Các mạng học sâu có hàng trăm lớp ẩn có thể được dùng để phân tích một vấn đề từ nhiều góc độ khác nhau.

Ví dụ: nếu bạn phải phân loại hình ảnh của một loài vật chưa xác định, bạn sẽ cần so sánh hình ảnh này với các loài vật đã biết. Chẳng hạn, bạn sẽ quan sát hình dáng của cặp mắt, đôi tai, kích thước, sô chi và hình mẫu lông của loài vật đó. Bạn sẽ cố gắng xác định các hình mẫu, chẳng hạn như sau:

- Loài vật có móng guốc nên nó có thể là bò hoặc hươu.
- Loài vật có mắt mèo nên nó có thể là một loài mèo hoang dã nào đó.

Các lớp ẩn trong mạng nơ-ron chuyên sâu hoạt động theo cùng một cách. Nếu một thuật toán học sâu đang cố gắng phân loại một hình ảnh động vật, mỗi lớp ẩn của thuật toán này sẽ xử lý một đặc điểm khác nhau của con vật và cố gắng phân loại chính xác nó.

Lớp đầu ra

Lớp đầu ra bao gồm các nút xuất dữ liệu. Các mô hình học sâu xuất ra đáp án "có" hoặc "không" chỉ có hai nút trong lớp đầu ra. Mặt khác, các mô hình xuất ra nhiều đáp án hơn sẽ có nhiều nút hơn.

2.1.3 Các loại Neural Network

Các mô hình deep learning có khả năng tự động học các tính năng từ dữ liệu, mang lại hiệu quả cho các tác vụ như nhận diện hình ảnh, nhận diện giọng nói và xử lý ngôn ngữ tự nhiên (NLP). Các kiến trúc nổi bật trong deep learning bao gồm mạng nơron truyền thẳng (feedforward neural networks - FNN), mạng nơron tích chập (Convolutional Neural Networks - CNN), và mạng nơron hồi quy (Recurrent Neural Networks - RNN).

- **Mạng nơron truyền thẳng (FNN):** Đại diện cho loại đơn giản nhất của ANN, có luồng thông tin tuyến tính trên toàn mạng. FNN được ứng dụng rộng rãi trong các nhiệm vụ như phân loại hình ảnh, nhận diện giọng nói, và xử lý ngôn ngữ tự nhiên.

- **Mạng nơron tích chập (CNN):** Được thiết kế đặc biệt cho các nhiệm vụ nhận diện hình ảnh và video, CNNs có khả năng tự động tìm hiểu các tính năng từ hình ảnh. Đặc điểm này giúp CNNs phù hợp với các nhiệm vụ như phân loại hình ảnh, phát hiện đối tượng và phân loại hình ảnh.
- **Mạng nơron hồi quy (RNN):** Thuộc loại mạng nơron có khả năng xử lý dữ liệu tuần tự, bao gồm chuỗi thời gian và ngôn ngữ tự nhiên. RNN có khả năng duy trì trạng thái bên trong để thu thập thông tin về các đầu vào trước đó, giúp chúng thích hợp với các nhiệm vụ như nhận diện giọng nói, xử lý ngôn ngữ tự nhiên và dịch thuật ngôn ngữ.

2.2 Thị giác máy tính và phát hiện đối tượng.

Khởi điểm của thị giác máy tính có thể được xác định lịch sử trở lại những năm 1960. Trong thời kỳ này, nhu cầu về việc giải mã và đánh giá dữ liệu hình ảnh đã dẫn đến sự ra đời của các kỹ thuật tiên phong, cho phép máy tính nhận diện và phân loại các mẫu và đối tượng trong các hình ảnh. Trung tâm của thị giác máy tính là các kỹ thuật như xử lý hình ảnh, phát hiện đối tượng và nhận dạng mẫu. Nguyên tắc cơ bản của thị giác máy tính nằm ở khả năng chuyển đổi hình ảnh thành dữ liệu số, sau đó được sử dụng như đầu vào tính toán. Việc chuyển đổi hình ảnh thành dữ liệu số bao gồm một số bước, bao gồm việc thu thập hình ảnh, tiền xử lý, trích xuất đặc trưng và thường là các phương pháp dựa trên học sâu.

Quá trình chuyển đổi được thực hiện thông qua việc sắp xếp các hình ảnh ở mức độ pixel, trong đó mỗi pixel có thể được đặc trưng bằng một giá trị số, hoặc là dạng xám hoặc là sự kết hợp của các giá trị số (ví dụ, 255, 0, 0 trong mô hình màu RGB). Hình ảnh xám biểu diễn mức độ sáng của mỗi pixel bằng một giá trị số duy nhất. Giá trị này thường nằm trong khoảng từ 0 (đen) đến 255 (trắng), với các bóng độ khác nhau của màu xám ở giữa. Hình ảnh xám thường được sử dụng vì tính đơn giản và hiệu quả trong xử lý hình ảnh. Hình ảnh màu RGB, bên cạnh đó, sử dụng một kết hợp của ba giá trị số cho mỗi pixel, biểu thị độ mạnh của các kênh màu đỏ, xanh lá cây và xanh lam.

Là một phân nhánh của trí tuệ nhân tạo, thị giác máy tính tạo điều kiện cho khả năng giải mã của máy móc và thực hiện hành động dựa trên thông tin xuất phát từ các nguồn hình ảnh như ảnh, video và các đầu vào dựa trên hình ảnh khác.

Một lĩnh vực cụ thể được quan tâm là phát hiện xe cộ, bao gồm việc xác định và theo dõi các phương tiện giao thông trong thời gian thực bằng các kỹ thuật thị giác máy tính. Một số phương pháp phát hiện xe cộ đã được đề xuất, bao gồm các phương pháp dựa trên đặc điểm, dựa trên diện mạo và dựa trên học sâu. Các phương pháp dựa trên đặc điểm dựa vào các đặc điểm được tạo thủ công như cạnh, góc và lược đồ màu để phát hiện xe cộ, trong khi các phương pháp dựa trên diện mạo sử dụng các mẫu hoặc bộ phân loại được huấn luyện trên các hình ảnh của xe cộ để phát hiện chúng. Ngược lại, các phương pháp dựa trên học sâu sử dụng các mạng nơ-ron để học các biểu diễn của các đặc điểm xe cộ trực tiếp từ dữ liệu hình ảnh thô.

Với sự tiến bộ trong công nghệ và sự tăng lên của lưu lượng giao thông trên đường, việc phát hiện xe cộ đã trở thành một phần quan trọng của an toàn giao thông. Các phương pháp phát hiện xe cộ hiện đại và phù hợp nhất bao gồm camera, drone và công nghệ LiDAR. Mỗi phương pháp có ưu điểm và hạn chế riêng, và phương pháp hiệu quả nhất phụ thuộc vào ứng dụng cụ thể và môi trường.

Camera là một trong những phương pháp phổ biến nhất cho việc phát hiện xe cộ. Chúng có thể được lắp đặt trên đường hoặc trên các phương tiện và chụp hình ảnh của đường. Các thuật toán thị giác máy tính sau đó được sử dụng để phát hiện xe cộ. Mặc dù phương pháp này hiệu quả trong điều kiện thời tiết tốt, nhưng có thể gặp khó khăn trong mưa lớn, tuyết rơi hoặc ánh nắng mặt trời chói chang.

Các thuật toán học máy cũng được phát triển cho việc phát hiện xe cộ. Các thuật toán này có thể học nhận diện các loại xe cộ khác nhau và thích nghi với các môi trường khác nhau. Thường thì chúng được kết hợp với camera, drone và công nghệ LiDAR để cải thiện độ chính xác của việc phát hiện xe cộ. Một trong những lợi ích của các thuật toán học máy là khả năng thích nghi với môi trường thay đổi. Chúng có thể học nhận diện các điều kiện thời tiết khác nhau, điều kiện ánh sáng và điều kiện đường và điều chỉnh phương pháp phát hiện của mình tương ứng. Điều này làm cho chúng đặc biệt hiệu quả trong các khu vực nơi điều kiện có thể thay đổi thường xuyên, như các tuyến đường cao tốc hoặc khu vực xây dựng. Một lợi ích khác của các thuật toán học máy là khả năng phát hiện các xe bị một phần che khuất hoặc khó nhìn. Ví dụ, chúng có thể phát hiện các xe bị che khuất một phần bởi cây cối hoặc tòa nhà hoặc đang di chuyển nhanh qua khu vực đông đúc. Điều này có thể cải thiện an toàn giao thông bằng cách cảnh báo tài xế về các nguy cơ tiềm ẩn mà họ có thể không nhận ra nếu không có hệ thống cảnh báo. Tuy nhiên, cũng có một số hạn chế đối với các thuật toán học máy. Chúng yêu cầu lượng dữ liệu lớn cho quá trình huấn luyện, điều này có thể tốn thời gian và chi phí. Ngoài ra, chúng có thể không chính xác bằng các phương pháp phát hiện xe cộ khác trong một số tình huống. Ví dụ, chúng có thể gặp khó khăn trong việc nhận diện các phương tiện giao thông có ngoại hình rất giống nhau, như hai chiếc ô tô giống hệt nhau đậu cạnh nhau.

Hai phương pháp dựa trên học sâu phổ biến cho việc phát hiện xe cộ là thuật toán You Only Look Once (YOLO) và thuật toán Mạng Nơ-ron Tích chập dựa trên Vùng (Fast R-CNN). YOLO sử dụng một mạng nơ-ron duy nhất để thực hiện phát hiện và phân loại đối tượng trong thời gian thực, trong khi thuật toán Faster R-CNN sử dụng một quy trình hai giai đoạn đầu tiên để đề xuất các vùng quan tâm và sau đó phân loại chúng là xe cộ hoặc không phải là xe cộ.

2.3 Quy trình hoạt động của các mô hình phát hiện đối tượng

Các mô hình được cung cấp các nhãn dán (label) được mã hoá thông qua các bộ mã hoá nhãn (label encoder) trong quá trình huấn luyện để tối ưu các dự đoán của mình sau mỗi lần đưa ra dự đoán nhờ vào hàm mát mát. Đồng thời, các đầu ra của mô hình cũng được đưa qua một bộ giải mã đầu ra (model's output decoder) để đưa ra các dự đoán có thể sử dụng được trong quá trình suy luận và đánh giá mô hình.

Mã hoá nhãn (label encoding) là quá trình ánh xạ các nhãn đối tượng thành các chỉ số dạng số nguyên để mô hình có thể hiểu được. Trong khi đó, giải mã đầu ra (model's output decoding) là quá trình chuyển đổi đầu ra của mô hình từ dạng số thành các đối tượng có ý nghĩa với con người, ví dụ như tên đối tượng và trong trường hợp này là vị trí và kích thước của các phương tiện trong hình ảnh.

2.4 Các hệ số đánh giá hiệu suất phát hiện đối tượng và loại bỏ không tối đa (Non-Maximum Suppression)

2.4.1 Các hệ số đánh giá

Hệ số đánh giá là công cụ chính để đánh giá độ chính xác và hiệu quả của các mô hình phát hiện đối tượng. Chúng làm rõ mức độ hiệu quả của một mô hình trong việc xác định và vị trí hóa các đối tượng trong hình ảnh. Ngoài ra, chúng giúp hiểu cách mô hình xử lý các dự đoán tích cực giả (false positive) và tiêu cực giả (false negative). Hiểu biết này rất quan trọng để đánh giá và cải thiện hiệu suất của mô hình.

Intersection over Union (IoU): IoU là thước đo định lượng sự chồng chéo giữa hộp giới hạn dự đoán (predicted bounding box) và ground truth bounding box (hộp giới hạn thực tế). Đây là cách để xem xét xem hộp giới hạn dự đoán có khớp với vị trí thực tế của hộp giới hạn không. Nó được sử dụng để xác định xem việc phát hiện có chính xác không dựa trên một ngưỡng (threshold) đã được xác định trước. Ngưỡng này giống như một tiêu chuẩn được sử dụng để quyết định liệu dự đoán có đủ tốt không. Nếu giá trị IoU đạt hoặc vượt qua ngưỡng này, chúng ta nói đó là một *true positive* có nghĩa là đó là một dự đoán chính xác. Tuy nhiên, nếu nó không đạt được tiêu chuẩn này, thì nó được xem như là một *false positive* một dự đoán sai. Lựa chọn của ngưỡng phụ thuộc vào nhiệm vụ cụ thể và có thể thay đổi dựa trên kỳ vọng về độ chính xác của mô hình. Trong các nhiệm vụ phát hiện đối tượng, ngưỡng IoU cũng giúp tìm ra độ chính xác, cho biết có bao nhiêu dự đoán tích cực chính xác (*true positive*) chúng ta nhận được trong số tất cả các dự đoán tích cực đã được thực hiện (bao gồm cả *true positive* và *false positive*).

$$IoU = \frac{\text{Intersection Area}}{\text{Union Area}} \quad (2.1)$$

Precision và Recall: Precision đo lường độ chính xác của các dự đoán tích cực của mô hình, trong khi recall đo lường tỷ lệ của các trường hợp tích cực thực sự mà mô hình xác định đúng. Thường có sự đánh đổi giữa *Precision* và *Recall*; ví dụ, tăng số lượng đối tượng được phát hiện (*Recall* cao hơn) có thể dẫn đến nhiều dự đoán sai (*Precision* thấp hơn). Để tính toán cho sự đánh đổi này, hệ số AP tích hợp đường cong *Precision-Recall*, mô tả *Precision* so với *Recall* cho các ngưỡng tin cậy khác nhau. Hệ số này cung cấp một đánh giá cân bằng giữa *Precision* và *Recall* bằng cách xem xét diện tích dưới đường cong *Precision-Recall*.

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2.2)$$

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2.3)$$

Điểm F1: Điểm *F1* là trung bình hài hòa của độ chính xác và độ hồi tưởng của mô hình, cung cấp

dánh giá cân bằng về hiệu suất của mô hình trong khi xem xét cả *False Positive* và *False Negative*.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2.4)$$

Độ chính xác trung bình (AP): AP tính toán diện tích dưới đường cong *Precision-Recall*, cung cấp một giá trị duy nhất gói gọn độ chính xác (*precision*) và hiệu suất thu hồi (*recall*) của mô hình.

$$AP = \sum_{k=0}^{n-1} [Recall_{(k+1)} - Recall_{(k)}] \times Precision_{(k+1)} \quad (2.5)$$

Trung bình cộng độ chính xác trung bình (mAP): mAP mở rộng khái niệm AP bằng cách tính toán các giá trị AP trung bình trên nhiều lớp đối tượng. Điều này rất hữu ích trong các tình huống phát hiện nhiều lớp đối tượng để cung cấp đánh giá toàn diện về hiệu suất của mô hình.

$$mAP = \frac{1}{n} \sum_{k=1}^n AP_k \quad (2.6)$$

2.4.2 Cách tính mAP

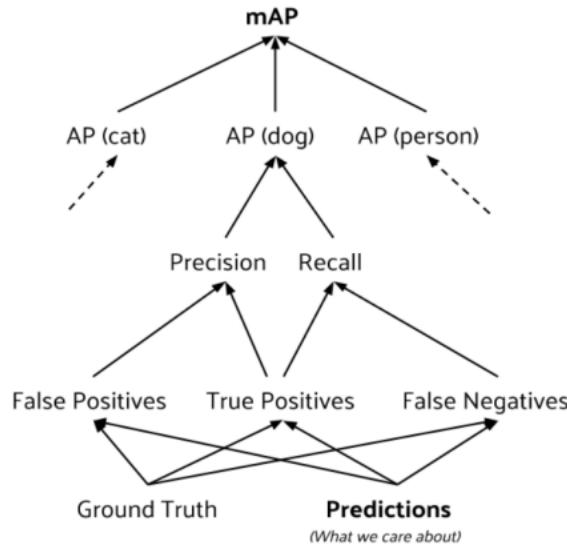
Để tính toán mAP, người ta xem xét một số yếu tố, bao gồm confusion matrix - một bảng cung cấp cái nhìn sâu sắc vào hiệu suất của thuật toán - cũng như IoU, đánh giá độ chính xác của thuật toán trong phát hiện đối tượng. Ngoài ra, tỷ lệ recall (số lượng true positive được xác định đúng) và precision (số lượng positive được xác định đúng trong số tất cả các positive được xác định) cũng được xem xét.

Điểm mAP không chỉ xem xét sự cân bằng giữa precision và recall, tìm kiếm một sự cân bằng tối ưu, mà còn giải quyết các lỗi như false positive (xác định một cái gì đó là đối tượng một cách không chính xác) và false negative (không phát hiện được một đối tượng thực sự).

Bằng cách tính toán tất cả các mặt này, điểm mAP cung cấp một đánh giá toàn diện về hiệu suất của thuật toán. Đây là lý do tại sao nó được rộng rãi chấp nhận bởi các nhà nghiên cứu trong lĩnh vực thị giác máy tính, phục vụ như một tiêu chuẩn đáng tin cậy để đánh giá sự mạnh mẽ và đáng tin cậy của các mô hình phát hiện đối tượng khác nhau.

Các bước để tính mAP:

- Sử dụng mô hình để tạo điểm dự đoán (prediction scores).
- Chuyển đổi các điểm dự đoán thành nhãn lớp (class label).
- Tính toán confusion matrix, bao gồm *True Positives* (TP), *False Positives* (FP), *True Negatives* (TN), và *False Negatives* (FN).
- Tính toán các chỉ số *Precision* và *Recall*.
- Tính toán diện tích dưới đường cong *Precision-Recall*.
- Đo lường độ chính xác trung bình (AP) cho mỗi lớp.



Hình 2.2: mAP flowchart

- Tính toán mAP bằng cách tìm độ chính xác trung bình cho mỗi lớp và sau đó tính trung bình qua số lượng lớp.

2.4.3 Hàm mất mát - Loss function

Trong hệ thống phát hiện đối tượng YOLO, hàm mất mát là một hàm đo lường sự khác biệt giữa các giá trị dự đoán và thực tế cho đối tượng, các lớp dự đoán, và tọa độ hộp giới hạn. Hàm mất mát được thiết kế để đánh trọng số cho các sai số trong các tọa độ hộp giới hạn và phát hiện đối tượng hơn là các sai số trong lớp dự đoán. Nó thường tích hợp sai số toàn phương trung bình (Mean Square Error) cho dự đoán hộp giới hạn và sai số hồi quy logistic cho dự đoán lớp và tinh đối tượng.

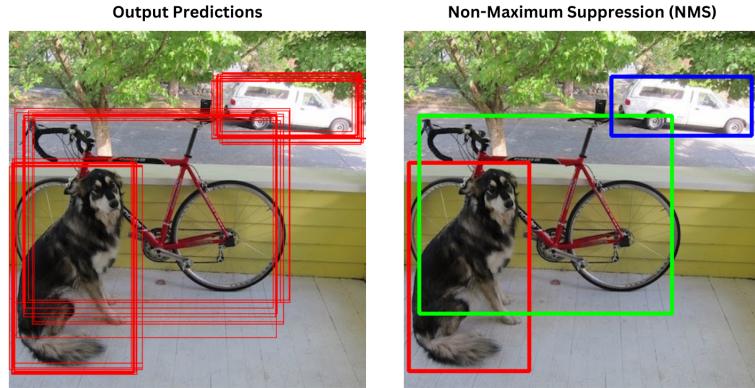
2.4.4 Loại bỏ sự chồng chéo không tối đa (Non-maximum Suppression)

Cơ chế Non-Maximum Suppression (NMS) [3] là một kỹ thuật xử lý sau khi dự đoán được sử dụng trong các thuật toán phát hiện đối tượng để giảm số lượng các hộp giới hạn trùng lặp và cải thiện chất lượng tổng thể của việc phát hiện. Các thuật toán phát hiện đối tượng thường tạo ra nhiều hộp giới hạn xung quanh cùng một đối tượng với các điểm số tin cậy khác nhau. NMS loại bỏ các hộp giới hạn trùng lặp và không liên quan, chỉ giữ lại những hộp giới hạn chính xác nhất. Thuật toán 1 mô tả quy trình. Hình 2.3 hiển thị kết quả điển hình của một mô hình phát hiện đối tượng chứa nhiều hộp giới hạn chồng chéo và đầu ra sau khi áp dụng NMS.

Algorithm 1 Thuật toán Loại bỏ Sự Chồng Chéo Không Tối Đa (Non-Maximum Suppression)

Require: Tập hộp giới hạn dự đoán B , điểm tự tin S , ngưỡng IoU τ , ngưỡng tự tin T
Ensure: Tập hộp giới hạn đã lọc F

```
1:  $F \leftarrow \emptyset$ 
2: Lọc các hộp giới hạn:  $B \leftarrow \{b \in B | S(b) \geq T\}$ 
3: Sắp xếp các hộp giới hạn  $B$  theo điểm tự tin giảm dần
4: while  $B \neq \emptyset$  do
5:   Chọn hộp  $b$  có điểm tự tin cao nhất
6:   Thêm  $b$  vào tập hộp giới hạn cuối cùng  $F$ :  $F \leftarrow F \cup \{b\}$ 
7:   Loại bỏ  $b$  khỏi tập hộp giới hạn  $B$ :  $B \leftarrow B - \{b\}$ 
8:   for all các hộp còn lại  $r$  trong  $B$  do
9:     Tính IoU giữa  $b$  và  $r$ :  $iou \leftarrow IoU(b, r)$ 
10:    if  $iou \geq \tau$  then
11:      Loại bỏ  $r$  khỏi tập hộp giới hạn  $B$ :  $B \leftarrow B - \{r\}$ 
```



Hình 2.3: NMS

2.5 Mạng Nơ-ron tích chập

Mạng Nơ-ron Tích chập (CNN) là một loại mô hình học sâu trong mảng Trí tuệ Nhân tạo thường được sử dụng cho các nhiệm vụ thị giác máy tính khác nhau, bao gồm phân loại hình ảnh, phát hiện đối tượng và phân đoạn hình ảnh. CNN được thiết kế đặc biệt để xử lý và phân tích dữ liệu hình ảnh, làm cho chúng hiệu quả cao trong các nhiệm vụ liên quan đến hình ảnh và mẫu hình ảnh. CNN được thiết kế một cách rõ ràng cho dữ liệu dạng lưới, như hình ảnh, nơi sắp xếp các pixel có ý nghĩa không gian. Một CNN học các mẫu trong hình ảnh bằng cách sử dụng sự phụ thuộc giữa các ô lân cận.

CNN bao gồm một lớp đầu vào, một lớp đầu ra và nhiều lớp ẩn nằm giữa. Các lớp này thực hiện các hoạt động thay đổi dữ liệu để học các đặc trưng cụ thể cho dữ liệu. Một số Khái niệm cơ bản trong CNN cho việc Phát hiện đối tượng bao gồm:

- **Các Lớp Tích chập:** Các lớp này sử dụng bộ lọc để quét một hình ảnh và xác định các mẫu hoặc đặc điểm cụ thể, như cạnh hoặc các cấu trúc texture. Các lớp tích chập trượt các bộ lọc

nhỏ qua hình ảnh đầu vào, tính toán tích vô hướng để phát hiện các mẫu. Mỗi bộ lọc chuyên biệt trong việc nhận dạng một đặc điểm cụ thể.

- **Hàm Kích hoạt:** Sau khi tích chập, một hàm kích hoạt được áp dụng vào các bản đồ đặc trưng kết quả, giới thiệu các phi tuyến tính để tăng cường khả năng của mạng trong việc nắm bắt các mối quan hệ phức tạp. Một hàm kích hoạt thường được sử dụng là ReLU, giữ các giá trị dương và đặt các giá trị âm thành không.
- **Các Lớp Gộp:** Các lớp gộp giảm các chiều không gian của các đặc trưng bằng cách tóm tắt thông tin trong các vùng nhỏ trong khi vẫn giữ lại các đặc trưng cần thiết.
- **Lớp Kết nối Đầu đuôi:** Lớp kết nối đầu đuôi thường được tìm thấy ở cuối của một CNN và chịu trách nhiệm cho việc phân loại. Nó nhận đầu vào từ các lớp trước đó và tạo ra đầu ra cuối cùng, gán xác suất cho các lớp đối tượng khác nhau.

Những khái niệm cơ bản này tạo thành một khái niệm nền tảng của các mô hình CNN cho việc phát hiện đối tượng. Chúng cho phép mạng trích xuất các đặc trưng liên quan, phân loại đối tượng và tạo ra dự đoán chính xác. Trong quá trình huấn luyện, CNN điều chỉnh các tham số của mình (trọng số và sai số) để giảm thiểu sự khác biệt giữa xác suất lớp được dự đoán và nhãn thực tế. Quá trình tối ưu hóa này, được gọi là lan truyền ngược, lặp lại cập nhật các tham số để cải thiện độ chính xác của mạng.

Trong các mô hình phát hiện đối tượng, các lớp CNN được sử dụng để trích xuất các đặc trưng từ hình ảnh đầu vào, với các lớp ban đầu phát hiện các mẫu đơn giản và các lớp sâu hơn nắm bắt các đặc trưng phức tạp hơn. Việc tích hợp CNN vào các mô hình phát hiện đã cho phép phát hiện đối tượng thời gian thực với các kiến trúc như YOLO (You Only Look Once) xử lý toàn bộ hình ảnh bằng một CNN duy nhất.

2.6 Các mô hình hai giai đoạn

2.6.1 Mạng Nơ-ron Tích chập Dựa trên Vùng (R-CNN)

R-CNN [4], hay Region-based Convolutional Neural Network, là một trong những mô hình CNN đầu tiên được thiết kế đặc biệt cho việc phát hiện đối tượng trong hình ảnh. Kiến trúc của R-CNN bao gồm ba mô-đun chính:

- Trích xuất đề xuất vùng (Region Proposal Network - RPN): Mô-đun này đề xuất một tập hợp các vùng đề xuất chứa đối tượng có thể trong hình ảnh. RPN sử dụng một mạng neural để dự đoán các vùng có khả năng chứa đối tượng dựa trên các đặc trưng hình ảnh.
- Biến đổi hình ảnh affine (RoI Pooling): Sau khi có được các đề xuất vùng, hình ảnh trong mỗi vùng được chọn được biến đổi thành một kích thước cố định để phù hợp với đầu vào của mạng CNN.
- Phân loại dựa trên lớp SVM (SVM-based classification): Các đặc trưng của các vùng được trích xuất từ mạng CNN và được sử dụng để phân loại các đối tượng trong các vùng này. Trong phiên bản ban đầu của R-CNN, một bộ phân loại dựa trên máy vector hỗ trợ (SVM) được sử dụng để phân loại các vùng.

Mặc dù R-CNN đã đạt được kết quả tốt trong việc phát hiện đối tượng, nó vẫn có nhược điểm là tốc độ chậm do phải tính toán các đặc trưng cho mỗi vùng đề xuất.

2.6.2 Mask R-CNN

Mask R-CNN [5] không chỉ là một phần mở rộng của R-CNN mà còn điều chỉnh kiến trúc để có thể dự đoán các mặt nạ phân đoạn ở mức pixel cho các đối tượng trong hình ảnh. Điều này làm cho mô hình có khả năng định vị và phân đoạn đối tượng cùng một lúc.

Mask R-CNN sử dụng một mạng lưới tương đối (Feature Pyramid Network - FPN) và một đường dẫn từ dưới lên để cải thiện việc trích xuất các đặc trưng ở các cấp độ khác nhau của hình ảnh. Điều này giúp cho việc nhận diện các đối tượng ở cả những vùng chi tiết nhỏ và những vùng lớn trong hình ảnh.

Kiến trúc của Mask R-CNN bao gồm ba nhánh chính:

- Dự đoán hộp giới hạn (Bounding Box Prediction): Nhánh này dự đoán các hộp giới hạn chứa đối tượng trong hình ảnh.
- Dự đoán lớp đối tượng (Object Class Prediction): Nhánh này dự đoán lớp của đối tượng trong mỗi hộp giới hạn được đề xuất.
- Tạo mặt nạ (Mask Generation): Nhánh này tạo ra các mặt nạ phân đoạn ở mức pixel cho các đối tượng trong hình ảnh, cho biết vị trí cụ thể của đối tượng trong không gian pixel.

Nhờ vào kiến trúc này, Mask R-CNN có khả năng đồng thời định vị, phân loại và phân đoạn đối tượng trong hình ảnh một cách chính xác và hiệu quả.

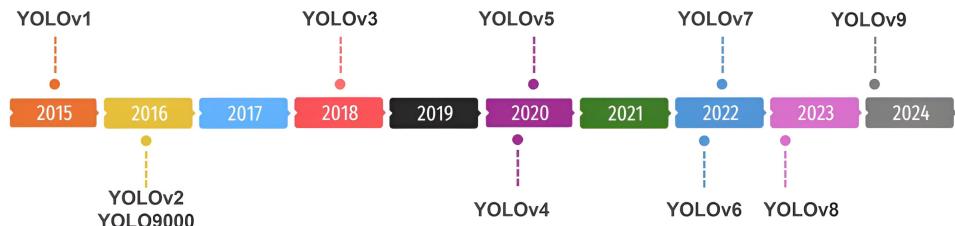
2.6.3 Faster R-CNN

Faster R-CNN [6] được giới thiệu bởi Ren và cộng sự vào năm 2017 trong bài báo của họ "Towards Real-Time Object Detection with Region Proposal Networks". Nó được phát triển để giải quyết vấn đề chậm trễ tính toán trong các hệ thống phát hiện đối tượng do các thuật toán xác định vùng có khả năng chứa đối tượng gây ra. Faster R-CNN là một thuật toán phát hiện đối tượng được sử dụng rộng rãi trong các nhiệm vụ thị giác máy tính. Nó bao gồm hai mô-đun: một mạng tích chập sâu hoàn toàn để đề xuất các vùng và một bộ phát hiện Fast R-CNN sử dụng các vùng được đề xuất. Mạng Đề Xuất Khu vực (RPN) là một mạng tích chập hoàn toàn dự đoán ranh giới đối tượng và điểm độ tin cậy của đối tượng tại mỗi vị trí. RPN được huấn luyện để tạo ra các đề xuất vùng chất lượng cao, sau đó được sử dụng bởi bộ phát hiện Fast R-CNN cho nhiệm vụ phát hiện đối tượng. RPN trong Faster R-CNN cho phép đề xuất khu vực gần như không tốn kém bằng cách chia sẻ đặc trưng tích chập với mạng phát hiện. Điều này giảm thiểu vấn đề chậm trễ tính toán của việc đề xuất khu vực và cho phép hệ thống chạy gần như ở tốc độ khung hình thời gian thực. Trong một mức độ chi tiết hơn thì mạng cơ sở được sử dụng để trích xuất các đặc trưng từ hình ảnh. Những đặc trưng này sau đó được đưa vào Mạng Đề Xuất Vùng (RPN) để tạo ra các khu vực đề xuất, có thể là vị trí hộp giới hạn cho các đối tượng.

2.7 Các mô hình dựa trên huật toán YOLO

YOLO là một thuật toán phát hiện đối tượng thời gian thực được giới thiệu vào năm 2015. Nó sử dụng một phương pháp hồi quy thay vì phân loại và liên kết xác suất để phát hiện các đối tượng bằng cách sử dụng một mạng nơ-ron tích chập duy nhất để tách các hộp giới hạn không gian và liên kết xác suất với mỗi đối tượng được phát hiện. Kiến trúc của YOLO bao gồm việc thay đổi kích thước của hình ảnh đầu vào, áp dụng các phép tích chập và sử dụng các kỹ thuật như chuẩn hóa batch và dropout để cải thiện hiệu suất. YOLO đã phát triển qua các năm, với các phiên bản từ v1 đến v9 giải quyết các hạn chế như phát hiện các đối tượng nhỏ hơn và hình dạng không bình thường. Nó đã trở nên phổ biến nhờ vào độ chính xác, khả năng tổng quát hóa và là mã nguồn mở. Trong nghiên cứu này, chúng tôi đã quyết định tiến hành với YOLOv8 và YOLOv9.

2.7.1 Các giai đoạn của YOLO



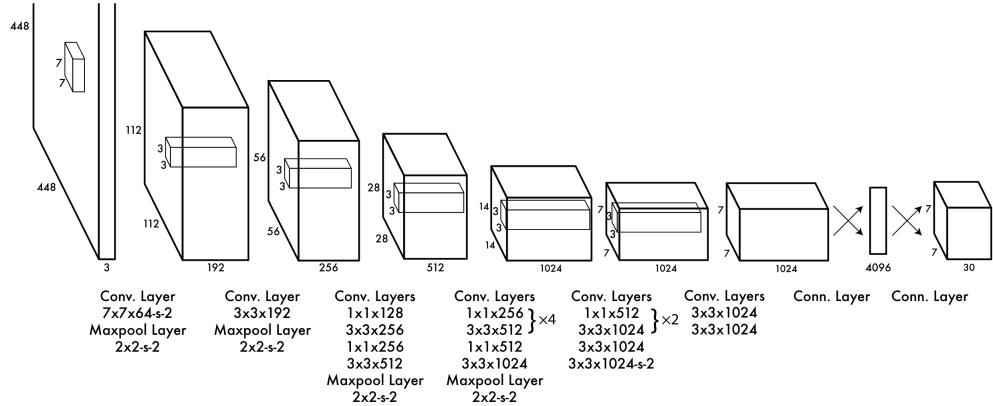
Hình 2.4: Sự phát triển của YOLO qua các năm

2.7.2 Xuất phát điểm - YOLOv1

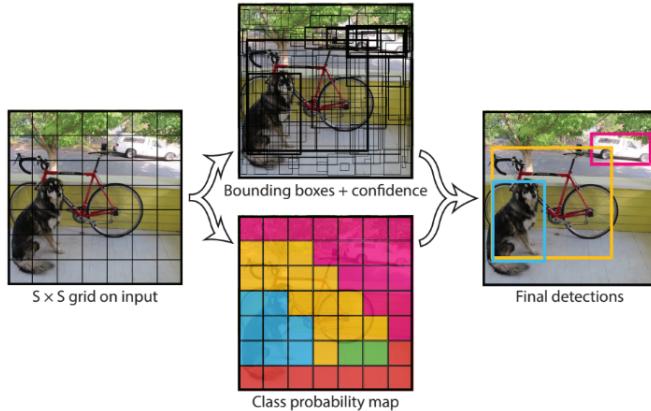
Phiên bản đầu tiên của YOLO [7], được phát hành vào năm 2015, đã là một bước đột phá trong việc phát hiện đối tượng. Phát triển bởi Joseph Redmon và các cộng sự, YOLOv1 sử dụng một mạng nơ-ron tích chập (CNN) để đồng thời dự đoán các hộp giới hạn (Bounding box) và xác suất của các lớp (Class) trực tiếp trong một lượt đi qua toàn bộ các vị trí có thể có trong lưới hình ảnh được tạo ra trong quá trình dự đoán.

Sự đổi mới chính của YOLOv1 nằm ở khả năng chia hình ảnh đầu vào thành một lưới và dự đoán các hộp giới hạn và xác suất lớp cho mỗi ô lưới. Phương pháp dựa trên lưới này giúp YOLO đạt được tốc độ và hiệu suất đáng kinh ngạc trong khi vẫn duy trì độ chính xác tương đối khi so với các mô hình thời gian thực hiện có tại cùng thời điểm. Hơn nữa, YOLOv1 không phụ thuộc vào các đề xuất vùng hoặc các bước xử lý sau phức tạp, làm cho nó đơn giản và hiệu quả hơn so với các phương pháp phát hiện đối tượng trước đó. Tuy nhiên, mô hình này cũng tồn đọng các nhược điểm và bất lợi như gặp khó khăn trong việc phát hiện các hình ảnh nhỏ hơn trong một nhóm và không thể phát hiện các hình dạng mới hoặc không bình thường.

Cho một hình ảnh đầu vào I , YOLO chia hình ảnh thành một lưới $S \times S$. Mỗi ô lưới chịu trách nhiệm dự đoán B hộp giới hạn và các điểm số tin cậy tương ứng $\text{Pr}(\text{object})$. Đối với mỗi hộp giới hạn, YOLO dự đoán 5 tham số: (x, y, w, h) cho tọa độ hộp giới hạn liên quan đến ô lưới và $\text{Pr}(\text{class}_i | \text{object})$ cho xác suất lớp có điều kiện.



Hình 2.5: Kiến trúc mô hình YOLOv1

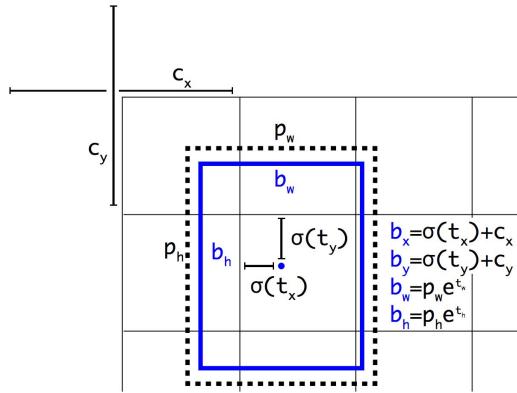


Hình 2.6: Cách thức hoạt động của YOLOv1

Điểm số tin cậy $\text{Pr}(\text{object})$ biểu thị xác suất rằng ô lưới chứa một đối tượng, trong khi $\text{Pr}(\text{class}_i|\text{object})$ chỉ ra xác suất rằng đối tượng phát hiện được thuộc lớp i . Các xác suất này được tính bằng phương pháp hồi quy logistic. Đầu ra cuối cùng của YOLOv1 là một tensor có hình dạng $S \times S \times (B \times 5 + C)$, trong đó C là tổng số lớp.

2.7.3 Tăng cường cải thiện - YOLOv2

YOLOv2 [8], còn được gọi là YOLO9000, giới thiệu một kiến trúc mạng mới được gọi là Darknet-19 cùng với nhiều cải tiến quan trọng đã cải thiện độ chính xác và tốc độ của việc phát hiện đối tượng. Trong đó nổi bật là đề xuất sử dụng các hộp giới hạn neo trước để dự đoán kích thước và vị trí của các hộp giới hạn, giúp cải thiện độ chính xác của việc dự đoán.



Hình 2.7: hộp giới hạn neo trong YOLOv2

Trong dự đoán của mỗi hộp giới hạn sẽ bao gồm các giá trị t_x, t_y, t_w, t_h tương ứng với giá trị phần bù cho vị trí của hộp dự đoán so với ô lưỡi và kích thước của hộp giới hạn dự đoán so với hộp giới hạn neo tiên nghiệm. Sau đó, vị trí và kích thước dự đoán của hộp sẽ được tính theo công thức như sau:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned}$$

Với σ là hàm *sigmoid*, (c_x, c_y) là toạ độ (x, y) của góc bên trái ô lưỡi, p_w và p_h là giá trị chuẩn hoá của kích thước hộp dự đoán so với kích thước hình ảnh đầu vào.

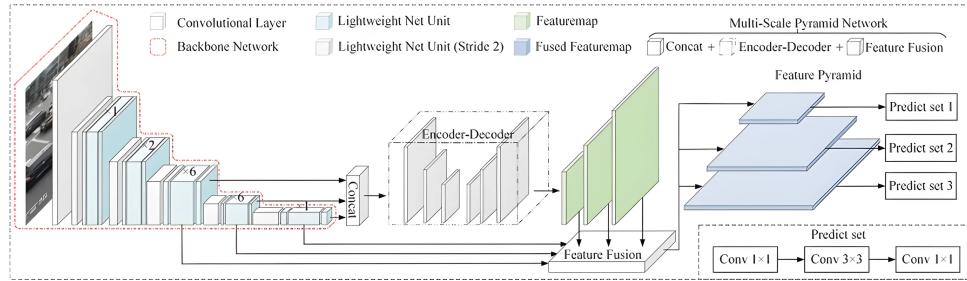
2.7.4 Nâng cấp - YOLOv3

YOLOv3 [9] xây dựng trên YOLOv2 bằng cách thực hiện dự đoán ở các tỷ lệ khác nhau tương tự như mạng đặc trưng kim tự tháp FPN (Feature Pyramid Networks) [10] đồng thời kết hợp phép gộp không gian kim tự tháp SPP (Spatial Pyramid Pooling) [11] và tận dụng liên kết dư (Residual Connection) [12], cho phép mô hình trích xuất các thông tin ngữ nghĩa tốt hơn và kết quả đầu có chất lượng cao hơn. Nó cũng đạt được tốc độ và độ chính xác tối ưu so với các phiên bản trước đó và các công cụ phát hiện đối tượng tiên tiến khác.

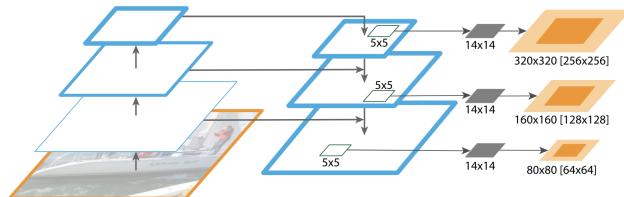
YOLOv3 mang khả năng trích xuất tính năng nhiều hơn với 53 lớp tích chập và các liên kết dư (Residual Connection) được sử dụng trên kiến trúc mạng Darknet-53 được mô tả như hình 2.8 dưới đây.

Kiến trúc của YOLOv3 cũng bao gồm một mạng đặc trưng kim tự tháp FPN (Feature Pyramid Network), là một bộ trích xuất đặc trưng có đầu ra là các bản đồ đặc trưng có kích thước với các tỷ lệ khác nhau, theo kiểu mạng nơ-ron tích chập hoàn toàn.

Bên cạnh các cải tiến trên, YOLOv3 cũng là phiên bản YOLO đầu tiên tích hợp liên kết dư (Residual Connection) vào backbone. Liên kết này giúp mạng có khả năng thành lập hàm định danh

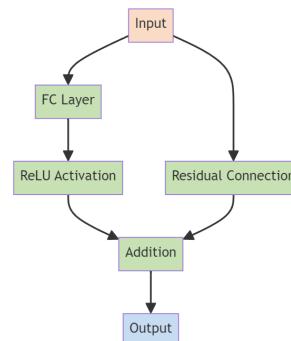


Hình 2.8: Kiến trúc mô hình YOLOv3



Hình 2.9: Mạng đặc trưng kim tự tháp FPN trong YOLOv3

nhờ bỏ qua các kết nối ở các tầng phía sau, giúp thông tin ở các tầng trước được đi sâu hơn vào trong mạng lưới.

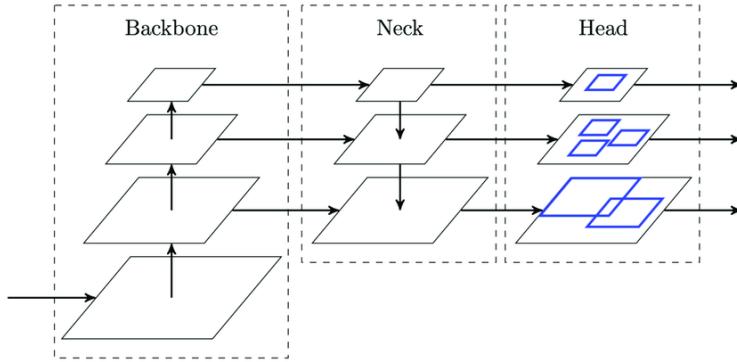


Hình 2.10: Liên kết dư được tích hợp trong YOLOv3

2.7.5 Mô-đun hóa các thành phần của mô hình YOLO

Kể từ YOLOv4 [13], kiến trúc mô hình phát hiện đối tượng đã được mô tả thành ba phần: backbone, neck và head, để phản ánh sự phát triển và tinh chỉnh của kiến trúc. Việc chia thành ba phần giúp tách biệt các chức năng và phần tử của mô hình, làm cho nó dễ hiểu và dễ quản lý hơn. Điều này cũng giúp cải thiện khả năng mở rộng và tinh chỉnh mô hình cho các ứng dụng cụ thể.

Backbone là phần chính của mạng, chịu trách nhiệm trích xuất các đặc điểm hữu ích từ hình ảnh đầu vào. Thông thường, đó là một mạng nơ-ron tích chập (CNN) được huấn luyện trên một nhiệm vụ phân loại hình ảnh quy mô lớn, như ImageNet [14]. Backbone bắt các đặc điểm phân cấp ở các tỷ lệ khác nhau, với các đặc điểm ở mức thấp hơn (ví dụ: cạnh và cấu trúc) được trích xuất ở các lớp



Hình 2.11: Kiến trúc tổng quan của mô hình YOLO kể từ phiên bản v4

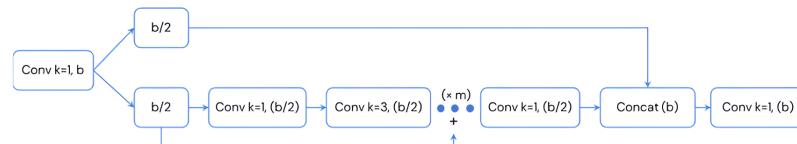
ban đầu và các đặc điểm ở mức cao hơn (ví dụ: các phần của đối tượng và thông tin ngữ nghĩa) được trích xuất ở các lớp sâu hơn

Neck là một phần trung gian kết nối giữa backbone và head. Nó tổng hợp và làm sạch các đặc điểm được trích xuất bởi backbone, thường tập trung vào việc cải thiện thông tin không gian và ngữ nghĩa qua các tỷ lệ khác nhau. Neck có thể bao gồm các lớp tích chập bổ sung, mạng phô cập tính năng (FPN), hoặc các cơ chế khác để cải thiện biểu diễn của các đặc điểm.

Head là thành phần cuối cùng của một bộ phát hiện đối tượng; nó chịu trách nhiệm đưa ra các dự đoán dựa trên các đặc điểm được cung cấp bởi backbone và neck. Thông thường, nó bao gồm một hoặc nhiều mạng con cụ thể cho từng nhiệm vụ như phân loại, định vị và gần đây là phân đoạn (segmentation) và ước lượng tư thế (pose estimation). Head xử lý các đặc điểm mà neck cung cấp, tạo ra các dự đoán cho mỗi đối tượng ứng viên.

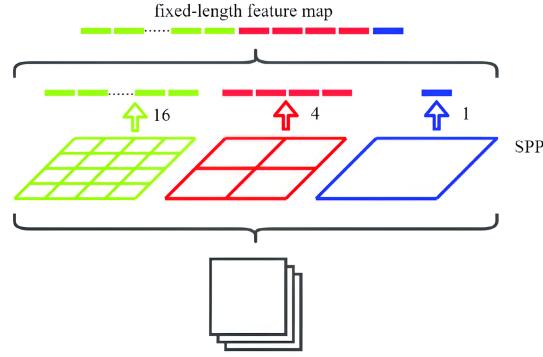
2.7.6 Tối ưu tốc độ và độ chính xác - YOLOv4

YOLOv4 [13] vẫn giữ nguyên triết lý cơ bản của YOLO - thời gian thực, mã nguồn mở và Darknet framework. YOLOv4 đã cố gắng tìm ra sự cân bằng tối ưu bằng cách thử nghiệm nhiều thay đổi và được phân loại thành hai loại: "bag-of-freebies" và "bag-of-specials". Bag-of-freebies là phương pháp chỉ thay đổi chiến lược huấn luyện (training strategy) và tăng chi phí huấn luyện (training cost) mà không làm tăng thời gian suy luận (inference time), phổ biến nhất là kỹ thuật tăng cường dữ liệu (data augmentation). Ngược lại, bag-of-specials là các phương pháp tăng một chút chi phí suy luận nhưng cải thiện đáng kể độ chính xác, phổ biến nhất là "*cross-stage partial connections*" và "*spatial pyramid pooling*".



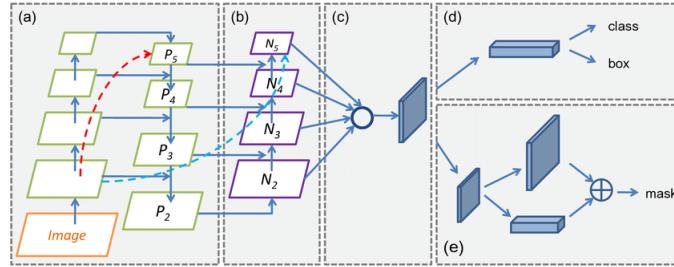
Hình 2.12: Liên kết CSP trong YOLOv4

Các thay đổi của YOLOv4 được tổng hợp như sau:



Hình 2.13: Phép gộp không gian kim tự tháp SPP trong YOLOv4

- Tác giả đã cải thiện kiến trúc bằng cách tích hợp với Bag-of-Specials(BoS), sử dụng Darknet-53 với Cross-Stage Partial Network (CSPNet) [15] và hàm kích hoạt Mish làm backbone. Dối với phần neck, họ sử dụng một phiên bản sửa đổi của Spatial Pyramid Pooling (SPP) từ YOLOv3-SPP và dự đoán đa tỉ lệ, kết hợp với một phiên bản sửa đổi của path aggregation network (PANet) [16] và mô-đun Spatial Attention (SAM). Về phần head sử dụng các anchor tương tự như YOLOv3, tạo ra mô hình được đặt tên là CSPDarknet53-PANet-SPP. CSP giảm tính toán trong khi vẫn giữ được độ chính xác, SPP nâng cao lĩnh vực nhận thức mà không ảnh hưởng đến tốc độ suy luận, và PANet được sửa đổi để nối các đặc trưng thay vì cộng chúng.



Hình 2.14: PAN (Path Aggregation Network) framework

- Tích hợp Bag-of-Freebies (BoF) để cải thiện phương pháp huấn luyện. Ngoài các kỹ thuật tăng cường thông thường như làm sáng, làm tương phản, tỉ lệ, cắt, lật và xoay ngẫu nhiên, các tác giả đã triển khai tăng cường mosaic kết hợp bốn hình ảnh thành một hình ảnh duy nhất cho phép phát hiện các đối tượng ngoài bối cảnh thông thường của chúng và cũng hạn chế phải sử dụng mini-batch có kích thước lớn cho chuẩn hóa cụm. Dối với điều chỉnh (regularization), họ đã sử dụng DropBlock và làm mờ nhãn lớp. Dối với bộ phát hiện (detector), họ đã thêm vào CIoU loss và Cross mini-batch normalization (CmBN) để thu thập thống kê từ toàn bộ cụm thay vì từ các mini-batch đơn lẻ.
- Để tìm ra các siêu tham số tối ưu được sử dụng trong quá trình huấn luyện, họ sử dụng thuật toán di truyền trong 10% đầu tiên của các chu kỳ và lập lịch giảm dần theo hàm cosine để điều chỉnh learning rate trong quá trình huấn luyện. Lập lịch bắt đầu giảm learning rate chậm rãi,

tiếp đó là giảm đột ngọt ở nửa chặng đường của quá trình huấn luyện và kết thúc bằng giảm nhẹ.

Backbone	Detector
Bag-of-Freebies	Bag-of-Freebies
Data augmentation	Data augmentation
- Mosaic	- Mosaic
- CutMix	- Self-Adversarial Training
Regularization	CIOU loss
- DropBlock	Cross mini-Batch Normalization (CmBN)
Class label smoothing	Eliminate grid sensitivity
	Multiple anchors for a single ground truth
	Cosine annealing scheduler
	Optimal hyper-parameteres
	Random training shapes
Bag-of-Specials	Bag-of-Specials
Mish activation	Mish activation
Cross-stage partial connections	Spatial pyramid pooling block
Multi-input weighted residual connections	Spatial attention module (SAM)
	Path aggregation network (PAN)
	Distance-IoU Non-Maximum Suppression

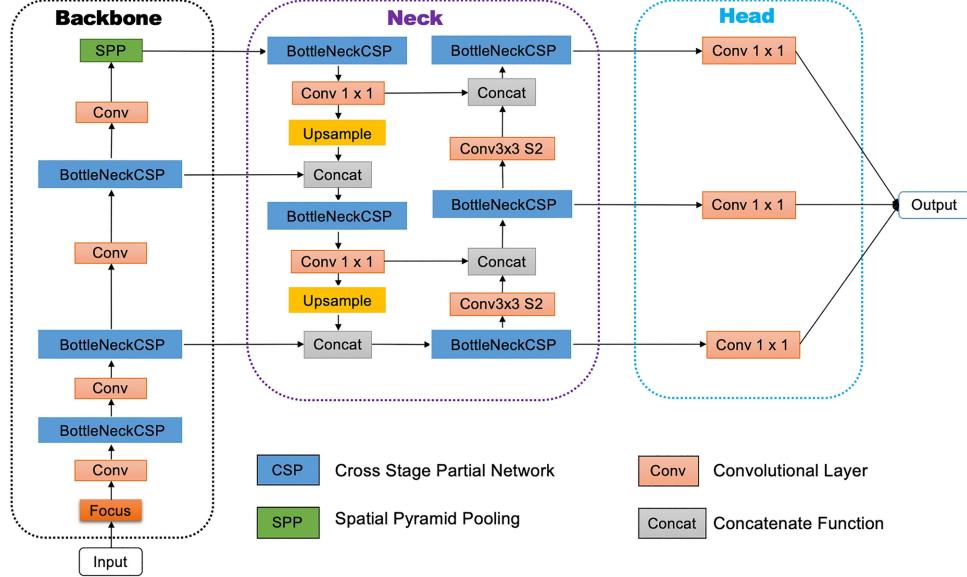
Hình 2.15: Các phương pháp cải thiện mô hình BoF và BoS

2.7.7 YOLOv5

YOLOv5 [17] ra mắt vài tháng sau YOLOv4 vào năm 2020 bởi Glen Jocher, người sáng lập và CEO của Ultralytics. Nó sử dụng nhiều cải tiến được mô tả trong phần YOLOv4 nhưng được phát triển trong Pytorch thay vì Darknet. YOLOv5 tích hợp một thuật toán của Ultralytics gọi là AutoAnchor. Công cụ tiền huấn luyện này kiểm tra và điều chỉnh các anchor nếu chúng không phù hợp với tập dữ liệu và cài đặt huấn luyện, như kích thước hình ảnh.

Backbone của YOLOv5 là một phiên bản được sửa đổi của CSPDarknet53, bắt đầu với một phần Stem, một lớp tích chập với bước nhảy và kích thước cửa sổ lớn để giảm bớt chi phí bộ nhớ và tính toán; tiếp theo là các lớp tích chập trích xuất các đặc trưng liên quan từ dữ liệu đầu vào là một hình ảnh. Lớp SPPF (spatial pyramid pooling fast) và các lớp tích chập tiếp theo xử lý các đặc trưng ở các tỷ lệ khác nhau, trong khi các lớp upsample tăng độ phân giải của các bản đồ đặc trưng. Lớp SPPF nhằm tăng tốc tính toán của mạng bằng cách gộp các đặc trưng ở các tỷ lệ khác nhau thành một bản đồ đặc trưng cố định. Mỗi lớp tích chập sau đó được tiếp tục bởi chuẩn hóa theo cụm (BN) và hàm kích hoạt SiLU. Phần neck sử dụng SPPF và một phiên bản được sửa đổi CSP-PAN, trong khi phần head giống với YOLOv3.

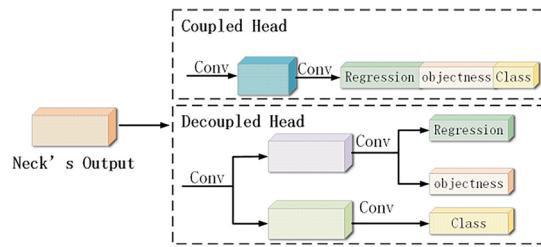
YOLOv5 sử dụng một số phương pháp tăng cường dữ liệu như Mosaic, sao chép vùng, MixUp, tăng cường HSV, lật ngang ngẫu nhiên, cũng như các phương pháp tăng cường bổ sung khác từ gói Albumentations. Nó cũng cải thiện độ nhạy của ô lưới để làm cho ô lưới ổn định hơn, giảm bớt hiện tượng biến mất gradient (Vanishing gradient).



Hình 2.16: Kiến trúc mô hình YOLOv5

2.7.8 Thiết kế thân thiện với phần cứng - YOLOv6

YOLOv6 [18] được công bố trên ArXiv vào tháng 9 năm 2022 bởi Bộ phận Trí tuệ Nhân tạo của Meituan Vision. Thiết kế mạng bao gồm một backbone với các khối RepVGG [19] hoặc CSPStackRep, một neck theo topology PAN, và một head phân tách (decouple head) với chiến lược kênh kết hợp. Ngoài ra, YOLOv6 còn giới thiệu các kỹ thuật lượng tử hóa cải tiến bằng cách sử dụng lượng tử hóa sau khi huấn luyện và truyền đạt theo chiều kênh (channel-wise distillation), dẫn đến các phát hiện đối tượng nhanh hơn và chính xác hơn. Tổng thể, YOLOv6 vượt trội hơn so với các mô hình tiên tiến trước đó về độ chính xác và tốc độ, chẳng hạn như YOLOv5, YOLOX và PP-YOLOE.

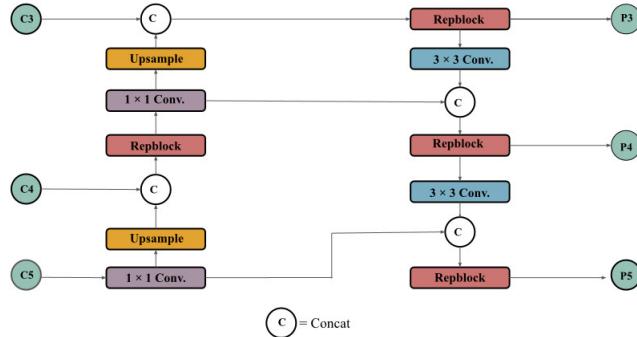


Hình 2.17: Coupled head và Decoupled head

Các điểm mới chính của mô hình này được tóm tắt dưới đây:

- Một backbone mới dựa trên RepVGG được gọi là EfficientRep sử dụng tối ưu hóa hơn so với các backbone YOLO trước đó có thể giúp giảm độ phức tạp tính toán và tăng tốc độ xử lý của mô hình. Đối với phần neck, họ sử dụng PAN được cải tiến với các khối RepBlocks hoặc CSPStackRep cho các mô hình lớn hơn. Dựa trên YOLOX, họ đã phát triển một head hiệu quả và độc lập. Đồng thời, YOLOv6 cũng là phiên bản đầu tiên của YOLO sử dụng mô hình không

có neo kết hợp với head phân tách.



Hình 2.18: EfficientRep

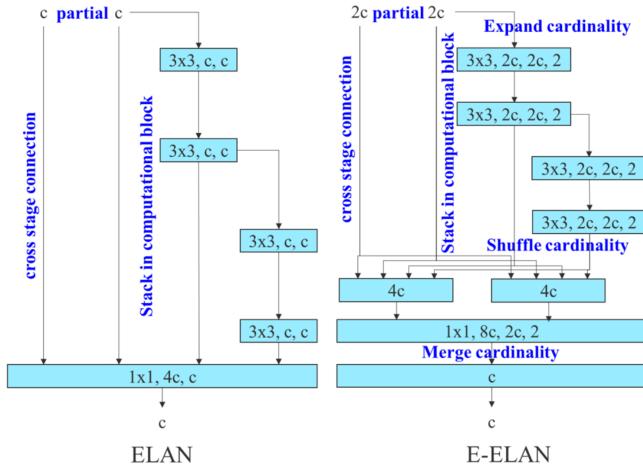
- Gán nhãn sử dụng phương pháp học điều chỉnh nhiệm vụ được giới thiệu trong TOOD.
- Các hàm mất mát phân loại và hồi quy mới. Họ đã sử dụng hàm mất mát phân loại VariFocal và SIoU/GIoU.
- Một chiến lược tự huấn luyện cho các nhiệm vụ hồi quy và phân loại.
- Cải thiện độ chính xác và tốc độ của việc phát hiện bằng cách sử dụng mô hình không có neo để nâng cao tốc độ phát hiện lên 51
- Lượng tử hóa cho phát hiện sử dụng RepOptimizer và tự truyền đạt theo chiều kênh (channel-wise distillation) đã giúp đạt được việc phát hiện nhanh hơn.

2.7.9 Trainable Bag-of-Freebies (BoF) - YOLOv7

YOLOv7 [20] được công bố trên ArXiv vào tháng 7 năm 2022. Vào thời điểm được công bố nó đã vượt qua mọi mô hình phát hiện đối tượng đã biết về tốc độ và độ chính xác. Giống như YOLOv4, nó đã được huấn luyện chỉ bằng tập dữ liệu MS COCO [21] mà không có các backbone được huấn luyện trước. YOLOv7 đề xuất một số thay đổi kiến trúc và một loạt các bag-of-freebies, giúp tăng độ chính xác và thời gian huấn luyện mà không ảnh hưởng đến tốc độ suy luận.

Các thay đổi kiến trúc của YOLOv7 bao gồm việc giới thiệu mạng ELAN [22] mở rộng (E-ELAN) [23] và chiến lược tăng cường cho các mô hình dựa trên ghép nối. E-ELAN được thiết kế để học và hội tụ hiệu quả hơn trong các mô hình sâu bằng cách kết hợp đặc trưng từ các nhóm khác nhau mà không phá hủy đường gradient ban đầu. Chiến lược tăng cường cho các mô hình dựa trên ghép nối giúp tạo ra các mô hình có kích thước khác nhau bằng cách điều chỉnh chiều sâu và chiều rộng của các khối với cùng một hệ số để duy trì cấu trúc tối ưu của mô hình. YOLOv7 được thiết kế nhằm mục tiêu cải thiện độ chính xác của việc phát hiện mà không tăng chi phí huấn luyện. Nó tập trung vào việc tăng cả tốc độ suy luận và độ chính xác của việc phát hiện, tạo ra một cải tiến đáng kể so với phiên bản trước đó.

Các phương pháp bag-of-freebies được sử dụng trong YOLOv7 bao gồm:



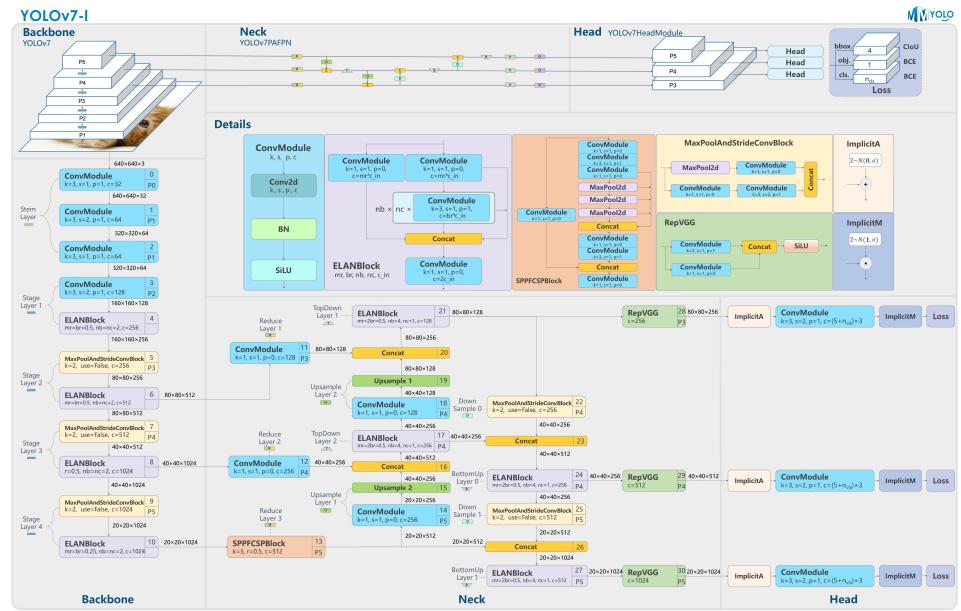
Hình 2.19: Kiến trúc của mạng ELAN và E-ELAN

- Điều chỉnh tham số của lớp tích chập: Giống với YOLOv6, kiến trúc của YOLOv7 cũng đều sử dụng một kỹ thuật gọi là tái tham số hóa tích chập (RepConv). Tuy nhiên, YOLOv7 có một cải tiến từ RepConv đó là RepConvN. RepConvN không bao gồm kết nối đồng nhất và được thiết kế để giảm thiểu ảnh hưởng tiêu cực lên hiệu suất của mô hình trong việc xử lý phần dư và sự nối liền.
- Batch normalization trong conv-bn-activation: kỹ thuật này tích hợp giá trị trung bình và phương sai của batch normalization vào trọng số và độ lệch của lớp tích chập. Điều này giúp giảm thiểu chi phí tính toán và tăng tốc độ quá trình suy luận.
- Implicit knowledge: Được lấy cảm hứng từ YOLOR.

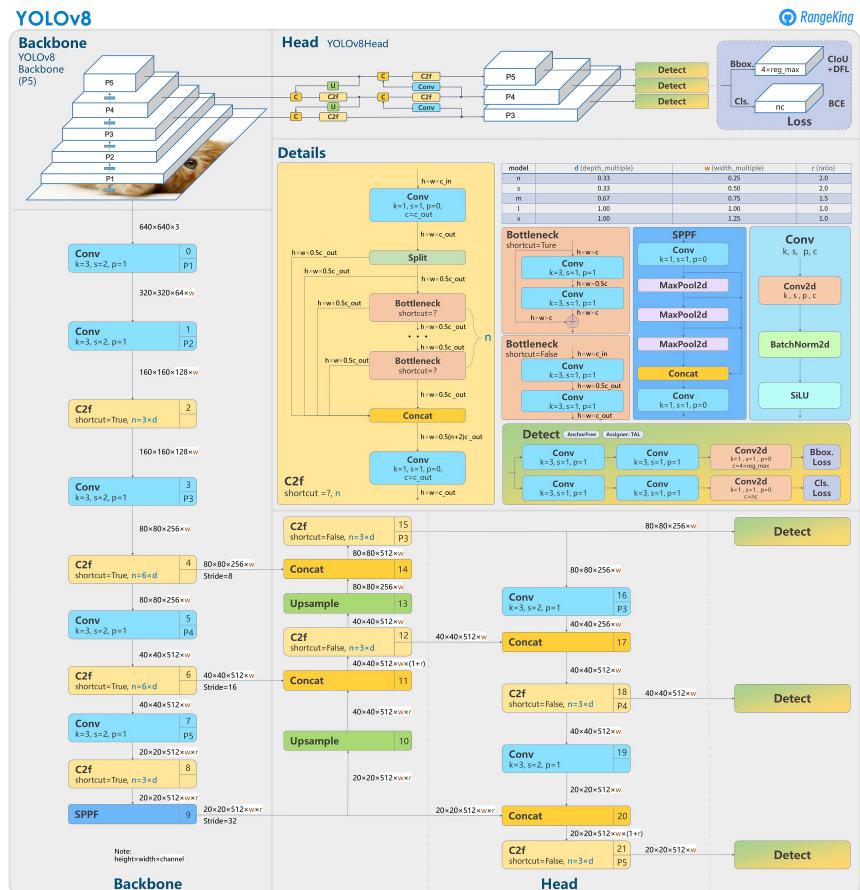
2.7.10 Hiệu suất và độ chính xác tối đa - YOLOv8

YOLOv8 [24] được phát hành vào tháng 1 năm 2023 bởi Ultralytics, công ty đã phát triển YOLOv5. YOLOv8 cung cấp năm phiên bản được điều chỉnh: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large) và YOLOv8x (extra large).

YOLOv8 sử dụng một backbone tương tự như YOLOv5 với một số thay đổi trên CSPLayer, hiện được gọi là mô-đun C2f. Mô-đun C2f (Cross-stage partial bottleneck với hai lớp tích chập) kết hợp các đặc trưng cao với thông tin ngữ cảnh để cải thiện độ chính xác của việc phát hiện. YOLOv8 sử dụng một mô hình không có anchor với một head phân tách để xử lý độ chắc chắn của đối tượng, phân loại và các nhiệm vụ hồi quy một cách độc lập. Thiết kế này cho phép mỗi nhánh tập trung vào nhiệm vụ của nó và cải thiện độ chính xác tổng thể của mô hình. Trong lớp đầu ra của YOLOv8, họ sử dụng hàm sigmoid làm hàm kích hoạt cho điểm chắc chắn của đối tượng (objectness score), đại diện cho xác suất rằng hộp giới hạn chứa một đối tượng. YOLOv8 sử dụng các hàm mất mát CIoU và DFL cho hàm mất mát của hộp giới hạn và hàm mất mát nhị phân cross-entropy cho hàm mất mát phân loại. Các hàm mất mát này đã cải thiện hiệu suất phát hiện đối tượng, đặc biệt là khi xử lý các đối tượng nhỏ.



Hình 2.20: Kiến trúc mô hình YOLOv7



Hình 2.21: Kiến trúc của YOLOv8

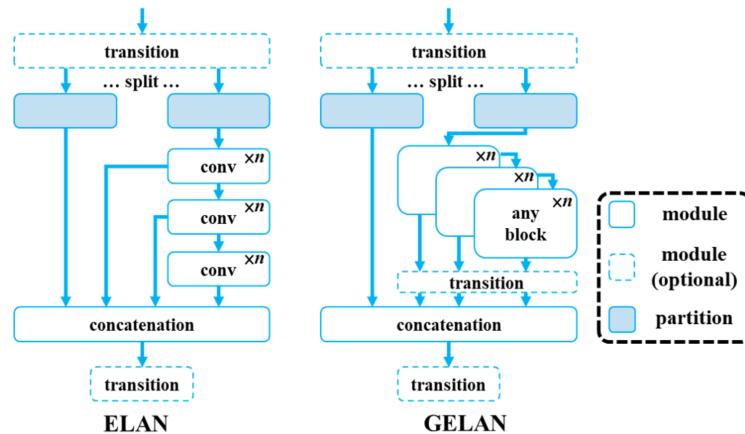
Sử dụng các cải tiến được thực hiện trong YOLOv7 như một nền tảng, YOLOv8 nhắm đến việc trở thành phiên bản tốt nhất từ trước đến nay trong việc phát hiện đối tượng trong hình ảnh hoặc video, hướng tới việc trở nên nhanh chóng và chính xác hơn bao giờ hết. YOLOv8 tuyệt vời ở nhiều điểm và một điểm cộng lớn là nó hoạt động tốt với tất cả các phiên bản YOLO cũ. Điều này làm cho việc thử nghiệm các phiên bản khác nhau trở nên đơn giản cho người dùng và dễ dàng xem xét phiên bản nào hoạt động tốt nhất, biến nó trở thành một lựa chọn hàng đầu cho những người muốn sử dụng các công cụ YOLO mới nhất nhưng cũng cung cấp sự lựa chọn cho những người muốn sử dụng các phiên bản cũ của mình.

2.7.11 Bước nhảy vọt đáng kể - YOLOv9

YOLOv9 [25] đánh dấu một bước tiến đáng kể trong việc phát hiện đối tượng thời gian thực, giới thiệu các kỹ thuật đột phá như Programmable Gradient Information (PGI) và Generalized Efficient Layer Aggregation Network (GELAN). Mô hình này thể hiện những cải tiến đáng kể về hiệu quả, độ chính xác và khả năng thích ứng, thiết lập các tiêu chuẩn mới trên bộ dữ liệu MS COCO [21]. Dự án YOLOv9, trong khi được phát triển bởi một nhóm mã nguồn mở riêng biệt, được xây dựng dựa trên cơ sở mã mạnh mẽ được cung cấp bởi Ultralytics YOLOv5, thể hiện tinh thần hợp tác của cộng đồng nghiên cứu AI.

YOLOv9 có hai điểm nhấn, cũng chính là hai thành phần cấu thành, bao gồm:

1. **GELAN (Generalized Efficient Layer Aggregation Network)**: Mô hình được nhóm tác giả đề xuất là sự kết hợp giữa hai kiến trúc mạng CSPNet và ELAN, đều được thiết kế dựa trên kỹ thuật Gradient Path Planning. Một điểm quan trọng được tác giả nhấn mạnh đến khả năng của GELAN là sự tối ưu về việc sử dụng trọng số đã giúp GELAN nhẹ hơn các mô hình khác, dẫn đến đạt được tốc độ inference vượt trội song vẫn giữ được độ chính xác cao. Theo đó, GELAN "tổng quát hóa" (Generalize) khả năng của ELAN, bằng cách cho phép thay thế việc sử dụng các layer convolutional với bất kỳ computational block nào khác (residual block, dense block...). Khả năng này giúp cho YOLOv9 trở nên linh hoạt và đa dạng hơn, có khả năng thích nghi với một loạt các ứng dụng khác nhau.



Hình 2.22: Kiến trúc khối GELAN và ELAN

2. **Programmable Gradient Information (PGI)**, là một kỹ thuật độc đáo được đề xuất để

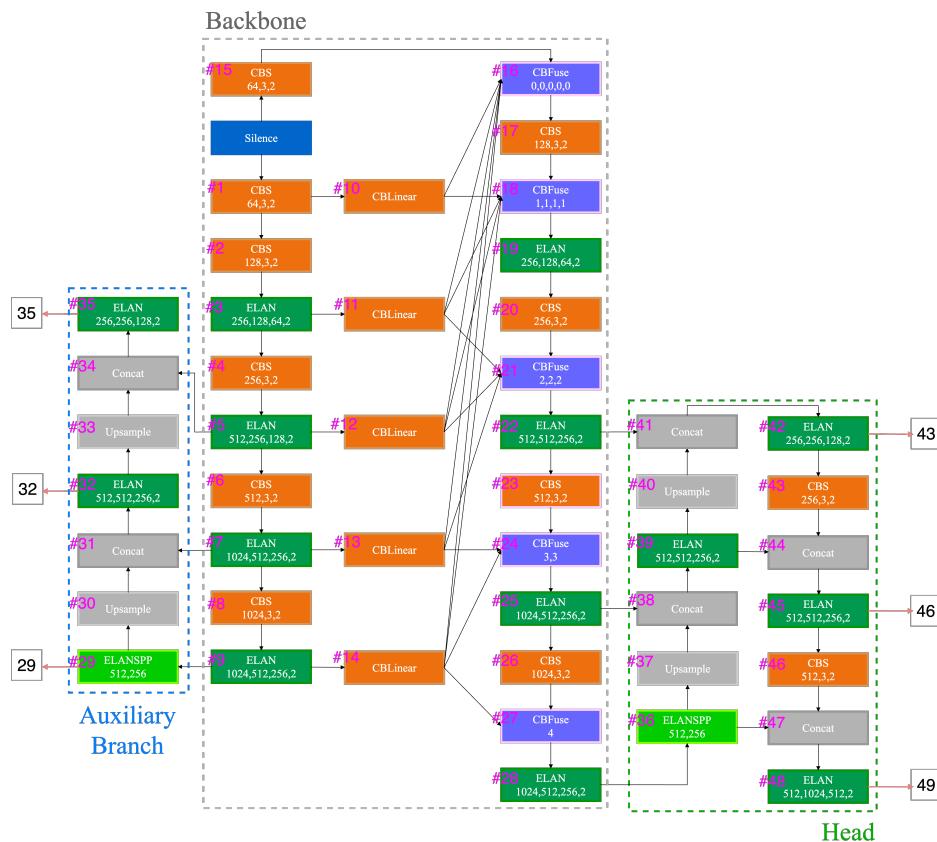
giải quyết vấn đề mất mát thông tin trong quá trình trích xuất đặc trưng và biến đổi không gian trong mạng neural network. Vấn đề này thường được gọi là Information Bottleneck, và PGI đặt ra hai nguyên tắc chính để giải quyết nó.

Nguyên tắc đầu tiên là Information Bottleneck, mục tiêu là giữ lại những thông tin quan trọng từ dữ liệu đầu vào, loại bỏ thông tin không cần thiết hoặc không quan trọng. Thứ hai là sử dụng Reversible Functions, có khả năng tái tạo lại dữ liệu đầu vào từ dữ liệu đầu ra mà không gây mất mát thông tin. Kết hợp hai nguyên tắc này, PGI có thể tạo ra các gradients chất lượng hơn trong quá trình training, đảm bảo rằng các đặc trưng quan trọng được giữ lại và không bị biến đổi quá mức.

PGI đã được chứng minh là hiệu quả hơn so với một số giải pháp hiện có như Reversible Architecture, Masked Modeling, và Deep Supervision. Điều này cho thấy tiềm năng của PGI trong việc cải thiện hiệu suất và độ chính xác của mạng neural network trong việc học và trích xuất thông tin từ dữ liệu.

Ngoài ra, YOLOv9 có 2 phiên bản được thiết kế dành cho các tác vụ yêu cầu nhu cầu phần cứng khác nhau, do đó các mô hình này cũng có độ phức tạp và khối lượng tham số khác nhau. Cụ thể YOLOv9 bao gồm YOLOv9-e (extra large) và YOLOv9-c (compact).

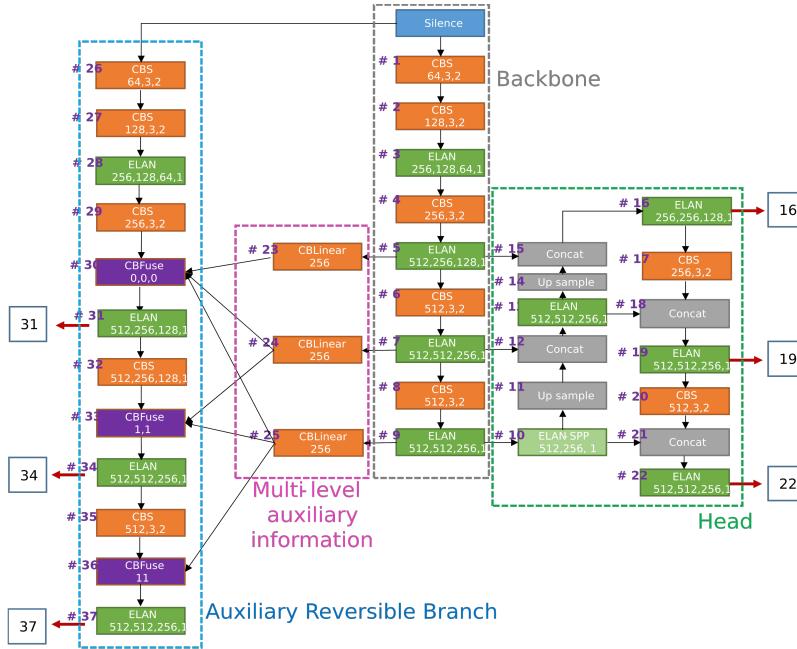
Trong luận văn này, chúng tôi lựa chọn mô hình YOLOv9-e làm mô hình cơ sở vì mô hình này có hiệu suất có thể cạnh tranh được so với các mô hình được thử nghiệm trong [1].



Hình 2.23: Kiến trúc mô hình YOLOv9-e

YOLOv9-e sử dụng Reversible Branch trong nhánh chính và loại bỏ thông tin đa tầng ở nhánh phụ trợ và số lượng các khối khôi có trong backbone và các khối lặp lại trong cái mạng cũng được tăng lên nhằm đáp ứng được các tác vụ phức tạp và đòi hỏi độ chính xác cao.

Trong khi đó, YOLOv9-c sử dụng Reversible Branch ở Auxiliary Branch, tăng số lượng các khối trong Auxiliary Branch nên lượng tham số và thời gian suy luận giảm đáng kể so với YOLOv9-e. Tuy nhiên, hiệu suất mô hình tương đối thấp hơn YOLOv9-e khi sử dụng trong các tác vụ phức tạp.



Hình 2.24: Kiến trúc mô hình YOLOv9-c

2.8 Các thuật toán theo dõi vật thể

2.8.1 Tổng quan về các thuật toán theo dõi vật thể

Khi triển khai các mô hình phát hiện vật thể ra ứng dụng thực tế, ta thường thấy độ bất ổn định qua các khung hình (frame) do nhiễu, ánh sáng thay đổi hoặc góc nhìn khác nhau. Thuật toán theo dõi (tracking algorithm) giúp duy trì vị trí của các vật thể được phát hiện giữa các khung hình, giảm bớt các sai sót hoặc mất mát ngắn hạn trong việc phát hiện từ đó tăng cường độ chính xác và ổn định cho mô hình phát hiện vật thể đang sử dụng.

Ngoài ra việc phát hiện vật thể ở mỗi khung hình đòi hỏi tính toán nhiều, làm tăng thời gian xử lý và yêu cầu tài nguyên. Bằng cách sử dụng thuật toán theo dõi, hệ thống có thể giảm tần suất thực hiện phát hiện, chỉ phát hiện lại khi cần thiết (ví dụ khi vật thể biến mất hoặc xuất hiện mới), từ đó giảm tải tính toán. Thuật toán theo dõi cũng cho phép gán ID duy nhất cho mỗi vật thể và theo dõi chúng qua nhiều khung hình. Điều này rất quan trọng trong các ứng dụng như giám sát an ninh,

theo dõi hành vi, hay đếm số lượng người/vật thể, nơi mà cần biết chính xác vị trí và hành động của từng đối tượng cụ thể qua thời gian.

Dòng thời, trong bối cảnh giám sát giao thông khi sử dụng camera fisheye, việc các phương tiện với kích thước lớn và vị trí thuận lợi hơn sẽ tạm thời che khuất các vật thể khác. Do đó, tận dụng việc dự đoán vị trí tiếp theo của vật thể sau khi bị che khuất, thuật toán theo dõi có thể đảm bảo rằng việc theo dõi các khung bị gián đoạn khi vật thể xuất hiện lại.

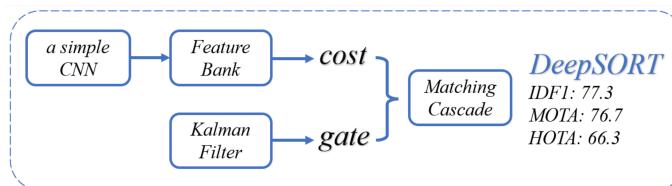
Trong luận văn này, chúng tôi quyết định sử dụng 2 thuật toán theo dõi DeepSORT và StrongSORT để kết hợp với các mô hình đề xuất nhằm tạo ra ứng dụng theo dõi các phương tiện giao thông trên camera fisheye với độ ổn định và chính xác cao.

2.8.2 DeepSORT

DeepSORT [26] (Deep Simple Online and Realtime Tracking) là một thuật toán theo dõi trực tuyến và thời gian thực dựa trên Deep Learning, được phát triển để cải thiện các thuật toán theo dõi truyền thống như SORT (Simple Online and Realtime Tracking). Trong việc sử dụng thuật toán theo dõi cho các ứng dụng giám sát giao thông, DeepSORT cho phép theo dõi chính xác và duy trì liên tục việc phát hiện các phương tiện và người đi bộ, hỗ trợ các ứng dụng như quản lý lưu lượng, giám sát an toàn giao thông, và phát hiện vi phạm.

Quy trình hoạt động của DeepSORT:

- Phát hiện đối tượng: Sử dụng mô hình phát hiện để nhận diện các đối tượng trong mỗi khung hình video.
- Trích xuất đặc trưng xuất hiện: Sử dụng mạng ReID để trích xuất các đặc trưng xuất hiện của từng đối tượng được phát hiện.
- Dự đoán vị trí: Sử dụng Kalman Filter để dự đoán vị trí của các đối tượng trong khung hình hiện tại.
- Liên kết đối tượng: Sử dụng Hungarian Algorithm để liên kết các phát hiện mới với các đường theo dõi hiện tại dựa trên ma trận chi phí.
- Cập nhật: Cập nhật trạng thái của các đường theo dõi hiện tại với các phát hiện mới.
- Khởi tạo và loại bỏ: Khởi tạo các đường theo dõi mới cho các phát hiện chưa được liên kết và loại bỏ các đường theo dõi không còn chính xác hoặc không còn tồn tại.



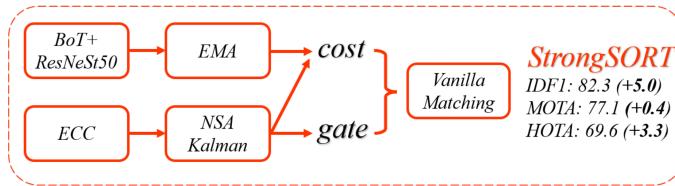
Hình 2.25: Các hoạt động của thuật toán DeepSORT

2.8.3 StrongSORT

StrongSORT [27] là một phiên bản nâng cao của DeepSORT, được thiết kế để cải thiện độ chính xác và độ ổn định trong việc theo dõi đối tượng. StrongSORT đặc biệt hiệu quả trong các tình huống phức tạp và động đúc như giám sát giao thông, nơi yêu cầu theo dõi chính xác các phương tiện và người đi bộ.

Quy trình hoạt động của StrongSORT:

- Phát hiện đối tượng: Sử dụng mô hình phát hiện tiên tiến để nhận diện các đối tượng trong mỗi khung hình video.
- Trích xuất đặc trưng xuất hiện: Sử dụng mạng ReID cải tiến để trích xuất các đặc trưng xuất hiện mạnh mẽ của từng đối tượng được phát hiện.
- Dự đoán vị trí và vận tốc: Sử dụng Kalman Filter cải tiến để dự đoán vị trí và vận tốc của các đối tượng trong khung hình hiện tại.
- Liên kết đối tượng: Sử dụng Hungarian Algorithm để liên kết các phát hiện mới với các đường theo dõi hiện tại dựa trên ma trận chi phí kết hợp.
- Cập nhật và quản lý đối tượng: Cập nhật trạng thái của các đường theo dõi hiện tại với các phát hiện mới và quản lý việc khởi tạo và loại bỏ các đường theo dõi.
- Học tập đa nguồn: Kết hợp thông tin từ nhiều nguồn để cải thiện độ chính xác và độ tin cậy của việc theo dõi.



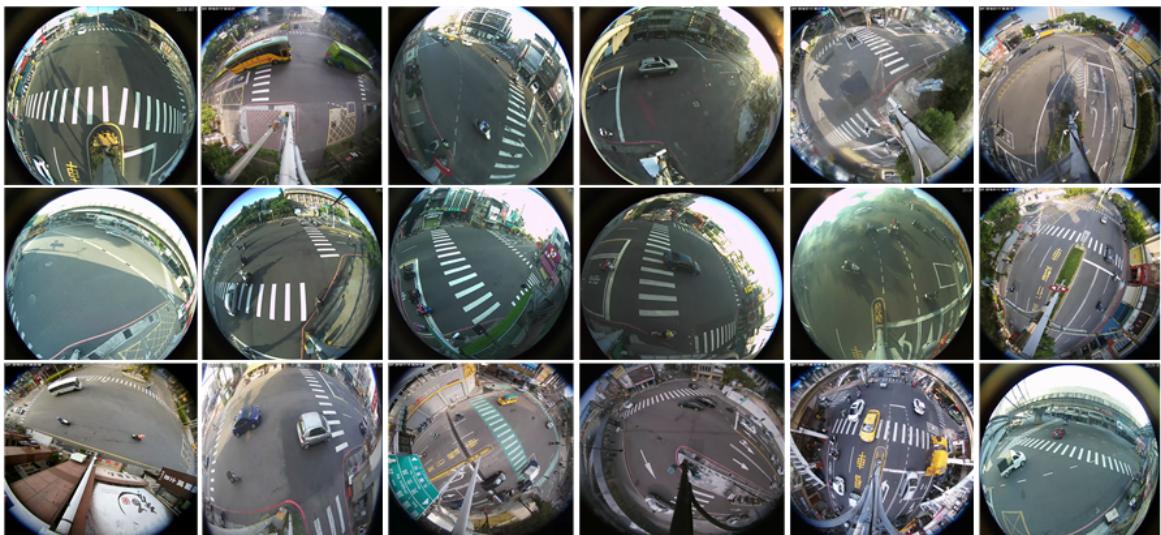
Hình 2.26: Các hoạt động của thuật toán StrongSORT

Chương 3

Phương pháp đề xuất

Chương này sẽ trình bày về cách tiếp cận dữ liệu, phân tích về tập dữ liệu đã chọn, kiến trúc mô hình cơ sở được sử dụng trong bài toán phát hiện đối tượng, phân tích các nghiên cứu liên quan, kiến trúc mô hình và hàm mất mát do chính chúng tôi đề xuất cải tiến dành cho hình ảnh thu từ fisheye camera và hướng triển khai mô hình sau huấn luyện.

3.1 Tập dữ liệu FISHEYE 8K

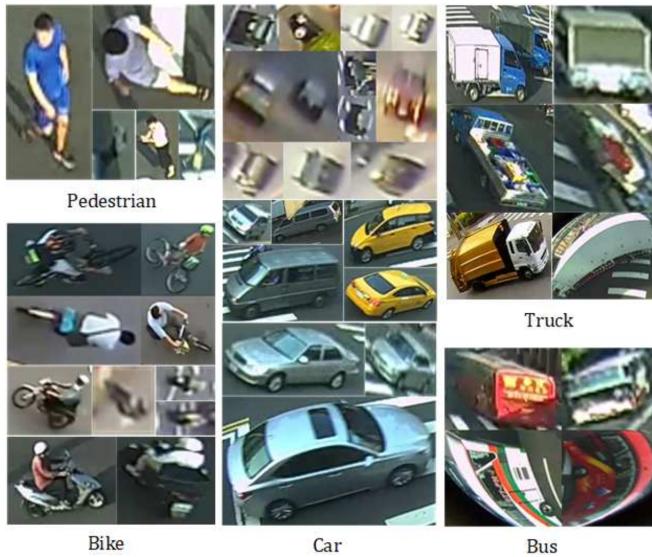


Hình 3.1: Tập dữ liệu FISHEYE8K

3.1.1 Tổng quan về tập dữ liệu

Bộ dữ liệu được sử dụng trong luận văn mang tên FishEye8K [1] được sử dụng cho việc huấn luyện và đánh giá nhiệm vụ phát hiện đối tượng trên đường. Bộ dữ liệu FishEye8K được đề xuất là bộ dữ liệu đầu tiên trong số các bộ dữ liệu mở, được thiết kế và xây dựng đặc biệt để phát triển và đánh giá khả năng phát hiện vật thể trên đường bằng camera fisheye. Bộ dữ liệu FishEye8K bao gồm 8.000 khung hình ảnh với 157.000 nhãn bounding box của 5 lớp đối tượng, bao gồm Người đi bộ, Xe đạp, Xe hơi, Xe tải và Xe buýt (Pedestrian, Bike, Car, Bus, and Truck). Dữ liệu này được thu thập từ 35 camera quan sát giao thông tại thành phố Hsinchu, Đài Loan với độ phân giải là 1080x1080 và 1280x1280. Bộ dữ liệu này bao gồm nhiều điều kiện giao thông khác nhau như đường cao tốc đô thị,

giao lộ, các điều kiện ánh sáng khác nhau và góc quay của các lớp đối tượng trên đường trong nhiều môi trường khác nhau.



Hình 3.2: Hình ảnh ghi lại các lớp trong bộ dữ liệu Fisheye8K

3.1.2 Quá trình Thu thập Video và Chuẩn bị Dữ liệu

Quá trình gán nhãn và xác thực dữ liệu đã mất nhiều thời gian và công sức, do hình ảnh siêu rộng và mắt kính của camera fisheye với độ biến dạng lớn được sử dụng trong môi trường có số lượng người tham gia giao thông đông đảo, đặc biệt là những người đi xe scooter. Các khung hình từ một camera cụ thể được gán vào tập huấn luyện và tập kiểm thử duy trì tỷ lệ 70:30 cho cả số lượng hình ảnh và hộp giới hạn trên từng lớp.

Quá trình thu thập liên quan đến việc ghi lại cảnh quan từ 20 máy ảnh giám sát tại 60 hình mỗi giây (FPS), kết quả trong hai bộ video: Bộ 1 với 30 video kéo dài từ 50-60 phút mỗi video, và Bộ 2 với 5 video kéo dài khoảng 20 phút mỗi video. Để tạo ra bộ dữ liệu, 18 video được lựa chọn, có đặc điểm là các điều kiện đường khác nhau, góc quay của máy ảnh và các tình huống ánh sáng khác nhau. Các video này đã được cắt thành các đoạn ngắn và khung hình đã được chọn mẫu để cho ra hơn 10.000 hình ảnh, được thay đổi kích thước để đảm bảo tính nhất quán.

3.1.3 Điều kiện môi trường

Bộ dữ liệu cũng bao gồm hình ảnh từ các loại giao lộ khác nhau, chẳng hạn như giao lộ T, giao lộ Y, giao lộ chéo, phần giữa đường, giao lộ cho người đi bộ và các giao lộ không thông thường. Các video được ghi lại dưới nhiều điều kiện ánh sáng khác nhau, bao gồm buổi sáng, buổi chiều, buổi tối và đêm, và các mức độ tắc đường đa dạng từ giao thông thông thoáng cho đến đông đúc và tắc nghẽn. Bên cạnh đó góc đặt camera và độ phân giải cũng rất đa dạng, bao gồm ảnh chụp từ bên cạnh và phía trước cũng như chất lượng video khác nhau.

3.1.4 Chú thích và Xác thực

Quá trình chú thích bao gồm việc đánh dấu thủ công tỉ mỉ hơn 10.000 khung hình bởi hai nhà nghiên cứu sử dụng chương trình chú thích DarkLabel. Để đảm bảo độ chính xác, các phương pháp xác thực bán tự động được áp dụng, bao gồm xác minh thủ công và phân tích các dương tính sai được tạo ra bởi các mô hình phát hiện. Phương pháp tiếp cận nghiêm túc này nhằm mục đích giảm thiểu lỗi của con người và đảm bảo tính đáng tin cậy của bộ dữ liệu.

3.1.5 Benchmark của tập dữ liệu

Nhóm tác giả cũng đã thực hiện đánh giá khả năng của các mô hình phát hiện đối tượng hiện nay khi làm việc với hình ảnh có độ méo cao từ camera fisheye. Điều này đặc biệt quan trọng trong các ứng dụng thực tế như giám sát giao thông và an ninh, nơi mà việc phát hiện chính xác các đối tượng trên đường phố là cần thiết. Các mô hình được đánh giá bao gồm YOLOv5, YOLOR, YOLO7, và YOLOv8. Mỗi mô hình được huấn luyện và kiểm tra trên cùng một tập dữ liệu để đảm bảo tính công bằng và so sánh chính xác. Kết quả cho thấy YOLOv8 và YOLOR có hiệu suất tốt nhất, với YOLOv8 đạt được độ chính xác cao nhất trên kích thước đầu vào 640×640 và YOLOR trên 1280×1280 . Các mô hình này đã được tối ưu hóa để xử lý đặc tính méo hình của camera fisheye và vẫn duy trì độ chính xác cao trong việc phát hiện đối tượng.

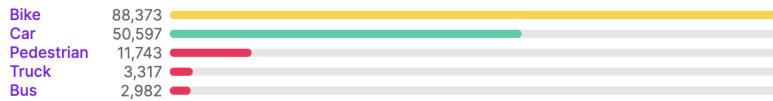
Model	Version	Input size	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	F1-score	Inference [ms]
YOLOv5	YOLOv5l6	1280	0.7929	0.4076	0.6139	0.4098	0.535	22.7
	YOLOv5x6	1280	0.8224	0.4313	0.6387	0.4268	0.5588	43.9
YOLOR	YOLOR-W6	1280	0.7871	0.4718	0.6466	0.4442	0.5899	16.4
	YOLOR-P6	1280	0.8019	0.4937	0.6632	0.4406	0.6111	13.4
YOLOv7	YOLOv7-D6	1280	0.7803	0.4111	0.3977	0.2633	0.5197	26.4
	YOLOv7-E6E	1280	0.8005	0.5252	0.5081	0.3265	0.6294	29.8
YOLOv7	YOLOv7	640	0.7917	0.4373	0.4235	0.2473	0.5453	4.3
	YOLOv7-X	640	0.7402	0.4888	0.4674	0.2919	0.5794	6.7
YOLOv8	YOLOv8l	640	0.7835	0.3877	0.612	0.4012	0.5187	8.5
	YOLOv8x	640	0.8418	0.3665	0.6146	0.4029	0.5107	13.4

Bảng 3.1: Kết quả của các mô hình phát hiện đối tượng YOLO được huấn luyện trên tập dữ liệu FishEye8K với hai kích thước đầu vào là 1280×1280 và 640×640 .

3.2 Phân tích tập dữ liệu FISHEYE 8K

3.2.1 Phân tích tổng quan

Trong tập dữ liệu này, các lớp chiếm ưu thế là Xe đạp (88.373) và Ô tô (50.597), có thể là do vị trí bán nhiệt đới của quốc gia nơi quay video. Mặt khác, các lớp Xe tải (3.317) và Xe buýt (2.982) có số lượng đối tượng thấp nhất, khiến tập dữ liệu mất cân bằng cao. Do đó cần phải sử dụng các phương pháp tối ưu khi huấn luyện và các mô hình có khả năng tổng quát hoá cao để xử lý tình trạng mất cân bằng này.

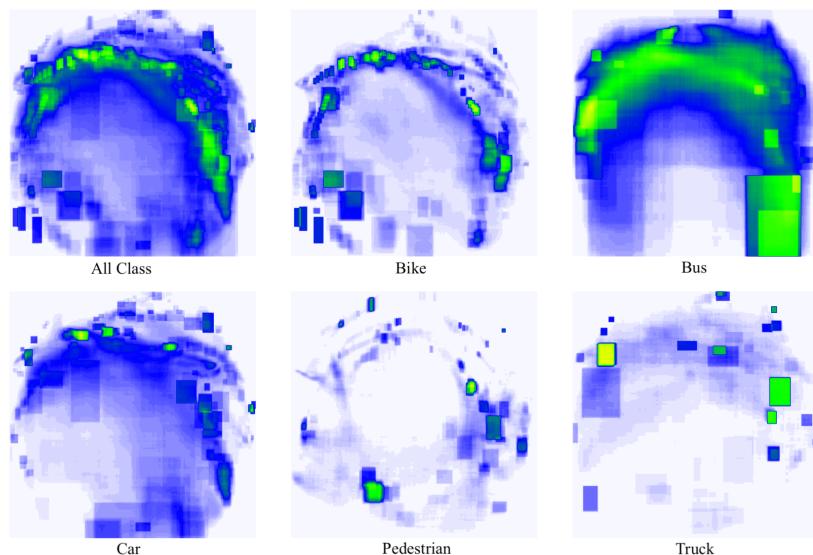


Hình 3.3: Sự phân bổ thực thể của các lớp trong tập dữ liệu Fisheye8K

Mặc dù tập dữ liệu chứa hình ảnh của 5 lớp phương tiện tham gia giao thông chính được chụp từ các góc khác nhau và trong các điều kiện ánh sáng khác nhau, nhưng nó thiếu sự đa dạng về điều kiện thời tiết, chẳng hạn như sương mù, mưa, tuyết và bão.

3.2.2 Sự phân bố của các thực thể

Thực hiện phân tích sự phân bố vị trí của các vật thể có trong tập dữ liệu bằng bản phương pháp đồ tập trung phân bố, thu được kết quả là thực thể của các lớp chủ yếu phân bố ở rìa ngoài ống kính, nơi mà các vật thể bị bẻ cong và biến thu phóng bất thường do đặc trưng của camera fisheye. Từ kết



Hình 3.4: Sự phân bố về vị trí của các lớp trên toàn bộ tập dữ liệu

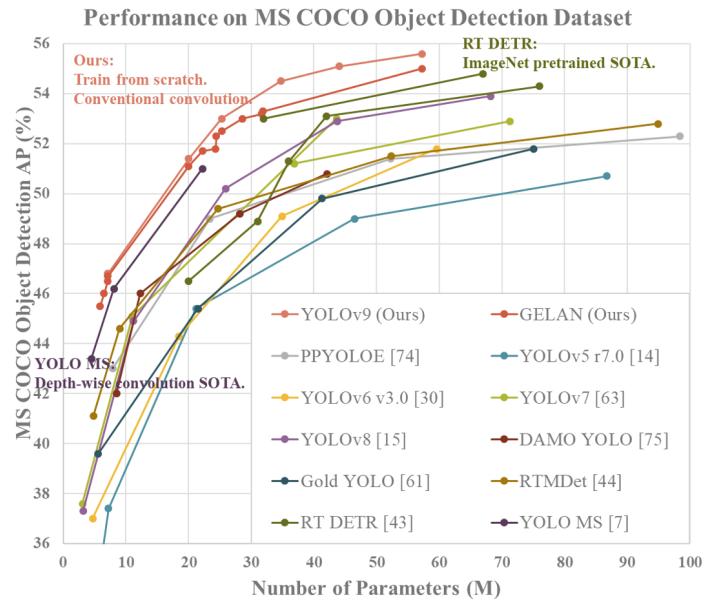
quả của phân tích này, chúng tôi nhận thấy là cần phải có các điều chỉnh về kiến trúc mô hình, các phương pháp tối ưu và tận dụng các phép biến đổi để thích ứng được sự bất thường của các vật thể về mặt hình ảnh.

3.3 Lựa chọn mô hình cơ sở

Hiện nay trong lĩnh vực thị giác máy tính nói chung và ứng dụng thị giác máy tính vào đời sống hằng ngày nói riêng, YOLO là phương pháp kinh điển để giải quyết các bài toán phát hiện vật thể trong thời gian thực, theo sau đó là R-CNN và các thuật toán dựa trên đặc trưng được tạo thủ công. Phần lớn các phiên bản trước của YOLO đã được chứng thực và sử dụng, ứng dụng trong các doanh

nghiệp cũng như trong các phương tiện sử dụng hằng ngày (Camera an ninh, Camera giám sát, ...). Nhận thấy điều đó, chúng tôi đề xuất sử dụng phiên bản mạnh mẽ nhất của YOLO ở hiện tại (Yolov9) để thực hiện việc huấn luyện, thí nghiệm và đề xuất các cải tiến để nâng cao hiệu suất mô hình, thích ứng với các đặc trưng của hình ảnh lấy từ camera fisheye.

YOLOv9 có hiệu suất vượt trội nhờ việc sử dụng GELAN và PGI trong các nhiệm vụ phát hiện đối tượng dựa trên bộ dữ liệu MS COCO [21] đã cho thấy kết quả vượt trội so với các phương pháp thay thế. Dáng chú ý, nó vượt trội về độ chính xác và tối ưu hóa tham số, vượt qua cả các mô hình được đào tạo trước trên các bộ dữ liệu mở rộng.



Hình 3.5: Hiệu suất vượt trội của YOLOv9 so với các mô hình khác trên tập COCO

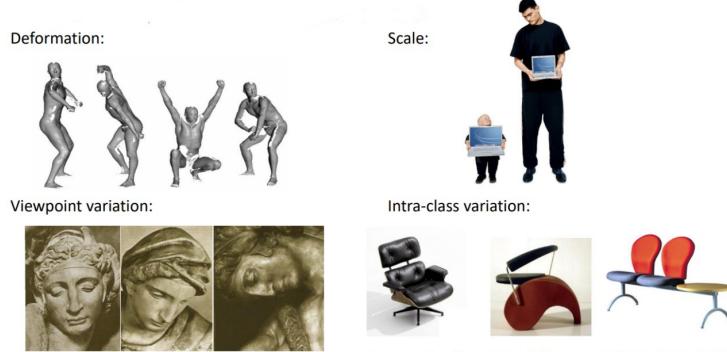
Trong luận văn này, mô hình YOLOv9-e (SOTA) được sử dụng để làm mô hình cơ sở cho các phương pháp đề xuất cải tiến của chúng tôi cũng như tích hợp thêm các tinh chỉnh từ các nghiên cứu trước đó để phù hợp với dữ liệu được thu từ camera fisheye.

3.4 Phân tích các nghiên cứu liên quan

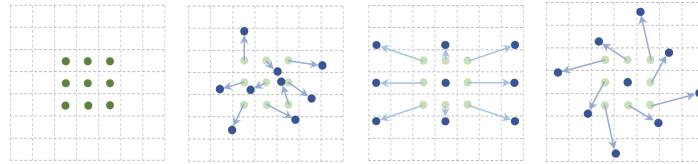
3.4.1 Phép tích chập biến dạng (Deformable Convolution)

Một trong những thách thức quan trọng trong việc nhận dạng hình ảnh là cách điều chỉnh các biến thể hình học hoặc mô hình hóa các biến đổi về tỷ lệ, tư thế, điểm nhìn và biến dạng của các đối tượng. Các biến thể của phép tích chập truyền thống được ra đời nhằm phần nào giải quyết được các vấn đề này. Trong đó phép tích chập biến dạng (deformable convolution) được giới thiệu dưới đây được xem là nổi bật nhất trong thời điểm hiện tại.

Phép tích chập biến dạng (Deformable Convolution) tương tự như phép tích chập thông thường nhưng vùng tiếp nhận (receptive fields) của nó bị biến dạng do các độ lệch không gian bổ sung được sử dụng trong quá trình lấy mẫu đầu vào. Phép tích chập biến dạng xuất phát ý tưởng cộng thêm



Hình 3.6: Các biến thể về mặt hình học của vật thể



Hình 3.7: Vị trí lấy mẫu của kernel trong phép tích chập biến dạng

vào các vị trí lấy mẫu trong các lớp tích chập truyền thống bằng một mảng độ lệch (offsets) 2 chiều. Điều này cho phép lớp tích chập lấy mẫu tại những vị trí đa dạng hơn. Đặc biệt các giá trị độ lệch (offsets) này điều được học từ các bản đồ đặc trưng trước đó thông qua các lớp tích chập nhờ vậy các giá trị độ lệch (offsets) linh hoạt thích ứng với dữ liệu đầu vào.

Phép tích chập 2D của deformable convolution bao gồm 2 bước: 1) lấy mẫu bằng lưới thông thường \mathcal{R} thông qua bản đồ đặc trưng đầu vào \mathbf{x} ; 2) Lấy tổng các giá trị lấy mẫu có trọng số bằng \mathbf{w} . Lưới \mathcal{R} xác định kích thước và độ giãn của vùng tiếp nhận. Ví dụ,

$$\mathcal{R} = \{(-1, -1), (-1, 0), \dots, (0, 1), (1, 1)\}$$

xác định một 3×3 kernel với độ giãn 1. Với mỗi vị trí \mathbf{p}_0 trên đầu ra của mỗi bản đồ đặc trưng \mathbf{y} , ta có

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n), \quad (3.1)$$

trong đó \mathbf{p}_n liệt kê các vị trí trong \mathcal{R} . Trong deformable convolution, lưới chính thức \mathcal{R} được tăng cường với độ lệch $\{\Delta\mathbf{p}_n | n = 1, \dots, N\}$, trong đó $N = |\mathcal{R}|$. Phương trình (3.1) trở thành

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n). \quad (3.2)$$

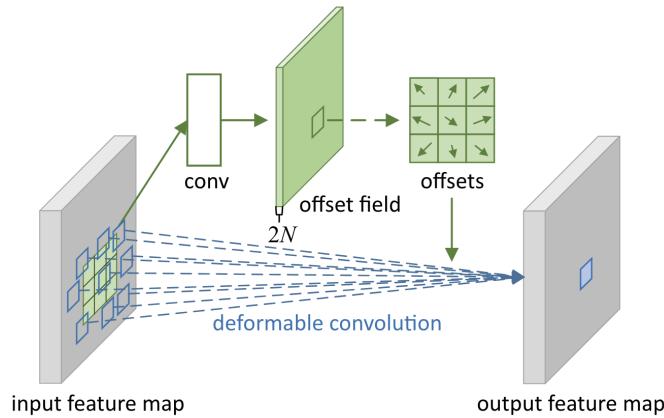
liệt kê tất cả các vị trí không gian tích hợp trong bản đồ đặc trưng. Vậy giờ, việc lấy mẫu sẽ ở các vị trí bất thường và vị trí lệch $\mathbf{p}_n + \Delta\mathbf{p}_n$. Với độ lệch $\Delta\mathbf{p}_n$ thường là các phân số, phương trình (3.2) được thực hiện thông qua nội suy song tuyến tính (bilinear interpolation) như sau

$$\mathbf{x}(\mathbf{p}) = \sum_{\mathbf{q}} G(\mathbf{q}, \mathbf{p}) \cdot \mathbf{x}(\mathbf{q}), \quad (3.3)$$

trong đó \mathbf{p} biểu thị một vị trí (phân số) tùy ý ($\mathbf{p} = \mathbf{p}_0 + \mathbf{p}_n + \Delta \mathbf{p}_n$ cho phương trình (3.2)), \mathbf{q} liệt kê tất cả các vị trí không gian tích hợp trong bản đồ đặc trưng \mathbf{x} , và $G(\cdot, \cdot)$ là kernel nội suy song tuyến tính (bilinear interpolation). Chú ý rằng G có hai chiều. Nó được tách thành hai kernel một chiều như sau

$$G(\mathbf{q}, \mathbf{p}) = g(q_x, p_x) \cdot g(q_y, p_y), \quad (3.4)$$

trong đó $g(a, b) = \max(0, 1 - |a - b|)$. Phương trình (3.3) được tối ưu tốc độ tính toán vì $G(\mathbf{q}, \mathbf{p})$ chỉ khác 0 (non-zero) với duy nhất một vài \mathbf{q} .



Hình 3.8: Cách hoạt động của phép tích chập biến dạng

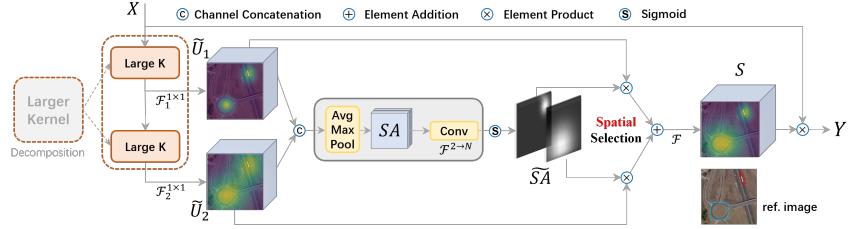
3.4.2 Large Selective Kernel Network

LSKNet [28] được thiết kế để phát hiện đối tượng trong hình ảnh từ cảm biến từ xa, như hình ảnh không gian được chụp từ máy bay hoặc vệ tinh. Đặc điểm nổi bật của LSKNet là khả năng điều chỉnh độ nhạy cảm không gian của nó để mô hình tốt hơn ngữ cảnh biến đổi của các đối tượng khác nhau trong các trường hợp cảm biến từ xa. LSKNet sử dụng các cơ chế lõi lớn và chọn lọc để đạt được hiệu suất cao trên các bộ kiểm tra tiêu chuẩn trong lĩnh vực phát hiện đối tượng từ cảm biến từ xa. Kiến trúc tổng thể của LSKNet được xây dựng dựa trên các cấu trúc phổ biến gần đây như Visual Attention Network, Pyramid Vision Transformer, Metaformer,...

Mỗi khối của LSKNet bao gồm hai residual sub-blocks: Large Kernel Selection (LK Selection) và khối con Feed-forward Network (FFN). Mô-đun (LSK) được tích hợp trong khối LK Selection. Nó gồm Large Kernel Convolutions và một cơ chế Spatial Kernel Selection.

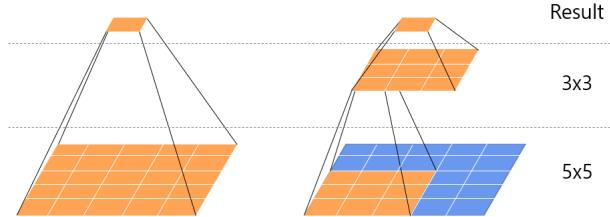
Large Kernel Convolutions

LSKNet có khả năng mô hình hóa một chuỗi các ngữ cảnh xa để lựa chọn và xử lý các ngữ cảnh xa của dữ liệu đầu vào nhờ vào việc sử dụng các phép tích chập với kernel lớn được phân rã thành chuỗi



Hình 3.9: Cách hoạt động của khối LSK

các phép tích chập theo chiều sâu (depth-wise convolution), giúp mô hình hiểu được và tập trung vào các đặc trưng quan trọng trong không gian hình ảnh một cách linh hoạt và hiệu quả.



Hình 3.10: Phân rã phép tích chập với kernel lớn thành chuỗi phép tích chập với kernel nhỏ hơn

Khi dữ liệu đầu vào thay đổi hoặc yêu cầu đặc biệt được đặt ra, mô hình có thể thích ứng bằng cách điều chỉnh các phép tích chập và các tham số liên quan để tạo ra các đặc trưng phù hợp với ngữ cảnh cụ thể đó. Cụ thể, việc tăng kích thước kernel k , tỷ lệ giãn nở (dilation rate) d và vùng tiếp nhận (Receptive Fields) RF của phép tích chập theo chiều sâu (Depth-wise Convolution) thứ i trong chuỗi được xác định như sau:

$$k_{i-1} \leq k_i; \quad d_1 = 1, \quad d_{i-1} < d_i \leq RF_{i-1} \quad (3.5)$$

$$RF_1 = k_1, \quad RF_i = d_i(k_i - 1) + RF_{i-1} \quad (3.6)$$

Việc tăng kích thước kernel k và tỷ lệ giãn nở d đảm bảo rằng vùng tiếp nhận RF có khả năng mở rộng đủ nhanh để mô hình tăng cường khả năng nhận biết các đặc trưng trong các không gian rộng lớn. Tác giả cũng đặt một giới hạn trên tỷ lệ giãn nở để đảm bảo rằng các phép tích chập giãn nở không tạo ra các khoảng trống giữa các bản đồ đặc trưng.

Để có được các đặc trưng có thông tin ngữ cảnh phong phú từ các phạm vi khác nhau cho đầu vào \mathbf{X} , một loạt các phép tích chập theo chiều sâu (depth-wise convolution) được phân rã với các trường tiếp nhận (Receptive Fields) khác nhau được áp dụng:

$$\mathbf{U}_0 = \mathbf{X}, \quad \mathbf{U}_{i+1} = \mathcal{F}_i^{dw}(\mathbf{U}_i), \quad (3.7)$$

trong đó $\mathcal{F}_i^{dw}(\cdot)$ là các phép tích chập theo chiều sâu (depth-wise convolutions) với kernel k_i và độ giãn nở (dilation) d_i . Giả định có N kernels được phân rã, mỗi kernel sau đó sẽ được xử lý với phép tích chập 1×1 $\mathcal{F}^{1 \times 1}(\cdot)$, cho phép trộn kênh cho từng vectơ đặc trưng không gian được miêu tả bằng công thức dưới đây:

$$\tilde{\mathbf{U}}_i = \mathcal{F}_i^{1 \times 1}(\mathbf{U}_i), \text{ for } i \text{ in } [1, N], \quad (3.8)$$

Có hai ưu điểm khi sử dụng Large Kernel Convolutions. Đầu tiên là chúng tạo ra nhiều đặc trưng với các vùng tiếp nhận lớn khác nhau, làm cho việc lựa chọn lõi sau này dễ dàng hơn. Ưu điểm thứ hai là khả năng phân rã tuần tự các phép tích chập mang lại hiệu quả tốt hơn so với việc sử dụng một kernel đơn lớn, giúp tối ưu hóa quá trình huấn luyện mô hình và giảm độ phức tạp tính toán cũng như số lượng tham số.

Spatial Kernel Selection

Spatial Kernel Selection được sử dụng để tăng cường khả năng của mạng trong việc tập trung vào các vùng ngữ cảnh không gian quan trọng nhất để phát hiện các mục tiêu. Mục tiêu của quá trình này là chọn ra các kernel (hoặc các phép tích chập) phù hợp nhất với các vùng không gian cụ thể trong dữ liệu. Điều này giúp mạng tập trung vào các đặc trưng quan trọng trong không gian của hình ảnh, giảm thiểu sự phân tán và tăng cường khả năng phát hiện và phân loại các đối tượng. Đầu tiên, chúng ta nối các đặc trưng thu được từ các kernel khác nhau, có phạm vi vùng tiếp nhận khác nhau:

$$\tilde{U} = [\tilde{U}_1; \dots; \tilde{U}_i]$$

và sau đó trích xuất mỗi quan hệ không gian một cách hiệu quả bằng cách áp dụng các phép gom nhóm trung bình và tối đa trên channel (được ký hiệu là $\mathcal{P}_{avg}(\cdot)$ và $\mathcal{P}_{max}(\cdot)$ cho \tilde{U}):

$$SA_{avg} = \mathcal{P}_{avg}(\tilde{U}), \quad SA_{max} = \mathcal{P}_{max}(\tilde{U}) \quad (3.9)$$

trong đó \mathbf{SA}_{avg} và \mathbf{SA}_{max} là bộ mô tả gom nhóm trung bình và tối đa.

Để thông tin có thể tương tác giữa các bộ mô tả, tác giả đã nối các đặc trưng được gom nhóm trong không gian và sử dụng một lớp tích chập $\mathcal{F}^{2 \rightarrow N}(\cdot)$ để biến đổi các tính năng được gom nhóm (với 2 channels) thành N bản đồ tập trung không gian:

$$\widehat{\mathbf{SA}} = \mathcal{F}^{2 \rightarrow N}([\mathbf{SA}_{avg}; \mathbf{SA}_{max}]). \quad (3.10)$$

Với mỗi bản đồ tập trung không gian, $\widehat{\mathbf{SA}}_i$, một hàm kích hoạt sigmoid được áp dụng để nhận được các mặt nạ không gian đơn cho từng kernel phân rã từ kernel lớn:

$$\widetilde{\mathbf{SA}}_i = \sigma(\widehat{\mathbf{SA}}_i), \quad (3.11)$$

trong đó $\sigma(\cdot)$ ký hiệu hàm sigmoid. Đặc trưng từ chuỗi các kernel phân rã từ kernel lớn sẽ được nhân với trọng số mặt nạ không gian đơn tương ứng của nó và kết hợp với một lớp tích chập $\mathcal{F}(\cdot)$ để nhận được space attention \mathbf{S} , công thức tổng hợp cuối cùng là:

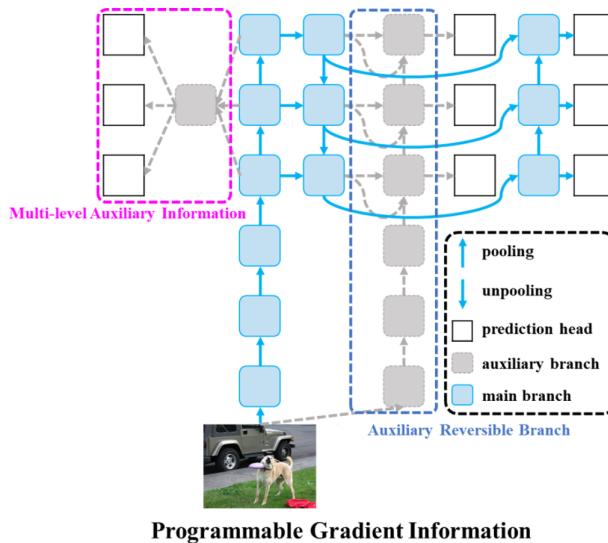
$$\mathbf{S} = \mathcal{F}\left(\sum_{i=1}^N (\widetilde{\mathbf{SA}}_i \cdot \widetilde{\mathbf{U}}_i)\right). \quad (3.12)$$

Tóm lại, LSKNet tốt trong việc phát hiện các vật thể nhỏ ở khoảng cách xa nhờ vào khả năng mô hình hóa các ngữ cảnh xa, sử dụng cơ chế chọn lọc không gian, và tính linh hoạt và hiệu suất của nó trong việc xử lý hình ảnh.

3.4.3 Programmable Gradient Information

Ý tưởng về thông tin gradient có thể lập trình PGI (Programmable Gradient Information) được đề ra để giải quyết những thay đổi khác nhau mà mạng có kiến trúc lớn yêu cầu nhằm mang lại nhiều lợi ích cho mô hình trong quá trình huấn luyện. PGI có thể cung cấp thông tin đầu vào dày đủ cho nhiệm vụ mục tiêu để tính toán hàm mục tiêu, nhờ đó có thể thu được thông tin gradient đáng tin cậy để cập nhật trọng số của mạng.

PGI chủ yếu bao gồm ba thành phần, đó là nhánh chính (main branch), nhánh đảo ngược phụ (auxiliary reversible branch) và thông tin phụ trợ đa cấp (multi-level auxiliary information).



Hình 3.11: Kiến trúc mô hình tích hợp PGI

Quá trình suy luận của PGI chỉ sử dụng nhánh chính và do đó không yêu cầu thêm bất kỳ chi phí suy luận nào. Đối với hai thành phần còn lại, chúng được sử dụng để giải quyết hoặc làm giảm bớt một số trở ngại phổ biến trong phương pháp học sâu. Trong số đó, nhánh đảo ngược phụ trợ được thiết kế để giải quyết các vấn đề gây ra bởi sự đào sâu của mạng lưới thần kinh. Việc đào sâu mạng sẽ gây ra tình trạng tắc nghẽn thông tin, khiến hàm mất mát không thể tạo ra gradient đáng tin cậy. Đối với thông tin phụ trợ đa cấp, nó được thiết kế để xử lý vấn đề tích lũy lỗi do giám sát sâu gây ra, đặc biệt đối với kiến trúc và mô hình gọn nhẹ của nhiều nhánh dự đoán.

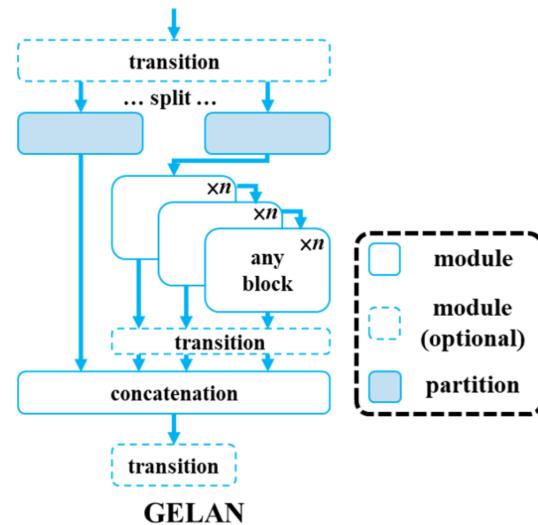
Nhánh đảo ngược phụ trợ đóng vai trò quan trọng trong việc tạo ra gradient đáng tin cậy và cập nhật các tham số mạng. Bằng cách cung cấp thông tin từ dữ liệu đến hàm mục tiêu, nó giúp hướng dẫn và tránh việc tìm ra các mối tương quan sai. Thông tin gradient đáng tin cậy từ nhánh phụ trợ này cũng thúc đẩy quá trình học tham số và hỗ trợ trích xuất thông tin quan trọng, cũng như giúp truyền đặc trưng hiệu quả hơn cho hàm mục tiêu. Đồng thời, nhánh phụ trợ có thể loại bỏ trong giai đoạn suy luận, giữ cho khả năng suy luận của mạng ban đầu không bị ảnh hưởng. Việc thêm một kiến trúc có thể đảo ngược vào nhánh chính sẽ tăng chi phí suy luận đáng kể, do thêm các kết nối từ các lớp sâu đến lớp nông, làm tăng thời gian suy luận thêm khoảng 20

Ý tưởng về thông tin phụ trợ đa cấp bao gồm chèn một mạng tích hợp giữa các lớp phân cấp

kim tự tháp đặc trưng (FPN) thuộc giám sát phụ trợ (auxiliary supervision) và nhánh chính (main branch), sau đó sử dụng nó để kết hợp các gradient được trả về từ các head dự đoán khác nhau. Sau đó, thông tin phụ trợ đa cấp sẽ tổng hợp thông tin gradient chứa tất cả các đối tượng mục tiêu và chuyển nó đến nhánh chính rồi cập nhật các tham số. Tại thời điểm này, các đặc trưng của hệ thống phân cấp kim tự tháp đặc trưng (FPN) của nhánh chính sẽ không bị chi phối bởi thông tin của một số đối tượng cụ thể. Nhờ đó, phương pháp này có thể giảm bớt vấn đề thông tin bị hỏng hóc trong quá trình giám sát sâu. Ngoài ra, bất kỳ mạng tích hợp nào cũng có thể được sử dụng thông tin phụ trợ đa cấp. Do đó, chúng ta có thể kế thừa trong các cấp độ ngữ nghĩa đa dạng để hướng dẫn các kiến trúc mạng có quy mô khác nhau học hiệu quả hơn.

3.4.4 Generalized ELAN (GELAN)

Generalized Efficient Layer Aggregation Network (GELAN) đã được nhóm tác giả của mô hình YOLOv9 đã tạo ra bằng cách kết hợp hai kiến trúc mạng thần kinh CSPNet và ELAN (đều được xây dựng dựa trên các nguyên tắc quy hoạch đường đi của gradient) sẽ được trình bày chi tiết ở phần này. GELAN mở rộng chức năng của ELAN trước đây bằng việc xếp chồng các lớp tích chập để có thể tận dụng được các khối tính toán và đường đi của gradient một cách hiệu quả. Sự cải tiến này cho phép GELAN tận dụng một loạt các đơn vị tính toán khác nhau, nâng cao tính linh hoạt và hiệu suất của nó.



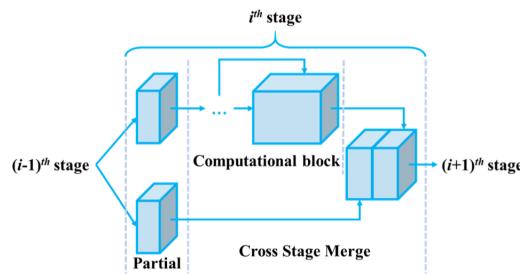
Hình 3.12: Kiến trúc mạng Generalized ELAN (GELAN)

CSPNet (Cross Stage Partial Networks)

CSPNet [15] được đề xuất vào năm 2019, đây là mạng dựa trên đường dẫn gradient ở cấp độ giai đoạn (stage-level). Giống như PRN [29] (Partial Residual Networks), CSP-Net dựa trên khái niệm tối đa hóa sự kết hợp gradient. Sự khác biệt giữa CSPNet và PRN là CSPNet tập trung vào việc xác

nhận sự cải thiện khả năng học của mạng bằng cách kết hợp gradient từ góc độ lý thuyết, trong khi CSPNet được thiết kế bổ sung để tối ưu hóa kiến trúc cho việc tận dụng tốc độ xử lý phần cứng.

CSPNet có hai thành phần chính: Cross stage partial operation và Gradient flow truncate operation. Thông qua các phân tích kiến trúc, tác giả nhận thấy rằng nguồn của gradient có thể được tối đa hóa khi mỗi channel có đường dẫn gradient khác nhau. Ngoài ra, từ góc độ tối đa hóa đầu thời gian độ dốc, số lượng timestamp gradient có thể được tối đa hóa khi mỗi channel có các khối tính toán với sâu khác nhau. Nhờ vào các phân tích này, thiết kế Cross stage partial operation của tác giả có thể tối đa hóa sự kết hợp của các gradient và tăng tốc độ suy luận mà không phá vỡ kiến trúc và có thể song song hóa. Kiến trúc này chia đặc trưng đầu vào thành hai phần, một phần được đi qua các khối tính toán và phần thứ hai được bỏ qua trực tiếp toàn bộ giai đoạn sau đó kết hợp với đầu ra của phần thứ nhất sau khi đã đi qua khối tính toán. Tác giả đã sử dụng thiết kế này để tăng số lượng nguồn gradient mà mạng có thể nhận và đồng thời giúp mô hình giảm một cách hiệu quả số lượng tham số, hoạt động, lưu lượng bộ nhớ và mức sử dụng tối đa bộ nhớ, cho phép hệ thống đạt được tốc độ suy luận nhanh hơn.

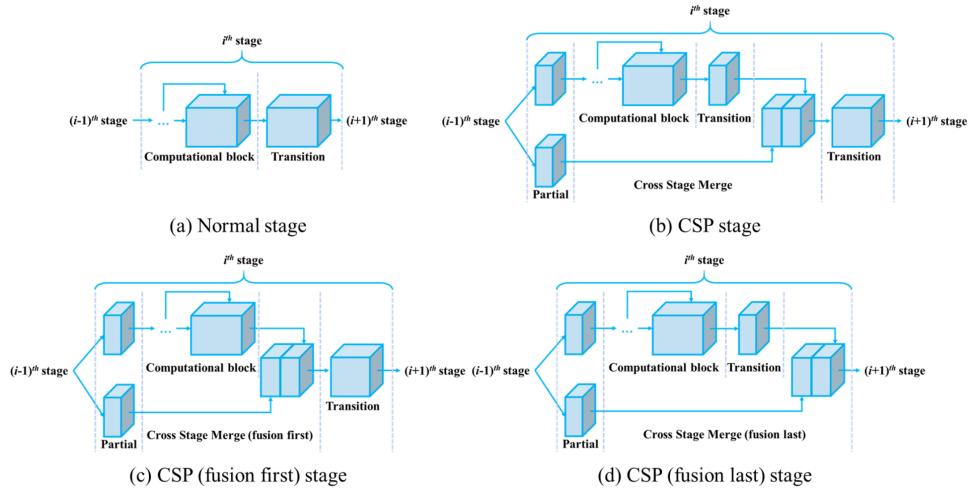


Hình 3.13: Cross stage partial operation

Gradient flow truncate operation đóng vai trò chèn một lớp chuyển tiếp (transition layer) ở cuối cả hai đường dẫn có chung nút gốc để cắt bớt luồng gradient trùng lặp vì các nguồn gradient được cung cấp từ hai đường dẫn có chung nút gốc chắc chắn sẽ chồng chéo lên nhau rất nhiều. Bằng cách trên, thông tin học được từ hai con đường và giai đoạn liền kề có sự đa dạng và rõ ràng hơn. Từ nguyên tắc trên, nhóm tác giả đã thiết kế được ba sự kết hợp khác nhau của Gradient flow truncate operation được mô tả bằng hình dưới đây.

ELAN (Efficient Layer Aggregation Networks)

Efficient Layer Aggregation Networks (ELAN) [22] được nhóm tác giả của mô hình YOLOv9 phát hành vào tháng 7 năm 2022. Mạng này thuộc danh mục mạng quy hoạch đường dẫn gradient ở cấp độ mạng. Mục đích chính của việc thiết kế ELAN là giải quyết vấn đề độ hội tụ của mô hình sâu sẽ giảm dần khi thực hiện việc mở rộng mô hình. Tác giả đã phân tích đường dẫn gradient ngắn nhất và đường dẫn gradient dài nhất qua từng lớp trong mạng tổng thể, nhận thấy rằng khi mạng tiếp tục mở rộng và đào sâu hơn thì thiết kế CSP fusion last (d) trong 3.14 cuối cùng sẽ có độ chính xác cao nhất, từ kết quả đó đã thiết kế kiến trúc tập hợp lớp với các đường truyền gradient hiệu quả. Mục đích thiết kế của nhóm tác giả là để tránh vấn đề sử dụng quá nhiều lớp chuyển tiếp (transition layer) và tránh làm cho đường dẫn gradient ngắn nhất của toàn mạng trở nên dài hơn.



Hình 3.14: Sự kết hợp từ các vị trí của Gradient flow truncate operation

GELAN: Sự kết hợp giữa CSP stage và ELAN với Computational Blocks

GELAN Kết hợp các điểm mạnh của CSP stage ((b) trong) 3.14 được biết đến với thiết kế gọn nhẹ và hiệu quả và kiến trúc mạng ELAN (tập trung vào quản lý đường dẫn gradient để việc học của mạng trở nên hiệu quả). GELAN cho phép sử dụng bất kỳ loại khối tính toán nào trong mạng, không chỉ bao gồm các lớp tích chập như ELAN (so sánh trong Hình 2.22). Điều này mang lại sự linh hoạt hơn trong việc thiết kế kiến trúc mạng hiệu quả.

Trong mô hình YOLOv9, các khối Computational Blocks trong GELAN được thay thế bằng khối RepNCSP. RepNCSP tận dụng thiết kế CSP fusion last ((d) trong 3.14), được mô tả như hình dưới đây:

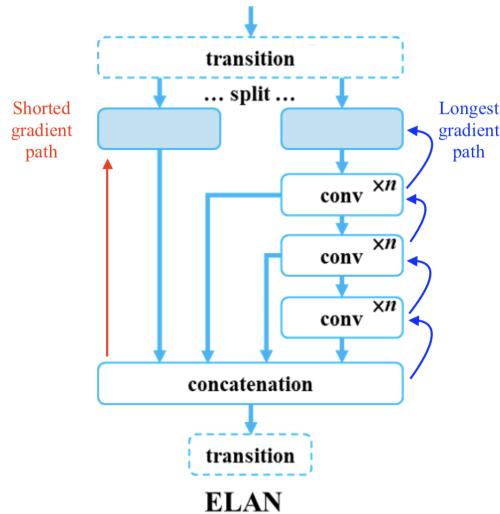
Bên cạnh đó, khối RepNBottleNeck trong khối RepNCSP 3.16 cũng sử dụng thiết kế Cross stage partial operation 3.13, được mô tả như hình sau:

3.5 Đề xuất các kiến trúc mô hình cải tiến

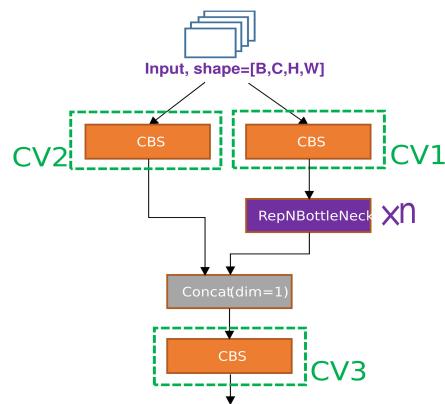
3.5.1 RepNDCNELAN4 - GELAN với Deformable Convolution

Trong nỗ lực tối ưu hóa và cải thiện khả năng phát hiện và theo dõi đối tượng trong camera fisheye, nhóm nghiên cứu của chúng tôi đã phát hiện ra một kỹ thuật mới đầy hứa hẹn: Deformable Convolution [30]. Khác với các phép tích chập truyền thống, Deformable Convolution có khả năng thay đổi vùng lấy mẫu, giúp mô hình học được các biến thể của đặc trưng từ các lớp phía trước, đồng thời làm cho vùng tiếp nhận (receptive fields) trở nên đa dạng hơn. Điều này giống như việc cung cấp cho mô hình khả năng linh hoạt, giúp nó có thể "nhìn thấy" và "hiểu" các đặc trưng của vật thể từ nhiều góc nhìn, tư thế và kích thước khác nhau.

Những lợi ích này đặc biệt phù hợp để giải quyết vấn đề mà chúng tôi gặp phải với hình ảnh fisheye. Vật thể trong những hình ảnh này thường bị bê cong và thay đổi kích thước khi nằm ở rìa ống kính, khiến việc nhận dạng trở nên khó khăn. Với sự linh hoạt trong vùng lấy mẫu của Deformable



Hình 3.15: Mạng ELAN với đường dẫn của gradient

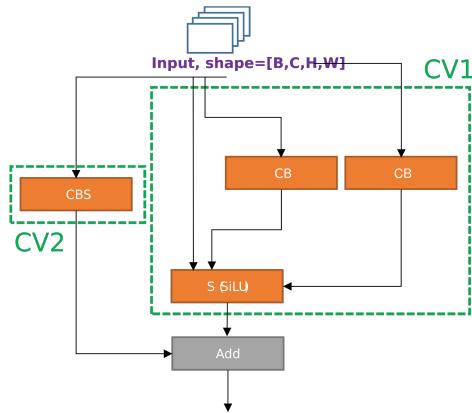


Hình 3.16: Kiến trúc khối RepNCSP

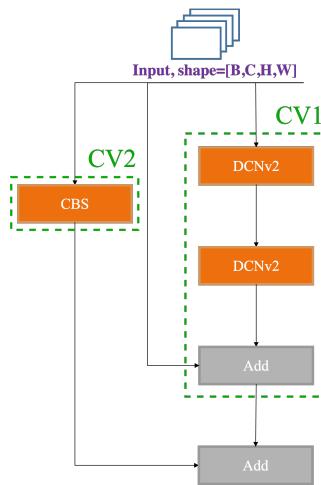
Convolution, chúng tôi tin rằng kiến trúc mạng với sự tích hợp Deformable Convolution có thể trích xuất được các đặc trưng từ những vật thể này một cách hiệu quả hơn, giúp nhận diện chúng dễ dàng hơn.

Lấy cảm hứng từ nghiên cứu của [31], nơi Deformable Convolution đã được tích hợp vào phiên bản YOLOv8 để giải quyết các vấn đề nhận dạng vật thể từ fisheye camera, chúng tôi quyết định áp dụng kỹ thuật này vào mô hình của mình. Chúng tôi đề xuất cải tiến mạng GELAN trong YOLOv9 bằng cách thay thế khối RepNBottleNeck (Hình 3.17) bằng khối RepNBottleNeckDCNv2 (Hình 3.18). Khối RepNBottleNeckDCNv2 này chủ yếu thay đổi ở phần BottleNeck bằng cách tích hợp liên tiếp hai lớp Deformable Convolution. Thay đổi này không chỉ mang lại lợi ích từ Deformable Convolution mà còn giữ được nguyên tắc quy hoạch đường đi gradient của kiến trúc GELAN.

Việc sử dụng các lớp BottleNeck với nhiều lớp Deformable Convolution liên tiếp giúp mạng lưới trở nên linh hoạt hơn rất nhiều so với chỉ sử dụng một lớp duy nhất. Điều này đã được tác giả của



Hình 3.17: Kiến trúc khối RepNBottleNeck



Hình 3.18: Kiến trúc khối RepNBottleNeckDCNv2

Deformable Convolution chứng minh bằng các kết quả vượt trội trong các thí nghiệm của họ [31].

Nhờ sự kết hợp này, RepNDCELAN4 không chỉ cải thiện hiệu suất nhận dạng đối tượng trong các hình ảnh fisheye mà còn mở ra nhiều tiềm năng mới trong việc áp dụng vào các hệ thống thị giác máy tính khác. Kết quả ban đầu cho thấy mô hình đã vượt qua nhiều thách thức, đặc biệt là trong việc phát hiện và nhận dạng các đối tượng nhỏ và bị che khuất.

3.5.2 RepNLSKELAN4 - GELAN với Large Selective Kernel Network

Trong quá trình nghiên cứu và cải tiến các mô hình phát hiện đối tượng, nhóm nghiên cứu của chúng tôi đã gặp phải một thách thức lớn: làm sao để cải thiện khả năng nhận dạng các đối tượng trong ảnh fisheye - những hình ảnh thường xuyên biến dạng với nhiều góc nhìn và điều kiện ánh sáng phức tạp. Mô hình YOLOv9 mà chúng tôi đang sử dụng đã đạt được nhiều thành tựu, nhưng vẫn còn nhiều khó khăn khi xử lý những hình ảnh này.

Khi đó, chúng tôi đã phát hiện một nghiên cứu đột phá từ nhóm tác giả của LSKNet [28]. LSKNet,

hay Large Selective Kernel Network, đã mở ra một hướng đi mới đầy tiềm năng. Điểm đặc biệt của LSKNet là khả năng mô hình hóa các ngữ cảnh xa để lựa chọn và xử lý thông qua các phép tích chập với kernel lớn, được phân rã thành chuỗi các phép tích chập theo chiều sâu (depth-wise convolution). Cơ chế này giúp mô hình tập trung vào các đặc trưng quan trọng trong không gian hình ảnh một cách linh hoạt và hiệu quả nhờ vào Spatial Selective.

Điều này giống như việc trao cho mô hình khả năng "quan sát" xa từng chi tiết nhỏ trong không gian hình ảnh, từ đó tăng cường khả năng phát hiện các mục tiêu bằng cách tập trung vào các vùng ngữ cảnh quan trọng nhất. Đặc biệt, Spatial Kernel Selection không chỉ giảm thiểu sự phân tán mà còn hiệu quả hơn nhiều so với phương pháp Channel Attention truyền thống.

Nhin thấy tiềm năng lớn từ LSKNet, chúng tôi nhận ra rằng đây chính là giải pháp cho những khó khăn mà mô hình cơ sở YOLOv9 đang gặp phải. Vì vậy, chúng tôi quyết định tích hợp khói LSKNet vào kiến trúc mạng GELAN của YOLOv9, tạo ra một phiên bản mới: RepNLSKELAN4. Khối LSKNet được thêm vào không chỉ để tận dụng những lợi thế của nó mà còn để đảm bảo rằng mô hình có thể xử lý hiệu quả các hình ảnh từ camera fisheye với các tinh chỉnh trên mạng GELAN. Hình ảnh từ loại camera này thường có góc rộng, chứa nhiều đối tượng, đa dạng ngữ cảnh, và các đối tượng thường xuất hiện ở các vị trí xa và rìa của ống kính, điều này làm cho việc nhận diện trở nên khó khăn.

Trong quá trình thực hiện, chúng tôi đã phác họa lại các thí nghiệm của nhóm tác giả LSKNet, nơi họ sử dụng camera từ góc trên cao để thu thập dữ liệu. Cảnh tượng này tương tự với những gì chúng tôi đối mặt với hình ảnh từ fisheye trên tập dữ liệu FishEye8K [1]: các đối tượng bị thu nhỏ, nằm rải rác và khó nhận diện. Chúng tôi tin rằng nếu LSKNet có thể hoạt động tốt trong các thí nghiệm từ xa của họ, thì nó cũng sẽ phù hợp với những hình ảnh fisheye mà chúng tôi đang xử lý.

Lấy cảm hứng từ nghiên cứu này, chúng tôi đã cài tiến mạng GELAN trong YOLOv9 bằng cách thay thế khối RepNBottleNeck (Hình 3.17) bằng khối RepNBottleNeckLSK (Hình 3.19). Kiến trúc mới này không chỉ mang lại những thế mạnh của LSKNet mà còn giữ được nguyên tắc quy hoạch đường đi gradient của GELAN, đảm bảo sự ổn định và hiệu quả trong quá trình huấn luyện.

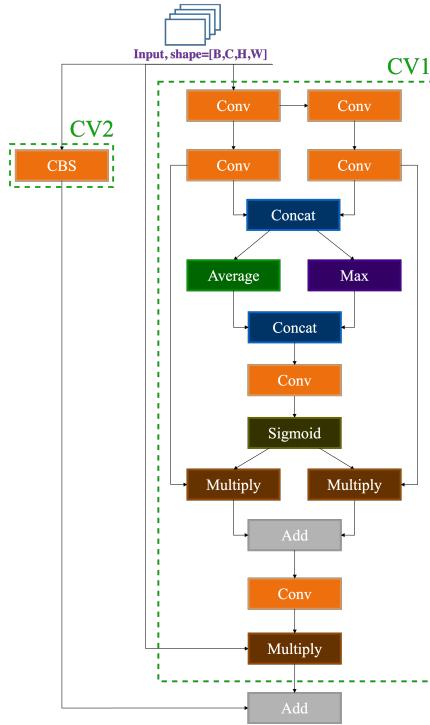
RepNLSKELAN4 đã trở thành một sự kết hợp độc đáo và mạnh mẽ giữa GELAN và LSKNet. Nhờ vào khả năng mô hình hóa các ngữ cảnh xa một cách hiệu quả, cùng với cơ chế chọn lọc không gian linh hoạt, mô hình này đã vượt qua nhiều thử thách trong việc xử lý hình ảnh fisheye. Điều quan trọng là, nó vẫn giữ được nguyên tắc quy hoạch đường đi gradient của toàn bộ kiến trúc mạng, từ đó tạo ra một bước tiến quan trọng trong hành trình nghiên cứu và phát triển của chúng tôi.

3.6 Đề xuất cải tiến hàm mất mát

3.6.1 Các hàm mất mát của YOLOv9

Hàm mất mát của YOLOv9 được chia thành hai phần tương ứng với các hàm tối ưu cho dự đoán Classification và Regression của mô hình:

- Classification loss: là hàm mất mát dành cho việc tối ưu các dự đoán phân loại của đối tượng. Classification loss trong YOLOv9 sử dụng hàm mất mát BCEWithLogitsLoss (Binary Cross Entropy with Logits Loss).



Hình 3.19: Kiến trúc mạng RepNBottleNeckLSK

- Regression loss: là hàm mất mát dành cho việc tối ưu dự đoán kích thước và vị trí của các bounding box. Regression loss trong YOLOv9 bao gồm IoU loss và DFL loss (Distribution Focal Loss).

Distribution Focal Loss (DFL)

Distribution Focal Loss (DFL) [32] ra đời nhằm giải quyết các khó khăn của việc dự đoán vị trí và kích cỡ của bounding box do sự phân phối không đồng đều của vị trí và kích cỡ của các bounding box cho từng object gây nên. Phương pháp này cho phép tạo ra một mô hình linh hoạt hơn có thể thích ứng với bất kỳ hình dạng phân bố nào mà không bị giới hạn ở các giả định cụ thể như phân bố Dirac delta hoặc Gaussian. Trước đây, các giả thuyết Dirac delta và Gaussian được đề ra để mô hình có thể học về sự phân phối thuộc dự đoán Regression của bounding box, tuy nhiên các giả định này thường gây ra sự mơ hồ và không chắc chắn trong những ngữ cảnh phức tạp của hình ảnh. Distribution Focal Loss [32] giúp mạng nhanh chóng tập trung vào việc học xác suất của các trị số xung quanh các vị trí liên tục của các hộp giới hạn mục tiêu (target bounding box) dưới sự phân bổ tùy ý và linh hoạt. DFL thúc đẩy mạng nhanh chóng tập trung vào các giá trị gần y bằng việc tăng thêm xác suất của y_i và y_{i+1} (2 điểm gần nhất của y , $y_i \leq y \leq y_{i+1}$). Vì việc học các hộp giới hạn chỉ dành cho các mẫu được gán nhãn dương (positive example), không có nguy cơ xảy ra vấn đề mất cân bằng lớp nên chỉ cần áp dụng phần entropy chéo hoàn chỉnh trong cho định nghĩa DFL, công thức của DFL được mô tả dưới đây:

$$\text{DFL}(\mathcal{S}_i, \mathcal{S}_{i+1}) = -((y_{i+1} - y) \log(\mathcal{S}_i) + (y - y_i) \log(\mathcal{S}_{i+1})). \quad (3.13)$$

Trong đó, \mathcal{S}_i và \mathcal{S}_{i+1} là các nghiệm toàn cục tối thiểu của DFL, trong YOLOv9 $\mathcal{S}_i = \frac{y_{i+1}-y_i}{y_{i+1}-y_i}$, $\mathcal{S}_{i+1} = \frac{y-y_i}{y_{i+1}-y_i}$. Các nghiệm toàn cục tối thiểu DFL có thể đảm bảo mục tiêu Regression có thể ước tính \hat{y} xấp xỉ với nhãn tương ứng y . Với $\hat{y} = \sum_{j=0}^n P(y_j)y_j = \mathcal{S}_i y_i + \mathcal{S}_{i+1} y_{i+1} = \frac{y_{i+1}-y_i}{y_{i+1}-y_i} y_i + \frac{y-y_i}{y_{i+1}-y_i} y_{i+1} = y$ đảm bảo cho sự chính xác của hàm mất mát.

CIOU loss

Trong các tình huống mà hộp giới hạn thực tế và dự đoán không chồng lên nhau thì hàm mất mát IoU không thể phân biệt giữa chúng, ngay cả khi dự đoán có thể gần với hộp giới hạn thực tế hơn so với dự đoán khác. Hạn chế này của hàm mất mát IoU có thể làm giảm hiệu suất quá trình huấn luyện, đặc biệt trong các nhiệm vụ phát hiện vật thể nơi mà việc xác định vị trí chính xác là quan trọng.

Để giải quyết hạn chế này, hàm mất mát CIOU [33] giới thiệu các thành phần bổ sung:

- Diện tích giao nhau giữa hộp dự đoán và hộp giới hạn thực tế - được gọi là hàm mất mát IoU.
- Khoảng cách chuẩn hóa giữa điểm trung tâm của hộp dự đoán và hộp giới hạn thực tế - được gọi là hàm mất mát DIoU.
- Tỷ lệ khung hình của hộp dự đoán và hộp thực tế.

Giống như hàm mất mát GIoU và DIoU [34], hàm mất mát CIOU di chuyển hộp giới hạn dự đoán về phía hộp giới hạn thực tế cho các trường hợp không giao nhau nhưng hội tụ nhanh hơn và chỉ cần ít vòng lặp hơn so với hàm mất mát IoU và GIoU. Nó cải thiện độ chính xác trung bình (AP) và độ bao phủ trung bình (AR) cho việc phát hiện và phân đoạn đối tượng. Công thức của CIOU loss là:

$$CIOU = 1 - IoU + \frac{d^2(b, b^{gt})}{C^2} + \alpha v \quad (3.14)$$

Trong đó: b và b^{gt} lần lượt là tâm của hộp giới hạn dự đoán và thực tế, $d(\cdot)$ là khoảng cách Euclidean, C đại diện cho độ dài đường chéo của hộp bao phủ nhỏ nhất chứa hai hộp, α là một tham số dương để cân đối, v đo lường sự nhất quán của tỉ lệ khung hình.

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2 \quad (3.15)$$

α được định nghĩa như sau:

$$\alpha = \frac{v}{(1 - IoU) + v} \quad (3.16)$$

Binary Cross Entropy Loss

Binary Cross-Entropy Error Loss (hay còn gọi là BCE loss) là một hàm mất mát được sử dụng phổ biến trong các bài toán phân loại nhị phân. Trong bài toán này, mỗi mẫu dữ liệu có thể được phân vào một trong hai nhóm (positive hoặc negative), và BCE loss được sử dụng để đo lường sự

khác biệt giữa phân phối xác suất dự đoán và phân phối xác suất thực tế của các nhóm. Công thức của BCE loss được tính như sau:

$$BCE(y, \hat{y}) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})) \quad (3.17)$$

Trong đó y là nhãn thực thể còn \hat{y} là nhãn dự đoán của mô hình.

3.6.2 Đề xuất hàm mất mát tối ưu cho mô hình phát hiện vật thể trên camera fisheye

Nhận thấy các khó khăn mà các hàm mất mát của mô hình cơ sở (YOLOv9) đang gặp phải trong giai đoạn học cũng như tối ưu các dự đoán của mình, chúng tôi đã đề xuất một hàm mất mát cải tiến mới nhằm tối ưu cho việc phát hiện vật thể cho camera fisheye đặc biệt là ngữ cảnh phát hiện và nhận dạng các phương tiện giao thông khi camera đặt trên cao. Ý tưởng cải tiến được xuất phát từ các phân tích của chúng tôi về sự ảnh hưởng của phân bố vị trí đến kích thước và hình dạng của các vật thể qua bộ dữ liệu Fisheye8K [1].

Các đề xuất trong hàm mất mát của chúng tôi chủ yếu thúc đẩy mô hình vượt qua được các rào cản, khó khăn mà các hàm mất mát đang có ở thời điểm hiện tại chưa thể thích ứng được khi xử lý các hình ảnh bị biến đổi do đặc tính của camera fisheye. Cụ thể, khi phân tích và đánh giá các sai số của mô hình cơ sở (YOLOv9), chúng tôi nhận thấy các sai số này chủ yếu phân bố ở gần rìa ảnh nơi các vật thể được thu lại bị bẻ cong và thu nhỏ hơn so với các vật thể nằm ở trung tâm ống kính như Hình 3.4. Đồng thời, các góc nhìn của vật thể khi nằm ở các vị trí khác nhau trong hình ảnh được thu lại từ camera fisheye cũng rất đa dạng nên mô hình rất khó nắm bắt được hình dạng tổng quát của từng vật thể của các lớp. Ví dụ như hình ảnh dưới đây được trích trong tập huấn luyện của [1] cho thấy người đi bộ với các vị trí khác nhau sẽ có các góc nhìn khác nhau từ đó hình dạng và kích thước cũng khác nhau:

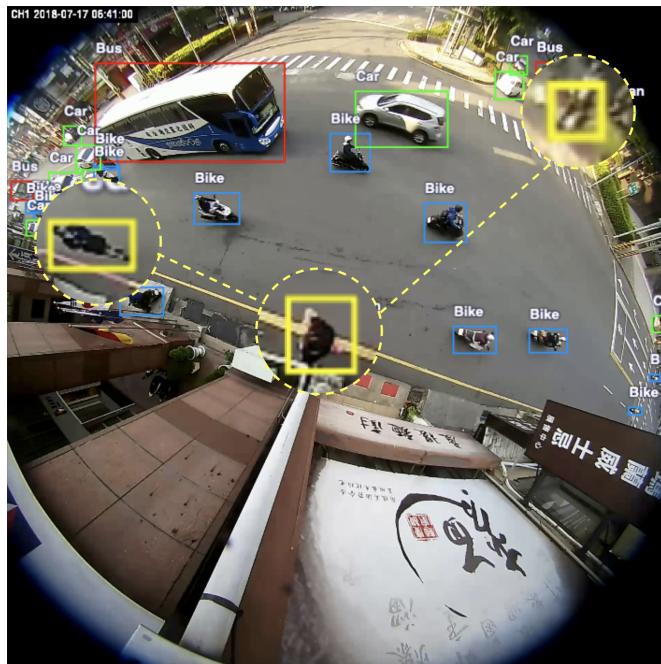
Distance-based Loss - Hàm loss dựa trên khoảng cách

Dựa vào phân tích về mặt hình ảnh về đặc tính của camera fisheye và các sai số của mô hình cơ sở trên bộ dữ liệu Fisheye8K [1], chúng tôi nhận thấy rằng: khi các vật thể càng xa tâm điểm (center point) của hình, các vật thể càng bị bẻ cong và thu nhỏ lại, làm cho mô hình dễ bị nhầm lẫn và không nhận dạng được. Đồng thời, sự phân bố của vật thể của các lớp cũng chủ yếu tập trung ở xung quanh rìa hình ảnh, nơi mà vật thể bị ảnh hưởng do đặc tính của camera fisheye. Do đó, chúng tôi đề xuất Distance-based Loss nhằm thúc đẩy mô hình ưu tiên học các vật thể bị biến dạng, tối ưu các dự đoán của mô hình khi gặp các vật thể ở vị trí rìa hình ảnh. Công thức của Distance-based Loss được mô tả dưới đây:

$$L_{\text{distance}} = \frac{1}{N} \sum_{i=1}^N w_d \cdot d_i \cdot L_i \quad (3.18)$$

Trong đó

- N là tổng số hộp giới hạn dự đoán
- Trọng số w_d điều chỉnh sự tập trung tối ưu của mô hình vào các vật thể xa tâm điểm



Hình 3.20: Người đi bộ ở các vị trí khác nhau trong tập dữ liệu Fisheye8K

- d_i là khoảng cách của hộp giới hạn dự đoán i với toạ độ tâm điểm hình, có công thức như sau:

$$d_i = \sqrt{(x_i - x_{\text{center}})^2 + (y_i - y_{\text{center}})^2}$$

với (x_i, y_i) là toạ độ của i -th hộp giới hạn dự đoán trên hình ảnh và $(x_{\text{center}}, y_{\text{center}})$ là toạ độ của tâm điểm hình.

- L_i là hàm binary cross-entropy loss cho hộp dự đoán i -th, có công thức:

$$L_i = -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i)$$

với y_i là nhãn thực tế cho hộp giới hạn thứ i và \hat{y}_i là nhãn dự đoán của hộp giới hạn thứ i bởi mô hình.

Khi các bounding box dự đoán có Entropy Loss cao và nằm xa toạ độ tâm điểm thì Distance-based Loss sẽ có giá trị lớn để thúc đẩy mô hình học các trường hợp này và ngược lại.

Angle-based Loss - Hàm loss dựa trên góc độ

Ngoài tối ưu dựa trên các đặc tính vốn có của camera fisheye, chúng tôi còn phân tích góc đặt camera cho các trường hợp sử dụng camera fisheye cho phát hiện phương tiện giao thông ở đường phố hoặc các giao lộ, từ đó đưa ra các đề xuất tối ưu dựa trên góc độ phân tích này. Nhận thấy rằng vị trí của vật thể cũng ảnh hưởng đến độ xoay độ vật thể so với các trục và đồng thời tạo ra các góc nhìn, biến thể khác của vật thể Hình 3.20 làm cho mô hình khó nhận diện hơn. Do đó, chúng tôi đã

để xuất hàm mất mát Angle-based Loss thúc đẩy mô hình nhận diện các biển thể này tốt hơn bằng cách xem xét góc được tạo ra giữa vật thể và trực hoành đi qua tâm hình. Công thức của Angle-based Loss được mô tả dưới đây:

$$L_{\text{angle}} = \frac{1}{N} \sum_{i=1}^N w_a (1 - \cos(\theta_i)) \cdot L_i \quad (3.19)$$

Trong đó

- $\theta_i = \arctan\left(\frac{y_i - y_{\text{center}}}{x_i - x_{\text{center}}}\right)$ là góc được tạo giữa hộp giới hạn dự đoán thứ i -th với trực hoành. với (x_i, y_i) là toạ độ của i -th hộp giới hạn dự đoán và $(x_{\text{center}}, y_{\text{center}})$ là toạ độ của tâm điểm hình
- Trọng số w_a điều chỉnh sự tập trung tối ưu của mô hình vào các biển thể và góc nhìn khác của vật thể
- L_i là binary cross-entropy loss cho hộp giới hạn dự đoán thứ i giống L_i trong Distance-based Loss tại Công thức 3.6.2

Khi các bounding box dự đoán có Entropy Loss cao và có góc tạo với toạ độ tâm điểm hình lớn thì Angle-based Loss sẽ có giá trị lớn để thúc đẩy mô hình học các trường hợp này và ngược lại.

Warp Loss - Hàm mất mát dành cho camera fisheye

Tổng hợp các điểm mạnh của 3.6.2 và 3.6.2, hàm mất mát Warp Loss thúc đẩy mô hình học các đặc trưng của camera fisheye thích ứng với đa dạng ngữ cảnh, biển thể của các vật thể có trong tập dữ liệu. Công thức của hàm mất mát Warped Loss được tổng hợp của Distance-based Loss 3.6.2 và Angle-based Loss 3.6.2 nhằm tạo ra một hàm mất mát duy nhất có thể tối ưu được cho camera fisheye. Công thức hoàn chỉnh của Warp Loss được mô tả sau đây:

$$L_{\text{warp}} = \frac{1}{N} \sum_{i=1}^N L_i \cdot [w_d \cdot d_i + w_a \cdot (1 - \cos(\theta_i))] \quad (3.20)$$

trong đó

- N là tổng số hộp giới hạn dự đoán
- L_i là hàm binary cross-entropy loss cho hộp dự đoán i -th, có công thức như sau:

$$L_i = -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i)$$

với y_i là nhãn thực tế cho hộp giới hạn thứ i và \hat{y}_i là nhãn dự đoán của hộp giới hạn thứ i bởi mô hình.

- Trọng số w_d điều chỉnh độ ảnh hưởng của Distance-based lên Warp Loss
- d_i là khoảng cách của hộp giới hạn dự đoán i với toạ độ tâm điểm hình, có công thức như sau:

$$d_i = \sqrt{(x_i - x_{\text{center}})^2 + (y_i - y_{\text{center}})^2}$$

với (x_i, y_i) là tọa độ của i -th hộp giới hạn dự đoán trên hình ảnh và (x_{center}, y_{center}) là tọa độ của tâm điểm hình.

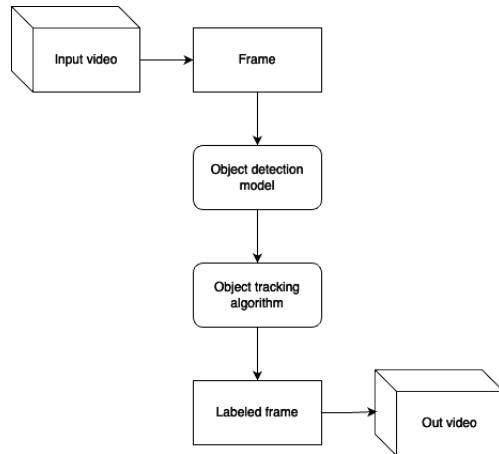
- Trọng số w_a điều chỉnh sự ảnh hưởng của Angle-based lên Warp Loss
- $\theta_i = \arctan\left(\frac{y_i - y_{center}}{x_i - x_{center}}\right)$ là góc được tạo giữa hộp giới hạn dự đoán thứ i -th với trục hoành. với (x_i, y_i) là tọa độ của i -th hộp giới hạn dự đoán và (x_{center}, y_{center}) là tọa độ của tâm điểm hình

3.7 Triển khai mô hình sau huấn luyện

Sau khi thu được các mô hình từ tất cả các phương pháp đề xuất, chúng tôi sẽ lựa chọn mô hình có kết quả tốt nhất từ các phương pháp đề xuất để so sánh kết quả với mô hình cơ sở (YOLOv9-e) đã được huấn luyện trên tập dữ liệu FishEye8K đồng thời tích hợp hợp các thuật toán theo dõi đối tượng để triển khai ra ứng dụng mô hình nhằm cung cấp một ứng dụng giám sát giao thông trong thời gian thực trên camera fisheye với độ chính xác và ổn định cao nhất hiện nay.

3.7.1 Tích hợp mô hình đã được huấn luyện với thuật toán theo dõi

Với việc theo dõi các đối tượng qua video hoặc thời gian thực bằng webcam, việc ghi lại ID của các đối tượng và cập nhật vị trí tương ứng của đối tượng với ID tương ứng là thực sự cần thiết cho mục đích giám sát. Do đó các thuật toán/mô hình giám sát theo sau mô hình phát hiện đối tượng là một trong những giải pháp phổ biến đã được áp dụng rộng rãi trong các ứng dụng giám sát hiện nay. Trong luận văn này, chúng tôi quyết định tích hợp và thử nghiệm hai mô đun DeepSORT và StrongSORT để có thể đưa ra các so sánh/ đánh giá và lựa chọn được thuật toán theo dõi tối ưu, phù hợp với ứng dụng.



Hình 3.21: Sơ đồ hoạt động của ứng dụng khi tích hợp thuật toán theo dõi

3.7.2 Nền tảng cung cấp môi trường triển khai ứng dụng

Hugging Face là một nền tảng mã nguồn mở nổi bật, chuyên cung cấp các công cụ và dịch vụ tiên tiến trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và trí tuệ nhân tạo (AI). Từ việc nổi tiếng với

việc phát triển các mô hình ngôn ngữ, Hugging Face đã mở rộng phạm vi hoạt động của mình bao gồm nhiều lĩnh vực khác như thị giác máy tính (Computer Vision) và học tăng cường (Reinforcement Learning), tạo nên một hệ sinh thái đa dạng và toàn diện cho các nhà phát triển và nhà nghiên cứu AI.

Hugging Face Spaces là một nền tảng đơn giản và tiện lợi để lưu trữ các ứng dụng demo học máy trên hồ sơ cá nhân hoặc tổ chức của bạn. Nó cho phép bạn tạo danh mục các dự án học máy, giới thiệu chúng tại các hội nghị hoặc cho các bên liên quan, và làm việc cùng những người khác trong hệ sinh thái học máy.

Spaces tích hợp sẵn hai SDK (Software Development Kit) mạnh mẽ là Streamlit và Gradio, giúp bạn xây dựng các ứng dụng Python hấp dẫn trong vài phút. Cả hai đều cho phép tạo giao diện người dùng trực quan và tương tác cho các mô hình học máy, làm cho việc trình bày và thử nghiệm mô hình trở nên dễ dàng hơn.

Ngoài ra, có thể sử dụng Docker để lưu trữ bất kỳ tệp Dockerfile nào, mang lại sự linh hoạt tối đa cho việc triển khai các ứng dụng phức tạp hoặc đặc thù. Cuối cùng, bạn cũng có thể tạo các Spaces tĩnh bằng JavaScript và HTML, phù hợp cho việc xây dựng các trang web tĩnh hoặc ứng dụng web đơn giản.

3.7.3 Framework hỗ trợ xây dựng ứng dụng

Gradio là một thư viện mã nguồn mở giúp tạo ra các giao diện web tương tác cho mô hình học máy và ứng dụng AI. Với Gradio, nhà phát triển và nhà nghiên cứu có thể nhanh chóng thiết lập các ứng dụng trực quan mà không cần kinh nghiệm phát triển web.

Gradio cho phép tạo giao diện người dùng đơn giản bằng vài dòng mã Python, hỗ trợ các loại đầu vào và đầu ra như văn bản, hình ảnh, âm thanh, và video. Điều này giúp việc trình bày và thử nghiệm mô hình dễ dàng hơn, cho phép người dùng trải nghiệm và đánh giá các tính năng của mô hình một cách trực quan nhất.

Gradio nổi bật với tính tương tác cao, cho phép nhập dữ liệu, hiển thị kết quả và điều chỉnh tham số mô hình trực tiếp trên giao diện web. Thư viện này tích hợp tốt với các nền tảng học máy phổ biến như TensorFlow, PyTorch và scikit-learn, giúp triển khai các mô hình đã huấn luyện một cách dễ dàng.

Gradio đồng thời hỗ trợ chia sẻ các giao diện qua URL duy nhất, giúp hợp tác và nhận phản hồi thuận tiện hơn. Với Gradio, việc tạo ứng dụng AI có giao diện người dùng trở nên dễ dàng, nhanh chóng và hiệu quả, lý tưởng cho cả người mới bắt đầu và chuyên gia AI.

Chương 4

Thực nghiệm, đánh giá và triển khai ứng dụng mô hình

Trong chương này, luận văn sẽ trình bày chi tiết mô hình thực nghiệm cũng như các thông số cấu hình mà luận văn đã sử dụng để tiến hành thực nghiệm. Đồng thời, các kết quả thực nghiệm cùng, các nội dung thảo luận về kết quả thực nghiệm của luận văn và phương pháp triển khai ứng dụng mô hình cũng sẽ được trình bày chi tiết trong chương này.

4.1 Chi tiết quá trình triển khai thử nghiệm

4.1.1 Mục đích thử nghiệm

So sánh kết quả đầu ra mô hình: Cho biết trước lớp đối tượng, đánh giá khả năng phát hiện đối tượng dựa trên phân lớp đối tượng đó. Điều này có ý nghĩa quan trọng trong việc đánh giá tính hiệu quả của mô hình trong các tác vụ nhận dạng đối tượng. Điều này sẽ cung cấp thông tin quan trọng về hiệu suất của mô hình và có thể hỗ trợ trong việc cải thiện và tối ưu hóa các phương pháp nhận dạng đối tượng kế tiếp.

4.1.2 Các triển khai thử nghiệm

Luận văn triển khai các thử nghiệm với việc sửa đổi mô hình cơ sở YOLOv9 dựa trên các đề xuất 3.5 và 3.6

- Triển khai 1: tích hợp YOLOv9-e và RepNDCNELAN4 làm tăng khả năng nhận diện các đối tượng ở rìa ảnh, nơi mà các đối tượng bị bóp méo trong tập dữ liệu FishEye8k.
- Triển khai 2: tích hợp YOLOv9-e và RepNLSKELAN4 làm tăng khả năng nhận dạng các đối tượng nhỏ trong ảnh.
- Triển khai 3: tích hợp YOLOv9-e từ hai kiến trúc mới là RepNDNCELAN4 và RepNLSKELAN4 làm tăng khả năng nhận dạng vật thể một cách tốt hơn.
- Triển khai 4: sử dụng hàm măt măt đề xuất Warp Loss 3.6 làm tăng khả năng nắm bắt được hình dạng của các vật thể một cách tổng quát hơn từ đó làm tăng hiệu suất mô hình.

4.2 Môi trường thử nghiệm

Trong quá trình thử nghiệm, các mô hình được huấn luyện trên tập dữ liệu FishEye8k. Số lượng ảnh ở tập train là 5288 ảnh, tập test là 2712 ảnh và phần validation được chia bằng 20% của tập training (1058 ảnh từ 5288 ảnh).

Luận văn triển khai phần thực nghiệm bằng việc sử dụng mô hình YOLOv9 là mô hình chính trong tác vụ nhận dạng. Chúng ta tải file pre-trained weights của model YOLOv9.

Mã nguồn huấn luyện thực nghiệm của luận văn được kế thừa và phát triển dựa trên mã nguồn mở của tác giả YOLOv9 [25], được cung cấp miễn phí trên [github](#).

Toàn bộ các mô hình thí nghiệm trong bài luận văn này được huấn luyện trên môi trường notebook của Kaggle với 2 core vCPUs, 29 GB Ram và 2 x GPU Tesla T4 và việc lựa chọn sử dụng một hay cả hai GPU phụ thuộc vào kiến trúc mô hình và số lượng batch size được chọn.

Trong quá trình huấn luyện thì hầu hết các tham số được cài đặt giống như mô hình cơ sở.

Một vài thông số đáng chú ý:

- Kích thước ảnh trong huấn luyện là 640
- Hệ số học 10^{-2} và sử dụng hàm tối ưu là SGD, batch size là 8.
- Động lượng (momentum) là 0.937
- Sự suy giảm trọng lượng (weight decay) là 0.0005.
- Ngưỡng IoU (Intersection over Union) là 0.5

Các thông số *iou* và *conf* được cài đặt tương tự trong bài báo [1] để đạt được sự công bằng trong việc đánh giá mô hình.

Tất cả mô hình được huấn luyện trên 120 epochs nhưng vào khoảng các epochs thứ 70 trở đi thì mô hình có khả năng bị overfit khá cao nên hầu như các mô hình từ epochs 60 trở đi đều được đánh giá cẩn thận để lựa chọn được mô hình tốt nhất.

Để tránh bị quá khớp chúng tôi có thực hiện một vài kỹ thuật tăng cường để giảm hiện tượng quá khớp nhưng trong quá trình thử nghiệm do nhận thấy mô hình hội tụ chậm và kết quả cũng không gây ra sự thay đổi nhiều nên chúng tôi cân nhắc bỏ qua việc tăng cường dữ liệu và chỉ sử dụng các thông số tăng cường mặc định của mô hình YOLOv9-e.

4.3 Các mô hình đề xuất

Các mô hình đề xuất đều được kết hợp dựa trên mô hình YOLOv9-e, YOLOv9-c và phần đề xuất 3.6, 3.5. Các thử nghiệm sẽ được kết hợp và tiến hành huấn luyện lần lượt để cho ra một kiến trúc tối ưu với hiệu suất cao nhất.

4.3.1 Hiệu suất của mô hình YOLOv9-e và YOLOv9-c huấn luyện với tập dữ liệu FishEye8K

Trong phần này, chúng tôi sẽ trình bày về quá trình kiểm tra hiệu suất của hai mô hình cơ sở là YOLOv9-e và YOLOv9-c khi huấn luyện với tập dữ liệu FishEye8K.

Mô hình đầu tiên mà chúng tôi triển khai là YOLOv9-e. Được biết đến với khả năng phát hiện đối tượng nhanh, chính xác và vượt trội so với các mô hình phát hiện đối tượng ở thời điểm thực hiện luận văn này, YOLOv9-e đã trở thành lựa chọn đầu tiên trong thử nghiệm của chúng tôi.

YOLOv9-e					
Classes	Precision	Recall	mAP _{0.5}	mAP _{0.5-.95}	F1-score
Bus	0.682	0.587	0.699	0.541	0.631
Bike	0.84	0.407	0.638	0.351	0.548
Car	0.873	0.612	0.757	0.531	0.72
Pedestrian	0.766	0.206	0.484	0.262	0.325
Truck	0.88	0.373	0.639	0.518	0.524
All	0.808	0.437	0.643	0.441	0.567

Bảng 4.1: Kết quả của mô hình YOLOv9-e base

Tiếp theo, chúng tôi triển khai mô hình YOLOv9-c để tạo ra cơ sở so sánh cho các mô hình rút gọn cải tiến sau này. Mục tiêu của chúng tôi là xác định hiệu suất của hai mô hình cơ sở trước khi áp dụng bất kỳ cải tiến nào.

YOLOv9-c					
Classes	Precision	Recall	mAP _{0.5}	mAP _{0.5-.95}	F1-score
Bus	0.909	0.549	0.743	0.608	0.6846
Bike	0.837	0.381	0.625	0.571	0.5236
Car	0.923	0.58	0.766	0.533	0.7123
Pedestrian	0.549	0.161	0.36	0.188	0.2489
Truck	0.717	0.367	0.544	0.424	0.4855
All	0.787	0.408	0.608	0.412	0.5374

Bảng 4.2: Kết quả của mô hình YOLOv9-c base

Bảng 4.1 và 4.2 cho biết kết quả của hai mô hình cơ sở YOLOv9-e và YOLOv9-c. Kết quả thực nghiệm ban đầu từ hai mô hình cơ sở đã cho chúng tôi một cái nhìn tổng quan về hiệu suất của từng mô hình trên tập dữ liệu FishEye8K. YOLOv9-e tỏ ra vượt trội hơn với độ chính xác (precision) và khả năng phát hiện (recall) cao hơn ở hầu hết các lớp so với YOLOv9-c. Đặc biệt, đối với các đối tượng như xe buýt và xe hơi, YOLOv9-e cho thấy hiệu suất vượt trội với mAP0.5 đạt lần lượt là 0.751 và 0.794.

Mặc dù vậy, cả hai mô hình vẫn còn tồn tại những hạn chế, đặc biệt là trong việc nhận dạng người đi bộ, với mAP0.5-0.95 khá thấp, cho thấy cần có những cải tiến để nâng cao hiệu suất. Chính vì lý do này, chúng tôi quyết định sử dụng kết quả của YOLOv9-e và YOLOv9-c làm cơ sở để đánh giá hiệu suất của các mô hình được đề xuất trong các phần tiếp theo.

4.3.2 RepNDCNELAN4

Sau khi đạt được những kết quả đáng khích lệ với các mô hình cơ bản YOLOv9-e và YOLOv9-c, chúng tôi bắt đầu tích hợp những cải tiến sâu hơn bằng cách tích hợp RepNDCNELAN4 vào YOLOv9-e. Quá trình này không chỉ giúp nâng cao hiệu suất mà còn mở ra những triển vọng mới trong việc tối ưu hóa mô hình.

Dựa vào các thí nghiệm trước đó trên mạng được tích hợp Deformable Convolution [30], các layer với Deformable Convolution được tích hợp vào mô hình ở các tầng ngữ nghĩa cao nhằm mang lại tính linh hoạt cho phép tích chập trên không gian ngữ nghĩa đầu vào đa dạng. Do đó, chúng tôi bắt đầu thử nghiệm bằng cách tích hợp RepNDCNELAN4 vào các layer ở tầng ngữ nghĩa cao (sau một vài

lớp Conv của mạng), đồng thời cũng xem xét sự kết hợp liên tiếp các layer này trên YOLOv9-e: layer thứ 5-7-9, 5-7, 5-9, 7-9, và layer thứ 9, thay thế cho RepNCPSELAN4. Mỗi phiên bản đều mang lại những kết quả đáng chú ý về hiệu suất của mô hình.

Version	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	$F1_{score}$
Layer 5-7-9	0.807	0.457	0.651	0.431	0.583
Layer 5-7	0.797	0.432	0.632	0.424	0.56
Layer 5-9	0.778	0.408	0.607	0.41	0.5352
Layer 7-9	0.805	0.441	0.643	0.445	0.5698
Layer 9	0.856	0.418	0.655	0.451	0.561

Bảng 4.3: Bảng kết quả thực nghiệm trên mô hình YOLOv9 kết hợp với RepNDCNELAN4

Khi nhìn vào bảng kết quả ở bảng 4.3, chúng tôi nhận thấy rằng phiên bản tích hợp RepNDCNELAN4 vào layer 5-7-9 và layer 9 đã thể hiện sự vượt trội so với mô hình YOLOv9-e ban đầu. Cụ thể, $mAP_{0.5}$ của cả hai phiên bản này đều cao hơn so với YOLOv9-e gốc (0.651 và 0.655 so với 0.647). Đặc biệt, layer 5-7-9 còn đạt điểm *Recall* là 0.457 so với 0.432 của YOLOv9-e, và điểm *F1-score* là 0.583 so với 0.5687. Tuy nhiên, nhìn vào kết quả tổng thể và các điểm số $mAP_{0.5}$, $mAP_{.5-.95}$ thì thí nghiệm trên layer 9 có phần ổn định và mang lại kết quả vượt trội nhất.

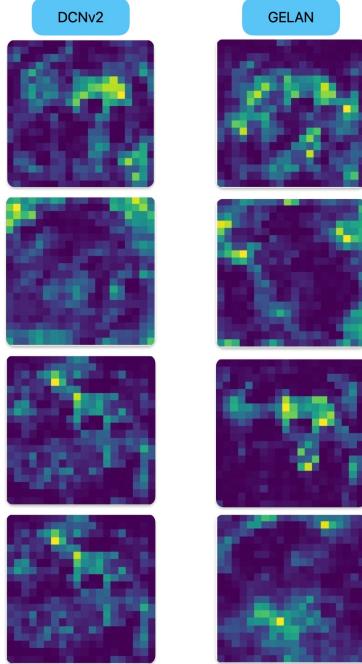
YOLOv9-e RepNDCNELAN4 layer 9					
Classes	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	$F1-score$
Bus	0.741	0.596	0.718	0.529	0.66
Bike	0.861	0.353	0.617	0.5	0.5
Car	0.927	0.612	0.784	0.55	0.737
Pedestrian	0.858	0.182	0.518	0.294	0.3
Truck	0.893	0.347	0.635	0.519	0.5
All	0.856	0.418	0.655	0.451	0.562

Bảng 4.4: Kết quả của mô hình YOLOv9-e kết hợp với RepNDCNELAN4 ở layer thứ 9

Từ kết quả trên các thí nghiệm tích hợp RepNDCNELAN4 vào YOLOv9-e, chúng tôi nhận thấy rằng Deformable Convolution cải thiện được khả năng phát hiện đối tượng trên camera fisheye khi được tích hợp vào các layer có tầng ngữ nghĩa cao (layer 9) trên backbone của mô hình tương tự như các thí nghiệm trước của tác giả [30] nhờ vào sự linh hoạt của vị trí lấy mẫu của kernel. Cụ thể, RepNDCNELAN4 được đặt tại layer 9 đã cải thiện việc phát hiện được các vật thể nhỏ như Pedestrian, Bike dựa trên điểm $mAP_{0.5}$, $mAP_{.5-.95}$ mà mô hình đạt được. Bảng 4.4.

Bên cạnh đó, việc tích hợp liên tiếp các layer này cũng mang lại kết quả khả quan song chúng tôi vẫn còn phải xem xét chi phí tính toán và lưu trữ bên cạnh thời gian suy luận của cả mô hình khi tích hợp RepNDCNELAN4 vào mô hình cơ sở để tạo nên một sự cải tiến mang lại ý nghĩa ở nhiều phương diện.

Chúng tôi cũng thực hiện việc trực quan hóa các đặc trưng trên mạng RepNDCNELAN4 bằng cách cho hình ảnh đi qua toàn bộ mạng và trực quan hóa toàn bộ các đặc trưng của từng chặng bằng bản đồ nhiệt để tạo ra các so sánh với mạng GELAN nhằm mục đích phân tích sự thay đổi khi tích hợp mạng này vào mô hình cơ sở. Từ hình ảnh trực quan hóa đặc trưng thu được, chúng tôi nhận thấy mạng RepNDCNELAN4 ở layer 9 trích xuất các đặc trưng của ảnh đầu vào mạnh mẽ hơn khi so sánh với GELAN cụ thể là các đặc trưng được phân bố một cách đa dạng và ở vùng rìa ảnh được tập trung hơn so với mạng GELAN ở mô hình cơ sở.



Hình 4.1: Khả năng trích xuất thông tin của RepNDCNELAN4 và GELAN.

4.3.3 RepNLSKELAN4

Tương tự với cách thực hiện thí nghiệm trên, chúng tôi cũng thử nghiệm việc kết hợp RepNLSKELAN4 vào mô hình cơ sở (YOLOv9-e). Chúng tôi tích hợp RepNLSKELAN4 vào các lớp khác nhau ở các lớp mang ngữ nghĩa cao nhất - các lớp cuối trên backbone tương ứng với các tầng khác nhau của mạng đặc trưng kim tự tháp [10] khi đưa vào neck của YOLOv9-e, nhằm tận dụng tối đa khả năng chọn lọc kích thước với các kernel lớn nhỏ khác nhau của LSKNet.

Chúng tôi bắt đầu thử nghiệm bằng cách tích hợp RepNLSKELAN4 vào các lớp ở tầng không gian ngữ nghĩa cao và ít thông tin về vị trí (các layer cuối của backbone), đồng thời cũng xem xét sự kết hợp liên tiếp các lớp này trên YOLOv9-e tại các pyramid thứ 3-4-5 tương ứng với layer 22, 25 và 28. Cụ thể, các thí nghiệm được tiến hành trên lớp thứ 22, 22-25, và 22-25-28, thay thế cho RepNCSELAN4.

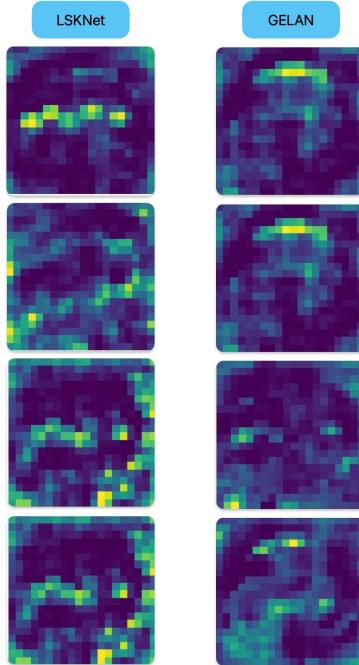
Version	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	$F1_{score}$
Layer 22-25-28	0.83	0.45	0.655	0.445	0.561
Layer 22-25	0.827	0.452	0.653	0.445	0.584
Layer 22	0.814	0.426	0.632	0.432	0.559

Bảng 4.5: Bảng kết quả thực nghiệm trên mô hình YOLOv9 kết hợp với RepNLSKELAN4

YOLOv9-e RepNLSKELAN4 layer 22-25-28					
Classes	Precision	Recall	mAP _{0.5}	mAP _{.5-.95}	F1-score
Bus	0.889	0.677	0.802	0.604	0.768
Bike	0.782	0.387	0.606	0.345	0.517
Car	0.92	0.618	0.785	0.549	0.739
Pedestrian	0.748	0.186	0.465	0.225	0.507
Truck	0.81	0.383	0.618	0.503	0.52
All	0.83	0.45	0.655	0.455	0.583

Bảng 4.6: Kết quả của mô hình YOLOv9-e kết hợp với RepNLSKELAN4 ở các layer 22-25-28

Dựa trên bảng kết quả trong bảng 4.5, chúng tôi thấy rằng việc tích hợp RepNLSKELAN4 vào các lớp 22-25-28 và lớp 22-25 của mô hình YOLOv9-e đã mang lại sự cải thiện đáng kể so với phiên bản ban đầu của YOLOv9-e. Cả hai phiên bản này đều có mAP_{0.5} cao hơn so với YOLOv9-e gốc (0.655 và 0.653 so với 0.647). Đáng chú ý, lớp 22-25-28 còn có Recall là 0.45 so với 0.432 của YOLOv9-e, và F1-score là 0.583 so với 0.5687. Tuy nhiên, tổng thể và các điểm số cho thấy rằng thí nghiệm trên lớp 22-25-28 có vẻ ổn định hơn các phiên bản còn lại và đem lại kết quả vượt trội nhất. Nhờ vào sự linh hoạt về kích thước của kernel, RepNLSKELAN4 mang lại các cải thiện trên các lớp có kích thước lớn như Bus, Truck, Car dựa vào điểm số mAP_{0.5}, mAP_{.5-.95} mà mạng đạt được trên phiên bản cải tiến ở liên tiếp các layer 22-25-28 trong bảng ??.



Hình 4.2: Khả năng trích xuất thông tin của RepNLSKELAN4 và GELAN.

Hình 4.2 so sánh kết quả trực quan hóa đặc trưng bằng bản đồ nhiệt từ mạng RepNLSKELAN4 tại layer 28 cho thấy sự cải thiện rõ rệt so với mạng GELAN dựa trên các đặc trưng được trích xuất ở các vùng rìa và trọng tâm hình ảnh, các đặc trưng trên mạng RepNLSKELAN4 được thể hiện một cách rõ ràng và chi tiết hơn. Điều này chứng minh rằng việc tích hợp RepNLSKELAN4 vào mô hình cơ sở giúp mô hình trích xuất và giữ lại thông tin của các đặc trưng quan trọng của ảnh đầu vào hiệu quả hơn. Sự tập trung tốt hơn ở các vùng rìa ảnh cũng góp phần cải thiện khả năng phát hiện các

đối tượng ở các vùng biên, điều mà mạng GELAN không làm tốt bằng. Điều này giúp nâng cao hiệu suất tổng thể của mô hình khi được tích hợp vào YOLOv9-e.

Từ kết quả các thí nghiệm tích hợp RepNLSKELAN4 vào YOLOv9-e, có thể kết luận rằng Large Selective Kernel Network đã cải thiện hiệu suất phát hiện đối tượng trên camera fisheye khi được tích hợp liên tiếp vào các lớp ở các tầng khác nhau của mạng đặc trưng kim tự tháp [10] nhờ vào sự linh hoạt về kích thước của các kernel. Việc tích hợp RepNLSKELAN4 vào mô hình cơ sở đã mang lại sự cải tiến đáng kể, tạo ra sự linh hoạt và mạnh mẽ hơn trong việc nhận diện và theo dõi các đặc trưng đặc thù trong ảnh được lấy từ camera fisheye.

4.3.4 Kết hợp RepNDCNELAN4 và RepNLSKELAN4

Từ hai kết quả trên của RepNDCNELAN4 và RepNLSKELAN4, chúng tôi quyết định sẽ tiến hành tích hợp cả hai kiến trúc này vào mô hình YOLOv9-e để tận dụng tối đa các ưu điểm của từng phương pháp. Các thử nghiệm bao gồm sự kết hợp của các layer khác nhau từ cả hai kiến trúc, nhằm tìm ra sự kết hợp tối ưu nhất.

Version	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	$F1_{score}$
Network A	0.776	0.427	0.62	0.421	0.55
Network B	0.844	0.435	0.649	0.449	0.574
Network C	0.809	0.455	0.653	0.450	0.582

Bảng 4.7: Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với RepNLSKELAN4 và RepNDCNELAN4

Network	RepNLSKELAN4	RepNDCNELAN4
A	Layer 22-25-28	Layer 5-7-9
B	Layer 22-25-28	Layer 9
C	Layer 22-25	Layer 9

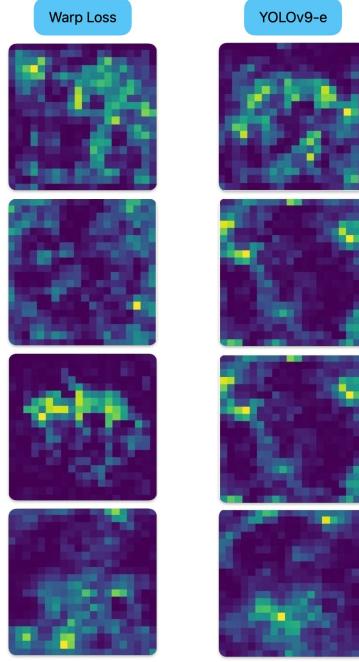
Bảng 4.8: Chi tiết về các layer được tích hợp giữa RepNLSKELAN4 và RepNDCNELAN4

Kết quả huấn luyện từ bảng 4.8 cho thấy network C, kết hợp từ các layer 22-25 của RepNLSKELAN4 và layer 9 của RepNDCNELAN4, đạt kết quả vượt trội so với network A và B. So sánh với kết quả bảng 4.1 và 4.2, network B và C đều thể hiện sự vượt trội, đặc biệt là network C với điểm *Recall* đạt 0.455 và $mAP_{.5-.95}$ đạt 0.450, cho thấy sự tương đồng và thậm chí có phần vượt trội so với các phiên bản trước. Sự tối ưu còn được thể hiện ở việc chúng tôi chỉ sử dụng một lớp RepNDCNELAN4, việc này làm tăng tốc độ suy luận và làm giảm chi phí tính toán của mô hình, làm cho mô hình nhẹ hơn nhưng vẫn đạt được một tốc độ suy luận tuyệt vời.

4.3.5 Tích hợp Warp Loss

Chúng tôi đã tiến hành huấn luyện các mô hình đạt hiệu suất cao nhất trước đó cùng với việc sử dụng hàm Warp Loss. Kết quả được thể hiện trong các bảng dưới đây.

Hình 4.3 là kết quả so sánh trích xuất đặc trưng giữa mô hình cơ sở và khi kết hợp với hàm Warp Loss được đề xuất. Dựa trên kết quả trực quan hóa, không khó để nhận thấy rằng mô hình kết hợp



Hình 4.3: Khả năng trích xuất thông tin của YOLOv9-e kết hợp Warp Loss và YOLOv9-e.

Warp Loss cải thiện đáng kể khả năng trích xuất đặc trưng với nhiều hơn các thông tin được tổng hợp và được trích xuất qua từng chặng được thể hiện, đặc biệt là ở các vùng rìa ảnh. Điều này cho thấy hàm Warp Loss giúp mô hình tập trung và phân bổ đặc trưng tốt hơn so với mô hình cơ sở, từ đó nâng cao hiệu quả nhận diện đối tượng. Những kết tích cực này khẳng định giá trị của việc tích hợp Warp Loss trong việc nâng cao hiệu suất của các mô hình phát hiện đối tượng.

YOLOv9-e + RepNDCNELAN4					
Version	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	F1-score
Layer 9	0.712	0.456	0.62	0.419	0.555

Bảng 4.9: Bảng kết quả thực nghiệm trên mô hình YOLOv9 kết hợp với RepNDCNELAN4 và sử dụng hàm Warp Loss.

YOLOv9-e + RepNLSKELAN4					
Version	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	F1-score
Layer 22-25-28	0.808	0.441	0.639	0.442	0.57

Bảng 4.10: Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với RepNLSKELAN4 và sử dụng hàm Warp Loss.

RepNLSKELAN4 + RepNDCNELAN4 + Warp Loss					
Version	Precision	Recall	$mAP_{0.5}$	$mAP_{.5-.95}$	F1-score
Network A	0.792	0.438	0.632	0.427	0.564
Network B	0.874	0.455	0.672	0.464	0.598
Network C	0.827	0.418	0.638	0.44	0.555

Bảng 4.11: Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với RepNLSKELAN4, RepNDCNELAN4 và sử dụng hàm Warp Loss.

Network B + Warp Loss					
Version	Precision	Recall	mAP _{0.5}	mAP _{.5-.95}	F1-score
Angle gain: 0.3 + Distance gain : 0.7	0.874	0.455	0.672	0.464	0.598
Angle gain: 0.7 + Distance gain : 0.3	0.874	0.455	0.672	0.464	0.598
Angle gain: 0.5 + Distance gain : 0.5	0.827	0.418	0.638	0.44	0.555

Bảng 4.12: Bảng kết quả thực nghiệm trên mô hình YOLOv9-e kết hợp với Network B và sử dụng hàm Warp Loss với các thông số khác nhau.

RepNLSKELAN4: 22-25-28 RepNDCNELAN4: 9 Warp Loss					
Classes	Precision	Recall	mAP _{0.5}	mAP _{.5-.95}	F1-score
Bus	0.923	0.629	0.785	0.623	0.7482
Bike	0.86	0.422	0.656	0.373	0.5661
Car	0.917	0.631	0.792	0.558	0.7476
Pedestrian	0.757	0.223	0.48	0.241	0.3445
Truck	0.91	0.372	0.649	0.523	0.5281
All	0.874	0.455	0.672	0.464	0.598

Bảng 4.13: Kết quả của network B sử dụng Warp Loss

Từ kết quả của các bảng trên, chúng tôi nhận thấy rằng:

1. Bảng 4.9 và 4.10: Mô hình YOLOv9-e kết hợp với các kiến trúc RepNDCNELAN4 và RepNLSKELAN4 và sử dụng hàm Warp Loss đã đạt được các kết quả tương đối tốt, nhưng chưa phải là tối ưu nhất.
2. Bảng 4.12: Mô hình YOLOv9-e kết hợp với cả RepNLSKELAN4 và RepNDCNELAN4 và sử dụng hàm Warp Loss đã cho kết quả tốt hơn so với các mô hình trước đó. Trong đó, phiên bản Network B đạt hiệu suất tốt nhất với Precision là 0.874, Recall là 0.455, mAP_{0.5} là 0.672, mAP_{.5-.95} là 0.464 và F1-score là 0.598.
3. Bảng 4.13: Khi điều chỉnh các siêu tham số của hàm Warp Loss trên mô hình Network B, chúng tôi nhận thấy rằng phiên bản với Angle gain: 0.7 và Distance gain: 0.3 đã đạt hiệu suất tốt nhất, giữ nguyên Precision và Recall so với phiên bản ban đầu, nhưng mAP_{0.5} và mAP_{.5-.95} đều được cải thiện nhẹ.
4. Bảng 4.13: Phân tích chi tiết theo từng lớp đối tượng cho thấy rằng mô hình đạt hiệu suất cải thiện đáng kể so với mô hình cơ sở ở tất cả các lớp, trong đó sự cải thiện rõ ràn nhất là trên lớp Bus với Precision là 0.923 và mAP_{0.5} là 0.785, trong khi đó lớp Pedestrian có sự giảm nhẹ hiệu suất với Precision là 0.757 và mAP_{0.5} là 0.48.

Kết luận, mô hình Network B sử dụng Warp Loss đã đạt hiệu suất cao vượt trội so với tất cả mô hình thử nghiệm còn lại và cả mô hình tốt nhất trong bài báo gốc [1] được huấn luyện với ảnh 640 × 640 là YOLOv8x. Mô hình Network B sử dụng Warp Loss vượt trội hoàn toàn ở toàn bộ chỉ số so với YOLOv8x.

YOLOv8x-640×640					
Classes	Precision	Recall	mAP _{0.5}	mAP _{.5-.95}	F1-score
Bus	0.9331	0.4796	0.7156	0.5419	0.6335
Bike	0.8035	0.377	0.6062	0.3208	0.5132
Car	0.9493	0.5331	0.749	0.5208	0.6827
Pedestrian	0.7785	0.1402	0.4596	0.2168	0.2376
Truck	0.7444	0.3028	0.5424	0.4141	0.4304
All	0.8418	0.3665	0.6146	0.4029	0.5106

Bảng 4.14: Kết quả mô hình YOLOv8x trên ảnh đầu vào 640 × 640.

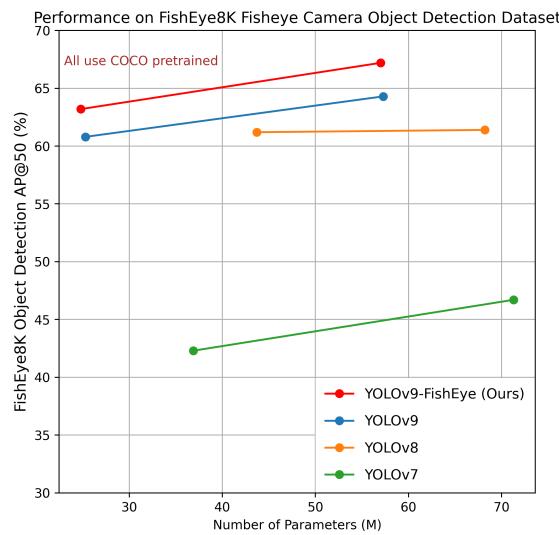
Bên cạnh đó, chúng tôi cũng đã tiến hành thử nghiệm các đề xuất và tích hợp vào mô hình YOLOv9-c để cung cấp thêm mô hình rút gọn dành cho các thiết bị có bộ nhớ với khả năng tính toán thấp. Cụ thể, RepNDCNELAN4 thay thế mạng Gelan (RepNCSPELAN4) ở lớp thứ 5 và RepNLSKELAN4 thay thế mạng Gelan (RepNCSPELAN4) ở lớp thứ 7 và 9, sau đó toàn bộ mạng cũng được huấn luyện với hàm mất mát tích hợp Warp Loss. Những cải tiến trên mô hình rút gọn (YOLOv9-c) này của chúng tôi được rút ra từ những thí nghiệm tương tự trên mô hình cơ bản (YOLOv9-e), do đó quá trình thử nghiệm và tích hợp các mạng trên từng layer được diễn ra khá thuận lợi. Các thí nghiệm trên mô hình rút gọn của chúng tôi cũng đạt được kết quả vượt trội so với mô hình rút gọn cơ sở (YOLOv9-c) nhằm phần nào khẳng định các đề xuất cải tiến cấu trúc các mạng và hàm mất mát của chúng tôi trên YOLOv9 cho hình ảnh từ camera fisheye là hiệu quả và có tính xác thực cao.

YOLOv9-c-640					
Classes	Precision	Recall	mAP _{0.5}	mAP _{.5-.95}	F1-score
Bus	0.739	0.662	0.758	0.567	0.698
Bike	0.844	0.386	0.628	0.341	0.529
Car	0.897	0.588	0.763	0.53	0.71
Pedestrian	0.722	0.143	0.433	0.23	0.239
Truck	0.745	0.342	0.577	0.475	0.469
All	0.79	0.424	0.632	0.429	0.552

Bảng 4.15: Kết quả mô hình rút gọn đề xuất với Warp Loss.

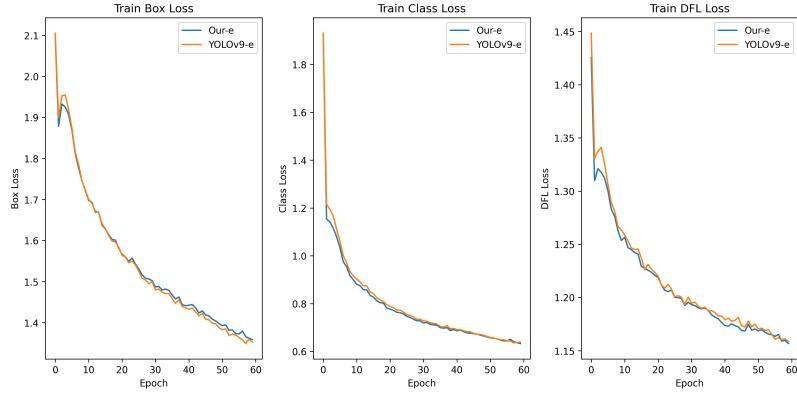
Hiệu năng của mô hình rút gọn đề xuất của chúng tôi cải thiện đáng kể so với mô hình rút gọn cơ sở YOLOv9-c (+2.4% mAP_{0.5} và +1.7% mAP_{.5-.95}), bên cạnh đó cũng mang lại hiệu năng tốt hơn đáng kể các mô hình với hình ảnh đầu vào có kích thước 640x640 trong thí nghiệm ở [1].

Sự cải thiện rõ rệt ở các mô hình đề xuất khi kết hợp RepNDCNELAN4, RepNLSKELAN4 và Warp Loss dễ quan sát nhất là chỉ số ở lớp Pedestrian có sự cải thiện đáng kể, biểu thị cho việc các thay đổi có ảnh hưởng tích cực đến các đối tượng nhỏ, ở rìa ảnh và khó phát hiện. Đồng thời, các lớp khác cũng nhận được sự cải thiện hiệu năng đáng kể qua các chỉ số ở tập thử nghiệm của mô hình đề xuất cơ bản và rút gọn khi so với vô cùng cơ sở.



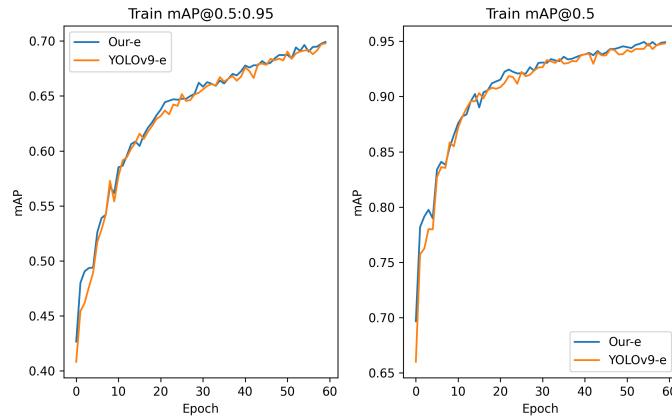
Hình 4.4: Hiệu năng của mô hình và phương pháp đề xuất của chúng tôi với các mô hình SOTA

Bên cạnh việc quan sát các điểm số trên tập thử nghiệm (test set), chúng tôi còn thực hiện các quan sát trên các hàm loss và các điểm số trên tập huấn luyện trong quá trình training để có cái nhìn trực quan hơn về sự tối ưu cũng như theo dõi sự ổn định của mô hình đề xuất khi so với mô hình cơ sở. Các giá trị trên các hàm loss nhận được từ mô hình đề xuất của chúng tôi có biểu đồ khá tương tự với giá trị trên mô hình cơ sở tuy nhiên giảm ổn định và có phần nhỉnh hơn trong quá trình huấn luyện nhờ vào sự kết hợp của Warp Loss với các Loss Function khác, giúp các Loss Function trong YOLOv9 tương tác và hỗ trợ lẫn nhau một cách tốt hơn.



Hình 4.5: Giá trị loss của mô hình YOLOv9-e so với mô hình đề xuất cỡ lớn.

Điểm số $mAP_{0.5}$ và $mAP_{.5-.95}$ trên mô hình đề xuất có sự vượt trội nhẹ cho thấy khả năng thích nghi của mô hình trên các dữ liệu phức tạp là cao hơn so với mô hình cơ sở và phần nào cho thấy các điểm số vượt trội nhận được trên tập thử nghiệm của mô hình đề xuất là hoàn toàn có cơ sở và hợp lý. Bên cạnh đó, biểu đồ điểm số $mAP_{0.5}$ và $mAP_{.5-.95}$ trên tập huấn luyện của mô hình đề xuất hội tụ và tăng ổn định hơn trong cả quá trình huấn luyện, phần nào giúp chúng tôi dễ dàng quan sát và theo dõi hơn khi so với biểu đồ của mô hình cơ sở.



Hình 4.6: Điểm số MAP của mô hình YOLOv9-e so với mô hình đề xuất cỡ lớn.

4.4 Mô hình đề xuất có hiệu năng tốt nhất

4.4.1 Hiệu năng các mô hình đề xuất với SOTA

Model	Size	Params	GFLOPs	mAP _{0.5}	mAP _{0.5-.95}	F1 _{score}
YOLOv5l6	1280 × 1280	76.8M	445.6G	0.6139	0.4098	0.535
YOLOv5x6	1280 × 1280	140.7M	839.2G	0.6387	0.4268	0.5588
YOLOR-W6	1280 × 1280	79.8M	453.2G	0.6466	0.4442	0.5899
YOLOR-P6	1280 × 1280	37.2M	325.6G	0.6632	0.4406	0.6111
YOLOv7-D6	1280 × 1280	154.7M	806.8G	0.3977	0.2633	0.5197
YOLOv7-E6E	1280 × 1280	151.7M	843.2G	0.5081	0.3265	0.6294
YOLOv7	640 × 640	36.9M	104.7G	0.6139	0.4098	0.535
YOLOv7-X	640 × 640	71.3M	189.9G	0.4235	0.2473	0.5453
YOLOv8l	640 × 640	43.7M	165.2G	0.612	0.4012	0.5187
YOLOv8x	640 × 640	68.2M	257.8G	0.6146	0.4029	0.5106
YOLOv9-C	640 × 640	25.3M	102.1G	0.608	0.412	0.537
YOLOv9-E	640 × 640	57.3M	189.0G	0.643	0.441	0.567
Our-C	640 × 640	24.7M	101.4G	0.632	0.429	0.551
Our-E	640 × 640	57.0M	186.3G	0.672	0.464	0.598

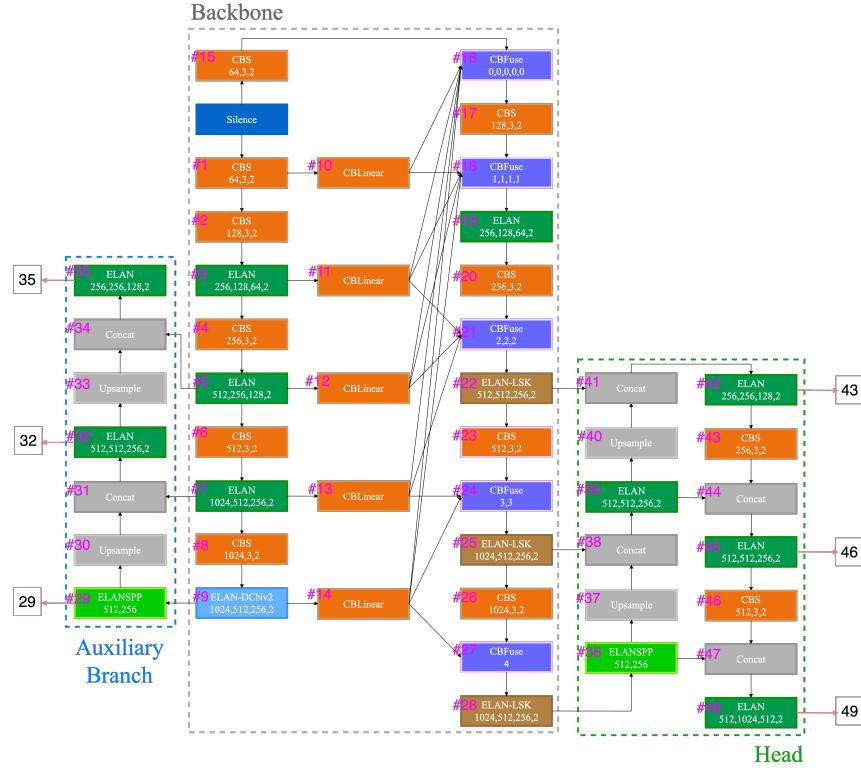
Bảng 4.16: Bảng kết quả hiệu năng các mô hình đề xuất với SOTA.

Như vậy, từ các quả của các thí nghiệm trên chúng tôi đã chọn lọc được mô hình đề xuất có hiệu năng tốt nhất cho cả phiên bản cỡ lớn (E-Extra large) với mAP@50 là 67.2% và phiên bản rút gọn (C-Compact) với mAP@50 là 63.2%.

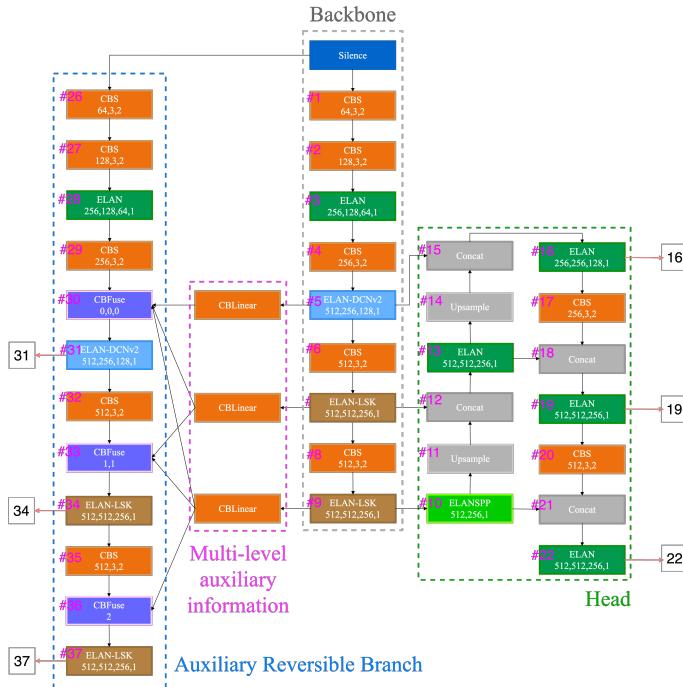
Ngoài việc các mô hình đề xuất cải thiện hiệu năng dựa trên mô hình cơ sở, các mô hình đề xuất của chúng tôi cũng đồng thời có số lượng tham số (Param) và phép tính toán (GFLOPs) thấp hơn giúp mô hình gọn nhẹ và tối ưu dung lượng bộ nhớ cũng như điện năng tiêu thụ khi thực hiện các suy luận hơn so với mô hình cơ sở.

4.4.2 Kiến trúc mô hình đề xuất có hiệu năng tốt nhất

Kiến trúc mô hình đề xuất cỡ lớn (Our-E) và rút gọn (Our-C) trong 4.16 có kết quả tốt nhất sau khi đã được chúng tôi tinh chỉnh và thử nghiệm được mô tả bằng các hình sau đây:



Hình 4.7: Kiến trúc mô hình cỡ lớn của chúng tôi dựa trên YOLOv9-e 2.23

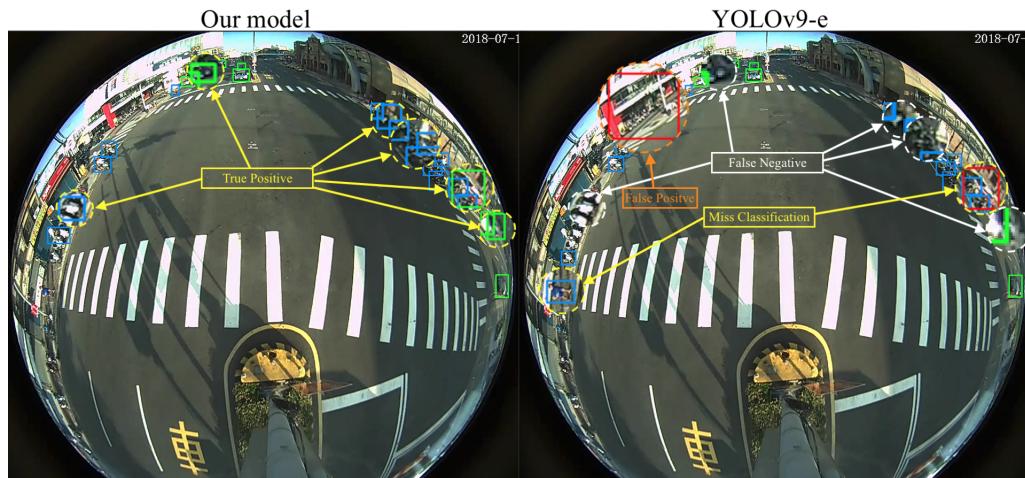


Hình 4.8: Kiến trúc mô hình rút gọn của chúng tôi dựa trên YOLOv9-c 2.24

4.4.3 Trực quan hoá và so sánh các kết quả trên tập thử nghiệm

Khi đã thu được kết quả đánh giá từ mô hình cơ sở và các mô hình đề xuất trên tập thử nghiệm (test set), chúng tôi đã lựa chọn NetworkB trong 4.13 có các số liệu đo lường tốt nhất để trực quan hoá các suy luận của mô hình cũng như so sánh với suy luận của mô hình cơ sở nhằm mang lại các phân tích ưu-nhược điểm của các mô hình một cách trực quan và rõ ràng hơn.

Từ các hình ảnh thu được qua trực quan hoá các suy luận của mô hình đề xuất và mô hình cơ sở, chúng tôi nhận thấy rằng mô hình đề xuất có khả năng đưa ra các suy luận chính xác hơn và bao phủ được nhiều vật thể của các lớp hơn so với mô hình cơ sở, đặc biệt là các vật thể nằm ở vị trí bị biến dạng nhiều do đặc tính của camera fisheye như giả thuyết chúng tôi đã đặt ra để cải thiện ở hàm mắt mè và các phương pháp cải tiến mô hình đã đề xuất. Trực quan hoá suy luận của 2 mô hình được thể hiện qua một ví dụ lấy từ ảnh trong tập thử nghiệm (test set) của FishEye8K [1] dưới đây:

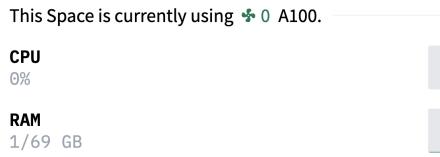


Hình 4.9: Trực quan hoá suy luận của 2 mô hình

4.5 Triển khai ứng dụng mô hình

4.5.1 Môi trường triển khai ứng dụng

Trong quá trình nghiên cứu và triển khai mô hình nhận dạng đối tượng trong giám sát giao thông thời gian thực, một trong những thách thức lớn là yêu cầu khối lượng tính toán và dung lượng bộ nhớ rất lớn, đòi hỏi sự hỗ trợ từ các GPU mạnh mẽ. Để giải quyết vấn đề này, chúng tôi đã sử dụng môi trường thử nghiệm được cung cấp bởi HuggingFace Space. Môi trường này được cung cấp đồng hỗ trợ với cấu hình mạnh mẽ gồm 12 vCPU, 69GB RAM và GPU Nvidia A100-40GB.

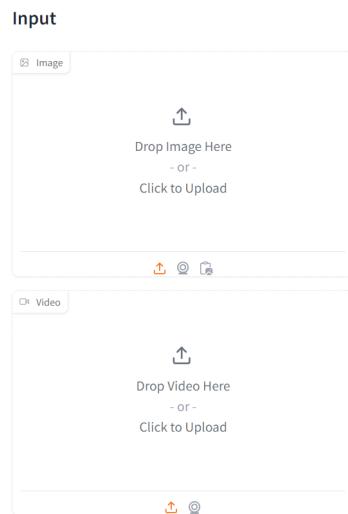


Hình 4.10: Cấu hình sử dụng trên HuggingFace Space

Với sự trợ giúp của HunggingFace Space, ứng dụng của chúng tôi đã được triển khai lên trang web nhằm cung cấp các suy luận trực tuyến cho người dùng tại địa chỉ <https://k20hcmus-fisheye8k.hf.space>.

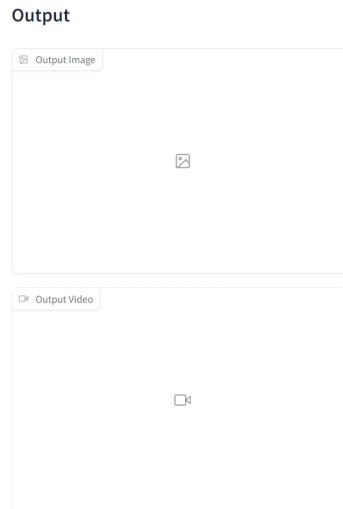
4.5.2 Thiết kế giao diện người dùng

Phần đầu vào (Input) của ứng dụng được thiết kế linh hoạt và tiện lợi, cho phép người dùng cuối dễ dàng nhập dữ liệu theo hai dạng: hình ảnh hoặc video, hình 4.11



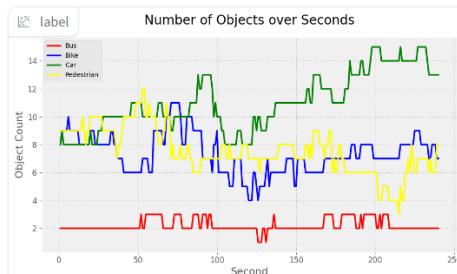
Hình 4.11: Phần dữ liệu đầu vào

Phần đầu ra (Output) hình 4.12 cho ra kết quả tương tự với loại dữ liệu đầu vào. Nếu là ảnh thì kết quả sẽ hiện trong khung đầu ra ảnh, nếu là video thì kết quả sẽ hiện trong khung đầu ra video.



Hình 4.12: Kết quả đầu ra sau khi áp dụng nhận dạng đối tượng

Phần đầu ra còn bao gồm biểu đồ trực quan về số lượng của từng lớp nhận dạng được. Đối với dữ liệu là ảnh, số lượng của từng lớp nhận dạng được sẽ được trực quan hóa thông qua biểu đồ cột (bar plot), hình 4.10. Biểu đồ này cho phép người dùng dễ dàng so sánh số lượng của các lớp khác nhau một cách trực quan và hiệu quả. Đối với dữ liệu là video, số lượng của từng lớp nhận dạng được sẽ được trực quan hóa thông qua biểu đồ đa đường (multiple line plot), hình 4.13. Biểu đồ này giúp người dùng theo dõi sự thay đổi của các lớp qua thời gian, cung cấp một cái nhìn toàn cảnh và chi tiết về quá trình phân tích video.

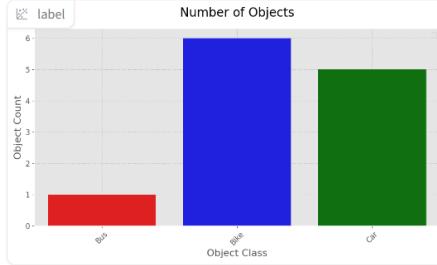


Hình 4.13: Ví dụ về biểu đồ đa đường trực quan cho dữ liệu là video

Biểu đồ này giúp người dùng theo dõi sự thay đổi của các lớp qua thời gian, cung cấp một cái nhìn toàn cảnh và chi tiết về quá trình phân tích video. Màu sắc của từng lớp trong các biểu đồ được thiết kế dựa theo màu của các lớp được mô tả trong bài báo [1], hình 4.15, đảm bảo tính nhất quán và dễ nhận diện.

Phần cấu hình (Configuration) hình 4.16 cho mô hình nhận dạng sẽ bao gồm tên mô hình sử dụng để tiến hành nhận dạng và thuật toán tracking có thể tùy chỉnh để phù hợp với nhu cầu nhận dạng của người dùng.

Cùng với đó chúng tôi cũng cung cấp hai phần dữ liệu ví dụ để người dùng có thể truy cập và tương tác với tính năng nhận diện camera mắt cá một cách dễ dàng cũng như hiểu rõ hơn về phương

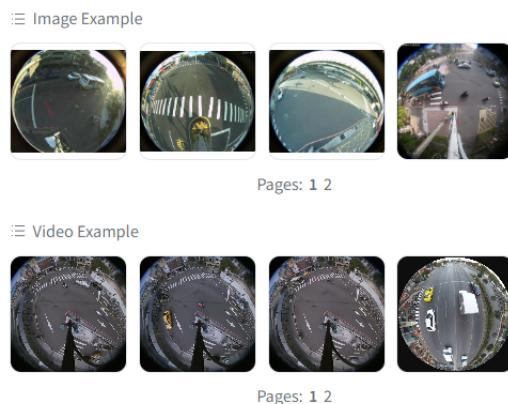


Hình 4.14: Ví dụ về biểu đồ cột trực quan cho dữ liệu là hình ảnh

■ Bus ■ Bike ■ Car ■ Pedestrian ■ Truck

Hình 4.15: Màu trực quan của từng lớp trong tập dữ liệu.

thức hoạt động của mô hình và thuật toán tracking trong các tình huống thực tế. Phần ví dụ gồm hai phần, phần hình ảnh và phần video. Các ví dụ đều được chúng tôi chắt lọc và trích xuất chi tiết để có thể tổng quát hóa lên hết sự khác biệt giữa các mô hình và thuật toán tracking, hình 4.17.



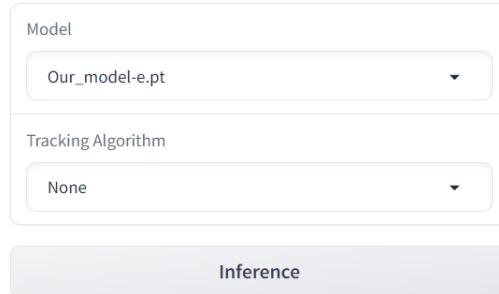
Hình 4.17: Các ví dụ trong ứng dụng.

Thời gian thực hiện suy luận của mỗi thuật toán tracking trên ứng dụng là khác nhau. Thời gian suy luận từ nhanh đến chậm của các thuật toán sẽ được sắp xếp như sau: Không sử dụng thuật toán tracking, sử dụng DeepSort, sử dụng StrongSort.

4.5.3 Thử nghiệm với dữ liệu thực tế

Phần ví dụ (example) trong ứng dụng được trích xuất từ ảnh trên tập thử nghiệm (test set) từ bộ liệu FishEye8K [3.1] và video từ R0 Fish Len Dataset Center Point [35]. Các video ví dụ được thu ngắn lại với thời lượng là 10 giây trong những lúc hoạt động giao thông đông đúc nhất để có thể đánh giá trực quan nhất về độ chính xác và ổn định của mô hình được chúng tôi đề xuất trong ứng dụng.

Configuration



Hình 4.16: Phần cài đặt mô hình và thuật toán tracking

Kết quả thu được từ ứng dụng với mô hình đề xuất được tích hợp thuật toán theo dõi đã cải thiện độ ổn định đáng kể khi các phương tiện bắt đầu thay đổi vị trí hoặc bị mờ vì di chuyển với tốc độ nhanh và bị che khuất một phần bởi các phương tiện khác.

4.5.4 Tốc độ phản hồi và đáp ứng của ứng dụng

Với môi trường triển khai trên nền tảng GPU, tốc độ suy luận trên các mô hình của chúng tôi được đáp ứng một cách tức thời và hình ảnh đầu ra từ mô hình được tải lên trên giao diện người dùng một cách nhanh chóng. Với tốc độ suy luận trên từng hình ảnh của mô hình phiên bản cỡ lớn (E-Extra large) là 34 ms tương ứng với 30 khung hình trên giây (fps) còn đối với phiên bản rút gọn (C-Compact) thời gian để mô hình đưa ra được suy luận là 17 ms tương ứng với 58 khung hình trên giây (fps).



Hình 4.18: Thời gian thực hiện các suy luận trên ứng dụng

Sau khi kết thúc việc thực hiện các suy luận, hệ thống sẽ tải lên hình ảnh hoặc video đầu ra từ mô hình, thuật toán theo dõi (nếu có) lên ứng dụng của chúng tôi. Thời gian tải kết quả lên ứng dụng còn tuỳ thuộc vào tốc độ và băng thông mạng được nền tảng Hugging Face cung cấp cho Space của chúng tôi nên khoảng thời gian này là không cố định, tuy nhiên việc này cũng không làm tăng đáng kể thời gian phản hồi của ứng dụng do hình ảnh và video có dung lượng không đáng kể so với trung bình tốc độ tải lên được cung cấp.

Chương 5

Kết luận

5.1 Kết luận chung

Trong bối cảnh đô thị hóa ngày càng phát triển, việc giám sát và quản lý giao thông một cách hiệu quả trở thành nhu cầu cấp thiết. Nghiên cứu này đã khám phá và áp dụng công nghệ tiên tiến trong lĩnh vực thị giác máy tính nhằm giải quyết các thách thức trong giám sát giao thông bằng camera mắt cá.

Bắt đầu từ việc xác định các yêu cầu và thách thức của bài toán, chúng tôi đã tiến hành phân tích và so sánh các mô hình hiện có, cuối cùng lựa chọn YOLOv9 làm nền tảng cho nghiên cứu. Với sự hướng dẫn tận tình của Tiến sĩ Huỳnh Thế Đăng và sự nỗ lực không ngừng nghỉ của nhóm nghiên cứu, chúng tôi đã thành công trong việc cải tiến mô hình này. Bằng cách tích hợp các cơ chế như RepNDCNELAN4, RepNLSKELAN4 và hàm Warp Loss, mô hình đã thể hiện hiệu suất vượt trội, khắc phục được nhiều hạn chế của các phương pháp trước đó.

Các kết quả thu được không chỉ chứng minh tính khả thi và hiệu quả của mô hình trong việc giám sát giao thông mà còn mở ra nhiều hướng nghiên cứu mới. Chúng tôi đã đạt được những bước tiến quan trọng trong việc nâng cao độ chính xác và tốc độ của mô hình phát hiện đối tượng, đồng thời tối ưu hóa khả năng theo dõi trong các điều kiện môi trường phức tạp của camera mắt cá. Những phát hiện này có thể được áp dụng rộng rãi trong thực tiễn, từ việc giám sát vi phạm giao thông đến quản lý lưu lượng phương tiện, góp phần quan trọng vào việc nâng cao chất lượng và an toàn giao thông đô thị.

Nhìn chung, nghiên cứu này không chỉ đóng góp vào kho tàng tri thức về thị giác máy tính và phát hiện đối tượng, mà còn khẳng định tiềm năng ứng dụng mạnh mẽ của công nghệ trong các hệ thống giám sát giao thông hiện đại. Chúng tôi hy vọng rằng, những kết quả và phương pháp mà chúng tôi đã phát triển sẽ trở thành nền tảng cho các nghiên cứu và ứng dụng tiếp theo, tiếp tục cải tiến và hoàn thiện để đáp ứng tốt hơn nhu cầu của thực tế.

Cuối cùng, chúng tôi xin gửi lời cảm ơn chân thành đến các thầy cô, bạn bè và gia đình đã luôn đồng hành và hỗ trợ chúng tôi trong suốt quá trình nghiên cứu. Sự động viên và góp ý quý báu của mọi người chính là động lực to lớn để chúng tôi hoàn thành luận văn này. Chúng tôi mong rằng nghiên cứu này sẽ nhận được sự đánh giá cao từ quý thầy cô và các bạn đồng nghiên cứu, đồng thời rất mong nhận được các ý kiến đóng góp để có thể tiếp tục hoàn thiện hơn trong tương lai.

5.2 Hướng phát triển

Một trong những yếu tố quan trọng cần được chú ý và phát triển là bộ dữ liệu FishEye8K. Bộ dữ liệu hiện tại có số lượng phân lớp đối tượng không đồng đều và sự mất cân bằng đáng kể giữa các

lớp. Đồng thời, số lượng lớp đối tượng trong bộ dữ liệu này còn khá ít (chỉ có 5 lớp đối tượng). Để giải quyết vấn đề này, việc thu thập và kết hợp thêm các nguồn dữ liệu để có được bộ dữ liệu tổng quát hơn, bao gồm nhiều lớp đối tượng tương tự như MS COCO [21], là cần thiết. Hiện nay, nguồn dữ liệu dành cho các mô hình nhận dạng vật thể qua camera fisheye còn khá khan hiếm, vì vậy, mở rộng và đa dạng hóa bộ dữ liệu sẽ giúp tăng độ chính xác và hiệu quả của các mô hình.

Một hướng phát triển khác là cải thiện việc xử lý hình ảnh từ ống kính fisheye. Ví dụ, làm rõ nét hình ảnh được cung cấp từ camera fisheye bằng các phương pháp như super-resolution và image deblurring trước khi đưa vào các mô hình phát hiện vật thể. Việc này sẽ giúp các mô hình nhận dạng vật thể hoạt động chính xác và hiệu quả hơn. Cụ thể, hình ảnh được xử lý rõ nét sẽ giúp giảm thiểu sự biến dạng và mất chi tiết, từ đó hỗ trợ các mô hình trong việc phát hiện và nhận dạng đối tượng một cách chính xác hơn.

Mặc dù trong luận văn này chúng tôi đã tích hợp hai thuật toán theo dõi là DeepSORT và StrongSORT, nhưng vẫn chưa đưa ra giải pháp tối ưu cho các thuật toán này dành cho mô hình đề xuất của chúng tôi và tổng quát hơn là theo dõi vật thể trên camera fisheye. Do đó, tối ưu các tham số và phát triển một thuật toán theo dõi chuyên biệt cho camera fisheye là một hướng đi cần thiết và quan trọng trong tương lai mà chúng tôi cần tập trung nghiên cứu và phát triển. Việc phát triển thuật toán theo dõi hiệu quả sẽ giúp cải thiện khả năng theo dõi liên tục và chính xác các đối tượng, đặc biệt là trong các điều kiện môi trường phức tạp và thay đổi.

Ngay tại thời điểm hoàn thành luận văn này, mô hình YOLOv10 [36] đã được công bố. Bằng cách loại bỏ triệt tiêu không tối đa (NMS) và tối ưu hóa các thành phần mô hình khác nhau, YOLOv10 đạt được hiệu suất hiện đại với chi phí tính toán giảm đáng kể. Đây là một hướng nghiên cứu đột phá do các nhà khoa học từ Đại học Thanh Hoa đề xuất. Những bước phát triển tiếp theo của chúng tôi sẽ được truyền cảm hứng từ mô hình tiên tiến này, nhằm tiếp tục cải tiến và nâng cao hiệu quả của các hệ thống giám sát giao thông.

Chúng tôi tin rằng việc áp dụng các công nghệ và mô hình mới như YOLOv10 vào nghiên cứu của mình sẽ mở ra nhiều cơ hội và thách thức mới. Sự tiến bộ không ngừng trong lĩnh vực học sâu và thị giác máy tính sẽ tiếp tục mang đến những giải pháp hiệu quả và đột phá cho bài toán phát hiện và nhận dạng đối tượng trong giám sát giao thông, góp phần quan trọng vào việc xây dựng các hệ thống giao thông thông minh và an toàn hơn trong tương lai.

Tài liệu tham khảo

Tiếng Anh

- [1] Munkhjargal Gochoo et al. “FishEye8K: a benchmark and dataset for fisheye camera object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 5304–5312.
- [2] Imran H. Sarker. “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions”. In: *SN Computer Science* 2.6 (2021). Epub 2021 Aug 18, p. 420. DOI: [10.1007/s42979-021-00815-1](https://doi.org/10.1007/s42979-021-00815-1).
- [3] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. “Learning non-maximum suppression”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4507–4515.
- [4] Ross Girshick et al. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587.
- [5] Kaiming He et al. “Mask r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.
- [6] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Advances in Neural Information Processing Systems* 28 (2015).
- [7] Joseph Redmon et al. In: *You Only Look Once: Unified, Real-Time Object Detection*. 2016, pp. 779–788.
- [8] Joseph Redmon and Ali Farhadi. “You Only Look Once: Unified, Real-Time Object Detection”. In: 2016, pp. 779–788.
- [9] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).
- [10] Tsung-Yi Lin et al. “Feature pyramid networks for object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.
- [11] Kaiming He et al. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.
- [12] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [13] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “YOLOv4: Optimal Speed and Accuracy of Object Detection”. In: *arXiv preprint arXiv:2004.10934* (2020).

- [14] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: [10.1109/CVPR.2009.5206848](https://doi.org/10.1109/CVPR.2009.5206848).
- [15] Chien-Yao Wang et al. “CSPNet: A new backbone that can enhance learning capability of CNN”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 390–391.
- [16] Shu Liu et al. “Path aggregation network for instance segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8759–8768.
- [17] Ultralytics LLC. *YOLOv5: A State-of-the-Art Object Detection Model*. <https://github.com/ultralytics/yolov5>. Accessed: April 29, 2024. 2021.
- [18] Chuiyi Li et al. “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications”. In: *arXiv preprint arXiv:2209.02976* (2022).
- [19] Xiaohan Ding et al. “Repvgg: Making vgg-style convnets great again”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 13733–13742.
- [20] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors”. In: *arXiv preprint arXiv:2207.02696* (2022).
- [21] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: [1405.0312 \[cs.CV\]](https://arxiv.org/abs/1405.0312).
- [22] Xindong Zhang et al. “Efficient long-range attention network for image super-resolution”. In: *European conference on computer vision*. Springer. 2022, pp. 649–667.
- [23] Chien-Yao Wang, Hong-Yuan Mark Liao, and I-Hau Yeh. “Designing network design strategies through gradient path analysis”. In: *arXiv preprint arXiv:2211.04800* (2022).
- [24] Glenn Jocher, Ayush Chaurasia, and Jian Qiu. *YOLO by Ultralytics*. <https://github.com/ultralytics/>. Accessed: February 30, 2023. 2023.
- [25] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. “YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information”. In: *arXiv preprint arXiv:2402.13616* (2024).
- [26] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. “Simple online and realtime tracking with a deep association metric”. In: *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017, pp. 3645–3649.
- [27] Yunhao Du et al. “Strongsort: Make deepsort great again”. In: *IEEE Transactions on Multimedia* (2023).
- [28] Yuxuan Li et al. “Large Selective Kernel Network for Remote Sensing Object Detection”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 16794–16805.

- [29] Chien-Yao Wang et al. “Enriching Variety of Layer-Wise Learning Information by Gradient Combination”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 2477–2484. DOI: [10.1109/ICCVW.2019.00303](https://doi.org/10.1109/ICCVW.2019.00303).
- [30] Jifeng Dai et al. “Deformable Convolutional Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 764–773.
- [31] Degang Yang et al. “PGDS-YOLOv8s: An Improved YOLOv8s Model for Object Detection in Fisheye Images”. In: *Applied Sciences* 14.1 (2024), p. 44. DOI: [10.3390/app14010044](https://doi.org/10.3390/app14010044). URL: <https://doi.org/10.3390/app14010044>.
- [32] Xiang Li et al. *Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection*. 2020. arXiv: [2006.04388 \[cs.CV\]](https://arxiv.org/abs/2006.04388).
- [33] Zhaohui Zheng et al. “Enhancing geometric factors in model learning and inference for object detection and instance segmentation”. In: *IEEE transactions on cybernetics* 52.8 (2021), pp. 8574–8586.
- [34] Zhaohui Zheng et al. “Distance-IoU loss: Faster and better learning for bounding box regression”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 07. 2020, pp. 12993–13000.
- [35] Wei Jan Ke, WEI-YU CHEN, and CHEN-KAI Sun. *fishlen traffic image dataset with center point annotation*. 2022. DOI: [10.21227/ksb2-ve17](https://doi.org/10.21227/ksb2-ve17). URL: <https://dx.doi.org/10.21227/ksb2-ve17>.
- [36] Ao Wang et al. “YOLOv10: Real-Time End-to-End Object Detection”. In: *arXiv preprint arXiv:2405.14458* (2024).