

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Đỗ Văn Hiệp - Nguyễn Hữu Thắng-Nguyễn
Minh Tâm

THỰC TẬP DỰ ÁN TỐT NGHIỆP

ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Tp. Hồ Chí Minh, tháng 3/2022

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Đỗ Văn Hiệp - 1712429

Nguyễn Hữu Thắng - 1712756

Nguyễn Minh Tâm - 1712746

Dò tìm đối tượng từ camera giám sát sử
dụng mô hình học sâu

ĐỒ ÁN TỐT NGHIỆP CỦ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

GIÁO VIÊN HƯỚNG DẪN

PGS.TS. Lê Hoàng Thái

Tp. Hồ Chí Minh, tháng 3/2022

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Tp Hồ Chí Minh, Ngày.... Tháng 3 Năm 2022.

Giảng viên hướng dẫn

NHẬN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN

Tp Hồ Chí Minh, Ngày.... Tháng 3 Năm 2022.

Giảng viên phản biện

Lời cảm ơn

Lời đầu tiên chúng em xin gửi lời cảm ơn chân thành đến PGS.TS Lê Hoàng Thái đã hỗ trợ và hướng dẫn chúng em thực hiện đề tài thực tập dự án tốt nghiệp này. Trong quá trình làm chúng em đã gặp phải rất nhiều những khó khăn những vướng mắc nhưng thầy luôn tận tình giải đáp những thắc mắc của chúng em, luôn quan tâm giúp đỡ tận tình, tâm huyết, kể cả là những thắc mắc nhỏ. Qua đó giúp chúng em có thể hiểu sâu hơn và rộng hơn về chủ đề được đề cập tới trong đề tài này. Qua đó giúp chúng em từng bước hoàn thành một cách tốt nhất đề tài tốt nghiệp này.

Do chưa có kinh nghiệm nghiên cứu cũng như là thiếu sót hạn chế về mặt kiến thức nên đôi chỗ có thể sai sót những chỗ chưa hợp lý hoặc là chưa đúng. Chúng em rất mong nhận được sự nhận xét, ý kiến đóng góp phê bình từ thầy cô để chúng em có thể hoàn thiện tốt hơn cũng như là giúp ích cho sau này.

Lời cuối cùng chúng em xin kính chúc thầy, cô thật nhiều sức khỏe, thành công trong công việc, cuộc sống cũng như là sự nghiệp trồm người.

Chúng xem xin chân thành cảm ơn.

Tp Hồ Chí Minh, Ngày.... Tháng 3 Năm 2022.

Nhóm sinh viên thực hiện
Đỗ Văn Hiệp -Nguyễn Hữu Thắng-
Nguyễn Minh Tâm

Đề cương chi tiết

Tên đề tài : Dò tìm đối tượng từ camera giám sát sử dụng
Giáo viên hướng dẫn :PST.TS Lê Hoàng Thái
Thời gian thực hiện: Tháng 9/2021 đến 25/2/2022
Nhóm sinh viên thực hiện:Nguyễn Hữu Thắng 1712756- Đỗ Văn Hiệp 1712429-Nguyễn Minh Tâm 1712746
Loại đề tài : Nghiên cứu
<p>Nội dung đề tài:</p> <ul style="list-style-type: none"> * Khảo sát các nghiên cứu sử dụng mạng học sâu cho phát hiện đối tượng * Các công nghệ và các mô hình chúng tôi sử dụng * Tiền xử lý dữ liệu và tăng cường hình ảnh * Kết quả thực nghiệm * Kết Luận và Phát triển
<p>Kế hoạch thực hiện:</p> <ul style="list-style-type: none"> * Tháng 9/2021- 10/2021: Tìm hiểu đề tài, tìm nguồn tài liệu tham khảo về các kiến trúc nhận dạng đối tượng Faster R-CNN, Yolov3, EffieciNet. * Tháng 10/2021 -11/2021: Thực hiện tiền xử lý dữ liệu , tăng cường dữ liệu cho các bộ dữ liệu. * Tháng 11/2021-12/2021: Thực hiện trích xuất khung chứa và phân lớp đối tượng trên bộ dữ liệu ENA-24 detection. Sau đó đánh giá kết quả thực nghiệm. * Tháng 1/2022- 2/2022: Thực hiện trích xuất khung chứa và phân lớp đối tượng trên bộ dữ liệu Citycam. Sau đó đánh giá kết quả thực nghiệm. * Tháng 2/2022-3/2022: Viết báo cáo và chuẩn bị cho bảo vệ đề tài.

Thiếu phương pháp đề xuất.

Bảng tra cứu các thuật ngữ

Anchor box	khung neo
Area of overlap	diện tích giao nhau
Area of union	diện tích phần hợp
Bounding box	khung chứa
Batch Normalization (BN)	Batch Normalization thực hiện việc chuẩn hóa (normalizing) và trừ cho trung bình mẫu dữ liệu trước khi đưa qua hàm kích hoạt
Convolutional Neural Network	mạng tích chập
Camera trap	bẫy camera
Epoch	khi chúng ta đưa hết tập dữ liệu vào huấn luyện một lần
Feature map	kết quả hiển thị sau mỗi lần trượt qua lớp đầu vào
Frame per second(FPS)	số khung hình trên giây
Ground-truth bounding box	khung chứa đúng
Gradient	vector độ biến thiên
High resolution classifier	bộ phân loại cao
Mini-batch	một số nào đó của tập dữ liệu khi người dùng chia tập dữ liệu r
Non-maxima suppression	triệt phi cực đại
Region of Interest(RoI)	vùng quan tâm
Residual block	tầng kết nối tắt
Support vector machine	máy vector hỗ trợ
Static monitoring camera	camera giám sát tĩnh
Tensor	là đối tượng hình học miêu tả quan hệ tuyến tính giữa các đại lượng vectơ, vô hướng,
Vanishing Gradient	Hiện tượng biến mất gradient: giá trị gradient sẽ tiến về 0 và bước cập nhật hệ số gradient sẽ trở nên vô nghĩa

Mục lục

Lời cảm ơn	iii
Đề cương chi tiết	iv
Mục lục	viii
Tóm tắt	xii
1 Giới thiệu đề tài	1
1.1 Bối cảnh đề tài	1
1.2 Thách thức của đề tài	1
1.2.1 Thách thức về mô hình	1
1.2.2 Thách thức về hình ảnh:	2
1.3 Phạm vi đề tài	2
1.4 Động lực nghiên cứu	3
1.5 Mục tiêu đề tài	4
1.6 Cấu trúc đề tài	4
2 Tổng quan	6
2.1 Các nghiên cứu liên quan	6
2.1.1 Khảo sát các nghiên cứu sử dụng mạng học sâu cho phát hiện đối tượng	6
3 Phương pháp	11
3.1 Các công nghệ mà mô hình sử dụng:	11
3.1.1 ResNet [6]	11
3.1.2 Faster R-CNN	13
3.1.3 Các mô hình YOLO V1,V2,V3:	18
3.1.4 YOLO V1:	18
3.1.5 YOLO V2:	19

3.1.6	YOLO V3	20
3.1.7	EfficientNet	26
3.2	Tiền xử lý dữ liệu và tăng cường ảnh:	33
3.3	Tăng cường dữ liệu:	34
3.4	Phương pháp đề xuất	38
4	Kết quả thực nghiệm	42
4.1	Giới thiệu tập dữ liệu	42
4.1.1	Ena24-LILABC [3]	42
4.1.2	Citycam-CMU [3]	43
4.2	Chi tiết quá trình thực nghiệm	43
4.2.1	Môi trường huấn luyện	43
4.2.2	Ngôn ngữ và thư viện	44
4.2.3	Hàm lỗi và độ đo chính xác:	44
4.2.4	Phương pháp đánh giá	47
4.2.5	Kết quả thu được	48
5	Kết luận và hướng phát triển	52
5.1	Kết luận	52
5.2	Hướng phát triển	52
Tài liệu tham khảo		54

Danh sách hình

1.1	Ảnh minh họa các thách thức về hình ảnh.	2
2.1	Mô hình RCNN	9
2.2	Mô hình Fast RCNN	10
3.1	Skip connection	12
3.2	Các phiên bản của resnet	12
3.3	Kiến trúc RPN	14
3.4	Rol pooling	16
3.5	Kiến trúc mạng Faster R-CNN	18
3.6	Kiến trúc YOLO V1	18
3.7	Kiến trúc YOLO V3	20
3.8	Kiến trúc darknet53	22
3.9	Kiến trúc chi tiết mô hình yolo v3	23
3.10	Tổng quan sơ qua về mô hình yolov3 spp	25
3.11	Khối spatial pyramid pooling(nguồn paper yolo v3 spp)	25
3.12	Biểu đồ so sánh các mô hình	27
3.13	Hình minh họa thí nghiệm thực hiện phép biến đổi tỷ lệ lên độ rộng, độ sâu hoặc độ phân giải của mạng ảnh hướng đến độ chính xác của mạng (EfficientNet-B0)	28
3.14	Bảng mô tả kiến trúc EfficientNet-B0	31
3.15	Hình minh họa khối MBConv3(k5x5)	32
3.16	Minh họa về Histogram Equalization	34
3.17	Hình ảnh khi chưa sử dụng cutout	35
3.18	Hình ảnh khi sử dụng cutout	36
3.19	Minh họa khi sử dụng mixup	37
3.20	Ảnh trước khi sử dụng grayscale	38
3.21	Ảnh sau khi sử dụng grayscale	38

3.22	Ảnh mô tả qui trình của phương pháp để xuất thực hiện đề tài	39
4.1	Biểu đồ các loại của bộ dữ liệu Ena24-LILABC	42
4.2	Biểu đồ các loại xe của bộ dữ liệu Citycam-CMU	43
4.3	Minh họa về IOU	44
4.4	Minh họa về công thức tính IOU	45
4.5	Minh họa về số liệu tính IOU	45
4.6	Các chỉ số của confusion matrix	46
4.7	Bảng kết quả	48
4.8	Ví dụ so sánh yolov3 và EfficientNet	49
4.9	So sánh yolov3 và EfficientNet	50
4.10	Ví dụ về phân lớp 2 stage yolov3 cho việc nhận dạng và EfficientNet cho việc phân lớp đối với tập dữ liệu citycam	51
4.11	Ví dụ khác về phân lớp 2 stage yolov3 cho việc nhận dạng và EfficientNet cho việc phân lớp đối với tập dữ liệu citycam	51

Danh sách bảng

4.1	Bảng so sánh	48
4.2	Bảng so sánh Yolov3 với Yolo V3 spp	49
4.3	So sánh phân lớp	49
4.4	Số liệu cụ thể việc nhận dạng trên tập citycam	50
4.5	Số liệu cụ thể việc phân lớp trên tập citycam	50

Chương 1

Giới thiệu đề tài

1.1 Bối cảnh đề tài

Trong kỉ nguyên cách mạng công nghệ, dữ liệu là nguồn tài nguyên chiến lược quan trọng. Việc thu thập và rút trích thông tin hữu ích từ dữ liệu trở thành đề tài hấp dẫn nhiều nhà nghiên cứu. Kể từ khi camera giám sát ra đời, việc thu thập dữ liệu trong thời gian dài trên quy mô lớn trở nên dễ dàng và giúp giảm thiểu nguồn nhân lực. Trong đó, camera giám sát “tĩnh” (static monitoring camera) được sử dụng rộng rãi nhưng thông tin do loại camera này thu nhận được gây nhiều khó khăn cho việc rút trích thông tin hữu ích.

Chính vì vậy, nhóm thực hiện đề tài với mong muốn ứng dụng một số phương pháp học sâu trong lĩnh vực thi giác máy tính nhằm rút trích thông tin hữu ích tự động để hỗ trợ tốt hơn cho các nhà nghiên cứu về loại camera trên.

1.2 Thách thức của đề tài

Có thể nói rằng, bài toán phát hiện đối tượng là một bài toán khó, chứa nhiều thách thức về việc phân tích cũng như xây dựng mô hình học. Nhóm thực hiện đề tài chia những thách thức này thành hai vấn đề chính:

1.2.1 Thách thức về mô hình

- Dữ liệu được dự đoán có thể có loại khác với dữ liệu được huấn luyện.
Ví dụ: Xuất hiện giống loài động vật mới chưa có trong dữ liệu được huấn luyện (đối với trường hợp bẫy camera).
- Phát hiện sai đối tượng vì nhầm lẫn với khung nền, gán nhãn sai.

- Mất cân bằng dữ liệu và số lượng ảnh trống (ảnh không chứa đối tượng) thường chiếm tỷ lệ lớn trong bộ dữ liệu.

1.2.2 Thách thức về hình ảnh:

- Chất lượng hình ảnh: ảnh có độ sáng thấp do chụp ở thời điểm thiếu sáng hoặc ban đêm, ảnh bị mờ do đối tượng chuyển động nhanh.
- Điều kiện thời tiết: điều kiện thời tiết bất lợi có thể gây cản trở ống kính như mưa, tuyết, sương mù, bụi,..
- Ảnh trống: ảnh không có đối tượng cần quan sát vì camera được kích hoạt bằng các nguyên nhân không phải đối tượng cần quan sát gây ra. Ví dụ: bẫy camera được kích hoạt bằng cảm biến chuyển động có thể chụp các ảnh do gió lay cây cối hoặc con người di chuyển.
- Khó phát hiện đối tượng: ảnh chụp đối tượng chỉ xuất hiện một phần cơ thể, bị che khuất bởi vật chắn, đối tượng nhỏ và nằm xa camera.



Hình 1.1: Ảnh minh họa các thách thức về hình ảnh.

1.3 Phạm vi đề tài

Đề tài sẽ tập trung xử lý với dữ liệu ảnh từ camera giám sát tĩnh. Camera giám sát “tĩnh” là kiểu camera giám sát được cố định tại một vị trí

và camera không tự chuyển động được. Kiểu camera này thường có mức khung hình thấp (1 FPS) vì cần thu thập dữ trong thời gian dài và hạn chế về tài nguyên xử lý và lưu trữ. Nhóm tập trung vào 2 kiểu camera đại diện cho loại camera trên:

- Camera giao thông: là kiểu camera giám sát được lắp đặt trên các tuyến đường và các giao lộ nhằm thu thập thông tin.
- Bẫy camera (camera trap): là kiểu camera giám sát được cố định tại một vị trí trong môi trường sống của động vật và có nhiều kiểu kích hoạt chế độ chụp ảnh khác nhau nhằm thích ứng với môi trường và hoạt động của các loài động vật cần quan sát.

1.4 Động lực nghiên cứu

Các phương pháp được đề xuất để giải quyết vấn đề ngày càng có hiệu suất và độ chính xác cao ra đời ngày càng nhiều. Tuy nhiên, việc áp dụng các phương pháp học sâu để nhận dạng và phân lớp tự động cho camera giám sát tĩnh thường chỉ giải quyết những vấn đề riêng biệt. Do đó, việc kết hợp các ưu điểm của các phương pháp học sâu để áp dụng những bài toán tổng quát hơn là vấn đề được các nhà nghiên cứu quan tâm.

Trong phạm vi đề tài được nêu ở phần 1.3, nhóm mong muốn xây dựng một phương pháp kết hợp một số mạng học sâu hiệu quả cho nhận dạng và phân lớp tự động có thể xử lý được các thách thức được nêu ở phần 1.2. Với hệ thống có thể nhận dạng và phân lớp tự động hiệu quả trên loại dữ liệu ảnh từ camera giám sát “tĩnh” sẽ mang lại nhiều lợi ích.

Đầu tiên, hệ thống sẽ gia tăng hiệu suất giám sát. Việc phải ngồi quan sát trong thời gian dài để tìm kiếm các thông tin về các đối tượng cần thiết dễ gây cảm giác mệt mỏi, nhảm chán, mất tập trung đến công việc. Hệ thống sẽ hỗ trợ tìm kiếm thông tin thay cho con người và giúp giảm thiểu các tác hại khi phải làm việc liên tục với máy tính, đặc biệt về mặt thị lực nhờ giảm thời gian cần quan sát dữ liệu bằng mắt thường. Từ đó, hệ thống sẽ giảm bớt gánh nặng nhân sự và khối lượng công việc cho con người.

Một lợi ích khác, các thông tin hữu ích được rút trích từ hệ thống có thể được sử dụng để phối hợp thêm để làm điều kiện cho các quyết định khác như xử phạt giao thông, điều hướng giao thông, ... (đối với camera giao thông); đưa ra các biện pháp hỗ trợ các loài động vật hoang dã đặc biệt cho các loài có nguy cơ tuyệt chủng như điều chỉnh môi trường sống, hỗ trợ y tế, ... (đối với bẫy camera). Giám sát đa dạng sinh học về mặt định lượng có thể giúp chúng ta hiểu các kết nối giữa sự suy giảm số lượng loài động vật với sự ô nhiễm do khai thác và đô thị hóa cùng với sự nóng lên toàn cầu để từ đó đưa ra các chính sách phát triển hợp lý và bền vững.

Ở viễn cảnh xa hơn, hệ thống có thể được ứng dụng trong nhiều lĩnh vực khác ngoài phạm vi giám sát. Hệ thống có thể được phát triển thêm nhằm tích hợp truy xuất thông tin, dự đoán hành động và tương tác với con người dễ dàng hơn nhờ hệ hỏi đáp trên ảnh (Image question answering). Khi đó, hệ thống có thể chuyển đổi thành các ứng dụng được cài đặt trên điện thoại thông minh cho phép các nhân viên có thể tự do di chuyển nhưng vẫn có thể nắm được các thông tin cần thiết.

1.5 Mục tiêu đề tài

Đề tài sẽ tìm hiểu một phương pháp nhận dạng và phân lớp tự động hiệu quả. Cụ thể là nhận dạng và phân lớp thông qua các phương pháp học sâu từ loại dữ liệu ảnh được chụp bởi kiểu camera giám sát tĩnh.

Cùng với sự phát triển mạnh mẽ của các phương pháp học sâu trên lĩnh vực Thị giác máy tính, đề tài hứa hẹn sẽ giải quyết một phần các khó khăn đã trình bày ở mục 1.2 và đề xuất kỹ thuật phối hợp một số mô hình học sâu hiện đại. Bên cạnh đó, đề tài cũng sử dụng một số phương pháp tiền xử lý và tham số hóa để tăng tính tổng quát và độ chính xác cho các mô hình được sử dụng.

1.6 Cấu trúc đề tài

Trong giai đoạn thực hiện đề tài, nhóm đã thực hiện một số công việc liên quan và được trình bày trong báo cáo như sau:

- Chương 1: Giới thiệu tổng quan về đề tài. Trong chương này, nhóm sẽ trình bày một cách tổng quát về đề tài, tiềm năng phát triển và khả năng ứng dụng trong tương lai.
- Chương 2: Tổng quan về các công trình nghiên cứu liên quan đến đề tài mà nhóm tìm hiểu được qua 3 phần: hướng tiếp cận, mô hình sử dụng và kết quả đạt được.
- Chương 3: Trong chương này chúng tôi giới thiệu qui trình của phương pháp đề xuất và các kiến thức nền tảng hỗ trợ cấu thành phương pháp mà nhóm sử dụng.
- Chương 4: Áp dụng phương pháp của nhóm vào bộ dữ liệu và đánh giá kết quả đạt được. Sơ lược về bộ dữ liệu nhóm sử dụng .
- Chương 5: Tổng kết những công việc nhóm đã làm được và các thách thức đã gặp phải. Bàn luận và đánh giá kết quả đạt được kết hợp định hướng phát triển đồ án trong tương lai.

Chương 2

Tổng quan

2.1 Các nghiên cứu liên quan

2.1.1 Khảo sát các nghiên cứu sử dụng mạng học sâu cho phát hiện đối tượng

Object detection là một nhiệm vụ đầy thử thách của thị giác máy tính liên quan đến việc dự đoán các vị trí của đối tượng trong hình ảnh và phân loại các đối tượng được phát hiện. Chủ đề nghiên cứu này nhận được rất nhiều sự quan tâm gần đây do khả năng ứng dụng thực tế rộng rãi như phát hiện khuôn mặt, phát hiện xe, đếm số người đi bộ, hệ thống bảo mật, xe tự hành, chú thích hình ảnh, nhận diện biển số xe, ... Mục tiêu của phát hiện đối tượng là phát hiện tất cả các trường hợp của các đối tượng được xác định trước và cung cấp vị trí của nó trong hình ảnh bởi các hộp giới hạn.

Các mô hình phát hiện đối tượng ban đầu được xây dựng như một tập hợp của công cụ trích xuất tính năng thủ công như Viola-Jones object detector, Histogram of Oriented Gradients (HOG), ... Những mô hình này có tốc độ chậm, không chính xác và hoạt động kém trên các tập dữ liệu không quen thuộc. Cùng với sự phát triển mạnh mẽ của các mạng học sâu và sự bùng nổ về dữ liệu, khi có đầy đủ dữ liệu ta có thể thực hiện các nhiệm vụ khó khăn của thị giác máy tính như phân loại ảnh, sinh mẫu ảnh và phát hiện ảnh giả, ... và phát hiện đối tượng.

Các mô hình phát hiện đối tượng thường có thể được chia thành hai loại: one-stage detection (single-stage detection) và two-stage detection. Dựa theo công trình khảo sát của Syed và các đồng nghiệp[20] , nhóm sẽ tổng hợp, giới thiệu một cách khái quát một số nhóm mô hình tiếp cận phổ biến hiện nay để đọc giả có cái nhìn tổng quan và sẽ không đi sâu vào

cụ thể từng phương pháp.

Một số công trình nghiên cứu cổ điển (Pioneer Work):

- **Viola-Jones object detector**[19] : là một khung phát hiện đối tượng được đề xuất vào năm 2001 bởi Paul Viola và Michael Jones với mục đích để phát hiện khuôn mặt. Vào thời điểm ấy, Viola-Jones có thể thực hiện trong thời gian thực cho độ chính xác cao nhờ kết hợp nhiều kỹ thuật như Haar Feature Selection, integral image, Adaboost và Cascading Classifiers. Đầu tiên là thuật toán tìm kiếm các đặc trưng Haar-Like bằng cách trượt một cửa sổ trên hình ảnh đầu vào và sử dụng hình ảnh tích hợp (integral image) để tính toán. Sau đó, nó sử dụng Adaboost huấn luyện và bộ phân loại Cascade để phân loại chúng.
- **HOG Detector**[2]: được đề xuất bởi Dalal và Triggs vào năm 2005 với mục đích phát hiện con người. Họ sử dụng phương pháp mô tả đặc trưng Histogram of Oriented Gradients (HOG) để trích xuất các tính năng để phát hiện đối tượng. HOG sử dụng thông tin về sự phân bố của các cường độ gradient (intensity gradient) hoặc của hướng biên (edge directins) để mô tả các đối tượng cục bộ trong ảnh. Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con, được gọi là “tế bào” (cells) và với mỗi cell, ta sẽ tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell. Sau khi ghép các histogram lại với nhau, ta sẽ có một biểu diễn cho bức ảnh ban đầu. Để tăng cường hiệu năng nhận dạng, các histogram cục bộ có thể được chuẩn hóa về độ tương phản bằng cách tính một ngưỡng cường độ trong một vùng lớn hơn cell, gọi là các khối (blocks) và sử dụng giá trị ngưỡng đó để chuẩn hóa tất cả các cell trong khối. Kết quả sau bước chuẩn hóa là một vector đặc trưng có tính bất biến cao hơn đối với các thay đổi về điều kiện ánh sáng và được đưa vào bộ phân loại SVM(support vector machine) tuyến tính để phát hiện.

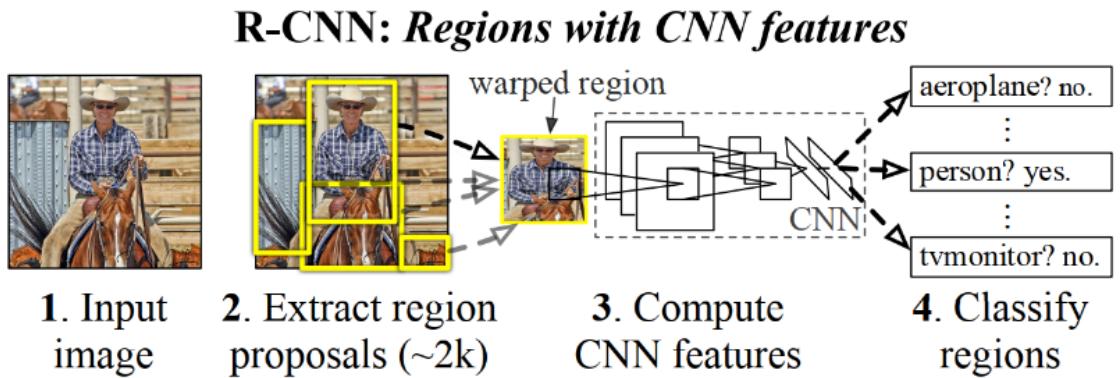
Two-stage detection:

Two-stage detection là nhóm mô hình cần 2 giai đoạn để phát hiện đối tượng, bao gồm: xác định vị trí hộp giới hạn của vật thể và phân loại vật thể. Two-stage detection thường sử dụng pipeline để kết hợp 2 thành phần mạng thực hiện các giai đoạn trên. Họ mô hình R-CNN [5] là lớp mô hình nổi bật và phổ biến cho kiểu phát hiện đối tượng này.

R-CNN:

Region-based Convolutional Neural Network (R-CNN) là bài báo tiên phong sử dụng mạng học sâu để cải thiện khả năng phát hiện đối tượng và cũng là mô hình mở đầu cho họ mô hình R-CNN. Cụ thể, R-CNN có bốn phần chính sau:

- Thuật toán sử dụng tìm kiếm chọn lọc[17] trên ảnh đầu vào để lựa chọn khoảng 2000 vùng đề xuất (proposed region). Các vùng đề xuất thông thường sẽ có nhiều tỷ lệ với hình dạng và kích thước khác nhau. Các nhãn và khung chứa sẽ được gán cho từng vùng đề xuất.
- Sau khi điều chỉnh kích thước phù hợp cho các vùng đề xuất, hệ thống trích xuất đặc trưng bằng một mạng CNN ở dạng tiền huấn luyện.
- Các đặc trưng và nhãn tương ứng của từng vùng đề xuất được kết hợp thành một mẫu để huấn luyện với SVM để thực hiện phân lớp đối tượng.
- Các đặc trưng và khung chứa được gán nhãn của mỗi vùng đề xuất được kết hợp thành một mẫu để huấn luyện mô hình hồi quy tuyến tính, để phục vụ dự đoán khung chứa nhãn gốc.

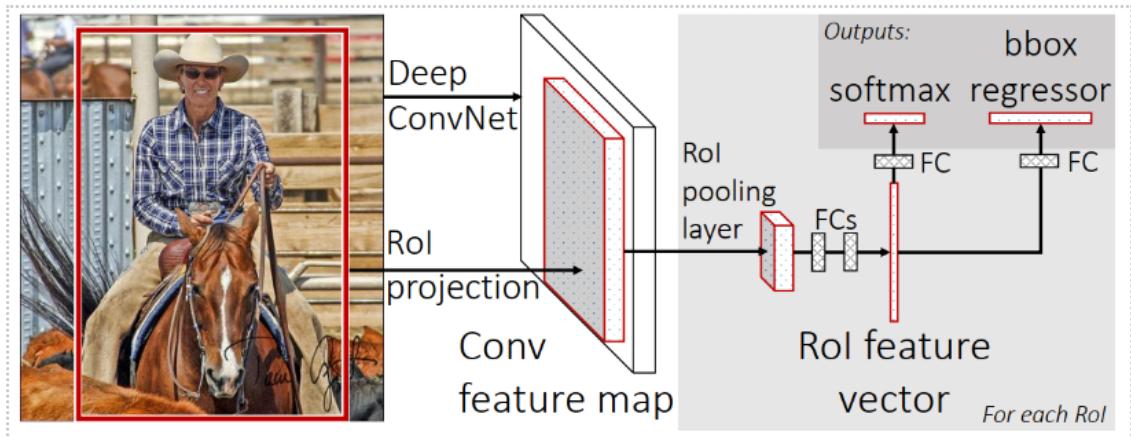


Hình 2.1: Mô hình RCNN

Mặc dù các mô hình R-CNN sử dụng các mạng CNN đã được tiền huấn luyện để trích xuất các đặc trưng ảnh một cách hiệu quả, điểm hạn chế chính yếu đó là tốc độ chậm (cần 47s để xử lý một ảnh). Khối lượng tính toán nặng nề khiến các mô hình R-CNN không được sử dụng rộng rãi trong các ứng dụng thực tế.

Fast R-CNN:

Fast R-CNN [4] ra đời năm 2015 nhằm giải quyết các hạn chế mà R-CNN gặp phải. Fast R-CNN sử dụng một mô hình đầu cuối duy nhất để huấn luyện cho phép chia sẻ sự tính toán và cập nhập trọng số giữa các lớp tích chập trong mạng. Hình ảnh được trích xuất đặc trưng thông qua một mạng CNN tiền huấn luyện. Sau đó, thuật toán sẽ chọn vùng đề xuất bằng thuật toán tìm kiếm chọn lọc trên fearture map khác biệt so với R-CNN cần thực hiện tách các vùng đề xuất ra rồi mới thực hiện trích xuất đặc trưng bằng CNN trên từng vùng đề xuất. Tuy nhiên, kích thước của các vùng đề xuất khác nhau nên ta cần một lớp có thể chuyển các vùng đề xuất về cùng kích thước. Để giải quyết vấn đề trên, tác giả Girshick đã đề xuất lớp mới được gọi là lớp RoI (Region of Interest) pooling và cách hoạt động của lớp sẽ được nêu rõ hơn ở phần 3.1.2. Sau khi trải qua lớp gộp RoI, các nhãn và vị trí khung chưa sẽ được dự đoán thông qua lớp kết nối đầy đủ. Fast R-CNN đã cải tiến đáng kể về tốc độ (2 giây mỗi hình ảnh, nhanh hơn 146 lần so với R-CNN) và độ chính xác so với R-CNN.



Hình 2.2: Mô hình Fast RCNN

Faster R-CNN : Faster R-CNN[11] là bước cải tiến vượt trội của Fast R-CNN. Mô hình được phát triển bởi Ren và cộng sự vào năm 2017. Mô hình có bước đột phá khi sử dụng mạng học sâu thay cho thuật toán tìm kiếm chọn lọc để đề xuất các vùng. Cụ thể về cơ sở lý thuyết của giải thuật sẽ được trình bày ở phần 3.1.2 và đây cũng chính là một hướng tiếp cận được áp dụng trực tiếp trong đề tài này.

One-stage detection:

Phổ biến trong đó có yolo v1, v2,v3 sẽ được đề cập chi tiết trong phần sau. Đề cập đến một loại mô hình phát hiện đối tượng là mô hình một giai đoạn, tức là mô hình bỏ qua giai đoạn đề xuất khu vực so với mô hình hai giai đoạn và chạy phát hiện trực tiếp qua lấy mẫu dày đặc các vị trí. Các loại mô hình này thường có suy luận nhanh hơn (có thể phải trả giá bằng hiệu suất).

Chương 3

Phương pháp

3.1 Các công nghệ mà mô hình sử dụng:

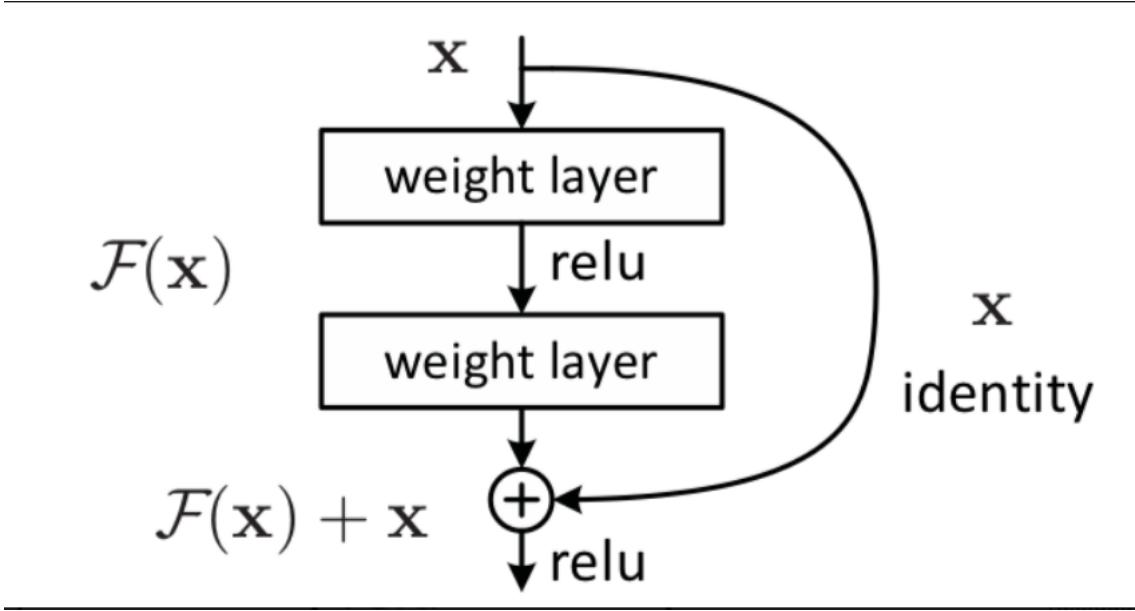
3.1.1 ResNet [6]

Trước thời điểm mạng ResNet ra đời, **Vanishing Gradient** là một nan đề khó giải cho các mạng học sâu. Gradient của các mạng học sâu thường có xu hướng nhỏ dần khi đi xuống các lớp thấp hơn. Vì thế, nếu độ sâu của mạng ngày càng tăng thì gradient của mạng ở các lớp thấp hơn sẽ càng nhỏ và có khả năng “biến mất”. Điều này làm độ chính xác của các mạng sẽ chuyển dần đến bão hòa và suy giảm. Để giải quyết vấn đề trên, vào năm 2015, nghiên cứu của Kaiming He và các đồng sự [1] đã đề xuất mạng ResNet (Residual Network) cho phép giải quyết **Vanishing Gradient**. Khi đó, ResNet đã giành được hạng nhất trong cuộc thi ILSVRC 2015 trong tác vụ phân lớp với mức lối là **3.57%** trên tập dữ liệu **ImageNet**. Tiếp nối thành công trên, ResNet còn giành được vị trí đầu tiên trong cuộc thi ILSVRC & COCO 2015 trong các tác vụ ImageNet detection, ImageNet localization, Coco detection và Coco segmentation. Giải pháp của ResNet là Residual Block thực hiện kết nối “tắt” giúp kết nối giá trị của lớp trước đó với lớp đầu ra tiếp theo. Theo bài báo gốc, có 2 cách để tính toán thực hiện kết nối “tắt” giữa các lớp.

Với trường hợp đầu vào và đầu ra của các lớp có cùng kích thước, giá trị x có thể được sử dụng trực tiếp theo công thức như sau:

$$y = \mathcal{F}(x, W_i) + x$$

Trong đó x và y là đầu vào và đầu ra của lớp; \mathcal{F} là hàm ánh xạ residual. Như trường hợp hình dưới 3.1 (~~sau này cần có thứ tự~~) có 2 lớp thì hàm kích hoạt có công thức là $F = W_2\sigma(W_1x)$ với σ đại diện cho hàm RELU.



Hình 3.1: Skip connection

Trong trường hợp đầu vào và đầu ra các lớp có sự khác biệt về kích thước, mô hình sẽ sử dụng phép chiếu tuyến tính W_s nhằm quy về cùng chiều tạo điều kiện để sử dụng công thức sau :

$$y = \mathcal{F}(x, W_i) + W_s x$$

Hiện tại, ResNet đã được phát triển rất nhiều phiên bản khác nhau về độ sâu và kiến trúc của các mô hình được thể hiện thông qua bảng 3.2 .

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
				3×3 max pool, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1			average pool, 1000-d fc, softmax		
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

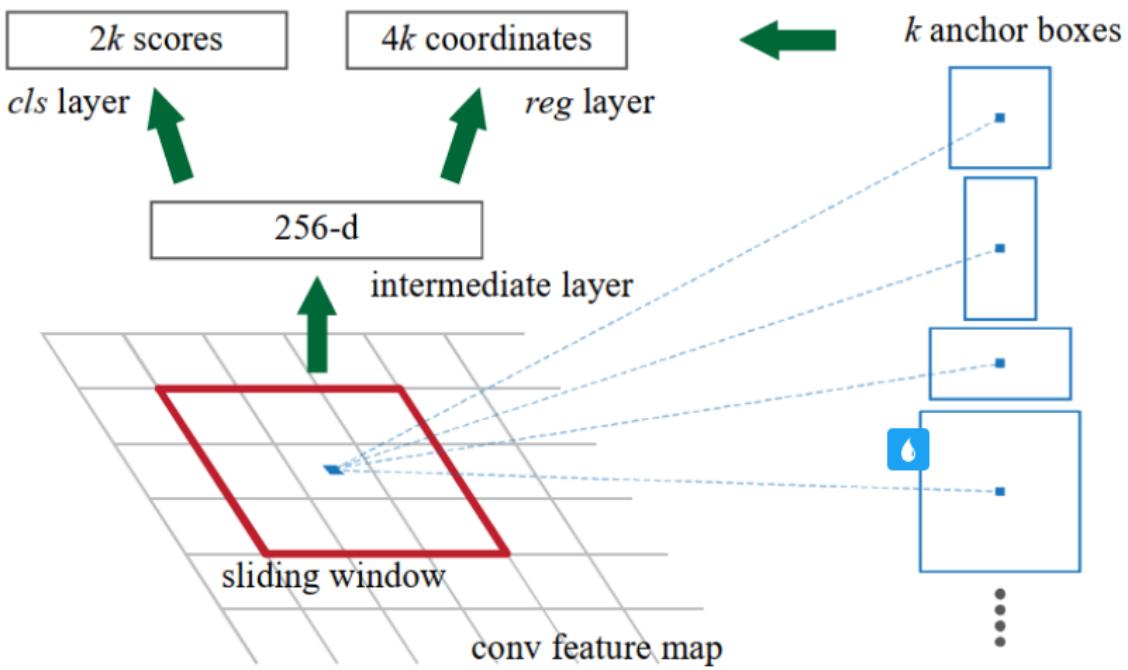
Hình 3.2: Các phiên bản của resnet

3.1.2 Faster R-CNN

Faster R-CNN là mạng học sâu thuộc họ R-CNN và được cải tiến từ Fast R-CNN. Trong các cuộc thi ILSVRC và COCO 2015, Faster R-CNN và RPN là nền tảng của các bài viết đoạt giải nhất trong một số cuộc thi. Fast R-CNN vẫn còn sử dụng thuật toán tìm kiếm chọn lọc gây ảnh hưởng tốc độ của mô hình. Vì thế, Faster R-CNN đề xuất sử dụng một mạng học sâu được gọi là dạng mạng đề xuất vùng (Region Proposal Network) để thay thế cho tìm kiếm chọn lọc. Đầu tiên cả bức ảnh được cho qua mô hình CNN đã tiền huấn luyện để lấy feature map. Sau đó, feature map được dùng cho RPN (Region Proposal Network) để lấy được các region proposal.

Theo bài báo gốc mô hình RPN khá đơn giản. Feature map được cho qua tầng tích chập 3×3 , 512 kernels. Không giống như những người tiền nhiệm của nó, RPN giới thiệu khung neo (anchor box). Khung neo được định nghĩa trước khi huấn luyện mô hình. Trong bài báo Faster-RCNN, khung neo được định nghĩa với 9 khung neo ứng với mỗi điểm pixel trên feature map. Tuy nhiên, việc tính toán tổng số lượng khung neo là dựa trên kích thước của feature map và kích thước cùng với tỷ lệ khung của khung neo phải tham chiếu ngược lại kích thước của ảnh gốc ban đầu.

Sơ lược về IoU (Intersection over Union): là tỷ số giữa diện tích giao nhau giữa khung neo và khung chứa với diện tích liên kết của hai hộp. IoU nằm trong khoảng 0.0 đến 1.0. Khi không có sự giao nhau thì diện tích IoU bằng 0, khi hai hộp gần nhau hơn, IoU tăng cho đến khi đạt 1.0 (khi hai hộp giống nhau 100%).



Hình 3.3: Kiến trúc RPN

Sau đó với mỗi khung neo lấy được ở trên RPN thực hiện 2 bước:

- Dự đoán xem khung neo đó có chứa vật thể hay không
- Dự đoán 4 giá trị thể hiện vị trí tâm (x_center , y_center) và kích thước ($width$, $height$) của các khung neo.

Các khung neo này được xét là dương tính (chứa đối tượng) hoặc âm tính (không chứa đối tượng) dựa vào chỉ số IoU của bản thân với khung chứa đúng (ground-truth bounding box) theo qui tắc sau:

- Các khung neo có tỉ lệ IoU lớn nhất với khung chứa đúng (ground-truth box) hoặc có tỉ lệ IoU lớn hơn hoặc bằng 0.7 sẽ được coi là dương tính.
- Các khung neo có tỉ lệ IoU nhỏ hơn 0.3 sẽ được coi là âm tính.
- Các khung neo nằm trong khoảng lớn hơn hoặc bằng 0.3 và nhỏ hơn 0.7 sẽ được coi là trung tính và không được sử dụng trong quá trình huấn luyện mô hình.

Tuy nhiên, số lượng khung neo dương tính không phải lúc nào cũng có đủ trong một ảnh. Vì vậy, trong bài báo gốc đưa ra hướng xử lý khi không đủ khung neo dương tính thì thay thế bằng các khung neo âm tính khác.

Sau khi thực hiện dự đoán khung neo, kết quả trả về có rất nhiều khung neo bị chồng lên nhau. Vì thế, mô hình sử dụng triệt phi cực đại (~~non-maxima suppression~~) được dùng để loại bỏ các khung neo chồng lên nhau.

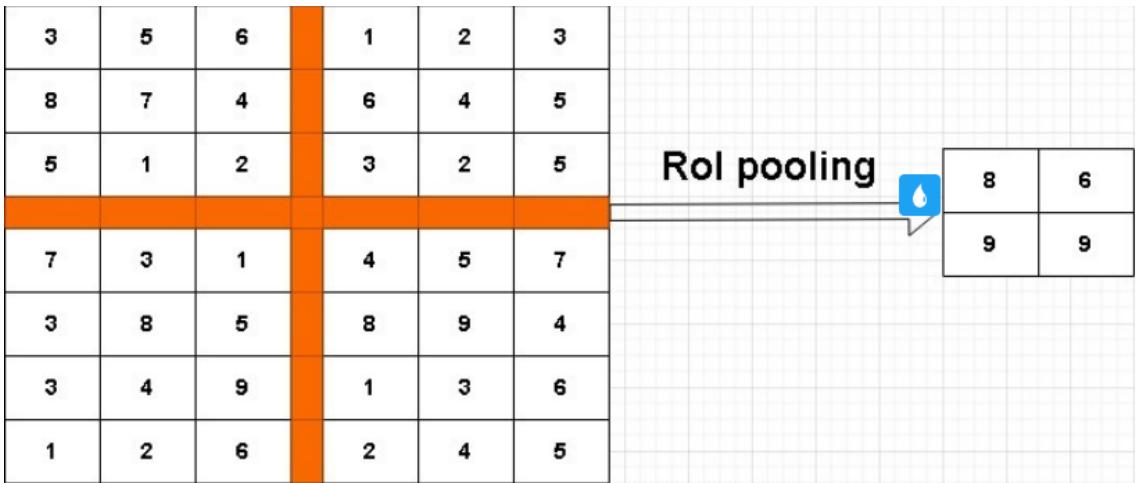
Non-maximum suppression

Triệt phi cực đại (~~Non-maxima suppression~~)

Để giữ lại N vùng đề xuất, nhóm sẽ sơ lược về các bước thực hiện thuật toán như sau:

- Bước 1: Chọn ra khung neo có xác suất chứa đối tượng lớn nhất (X) trong tập Input.
- Bước 2: Thêm X vào tập Ouput. Sau đó, loại bỏ X và các khung neo trong tập Input mà có hệ số IoU với X lớn hơn 0.5 ra khỏi tập Input.
- Bước 3: Kiểm tra nếu tập Input rỗng hoặc tập Output đủ N khung neo thì dừng lại, nếu không quay lại bước 1.

Sau khi thực hiện triệt phi cực đại, RPN sẽ lấy N khung neo để làm vùng đề xuất, thuật toán cần resize các vùng đề xuất về cùng kích thước trước khi phân lớp đối tượng. Tuy nhiên ở feature map ta không thể resize được, nên cần phải có cách gì đây để chuyển các vùng đề xuất (region proposal) trong feature map về cùng kích thước. Vì thế, Region of Interest (ROI) pooling ra đời để giải quyết vấn đề.



Hình 3.4: ROI pooling

Điểm khác so với max pooling hay average pooling là bất kể kích thước của input tensor, ROI pooling luôn cho ra output có kích thước cố định được định nghĩa trước. ROI pooling sử dụng nguyên tắc Dirichlet để làm tròn số khi chia vùng đề xuất thành các phần. Sau khi chia, thuật toán sử dụng max pooling cho từng vùng và trả về kết quả. Ví dụ: đầu vào của ROI pooling kích thước 6×7 và đầu ra có kích thước 2×2 . Ta chia chiều rộng thành 2 phần, mỗi phần có kích thước $6/2 = 3$. Tuy nhiên, khi ta chia chiều dài thành 2 phần; thay vì giữ kích thước mỗi phần là $7/2 = 3.5$ thì thuật toán sẽ làm tròn kích thước một phần về 3, phần còn lại có kích thước bằng 4. Sau đó với mỗi khối được tạo ra bằng các đường màu cam, ta thực hiện max pooling lấy ra 1 giá trị.

Loss function (hàm lỗi) của mô hình Faster R-CNN sẽ được định nghĩa với công thức như sau:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda * \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

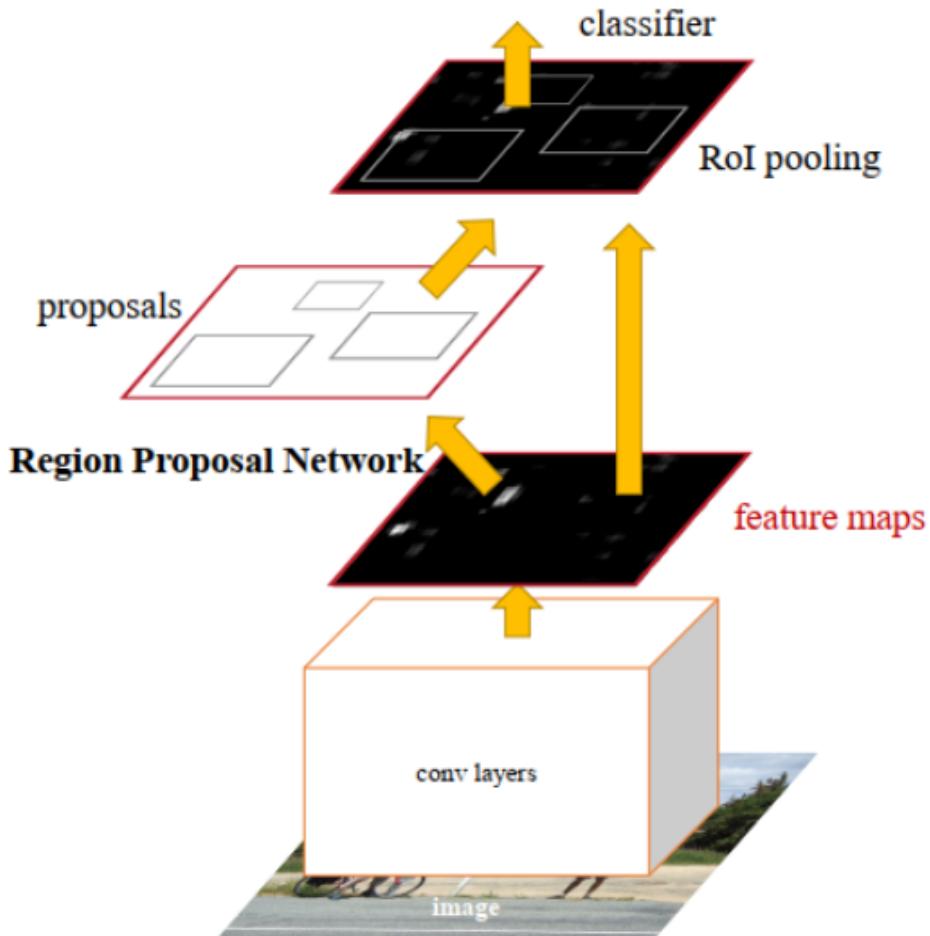
Trong đó i là index của khung neo trong một mini-batch và p_i là xác suất dự đoán cho khung neo tại index thứ i là đối tượng. Chú ý, loss cho A chỉ được tính khi khung neo được xác định là dương tính.

L_{cls} ứng với binary cross entropy lớp cls của RPN để xác định khung neo có chứa đối tượng hay không.

L_{reg} là loss tính cho hồi qui khung chứa. Có thể dùng L_2 hoặc L_1 loss, tuy nhiên trong paper có đề cập sử dụng hàm loss Smooth L_1 Loss. Smooth L_1 Loss có gradient tăng ổn định khi x lớn với L_1 loss và gradient ít dao động khi x nhỏ (L_2 loss). Công thức của Smooth L_1 loss dạng đơn giản:

$$L_{1;smooth} = \begin{cases} |x|, & \text{nếu } |x| > a, \\ \frac{1}{|a|}x^2, & \text{nếu } |x| \leq a. \end{cases}$$

Kết quả thu được chỉ ra rằng Faster R-CNN đã cải thiện đáng kể cả độ chính xác và hiệu quả phát hiện. Trên PASCAL Bộ thử nghiệm VOC 2007, Faster R-CNN đạt mAP 69,9% như so với Fast R-CNN là 66,9%. Đồng thời, tổng thời gian chạy Faster R-CNN (198ms) thấp hơn gần 10 lần so với Fast R-CNN (1830ms) với cùng mạng trục VGG [12] và tốc độ xử lý là 5 khung hình / giây so với 0,5 khung hình / giây. Faster R-CNN là thành phần quan trọng để tạo nên mô hình giành được hạng nhất trong cuộc thi ILSVRC 2015 object detection [6].

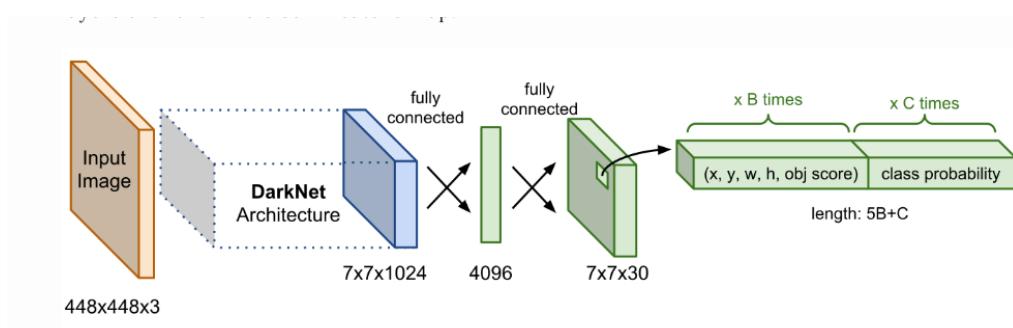


Hình 3.5: Kiến trúc mạng Faster R-CNN

3.1.3 Các mô hình YOLO V1, V2, V3:

Trong những năm gần đây, với sự phát triển mạnh mẽ của computer vision. Các bài toán ngày càng yêu cầu độ chính xác cao, khả năng ứng dụng được trong thực tế. Vì vậy có nhiều thuật toán cũng như mô hình ra đời ngày càng cải thiện độ chính xác so với những thuật toán hay là những mô hình cổ điển trước đây. Một trong số đó là you only look one hay người ta còn gọi tắt là YOLO. Hiện có năm phiên bản của YOLO đó là V1, V2, V3, V4, V5. Dưới đây giới thiệu phiên bản YOLO V1[10] (version 1) và YOLO v2 (version 2) [8] và YOLOV3 (version3)[9].

3.1.4 YOLO V1:



Hình 3.6: Kiến trúc YOLO V1

Yolo chia mỗi bức ảnh thành các vùng lưới (grid cell) thông thường là 5x5 hoặc là 7x7, trọng tâm của các vật thể được tìm trong các vùng lưới đó thông qua quá trình đào tạo.

Output của mô hình là $S * S * (5 * B + C)$.

Üng với mỗi box trong B bounding box này sẽ là 5 tham số x, y, w, h, confidence, lần lượt là tọa độ tâm (x, y), chiều rộng, chiều cao và độ tự tin của dự đoán. Với grid cell trong lưới SxS kia, mô hình cũng dự đoán xác suất rơi vào mỗi C class.

Loss function

$$\begin{aligned}
& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} (C_i - C_i i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
& + \sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_c (p_i(c) - \hat{p}_i(c))^2
\end{aligned}$$

- Ba hằng số λ chỉ là các hằng số để xem xét thêm một khía cạnh của hàm mất mát. Trong bài báo, λ_{coord} là cao nhất để có tầm quan trọng hơn trong thuật ngữ đầu tiên.
- Dự đoán của YOLO là vectơ $S * S * (B * 5 + C)$: dự đoán B bounding box cho mỗi ô lưới và dự đoán lớp C cho mỗi ô lưới (với C là số lớp). 5 đầu ra bounding box của hộp j của ô i là tọa độ của tâm của bounding box x_{ij} y_{ij} , chiều cao h_{ij} , chiều rộng w_{ij} và độ tin cậy C_{ij} .
- 1_i^{obj} là 1 nếu đối tượng thuộc về ô lưới và bằng 0 nếu ngược lại.
- C_i là đối tượng nghĩa là điểm tin cậy của đối tượng đó có hay không.
- $p_i(c)$ là sự mất mát của phân lớp .

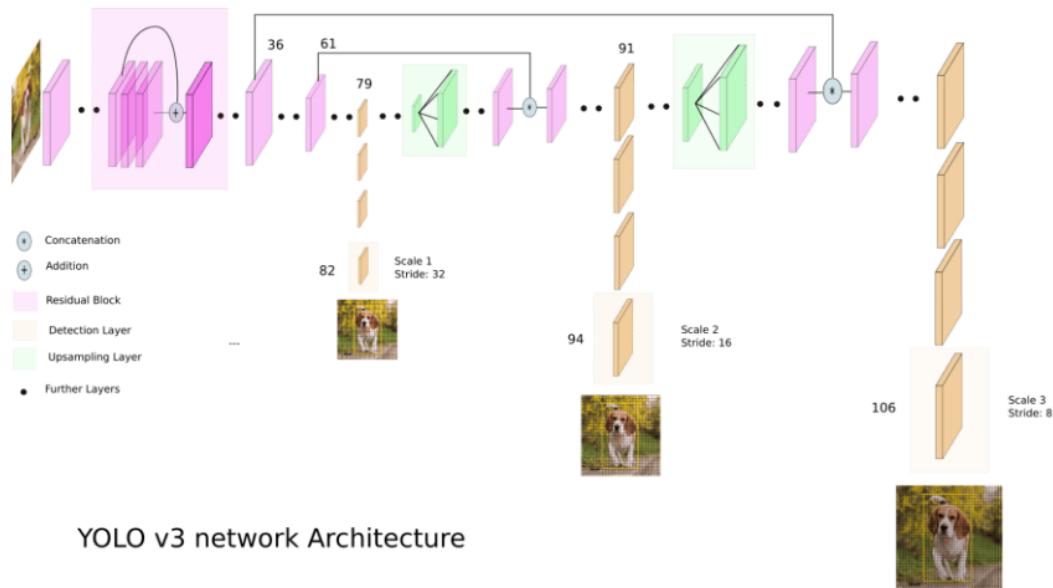
3.1.5 YOLO V2:

Do yolo v1 gây ra rất nhiều lỗi về việc xác định vị trí của vật thể và giá trị recall rất thấp. Vì vậy, Yolo v2 ra đời nhằm bổ sung và cải tiến những cái của yolov1, với những cải tiến như sau:

- BatchNormalization (BN): thêm BN vào tất cả các lớp chập, điều này giúp chúng ta làm giảm đi sự phụ thuộc vào giá trị khởi tạo của các tham số. Và sau khi áp dụng BN thì giá trị mAP của model đã tăng thêm 2%.
- Bộ phân loại phân giải cao (High Resolution Classifier): Việc sử dụng High Resolution Classifier này giúp tăng mAP lên tối đa 4%.

- Phát hiện nhiều đối tượng trong một ô lưới (gird cell).
- Khung neo: nó là các bounding box nhưng được tạo ra sẵn (còn bounding box là do model dự đoán). Với một ô lưới sẽ có trách nhiệm tạo ra một số K các khung neo với các kích thước khác nhau.
- Được đào tạo trên nhiều kích thước ảnh đầu vào .
- Darknet19: với YOLOv1 sử dụng backbone là 24 lớp convolution thì sang YOLOv2 lại được sử dụng Darknet19 với 19 lớp convolution cùng với 5 lớp max pooling. Darknet19 xử lý rất nhanh trong việc nhận dạng vật thể nên rất có ý nghĩa trong việc xử lý ở thời gian thực[15].

3.1.6 YOLO V3



Hình 3.7: Kiến trúc YOLO V3

Tổng quan về kiến trúc :

- YOLO v3 sử dụng một biến thể của Darknet, ban đầu có mạng 53 lớp được đào tạo trên Imagenet.
- Đối với nhiệm vụ phát hiện, 53 lớp khác được xếp chồng lên nó, mang lại cho chúng ta một kiến trúc cơ bản hoàn toàn phức hợp 106 lớp cho YOLO v3.

- Trong YOLO v3, việc phát hiện được thực hiện bằng cách áp dụng các hạt nhân phát hiện (kernels) 1 x 1 trên các bản đồ đối tượng (feature maps) có ba kích thước khác nhau tại ba vị trí khác nhau trong mạng.
- Hình dạng của hạt nhân(kernels) phát hiện là $1 * 1 * (B * (5 + C))$. Ở đây B là số ô giới hạn mà một ô trên bản đồ đối tượng(feature maps) có thể dự đoán, '5' là dành cho 4 thuộc tính ô giới hạn(bounding boxes) và độ tin cậy của một đối tượng và C là số lớp của các lớp học.
- YOLO v3 sử dụng entropy chéo nhị phân(binary cross entropy) để tính toán tổn thất phân loại cho mỗi nhãn trong khi độ tin cậy của đối tượng và dự đoán lớp được dự đoán thông qua hồi quy logistic[16]
- .

Các siêu tham số được sử dụng :

- Class_threshold:Xác định ngưỡng xác suất cho đối tượng được dự đoán
- Non-Maximum Suppression: Nó giúp khắc phục vấn đề phát hiện một đối tượng nhiều lần trong một bức ảnh. Nó thực hiện điều này bằng cách lấy các hộp có xác suất tối đa và loại bỏ các hộp gần nhau có xác suất không tối đa (nhỏ hơn ngưỡng được xác định trước).
- input_height và input_shape: đầu vào của hình ảnh.

Kiến trúc của Darknet53:

- Darknet-53 ~~đ~~được sử dụng như một công cụ trích xuất đặc trưng.
- Darknet-53 chủ yếu bao gồm các bộ lọc 3 x 3 và 1 x 1 với các kết nối bỏ qua(skip connections) giống như mạng còn lại trong ResNet.

Type	Filters	Size	Output
Convolutional	32	3×3	256×256
Convolutional	64	$3 \times 3 / 2$	128×128
1x	Convolutional	32	1×1
	Convolutional	64	3×3
	Residual		128×128
2x	Convolutional	128	$3 \times 3 / 2$
	Convolutional	64	1×1
	Convolutional	128	3×3
8x	Residual		64×64
	Convolutional	256	$3 \times 3 / 2$
	Convolutional	128	1×1
8x	Convolutional	256	3×3
	Residual		32×32
	Convolutional	512	$3 \times 3 / 2$
8x	Convolutional	256	1×1
	Convolutional	512	3×3
	Residual		16×16
4x	Convolutional	1024	$3 \times 3 / 2$
	Convolutional	512	1×1
	Convolutional	1024	3×3
	Residual		8×8
Avgpool		Global	
Connected		1000	
Softmax			

Hình 3.8: Kiến trúc darknet53

Kiến trúc chi tiết mô hình YOLOV3:

Layer	Filters size	Repeat	Output size
Image			416×416
Conv	$32 \times 3/1$	1	416×416
Conv	$64 \times 3/2$	1	208×208
Conv	$32 \times 1/1$	Conv	208×208
Conv	$64 \times 3/1$	Conv $\times 1$	208×208
Residual		Residual	208×208
Conv	$128 \times 3/2$	1	104×104
Conv	$64 \times 1/1$	Conv	104×104
Conv	$128 \times 3/1$	Conv $\times 2$	104×104
Residual		Residual	104×104
Conv	$256 \times 3/2$	1	52×52
Conv	$128 \times 1/1$	Conv	52×52
Conv	$256 \times 3/1$	Conv $\times 8$	52×52
Residual		Residual	52×52
Conv	$512 \times 3/2$	1	26×26
Conv	$256 \times 1/1$	Conv	26×26
Conv	$512 \times 3/1$	Conv $\times 8$	26×26
Residual		Residual	26×26
Conv	$1024 \times 3/2$	1	13×13
Conv	$512 \times 1/1$	Conv	13×13
Conv	$1024 \times 3/1$	Conv $\times 4$	13×13
Residual		Residual	13×13

Hình 3.9: Kiến trúc chi tiết mô hình yolo v3

Tầng tích chập trong yolo v3

- Chứa 53 lớp tích chập, mỗi lớp tiếp theo là lớp chuẩn hóa (batch normalization) hàm kích hoạt (Leaky ReLU activation.).
- Lớp Convolution được sử dụng để kết hợp nhiều bộ lọc trên hình ảnh và tạo ra nhiều bản đồ đối tượng(feature maps).
- Không có hình thức tổng hợp nào được sử dụng và một lớp tích chập có bước 2(stride=2) được sử dụng để lấy mẫu bản đồ đối tượng địa lý. Nó giúp ngăn ngừa việc mất các tính năng cấp thấp thường được cho là do gộp chung(pooling).

YOLOv3 là phiên bản cải tiến của YOLOv2. Những cải tiến so với yolo v2 là :

- Logistic regression cho điểm tự tin(confidence score): sử dụng sử dụng logistic regression YOLOv3 predict độ tự tin của bounding box (có chứa vật hay không).

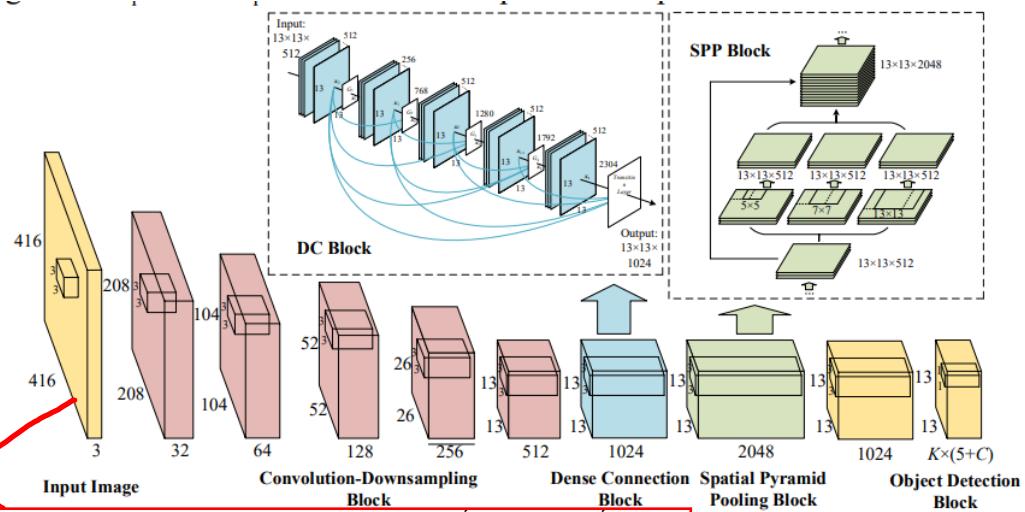
- Thay softmax bằng các logistic classifier rời rạc: YOLOv3 sử dụng các logistic classifier thay vì softmax cho việc phân loại đối tượng. Việc này cho hiệu quả tốt hơn, nghĩa là có thể có đối tượng cùng thuộc 2 hay nhiều class khác nhau.
- Backbone mới Darknet-53: Backbone được thiết kế lại với việc thêm các residual blocks (kiến trúc sử dụng trong ResNet).
- Multi-scale prediction: YOLOv3 sử dụng kiến trúc Feature Pyramid Networks (FPN) để đưa ra các dự đoán từ nhiều scale khác nhau của feature map. Việc này giúp YOLOv3 tận dụng các feature map với độ thô - tinh khác nhau cho việc dự đoán.

Theo kết quả thí nghiệm trên tập dữ liệu MS COCO, YOLOv3 (AP: 33%) hoạt động ngang bằng với biến thể SSD (DSSD513: AP: 33,2%) theo chỉ số MS COCO chưa đến 3 lần nhanh hơn DSSD trong khi thua kém RetinaNet một chút (AP: 40,8%). Nhưng sử dụng chỉ số phát hiện cũ của mAP tại $IOU = 0,5$ (hoặc AP50), YOLOv3 có thể đạt được 57,9% mAP khi so với DSSD513 là 53,3% và RetinaNet là 61,1%.

Do những ưu điểm của dự đoán đa quy mô, YOLOv3 có thể phát hiện các vật thể nhỏ nhiều hơn nhưng tương đối kém hơn hiệu suất trên các đối tượng có kích thước trung bình và lớn hơn .

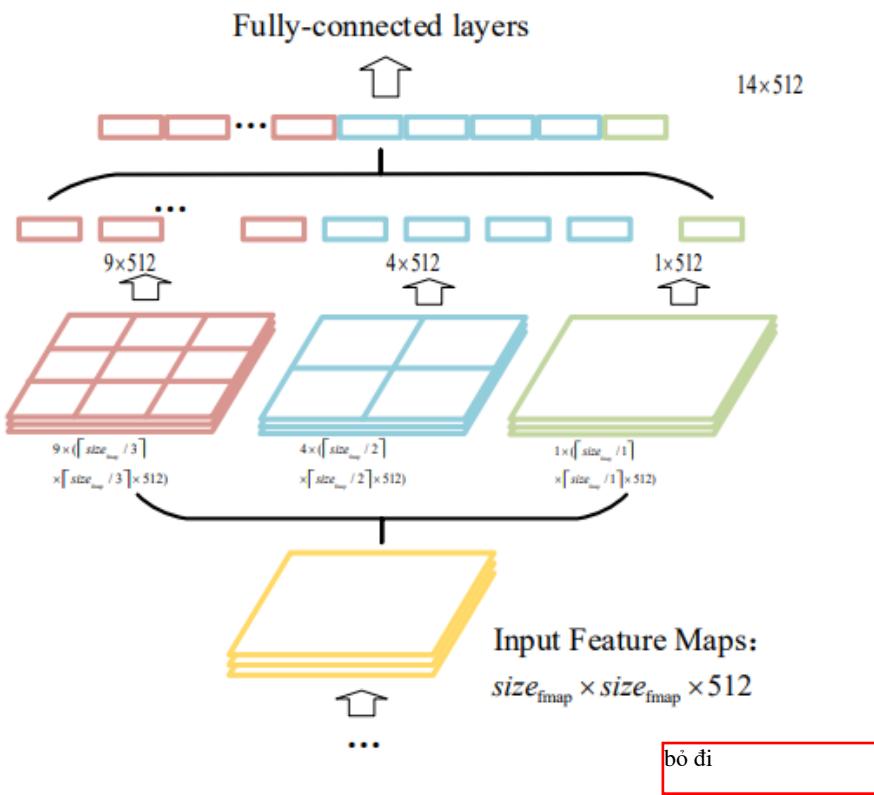
YOLO SPP(spatial pyramid pooling)

Trong phần này không đi quá sâu vào chi tiết mô hình YOLO SPP mà chỉ là nêu những điểm khác biệt so với yolo v3 , từ đó giúp cải thiện hơn so với yolov3.



Hình này của YOLOv2- SPP chứ YOLOv3 gì. Đọc kĩ lại bài báo trước khi lấy hình chứ mấy bạn. Đùa nhau ak. <https://arxiv.org/ftp/arxiv/papers/1903/1903.08589.pdf>. Bài này nó bảo nó improve YOLOv2 bằng khối SPP và dense convolution. May mà thầy ko đê ý.

Hình 3.10: Tổng quan sơ qua về mô hình yolov3 spp



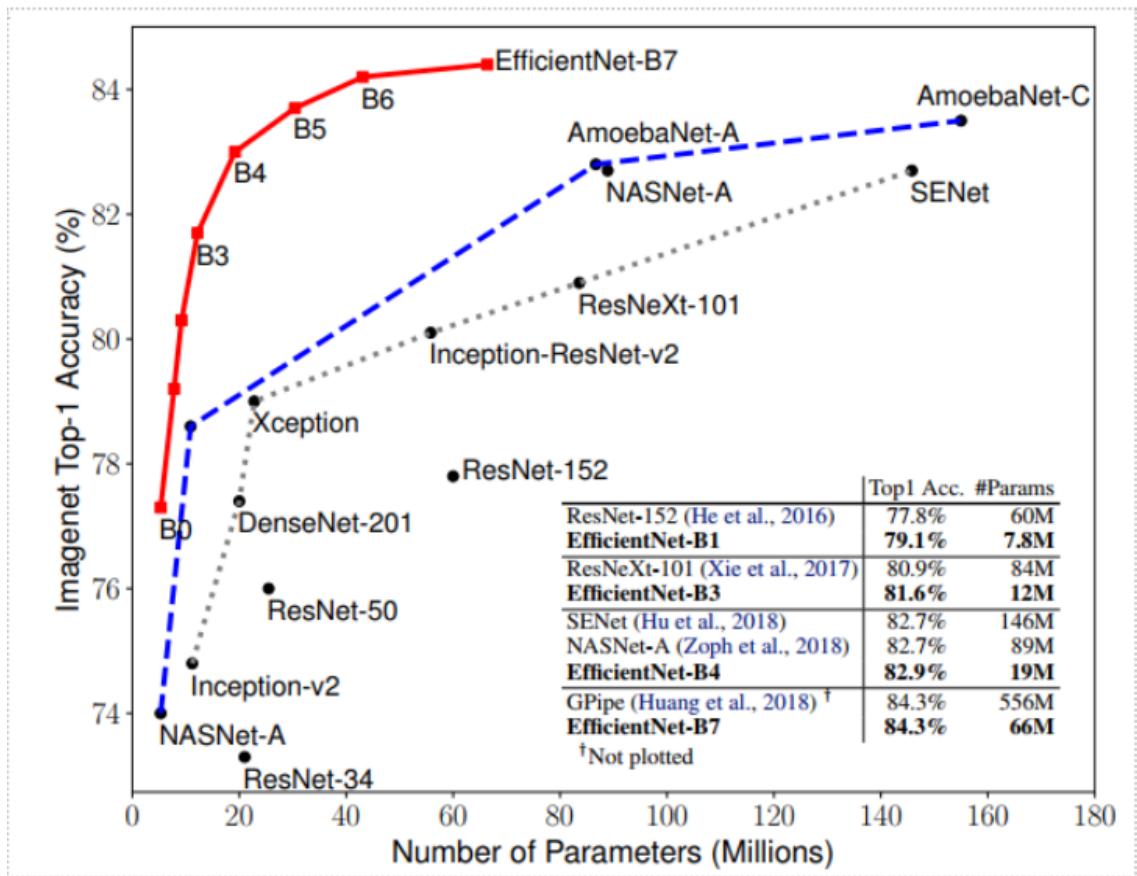
Hình 3.11: Khối spatial pyramid pooling(nguồn paper yolo v3 spp)

Trong khối spatial pyramid pooling sẽ chia các đặc trưng đầu vào(input feature maps) thành $a_i = n_i * n_i$ theo các thang đo đại diện cho các lớp khác nhau của các đặc trưng của hình chép (feature pyramid), a_i biểu diễn số lớp chồng lên nhau của lớp thứ i . Sau đó các bộ lọc sẽ trượt trên các

lớp đặc trưng của hình chóp (feature pyramid) với kích thước tương ứng và sau cùng ta thu được một lớp kết nối đầy đủ (fully connected layers).

3.1.7 EfficientNet

Các mạng CNN thường sẽ gia tăng độ chính xác khi có kích thước tham số càng lớn. Đi kèm với sự gia tăng kích thước đó, chúng ta cần nhiều tài nguyên hơn để huấn luyện. Tuy nhiên, trong thực tế, các mạng CNN thường được cung cấp nguồn tài nguyên cố định để phát triển. Vì thế, khi nhóm tác giả Mingxing Tan và Quoc V. Le đã về vấn đề cân bằng giữa hiệu suất và độ chính xác, họ nhận thấy rằng việc cân bằng một cách có hệ thống độ sâu, chiều rộng và độ phân giải mạng CNN có thể mang đến hiệu suất tốt hơn. Từ kết quả trên, nhóm tác giả nghiên cứu Mingxing Tan và Quoc V. Le đã đề xuất ra mạng EfficientNet[13] đạt được nhiều thành tựu nổi bật. Tại thời điểm bài báo ra đời, EfficientNet-B7 đã đạt được độ chính xác 84,3% nằm trong nhóm mô hình hiện đại có độ chính xác hạng nhất trên tập dữ liệu ImageNet. Mô hình EfficientNet-B7 cho độ chính xác tương đương mô hình GPipe của Huang và các đồng sự [15] đề xuất nhưng lại có kích thước nhỏ hơn 8.4 lần và nhanh hơn 6.1 lần. Nhóm mô hình EfficientNet còn đạt được các độ chính xác cao trên các tập dữ liệu khác như CIFAR-100(91,7%), Flowers (98,8%), ...



Hình 3.12: Biểu đồ so sánh các mô hình

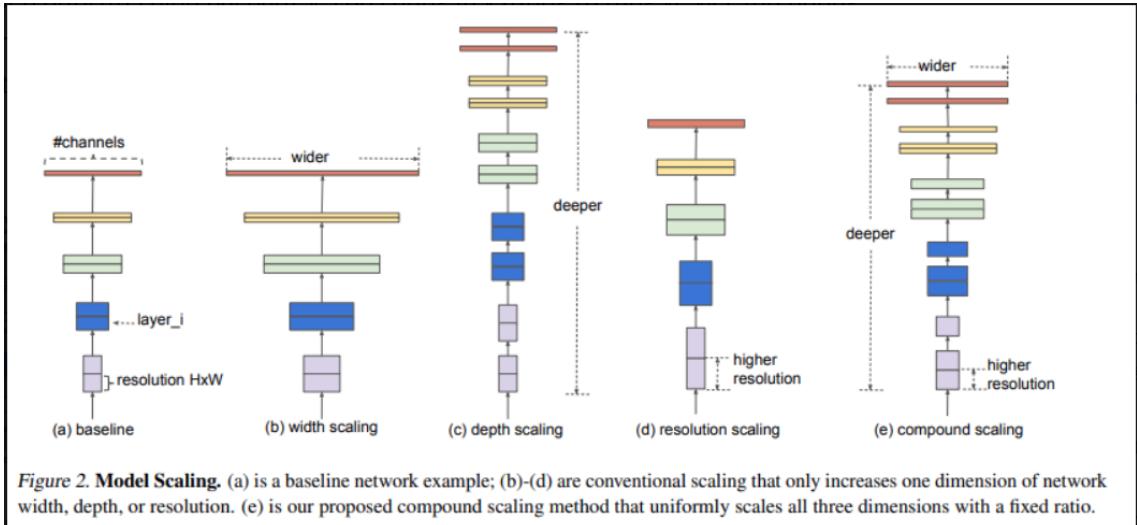
Thành công của EfficientNet được thể hiện qua 2 thành phần chính: phương pháp compound scaling và kiến trúc của EfficientNet.

Trước khi EfficientNet ra đời, cách phổ biến nhất để mở rộng ConvNets là theo một trong ba kích thước tỷ lệ: depth (độ sâu của mạng tương đương với số lớp), width (độ rộng của mạng tương ứng với số kênh) hoặc resolution (độ phân giải hình ảnh).

Sau khi nghiên cứu và thực hiện các thí nghiệm về việc biến đổi tỷ lệ kích thước mạng sẽ ảnh hưởng đến độ chính xác của mô hình như thế nào, nhóm tác giả đã rút ra được 2 kết luận:

- Thực hiện phép biến đổi tỷ lệ lên bất kỳ kích thước nào về độ rộng, độ sâu hoặc độ phân giải của mạng sẽ cải thiện độ chính xác, nhưng độ chính xác sẽ giảm đối với các mô hình có kích thước lớn hơn.
- Để đạt được độ chính xác và hiệu quả tốt hơn, điều quan trọng là phải cân bằng tất cả các kích thước của độ rộng, độ sâu và độ phân

giải mạng trong quá trình thực hiện phép biến đổi tỷ lệ trên mạng CNN



Hình 3.13: Hình minh họa thí nghiệm thực hiện phép biến đổi tỷ lệ lên độ rộng, độ sâu hoặc độ phân giải của mạng ảnh hưởng đến độ chính xác của mạng (EfficientNet-B0).

Đầu tiên, nhóm tác giả đã tổng quát hóa bài toán và định nghĩa lại mạng CNN theo công thức sau gọi là công thức (1) :

$$\mathcal{N} = \bigodot_{i=1 \dots s} F_i^{L_i} (X_{\langle H_i, W_i, C_i \rangle})$$

Trong đó: \mathcal{N} đại diện cho mạng CNN, $F_i^{L_i}$ biểu thị kiến trúc lớp F_i được lặp lại L_i lần trong stage i X là tensor đầu vào với kích thước $\langle H_i, W_i, C_i \rangle$ với H_i, W_i là spatial dimension và C_i là channel dimension.

Ý tưởng của compound scaling là cố gắng thực hiện phép biến đổi tỷ lệ chiều dài mạng (L_i), chiều rộng (C_i) và hoặc độ phân giải (H_i, W_i) mà không thay đổi F_i được xác định trước trong mạng. Nhằm thu hẹp không gian tìm kiếm, nhóm tác giả đã hạn chế rằng tất cả các lớp phải được biến đổi tỷ lệ một cách đồng nhất với tỷ lệ không đổi. Mục tiêu của nhóm nghiên cứu là tối đa hóa độ chính xác của mô hình cho bất kỳ lượng hạn chế tài nguyên nhất định. Từ đó, vấn đề này có thể được xem như một bài toán tối ưu hóa được mô tả theo công thức sau gọi là công thức (2):

$$\max_{d,w,r} \text{Accuracy}(\mathcal{N}(d, w, r)) \text{ s.t. } \mathcal{N}(d, w, r) = \bigodot \hat{F}_i^{d \cdot \hat{L}_i} (X_{\langle r \cdot \hat{H}_i, r \cdot \hat{W}_i, w \cdot \hat{C}_i \rangle})$$

$$\text{Memory}(\mathcal{N}) \leq \text{target_memory}$$

$$\text{FLOPS}(\mathcal{N}) \leq \text{target_flops}$$

Trong đó với w, d, r là các hệ số để chia tỷ lệ chiều rộng, chiều sâu và độ phân giải của mạng $\widehat{H}_i, \widehat{W}_i, \widehat{C}_i$ là các tham số được xác định trước trong mạng cơ sở.

Từ đó, nhóm tác giả đề xuất phương pháp compound scaling, sử dụng hệ số kép ϕ để biến đổi tỷ lệ một cách đồng nhất độ rộng, độ sâu và độ phân giải của mạng theo các công thức và nguyên tắc sau gọi là công thức (3):

$$d = \alpha^\phi$$

$$w = \beta^\phi$$

$$r = \gamma^\phi$$

$$\text{st}\alpha^\phi * \beta^\phi * \gamma^\phi \approx 2$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$$

.

Trong đó:

- d, w, r lần lượt là độ rộng, độ sâu và độ phân giải của mạng.
- α, β, γ là các hằng số có thể được xác định bằng small grid search.
- ϕ là hệ số do người dùng chỉ định để kiểm soát số lượng tài nguyên khác có sẵn để thực hiện phép biến đổi tỷ lệ cho mô hình.

Tuy nhiên, FLOPS của một mạng CNN thông thường tỷ lệ với d, w^2, r^2 tức là, độ sâu mạng tăng gấp đôi sẽ tăng gấp đôi FLOPS, nhưng tăng gấp đôi độ rộng hoặc độ phân giải của mạng sẽ tăng FLOPS lên bốn lần. Từ đó việc thực hiện phép biến đổi tỷ lệ với phương trình trên sẽ gia tăng tổng FLOPS khoảng $(\alpha * \beta^2 * \gamma^2)^\phi$. Trong bài báo gốc, nhóm tác giả ràng buộc $\alpha * \beta^2 * \gamma^2 \approx 2$ nhằm khi thực hiện các thay đổi tham số α, β, γ nào thì tổng FLOPS của mạng sẽ gia tăng xấp xỉ khoảng 2^ϕ lần. Đi kèm với phương pháp compound scaling, nhóm tác giả cũng đã phát triển một mạng CNN

được gọi là EfficientNet. Lấy cảm hứng từ các nghiên cứu trước, nhóm tác giả phát triển mạng CNN của mình bằng cách sử dụng multi-objective neural architecture search nhằm tìm kiếm kiến trúc của lớp F_i có khả năng tối ưu hóa cả độ chính xác và FLOPS cho mô hình. Cụ thể, nhóm nghiên cứu khởi tạo không gian tìm kiếm tương tự không gian được đề cập trong mô hình MnasNet[14] của Mingxing Tan, Quoc V. Le và các đồng sự[7]. Hàm mục tiêu tối ưu hóa của không gian tìm kiếm có công thức như sau:

$$ACC(m) * \left[\frac{FLOPS(m)}{T} \right]^w$$

Trong đó:

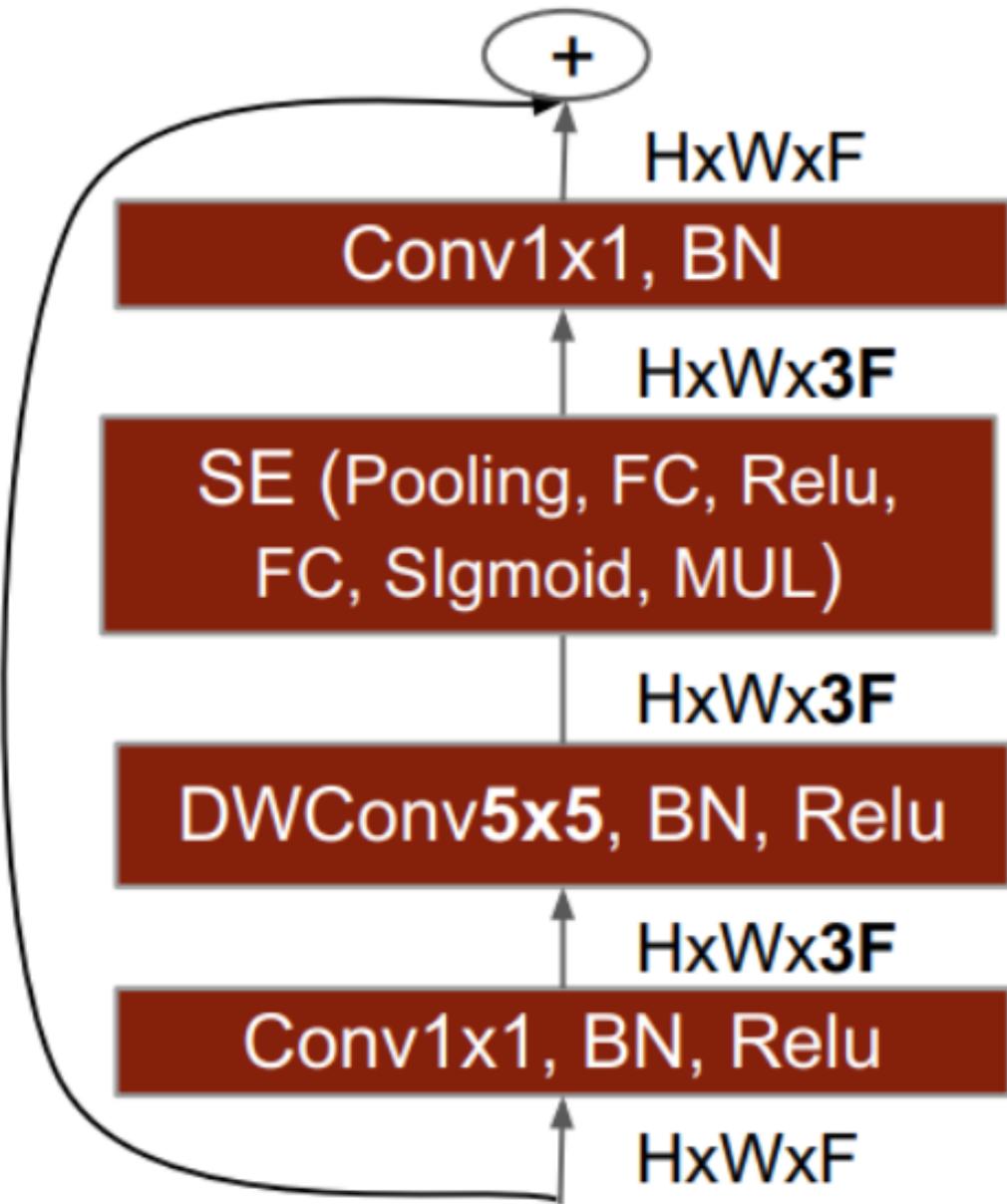
- $ACC(m)$ và $FLOPS(m)$ biểu thị độ chính xác và FLOPS của mô hình m
- T đại diện cho giới hạn mục tiêu FLOPS của không gian tìm kiếm.
- $w = -0,07$ là siêu tham số để kiểm soát sự đánh đổi giữa độ chính xác và FLOPS.

Điểm khác biệt so với mô hình MnasNet, nhóm tác giả tối ưu hóa FLOPS thay vì độ trễ(latency) vì họ không nhắm mục tiêu bất kỳ thiết bị phần cứng cụ thể nào. Sau quá trình tìm kiếm trên, nhóm tác giả đã tìm ra kiến trúc mạng CNN hiệu quả, thỏa mãn các mục tiêu được đề ra và đặt tên EfficientNet-B0.

Stage	Operator	Resolution	#channels	#layers
1	Conv3x3	224x224	32	1
2	MBConv1,k3x3	112x112	16	1
3	MBConv6,k3x3	112x112	24	2
4	MBConv6,k5x5	56x56	40	2
5	MBConv6,k3x3	28x28	80	3
6	MBConv6,k5x5	14x14	112	3
7	MBConv6,k5x5	14x14	192	4
8	MBConv6,k3x3	7x7	320	1
9	Conv1x1/Pooling/FC	7x7	1, 280	1

Hình 3.14: Bảng mô tả kiến trúc EfficientNet-B0

Điều đặc biệt ở kiến trúc EfficientNet là kiến trúc các khối mobile inverted bottleneck (MBConv).



Hình 3.15: Hình minh họa khối MBConv3(k5x5)

Khối MB Conv có sử dụng tích chập tách biệt chiều sâu (depthwise separable convolution) nhằm làm giảm số lượng tham số và chi phí tính toán. Kết hợp với phép tích chập trên, khối MB Conv còn được bổ sung khối squeeze-and-excitation có tác dụng tăng cường thông tin giữa các kênh(channel) nhằm tăng cường chất lượng tính năng. Qua đó, mô hình sẽ vừa có kích thước nhỏ gọn nhưng vẫn có độ chính xác phù hợp với mục tiêu.

Cuối cùng, nhóm tác giả áp dụng phương pháp compound scaling lên kiến trúc mạng EfficientNet-B0 theo hai bước sau:

- Đặt cố định giá trị của ϕ bằng 1, nhóm nghiên cứu đã thực hiện small grid search kết hợp biểu thức (2) và (3) và thu được bộ giá trị tối ưu $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$ theo ràng buộc của $\alpha * \beta^2 * \gamma^2 \approx 2$ cho mạng EfficientNet-B0.
- Cố định α, β, γ dưới dạng các hằng số và áp dụng compound scaling lên mạng EfficientNet-B0 với các ϕ khác nhau, từ đó nhóm nghiên cứu thu được các mô hình mới từ EfficientNet-B1 đến EfficientNet-B7.

Các mô hình tiền huấn luyện thuộc họ EfficientNet luôn đảm bảo độ chính xác và tốc độ cho các tác vụ thị giác máy tính.

3.2 Tiềm xử lý dữ liệu và tăng cường ảnh:

Tiềm xử lý là một bước rất quan trọng trong việc giải quyết bất kì vấn đề nào trong lĩnh vực của học máy nói chung cũng như là deep learning nói riêng. Hầu hết các bộ dữ liệu được sử dụng trong các bài toán liên quan đến học máy cũng như học sâu cần được xử lý, biến đổi làm sạch biến đổi trước khi đưa vào mô hình học sâu huấn luyện. Trong bài này sẽ giới thiệu một vài phương pháp tiềm xử lý dữ liệu điển hình như là CLAHE (Contrast Limited Adaptive Histogram Equalization).

CLAHE (Contrast Limited Adaptive Histogram Equalization).

Ràng buộc trong hình ảnh: là sự khác nhau giữa màu sắc và độ sáng là yếu tố làm cho một đối tượng có thể phân biệt với một đối tượng khác trong cùng một trường nhìn.

Histogram Equalization :

Trong phạm vi của phần này chúng ta chỉ giới thiệu qua chứ không đi sâu vào chi tiết của thuật toán.

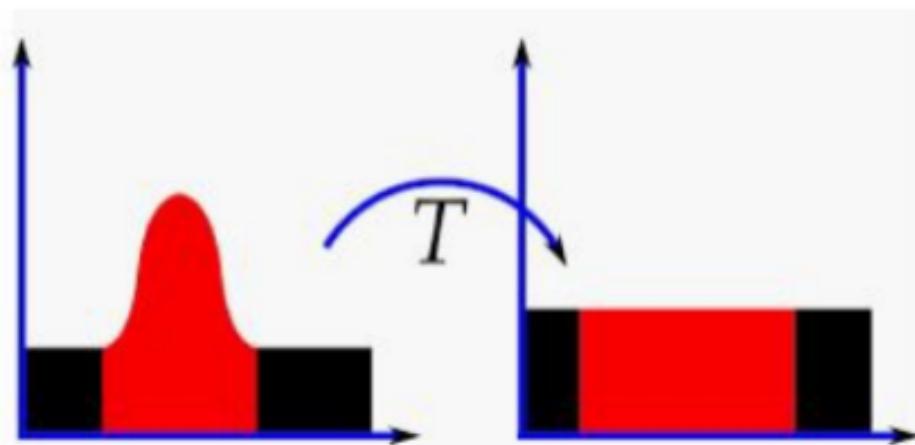
Histogram Equalization:

- Dùng để tăng độ tương phản của ảnh.

- Bằng cách thay đổi phân bố histogram của ảnh.

Để thay đổi histogram, cách duy nhất chính là thay đổi intensity của các điểm ảnh. Nói một cách đơn giản, ta đi tạo một bảng màu mới sao cho khoảng màu mới rộng hơn khoảng màu cũ.

Từ equalization mang nghĩa sự chia đều, làm cho bằng nhau. Vậy histogram equalization có nghĩa là làm cho histogram đồng đều, bằng nhau. Cụ thể hơn, chính là làm cho histogram cho hình dáng về gần một đường ngang nhất có thể (làm số pixels của mỗi một intensity gần bằng nhau). Ví dụ, với ảnh 8 bit ($0 \rightarrow 255$) có độ phân giải 20×40 , 800 pixels đó của ảnh chỉ tập trung trong khoảng từ $120 \rightarrow 150$, histogram equalization sẽ tìm cách phân đều 800 pixels đó sao cho cường độ sáng nhỏ nhất là 0 và lớn nhất là 255.



Graphical Representation of Histogram Equalization

Hình 3.16: Minh họa về Histogram Equalization

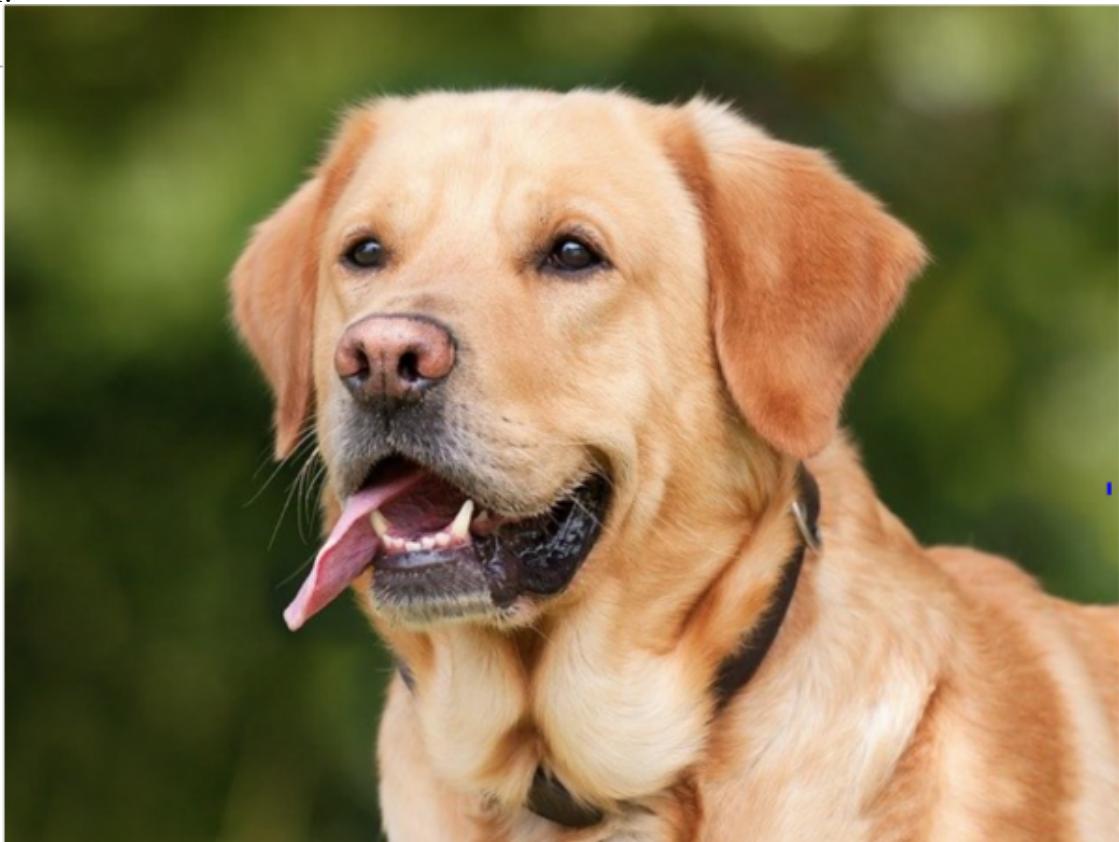
3.3 Tăng cường dữ liệu:

Trong thời đại số hiện nay, người ta nói, có dữ liệu như là nắm được vàng trong tay, dữ liệu càng ngày càng quan trọng đối với thời đại công

nghệ số. Trong học sâu cũng vậy, nếu một mô hình có quá ít dữ liệu thì sẽ không thể có kết quả tốt được, vì vậy chúng ta cần những kĩ thuật để làm giàu, tăng cường dữ liệu, sau đây là một vài phương pháp tăng cường dữ liệu.

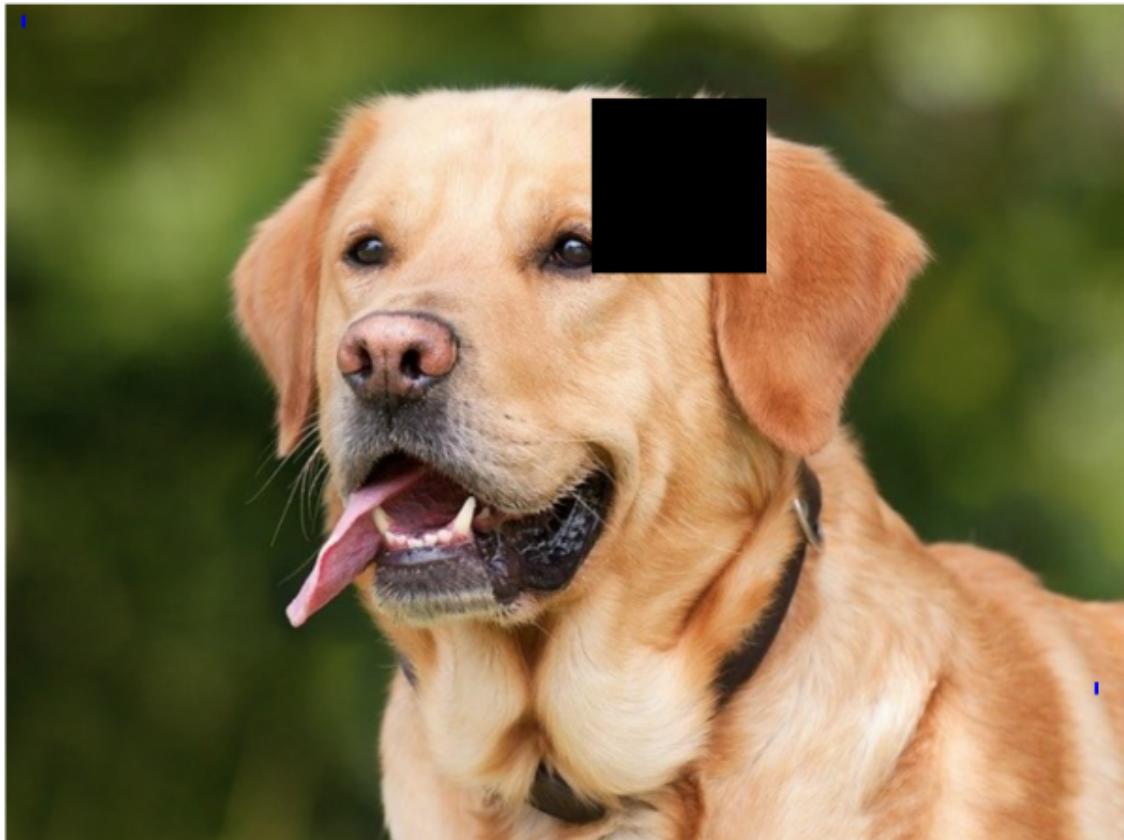
Cutout:

Ý tưởng của phương pháp này là từ một hình ảnh bất kì trong tập dữ liệu, ta cắt random một vùng ngẫu nhiên trên bức ảnh để tạo ra bức ảnh mới.



Hình 3.17: Hình ảnh khi chưa sử dụng cutout

Sau khi sử dụng cutout ta được một ảnh mới và có thể được rất nhiều ảnh khi ta chọn ngẫu nhiên để cắt.



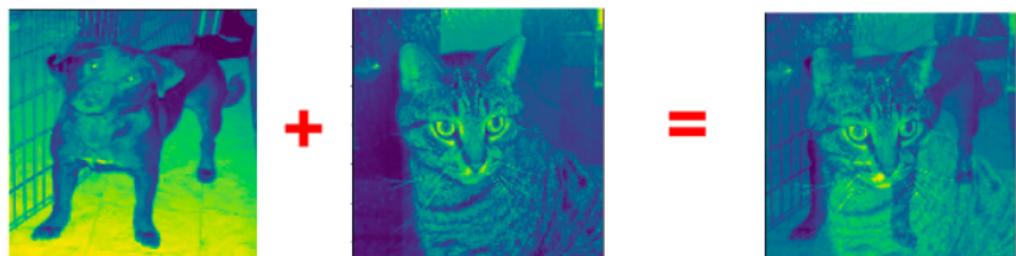
Hình 3.18: Hình ảnh khi sử dụng cutout

Mix up

Được giới thiệu vào năm 2018 và mang lại nhiều hiệu quả ý tưởng của phương pháp này là kết hợp hai ảnh lại với nhau giúp làm giảm khả năng ghi nhớ của các nhãn bị hỏng, cài đặt đơn giản và tăng tốc độ huấn luyện. Dưới đây là công thức của Mix up.

$$\text{newImage} = \alpha * \text{image1} + (1 - \alpha) * \text{image2}$$

$$\text{newTarget} = \alpha * \text{target1} + (1 - \alpha) * \text{target2}$$



Hình 3.19: Minh họa khi sử dụng mixup

Tại sao lại sử dụng mixup:

- MixUp giúp dễ dàng trong việc điều chỉnh các mô hình Machine Learning cho các tác vụ Thị giác máy tính. Bạn có thể đào tạo DNN trên một GPU duy nhất trong 6 phút và vẫn nhận được 94% trên tập dữ liệu CIFAR 10 .
- Tính đơn giản và tốc độ giúp người học có thể dễ dàng nắm bắt, đơn giản mà hiệu quả ,giúp tăng tốc độ.

Grayscale

Tăng thang độ xám ngẫu nhiên khiến hình ảnh màu đầu vào được chuyển đổi thành hình ảnh đầu ra có thang độ xám.

$$grayscale = \frac{R + G + B}{3}$$



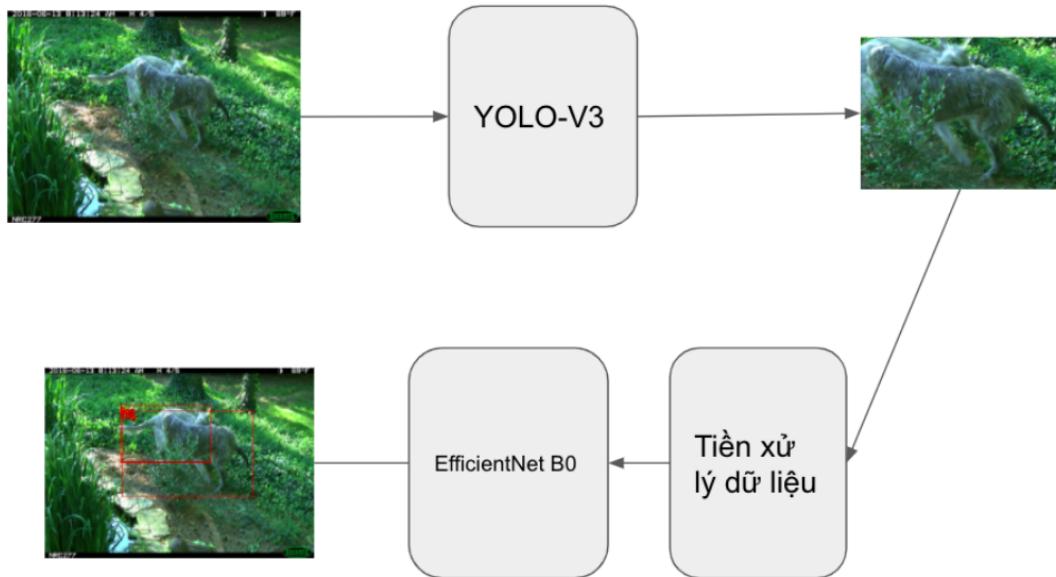
Hình 3.20: Ảnh trước khi sử dụng grayscale



Hình 3.21: Ảnh sau khi sử dụng grayscale

3.4 Phương pháp đề xuất

Phương pháp của chúng tôi được phát triển dựa trên công trình nghiên cứu của Abulikemu và các đồng sự [1]. Chúng tôi mô tả bài toán và qui trình của phương pháp đề xuất thực hiện đề tài trong hình 3.3.1. Qua đó, đầu vào sẽ là một ảnh được chụp từ loại camera giám sát tĩnh. Ở giai đoạn phát hiện đối tượng, mô hình sẽ cố gắng xác định trong ảnh có đối tượng hay không và vị trí các đối tượng trong ảnh. Sau đó, mô hình sẽ thực hiện phân lớp đối tượng ở giai đoạn tiếp theo. Cuối cùng, kết quả trả về sẽ là vị trí và nhãn của các đối tượng với kỳ vọng nhãn dự đoán và vị trí đối tượng sẽ chính xác với đối tượng trong ảnh đầu vào.



Hình 3.22: Ảnh mô tả qui trình của phương pháp đề xuất thực hiện đề tài

Trong giai đoạn phát hiện đối tượng, chúng tôi nhận thấy các ảnh của kiểu camera giám sát tĩnh thường gặp phải nhiều thách thức được nêu ở phần 1.2 gây khó khăn cho các mô hình. Vì thế, chúng tôi đã cần lựa chọn nhiều mô hình khác nhau vừa phải đảm bảo yếu tố hoạt động tốt trên dạng dữ liệu mà đề tài nhắm đến và mô hình phải phổ biến, dễ cài đặt để phù hợp với thời gian đồ án. Sau quá trình tìm kiếm, tham khảo và thử nghiệm, chúng tôi quyết định chọn ra 3 mô hình đã được tiền huấn luyện: Faster R-CNN (backbone ResNet50), Faster R-CNN (backbone ResNet101) và YOLOv3 vì:

- Các mô hình trên là các mô hình phát hiện đối tượng hiện đại đạt nhiều kết quả cao trên các tập dữ liệu lớn uy tín về Thị giác máy tính.
- Các mô hình trên đã được tiền huấn luyện trên các tập dữ liệu lớn. Chúng tôi có thể tận dụng chúng mà không cần mất nhiều thời gian và tài nguyên cho việc huấn luyện.

Sau đó, chúng tôi sẽ thử nghiệm chạy các mô hình với các tập dữ liệu đại diện cho kiểu camera giám sát tĩnh và so sánh hiệu suất để chọn ra mô hình phù hợp nhất.

Ở giai đoạn phân lớp đối tượng, chúng tôi nhận thấy nhiều mô hình phát hiện đối tượng kém hiệu suất khi phân lớp đối tượng cho dạng dữ liệu ảnh từ camera giám sát tĩnh. Đặc biệt, với dữ liệu của bầy camera, chúng tôi nhận thấy có rất nhiều ảnh trống (ảnh không chứa đối tượng) dễ gây ảnh hưởng chất lượng của mô hình. Vì thế, chúng tôi đề xuất cắt các đối tượng trong ảnh và tạo thành ảnh mới cho bước phân lớp đối tượng kết hợp với loại bỏ các ảnh trống. Sau đó, chúng tôi bổ sung thêm một phương pháp tiền xử lý hình ảnh và tăng cường dữ liệu với gia tăng lượng thông tin có ích mà mô hình có thể học được. Cuối cùng, chúng tôi chọn lựa mô hình EfficientNet-B0 làm mô hình phân lớp phục vụ giải quyết bài toán vì:

- Đây là một nghiên cứu được công bố gần đây (năm 2019) trong một hội nghị uy tín - International Conference on Machine Learning.
- Mô hình đạt được nhiều thành tựu nổi bật đã được nêu trong phần 3.1.7.
- Mô hình có các dạng tiền huấn luyện trên các tập dữ liệu lớn như ImageNet, ... cho phép chúng tôi có thể thực hiện huấn luyện tăng cường chuyển giao (transfer learning) nhằm tận dụng sức mạnh của mô hình và phù hợp với tài nguyên, thời gian để hoàn thành đồ án.

Cụ thể hơn, chúng tôi sử dụng thuật toán CLAHE nhằm cải thiện chất lượng hình ảnh nhưng không áp dụng trên toàn bộ ảnh mà chỉ áp dụng lên một số trường hợp với xác suất là 0.2. Chúng tôi kì vọng sẽ tạo nên một phần dữ liệu có kết quả phân loại tốt để giảm sự ảnh hưởng của các kết quả âm tính lên mô hình. Tương tự, chúng tôi cũng áp dụng grayscale vào bộ dữ liệu với xác suất là 0.1 nhằm mô phỏng điều kiện chụp ảnh vào ban đêm giúp làm phong phú bộ dữ liệu. Để khắc phục trường hợp quá khớp dữ liệu(over-fitting), chúng tôi sử dụng một số biện pháp tăng cường dữ liệu được nêu ở phần 3.3. Ngoài ra, chúng tôi còn kì vọng mô hình sẽ học được nhiều đặc trưng hơn cho từng lớp để nhận dạng chính xác hơn khi gấp các trường hợp thách thức về ảnh được nêu ở phần 1.2. Ví dụ: ảnh được cutout sẽ cho ra trường hợp đối tượng bị mất một phần và từ đó mô hình cần học được nhiều đặc trưng hơn để phân loại đúng. Bên cạnh

đó, chúng tôi lựa chọn thuật toán Adam làm thuật toán tối ưu vì Adam là sự kết hợp của Momentum và RMSprop. Nếu giải thích theo hiện tượng vật lí thì Momentum giống như 1 quả cầu lao xuống dốc, còn Adam như 1 quả cầu rất nặng có ma sát, vì vậy nó dễ dàng vượt qua local minimum tới global minimum và khi tới global minimum nó không mất nhiều thời gian dao động qua lại quanh đích vì nó có ma sát nên dễ dừng lại hơn. Thuật toán Adam hoạt động khá tốt, tiến nhanh tới global minimum hơn các phương pháp khác. Cuối cùng, sau các bước lựa chọn trên, chúng tôi kì vọng tạo ra một mô hình có độ chính xác cao đáp ứng được mục tiêu đề tài trong phần 1.5.

Chương 4

Kết quả thực nghiệm

4.1 Giới thiệu tập dữ liệu

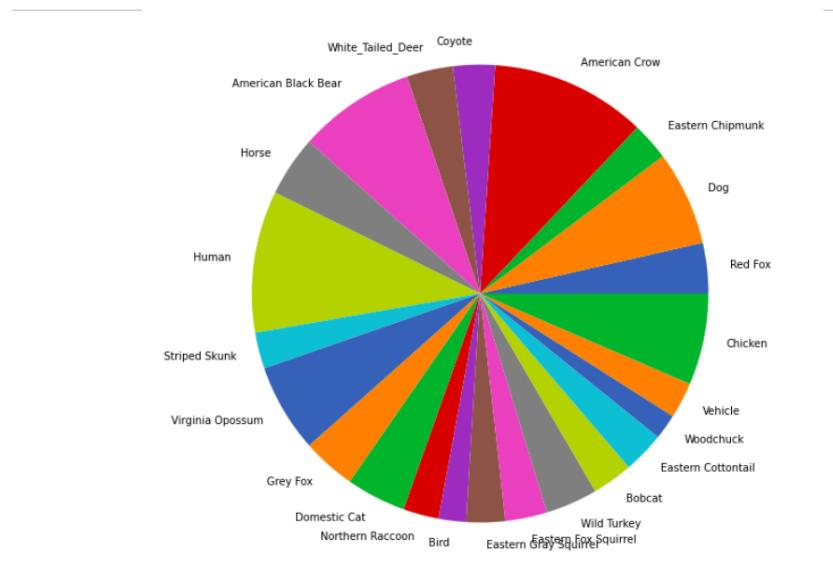
Đề tài sử dụng 2 bộ dữ liệu đại diện cho 2 kiểu camera được đề cập trong phạm vi đề tài mục 1.3

4.1.1 Ena24-LILABC [3]

Bộ dữ liệu ENA24-detection chứa 11596 ảnh từ 23 loài động vật (chưa tính con người và xe cộ) được chụp ở vùng Đông Bắc nước Mỹ. Quạ đen, gấu đen và chó là các loài động vật phổ biến trong tập dữ liệu.

Dịnh dạng của những hộp bao quanh đối tượng được định dạng theo dữ liệu của COCO.

Hình bên dưới biểu diễn tỉ lệ các loài động vật có trong dữ liệu:

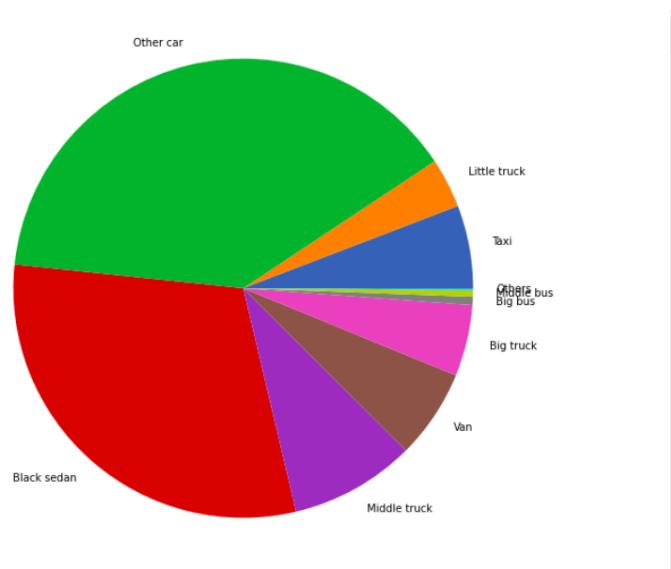


Hình 4.1: Biểu đồ các loại của bộ dữ liệu Ena24-LILABC

4.1.2 Citycam-CMU [3]

Bộ dữ liệu citycam được **trình** từ những camera tại nhiều địa điểm ở New York, Mỹ. Chứa khoảng 60000 hình với 900000 khung đánh dấu xe cộ. Định dạng của những hộp bao quanh đối tượng được định dạng theo file XML.

Hình bên dưới biểu diễn tỉ lệ loại các xe cộ có trong dữ liệu:



Hình 4.2: Biểu đồ các loại xe của bộ dữ liệu Citycam-CMU

4.2 Chi tiết quá trình thực nghiệm

4.2.1 Môi trường huấn luyện

Toàn bộ quá trình huấn luyện mô hình và định danh chúng tôi thực hiện với Google Colab.

Những quá trình như cắt hình, chuyển đổi định dạng thì được thực hiện trên máy tính cá nhân. Do số lượng ảnh khá lớn nên việc thực hiện đọc ghi file trên Google Colab vào Google Drive bị hạn chế do chính sách của Google. Máy tính cá nhân chúng tôi sử dụng cấu hình CPU Intel Core I7 thế hệ thứ 8, RAM 16GB, MacOS 11.4.

4.2.2 Ngôn ngữ và thư viện

Chúng tôi đã sử dụng Python làm ngôn ngữ chính cho thực hiện đề tài này. Trong đó chúng tôi xây dựng mô hình bằng thư viện tensorflow, pytorch là chính.

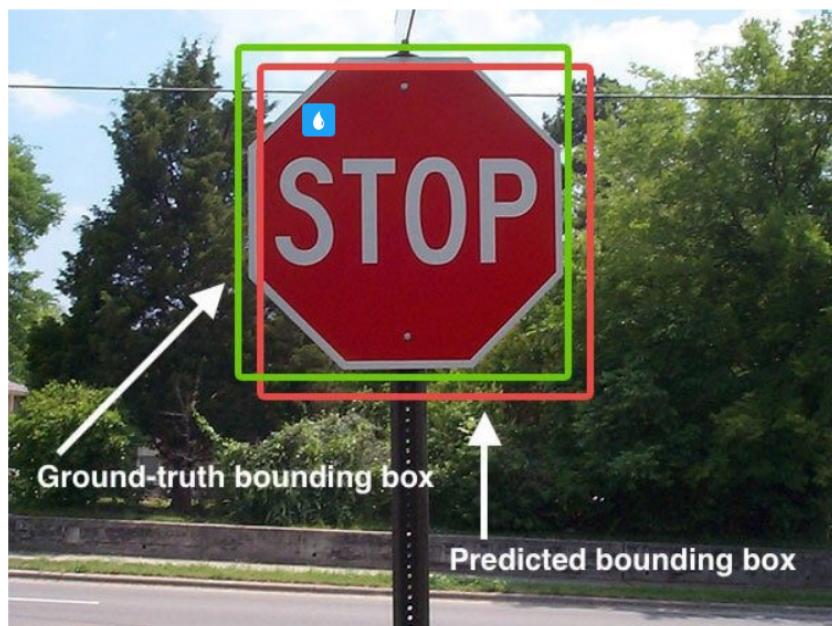
Trong đó khi thử nghiệm mô hình F-RCNN chúng tôi sử dụng thư viện tensorpack, YoloV3 chúng tôi sử dụng thư viện của ultralytics [18] .

4.2.3 Hàm lõi và độ đo chính xác:

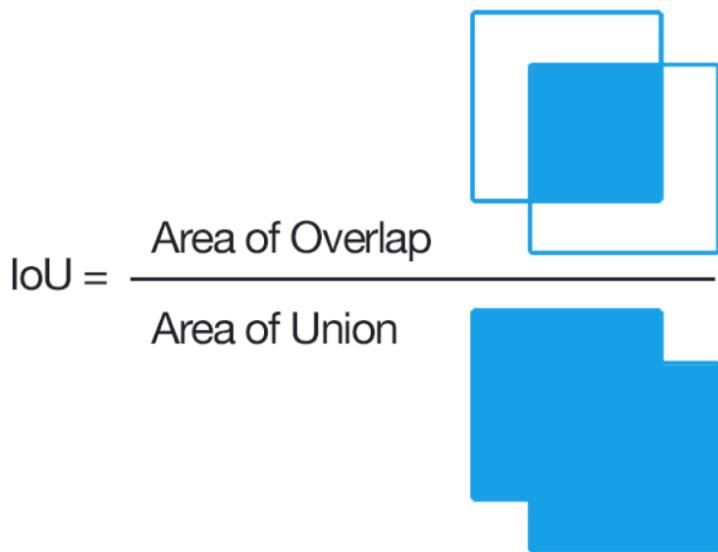
Intersection over Union (IoU)

IoU được sử dụng trong bài toán phát hiện đối tượng, để đánh giá xem bounding box dự đoán đối tượng khớp với ground truth thật của đối tượng. Ví dụ về hệ số IoU, nhận xét:

- Chỉ số IoU trong khoảng $[0,1]$.
- IoU càng gần 1 thì bounding box dự đoán càng gần ground truth .



Hình 4.3: Minh họa về IOU



Hình 4.4: Minh họa về công thức tính IOU



Hình 4.5: Minh họa về số liệu tính IOU

Confusion Matrix

Là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại.

Để cho dễ hiểu ta lấy ví dụ về bài toán ung thư :trong bài toán ung thư ta có 2 lớp: lớp bị ung thư được chuẩn đoán Positive và lớp không bị ung thư được chuẩn đoán là Negative:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Hình 4.6: Các chỉ số của confusion matrix

- True Positive (TP): những bệnh nhân ta đoán là có bệnh đúng là đang mang bệnh.
- True Negative (TN): những bệnh nhân ta đoán là không có bệnh đúng là đang khỏe mạnh.
- False Positive (FP): những bệnh nhân ta đoán là có bệnh thật ra đang khỏe mạnh.
- False Negative (FN): những bệnh nhân ta đoán là không có bệnh thật ra đang mang bệnh.

Precision: đây là tỷ lệ giữa những người thật sự có bệnh so với tất cả các ca được dự đoán là có bệnh. Nói cách khác, có bao nhiêu dự đoán “positive” là thật sự “true” trong thực tế.

$$Precision = \frac{TP}{TP + FP}$$

Recall (đôi khi còn được gọi là Sensitivity): trong những người thực sự có bệnh, bao nhiêu trong số họ được dự đoán đúng bởi mô hình . Nói cách khác, có bao nhiêu dự đoán “positive” đúng là do mô hình đưa ra.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

Tuy nhiên, chỉ có Precision hay chỉ có Recall thì không đánh giá được chất lượng mô hình.

- Chỉ dùng Precision, mô hình chỉ đưa ra dự đoán cho một điểm mà nó chắc chắn nhất. Khi đó Precision = 1, tuy nhiên ta không thể nói là mô hình này tốt.
- Chỉ dùng Recall, nếu mô hình dự đoán tất cả các điểm đều là positive. Khi đó Recall = 1, tuy nhiên ta cũng không thể nói đây là mô hình tốt.

Khi đó F1-score được sử dụng. F1-score là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0). F1-score được tính theo công thức:

$$F1 * (Precision + Recall) = 2 * Precision * Recall$$

4.2.4 Phương pháp đánh giá

Khi so sánh mô hình F-RCNN và YOLO-V3 cho việc nhận dạng:

Đánh giá 2 model trên tập động vật ena24.

Khi huấn luyện mỗi epochs chúng tôi tính ra độ lỗi nhận dạng vật bằng công thức của IoU (box_loss), độ lỗi đánh nhãn (label_loss) theo công thức binary cross-entropy.

Thời gian huấn luyện cũng là một yếu tố chúng tôi dùng để đánh giá 2 mô hình.

Khi so sánh giữa mô hình yolov3 thường và yolov3 có thêm khối spp:

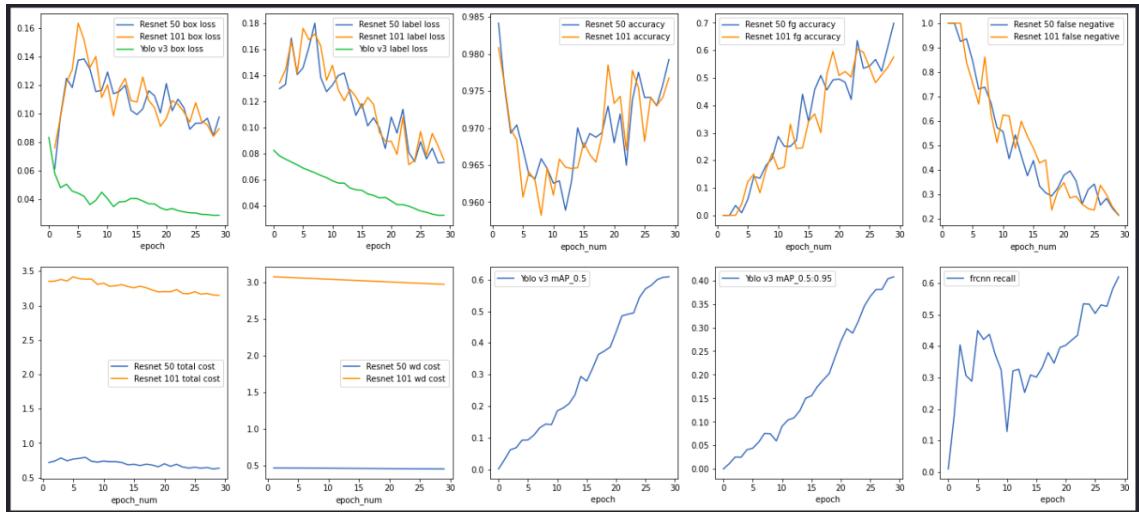
Đánh giá 2 model trên tập động vật ena24.

Khi huấn luyện xong tập train, chúng tôi chạy qua tập test để tính số điểm mAP.

Thời gian huấn luyện cũng là một yếu tố chúng tôi dùng để đánh giá 2 mô hình

4.2.5 Kết quả thu được

So sánh mô hình Faster RCNN (resnet50 và resnet101 backbone) và YOLO-V3 cho việc nhận dạng đối tượng:



Hình 4.7: Bảng kết quả

Như trên biểu đồ ta thấy sau 30 epochs thì khi huấn luyện độ lỗi nhận dạng (box_loss) hay độ lỗi đánh nhãn của mô hình YoloV3 luôn cho kết quả thấp hơn so với F-RCNN (cả resnet50 và resnet101 backbone).

Bảng 4.1: Bảng so sánh

30 epochs	F-RCNN(resnet50)	F-RCNN(resnet101)	Yolo-V3
mAP(0.5:0.95)	0.51	0.53	0.609
mAP(0.5)	0.12	0.15	0.407
Tổng thời gian huấn luyện	3.45 giờ	4.23 giờ	2.137 giờ

Dựa vào các kết quả này chúng tôi quyết định sử dụng mô hình YoloV3 cho việc nhận dạng đối tượng. Nhưng yolov3 có phiên bản thêm vào khối spp nên thí nghiệm kế tiếp chúng tôi so sánh giữ YoloV3 thường và YoloV3 có khối SPP.

Dựa vào các kết quả này chúng tôi quyết định sử dụng mô hình YoloV3-SPP cho việc nhận dạng đối tượng.

Một số ảnh minh họa việc so sánh (Nhãn trên hộp nhận dạng là từ YoloV3, nhãn góc trái phía trên là từ EfficientNet)

So sánh mô hình YoloV3 thường và YoloV3 có khối spp:

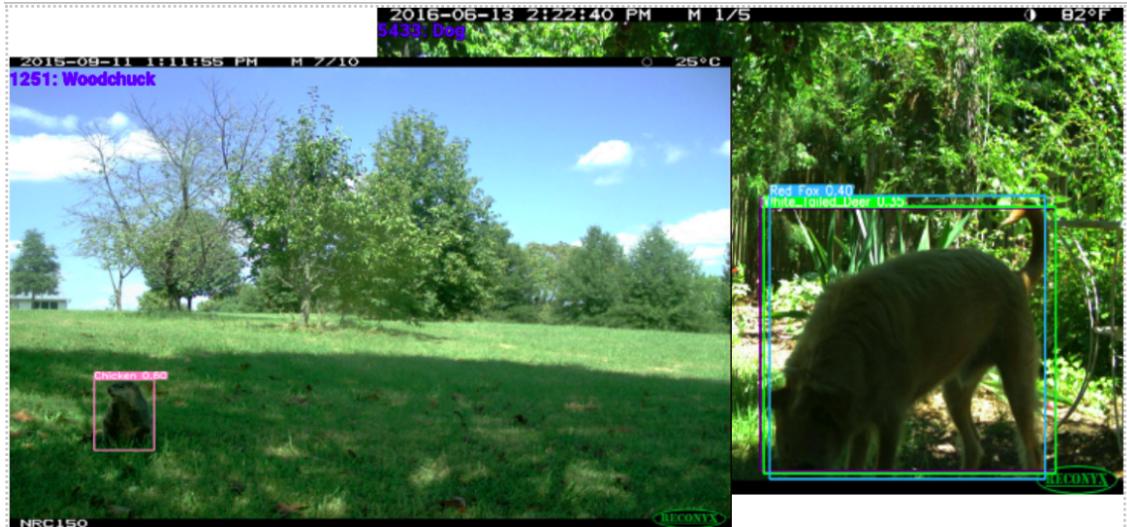
Bảng 4.2: Bảng so sánh Yolov3 với Yolo V3 spp

30 epochs	Yolo v3	Yolov3 -spp
mAP(0.5:0.95)	0.609	0.796
mAP(0.5)	0.407	0.536
tổng thời gian huấn luyện	2.137 giờ	2.834 giờ

So sánh việc phân lớp bằng 2 stage yolov3 cho việc nhận dạng và EfficientNet cho việc phân lớp với 1 stage chỉ yolov3:

Bảng 4.3: So sánh phân lớp

classification	Yolo v3-spp	Yolov3 -spp+efficientNet
F1-score	0.763	0.677



Hình 4.8: Ví dụ so sánh yolov3 và EfficientNet



Hình 4.9: So sánh yolov3 và EfficientNet

Khi so sánh 2 phương pháp, chúng tôi nhận thấy việc nhận dạng bằng YoloV3 rồi phân lớp bằng effecientnet (2 stage) gán nhãn chính xác hơn sử dụng YoloV3 cho cả việc nhận dạng và phân lớp (1 stage).

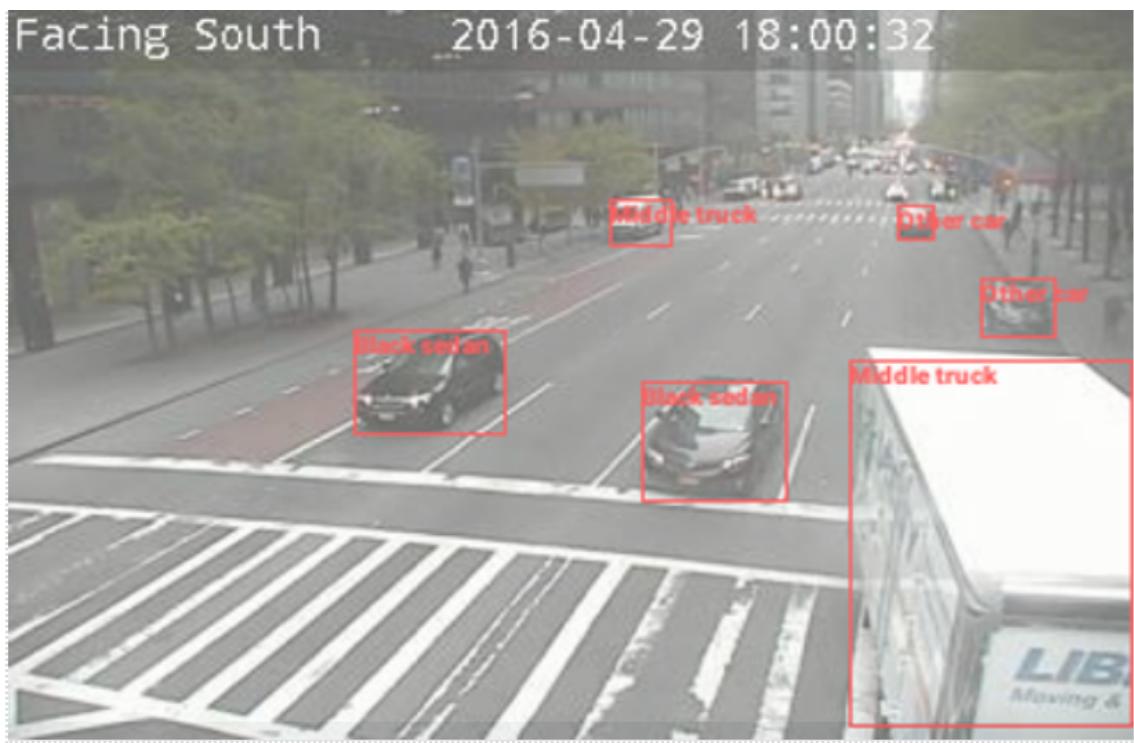
Phương pháp 2 stage yolov3 cho việc nhận dạng và EfficientNet cho việc phân lớp đối với tập dữ liệu citycam

Bảng 4.4: Số liệu cụ thể việc nhận dạng trên tập citycam

độ đo	Yolov3-spp
mAP(0.5:0.95)	0.497
mAP(0.5)	0.349

Bảng 4.5: Số liệu cụ thể việc phân lớp trên tập citycam

độ đo	EfficientNet
F1-score	0.477



Hình 4.10: Ví dụ về phân lớp 2 stage yolov3 cho việc nhận dạng và EfficientNet cho việc phân lớp đối với tập dữ liệu citycam



Hình 4.11: Ví dụ khác về phân lớp 2 stage yolov3 cho việc nhận dạng và EfficientNet cho việc phân lớp đối với tập dữ liệu citycam

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Qua kết quả thu được, có thể thấy được rằng trong việc nhận dạng vật thể, YoloV3 thực hiện tốt hơn so với Faster R-CNN trong lĩnh vực camera trap. Đối việc phân lớp thì việc nhận dạng bằng YoloV3 rồi phân lớp bằng effecitienet (2 stage) gán nhãn chính xác hơn sử dụng YoloV3 cho cả việc nhận dạng và phân lớp (1 stage).

Chúng tôi tự tin rằng phương pháp 2 stage có thể sử dụng cho camera trap để ghi lại hay theo dõi các hành vi của động vật. Tuy nhiên đối với việc theo dõi giao thông bằng phương pháp này chưa đem lại kết quả tốt.

5.2 Hướng phát triển

Do đặc trưng của kiểu camera giám sát tĩnh là môi trường ít thay đổi môi trường nền xung quanh nên chưa chắc rằng đối với môi trường mới thì kết quả vẫn tốt như kỳ vọng. Vì thế, chúng tôi phải huấn luyện thêm dữ liệu hoặc có cách điều chỉnh mô hình có khả năng thích nghi môi trường mới. Đặc biệt, kiểu dữ liệu do loại camera này mang lại thường bị mất cân bằng dữ liệu, nhóm sẽ cần xử lý vấn đề này khi tiếp tục phát triển mô hình.

Đối với việc theo dõi giao thông đô thị, phương pháp vẫn chưa cho kết quả tốt, có một số nghi vấn như tập dữ liệu huấn luyện chưa rõ nét, quá nhiều vật thể trong cùng một khung hình, những vật thể ở xa chưa phát hiện do hình ảnh bị mờ. Do vậy có thể cải thiện mô hình để có thể phát hiện nhiều vật thể ở trong cùng một mô hình tốt hơn.

Nhóm chúng tôi sử dụng dữ liệu là hình ảnh tĩnh, có thể cải tiến phương pháp để có thể thực hiện với nhóm dữ liệu video hay livestream bằng việc

tách frame. Hơn thế nữa, chúng tôi có thể kết hợp mô hình đề xuất với mô hình nhận dạng giọng nói để trở thành hệ hỏi đáp trên ảnh (Image question answering). Sự kết hợp cả 2 miền tri thức thị giác máy tính và xử lý ngôn ngữ tự nhiên là xu hướng phát triển trong tương lai gần.

Tài liệu tham khảo

Tiếng Anh

- [1] Abuduweili, Abulikemu, Wu, Xin, and Tao, Xingchen. *Efficient Method for Categorize Animals in the Wild*. 2019. arXiv: 1907.13037 [cs.CV].
- [2] Dalal, N. and Triggs, B. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.
- [3] Dataset. https://github.com/Microsoft/CameraTraps/blob/main/data_management/README.md#coco-cameratraps-format. Accessed: 2022-02-12.
- [4] Girshick, Ross. *Fast R-CNN*. 2015. arXiv: 1504.08083 [cs.CV].
- [5] Girshick, Ross et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014. arXiv: 1311.2524 [cs.CV].
- [6] He, Kaiming et al. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- [7] Huang, Yanping et al. *GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism*. 2019. arXiv: 1811.06965 [cs.CV].
- [8] Redmon, Joseph and Farhadi, Ali. *YOLO9000: Better, Faster, Stronger*. 2016. arXiv: 1612.08242 [cs.CV].
- [9] Redmon, Joseph and Farhadi, Ali. *YOLOv3: An Incremental Improvement*. 2018. arXiv: 1804.02767 [cs.CV].
- [10] Redmon, Joseph et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

- [11] Ren, Shaoqing et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. arXiv: 1506 . 01497 [cs.CV].
- [12] Simonyan, Karen and Zisserman, Andrew. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409 . 1556 [cs.CV].
- [13] Tan, Mingxing and Le, Quoc V. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. arXiv: 1905 . 11946 [cs.LG].
- [14] Tan, Mingxing et al. *MnasNet: Platform-Aware Neural Architecture Search for Mobile*. 2019. arXiv: 1807 . 11626 [cs.CV].
- [15] *Tim hieu mo hinh yolov2*. <https://aicurious.io/posts/tim-hieu-yolo-cho-phat-hien-vat-tu-v1-den-v3/>. Accessed: 2010-09-30.
- [16] *Tim hieu mo hinh yolov3*. <https://aicurious.io/posts/tim-hieu-yolo-cho-phat-hien-vat-tu-v1-den-v3/>. Accessed: 2010-09-30.
- [17] Uijlings, Jasper et al. “Selective Search for Object Recognition”. In: *International Journal of Computer Vision* 104 (Sept. 2013), pp. 154–171. DOI: 10.1007/s11263-013-0620-5.
- [18] *Ultralytics yolov3 Description*. <https://github.com/ultralytics/yolov3>. Accessed: 2010-09-30.
- [19] Viola, P. and Jones, M. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.
- [20] Zaidi, Syed Sahil Abbas et al. *A Survey of Modern Deep Learning based Object Detection Models*. 2021. arXiv: 2104 . 11892 [cs.CV].